Statistical Physics of Design

by

Andrei A. Klishin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Physics)
in The University of Michigan
2020

Andrei A. Klishin

aklishin@umich.edu

ORCID iD: 0000-0002-5740-8520

ACKNOWLEDGEMENTS

I went to graduate school hoping to become a complex systems scientist. That more or less happened. However, the path towards that result has been incredibly winding and very different from what I could project going in. Most importantly, I met an incredible cast of people who in many ways helped me to make it through.

The work in this dissertation is chiefly a collaboration with my advisor, Greg van Anders. With Greg's guidance, I understood how to efficiently expand the boundaries of what we call physics, or science in general, and how to sell that understanding to broader community. Greg extensively drilled me with writing and re-writing my texts, drawing and re-drawing my figures in the interest of distilling the message to its clearest form. I am grateful for his comprehensive program of training the next generation of scientists. I am also deeply grateful to him for morally and financially supporting my outreach activities during graduate school, even though they did not directly contribute to this research.

I am also fortunate to have deep connections with the Michigan faculty who formed my dissertation committee. Mark Newman taught me everything I know about networks, and provided formal support in a crucial moment by becoming a co-chair of my dissertation. Dave Singer taught me everything I know about Naval Engineering, and on many occasions defended my work more valiantly than I could have done myself. Sharon Glotzer drove a lot of my thinking about self-assembly, served as a great role model in how to weave together scientific narratives, and taught me the truth about entropy. Roberto Merlin provided an important perspective on graduate school and the academic world that was largely agnostic of my research subject and thus not tainted by it.

I would also like to thank Michael Brenner for giving me an opportunity to spend two years at Harvard School of Engineering and Applied Sciences, a rich academic environment complementary to that of Michigan. Michael taught me a lot about self-assembly (again), and Chapter VII of this dissertation largely builds on the work developed in his group. I am grateful to the Brenner group for providing me with an intellectual home during this time and exposing me to many diverse scientific ideas.

I would like to thank my collaborators on projects presented here. Colin Shields and Conner Goodrum filled in a lot of Naval Engineering and design context in my work. Alec Kirkley assisted with computations in Chapter V. Adam Jermyn introduced me to the tensor network methods and provided the backend software package `PyTNR` for tensor network computations. Norman Mackay helped with the `TenZ` package that wraps physics around the `PyTNR` core.

I would like to thank James Sethna and Nigel Goldenfeld for useful discussions on the history and philosophy of certain statistical mechanics concepts.

I would like to thank the Office of Naval Research and specifically Kelly Cooper for funding this research. During my studies I enjoyed an ability to travel to a variety of professional events, and I don't take that opportunity for granted.

I would like to thank people who made my time at Michigan much more pleasant. I was attracted to Michigan by the Center for Studies of Complex Systems, led by Charlie Doering and his staff, who do a superb job at attracting resident faculty and guest speakers, as well as providing systematic complex systems training to students of different levels. On the Physics Department side, I was supported multiple times by Bradford Orr during his time as department chair. Socially, I greatly enjoyed the company of Harry Liu, Steve Novakov, Yuri Popov, and Markus Wallerberger over the years.

I would like to thank the teachers who initially put me on the path towards becoming a physicist during my years in high school and early years of university in Belarus. While the things they taught me are perhaps too elementary to be cited in this dissertation, I still firmly believe that I would not be the scientist I am today if all of my education happened in the United States. Belarus might be in political turmoil now, but I still believe that it has a huge reservoir of untapped human talent that will make itself known. Жыве Беларусь!

I would like to thank my parents who let me go out into the world and make my own decisions, but supported all of my decisions nonetheless.

Lastly, I would like to thank Chrisy Xiyu Du, my partner in all things, who has been by my side for almost the whole time in graduate school.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Modern life increasingly relies on complex products that perform a variety of functions. The key difficulty of creating such products lies not in the manufacturing process, but in the design process. However, design problems are typically driven by multiple contradictory objectives and different stakeholders, have no obvious stopping criteria, and frequently prevent construction of prototypes or experiments. Such ill-defined, or “wicked” problems cannot be “solved” in the traditional sense with optimization methods. Instead, modern design techniques are focused on generating knowledge about the alternative solutions in the design space.

In order to facilitate such knowledge generation, in this dissertation I develop the “Systems Physics” framework that treats the emergent structures within the design space as physical objects that interact via quantifiable forces. Mathematically, Systems Physics is based on maximal entropy statistical mechanics, which allows both drawing conceptual analogies between design problems and collective phenomena and performing numerical calculations to gain quantitative understanding. Systems Physics operates via a Model–Compute–Learn loop, with each step refining our thinking of design problems.

I demonstrate the capabilities of Systems Physics in two very distinct case studies: Naval Engineering and self-assembly. For the Naval Engineering case, I focus on an established problem of arranging shipboard systems within the available hull space. I demonstrate the essential trade-off between minimizing the routing cost and maximizing the design flexibility, which can lead to abrupt phase transitions. I show how the design space can break into several locally optimal architecture classes that have very different robustness to external couplings. I illustrate how the topology of the shipboard functional network enters a tight interplay with the spatial constraints on placement. For the self-assembly problem, I show that the topology of self-assembled structures can be reliably encoded in the properties of the building blocks so that the structure and the blocks can be jointly designed.

The work presented here provides both conceptual and quantitative advancements. In order to properly port the language and the formalism of statistical mechanics to the design domain, I critically re-examine such foundational ideas as system–bath coupling, coarse graining, particle distinguishability, and direct and emergent interactions. I show that the design space can be packed into a special information structure, a tensor network, which allows seamless transition from graphical visualization to sophisticated numerical calculations.

This dissertation provides the first quantitative treatment of the design problem that is not reduced to the narrow goals of mathematical optimization. Using statistical mechanics perspective allows me to move beyond the dichotomy of “forward” and “inverse” design and frame design as a knowledge generation process instead. Such framing opens the way to further studies of the design space structures and the time- and path-dependent phenomena in design. The present work also benefits from, and contributes to the philosophical interpretations of statistical mechanics developed by the soft matter community in the past 20 years. The discussion goes far beyond physics and engages with literature from materials science, naval engineering, optimization problems, design theory, network theory, and economic complexity.

CHAPTER I

Introduction

1.1 What is design?

In our daily life, we are surrounded by a variety of products that perform various useful functions. These products are complex arrangements of different atoms on many scales, which very often we didn't arrange ourselves. Instead, this arrangement was done by a manufacturing process such as baking, cutting, casting, or printing. The process might have taken place at a factory on the other side of the planet, in which case our ability to use the product output also depends on the logistical network that delivered it to our hands. However, even before the product could be shipped or manufactured, it had to be conceived by someone, or *designed*.

Design is a curious inversion of science. In science, we attempt to take the role of impartial observers of a system who perform measurements and analyses of system's output in order to test theoretical hypotheses about system's behavior. The hypotheses of most generality that stand the test of empirical data get entrenched as scientific laws that are true regardless of our actions.[1] In contrast, design is all about intervention, about harnessing some principles of the natural world to serve our interests.

Design starts with a desire for some function embodied in a product. Such desire might be driven by an individual's idea, by the market demand, or an explicit brief from an employer or client.[2] At the outset, it is often unclear whether such a function is even physically possible; it would be nice to have a perpetual motion machine, but that is unfortunately impossible. Even if realizing the desire is physically possible, it is not immediately clear how exactly to do that. Design is the process of figuring out if and how the desire can be realized.

Of course, design exists not in vacuum but in a society, and is subject to political and economic drivers, availability of technology, and existence of knowledge and knowhow.[3] These constraints together limit the complexity of products that can be developed in a given geographical area and explain the existence of technology hubs.[4] For the most complex products, design is the most expensive part of the life cycle, far surpassing the cost of crude materials and manufacturing that become cheap with automation and economies of scale.[5, 6]

The design complexity of modern products is driven by their need to perform many functions. The device now commonly called just a "phone" can likely exchange information in several electromagnetic frequency bands, record and produce sounds, take photos and videos, display colored images, measure orientation and acceleration, scan fingerprints, and store enough energy to support all the previous functions. Creating such a product is a task of systems integration: not only each function needs to have adequate performance, but the different functions need to smoothly communicate with one another, and the modules that perform them need to fit within the available space. Each of the modules is frequently designed by a separate human designer or team, thus requiring further negotiation of the product architecture.

As a result of these complications, design is not a noun but a verb. The initial idea does not become the final product. Instead, the initial idea spurs an investigation of options, clarification of goals and constraints, integration of multiple functionalities. The output of this process is not a singular solution, but the whole trove of created knowledge. In order to make this characterization of design more specific, I would adopt the following operational definition of design from Ref. [7], itself distilled from preceding studies:

Design is the ***act*** *of generating* ***knowledge*** *for* ***decision-making*** *through* ***time****.*

The bold emphases allow me to both highlight the key tenets of the definition, and later to draw analogies with the natural science investigations. The design process consists of *actions*, or deliberate efforts by the designers to conduct analysis, modeling, or computation. The output of these efforts provides us with some *knowledge*, or organized information about the designed system. In order to narrow down the range of considered solutions for the system, we need to *make decisions* about certain elements, after which the elements are not considered variable anymore. Finally, the investigation proceeds throughout *time*, since at any moment we can only operate based on the knowledge and decisions preceding the moment.

This definition recognizes design as a study area in its own right, beyond the design of any specific product. Design is understood to be an essentially "wicked" problem, with associated difficulties discussed below. The wickedness can be mitigated by various approaches that recognize the inherent complex and collective nature of the problem. In this dissertation I draw an analogy between design problems and physical phenomena, specifically those of statistical physics. For one of my case studies, Naval Engineering, the analogy is quite nontrivial and needs to be developed gradually. The other case study, self-assembly, is already known as a physical phenomenon, but a lot of questions of self-assembly design remain open. For both case studies, I highlight to what degree the design phenomena can be represented in quantitative terms, and develop dedicated mathematical tools to analyze them. This investigation both shows novel physical phenomena in design spaces, and gives general recommendations for design analysis.

1.2 Central problems

The discussion of existing design practices allows me to formulate a list of central problems to address in this dissertation.

First and foremost, I need to formulate a consistent mathematical framework for design problems that is alternative to optimization. If we don't focus exclusively on the absolute minimum of the objective function landscape, then how do we navigate this landscape? I show that this can be done in statistical physics terms that reduce to well-defined computations.

Second, I need to devise a way to automate some of the computations by introducing the appropriate information structures. In the earlier chapters IV-V a lot of computations can be performed manually or with *ad hoc* methods, but for the later chapters VI-VII the mathematical expressions become too bulky and require invention of new types of graphical and analytical notation.

Third, I aim to investigate the trade-offs in pursuing different design objectives simultaneously. A discovered trade-off informs us that achieving two or more outcomes simultaneously is impossible and therefore reveals a choice between them. In case of Naval Engineering, one important and incompletely understood trade-off is between the cost of a solution and its flexibility. In case of self-assembly, the off-target binding strength and bending entropies puts significant limits on the size of structures that can be encoded and assembled with high yields.

Fourth, I want to find out whether the design spaces of my case studies contain phase transitions, or large-scale reorganizations of the design ensemble. Phase transitions are typically associated with large fluctuations of the optimal solution and thus high uncertainty of design, but can belong to different classes.

Fifth, I aim to focus at intermediate-scale structures in the design ensemble and study their robustness to the changes in the problem statement. Having quantitative robustness metrics would allow a designer to identify aspects of the solution that remain stable as the problem changes.

Sixth, I am interested in integrating the design degrees of freedom of qualitatively different nature. Within the Naval Engineering case study, the ship layouts have to observe both spatial and topological constraints that result in nontrivial couplings and require special accounting techniques and information structures. Within the self-assembly case study, structures can be described at different resolution level via their topology, linear size, or particular element sequences, and efficient design should be able to control high-level topology via low-level binding energies.

Seventh, I need to relate the design variables that are most convenient for theory to those measurable in experiments or numerical simulations. While the sheer scale of ship design problems practically precludes experimentation, the state-of-the-art of self-assembly experiments is constantly improving. Serving the interest of these possible experiments would require discussion of which variables are actually control knobs available to an experimentalist, and which are theoretical constructs.

1.3 Overview of the dissertation

This dissertation presents a theoretical study of design problems with both analytical and numerical means, and addresses two case studies: Naval Engineering and self-assembly.

The central theoretical contribution of the dissertation is the Systems Physics framework that treats emergent structures in design ensemble as physical objects that interact via measurable forces. Systems Physics is based on statistical physics ideas and generalizes and reinterprets many statistical physics quantities. The central focus of Systems Physics is the behavior of the design ensemble, as opposed to a particular "optimal" solution. In this way, the framework provides a set of tools to realize the Set-Based Design as a design method by executing the Model–Compute–Learn loop.

Chapter II develops the qualitative background on the nature of the design problem. I argue that design problems cannot be solved with optimization techniques and provide an alternative framework: Systems Physics. In order to justify this claim, I review the "wicked problem" nature of design and the ways to manage this wickedness that designers have come up with. I argue that design shares essential features with physics of collective phenomena, especially well illustrated by design of self-assembling systems. I develop the qualitative structure of Systems Physics in form of the Model–Compute–Learn loop and explain the role of each step in that loop.

Chapter III provides a background on the mathematical methods used in the dissertation. I review the key tenets of statistical physics from both the Maximal Entropy and thermal bath perspectives, as well their interpretation in design terms and the computation of design outcomes as thermodynamic observables. I introduce tensor networks, a tool that recently became popular across many fields, and show how they can serve as information structures for problems where spatial and topological effects interplay, and for statistical mechanics more broadly. I discuss several conceptual questions that arise in the classical statistical mechanics of self-assembly problems, show how these questions are persistently not addressed in recent textbooks, and lay out a research program towards a consistent classical theory.

Chapter IV introduces the foundations of the Systems Physics approach and demonstrates it on an example arrangement problem. I show that small changes in the design objectives can result in large-scale phase transitions within the solution space. When two different design objectives apply simultaneously, they give rise to a more complex phase diagram with a variety of design phases. This chapter also introduces the notions of design stress and strain that drive the design solutions at the intermediate scale.

Chapter V studies the patterns of design stress and strain further. Instead of focusing on whole-ensemble behaviors, I instead identify discrete architecture classes that demonstrate specific design stress–strain responses. I compare these stress–strain relationships to those found in materials physics. These comparisons show that robustness is not a one-dimensional, but a two-dimensional quantity. I provide the diagrams of this two-factor robustness for several design objectives.

Chapter VI focuses on the interplay of two types of ship architecture: Logical and Physical. The Logical Architecture is represented by the topology of functional connections among shipboard system functional units, and the Physical Architecture is represented by the spatial location of those units. I show that the whole design space is efficiently encoded in tensor networks that serve as information structures. Specific queries to those information structures can be both shown graphically and performed numerically as tensor network operations. These operations reveal a variety of emergent couplings between the two Architectures in the design process.

Chapter VII switches to the self-assembly case study and builds a framework for encoding the target structures in specific interaction matrices. I study self-assembly of two classes of structures of distinct topology: linear chains and rings. I show how the set of building blocks can be characterized and designed collectively by studying the spectra of interactions matrices. Via combinatorial tricks I count all possible self-assembled structures, derive the partition function, and compute yields of target structures. This work illustrates many fundamental trade-offs between interaction cross-talk and size and complexity of target structures. Additionally, it provides an attempt at constructing a classical statistical mechanical theory and gives extensive interpretations of statistical weights, partition functions, and divergences.

Chapter VIII summarizes the dissertation by detailing what we learned about each stage of the Model–Compute–Learn loop. I propose several extensions to the presented work, and introduce newly opened questions about quantitative studies of design.

CHAPTER II

Design is not an Optimization Problem

2.1 Wicked problems in design

[1]In Chapter I describe above the general traits of the design problem. But what kind of a problem is it? Is it addressable with quantitative techniques we have in physics and math? Is it similar to problems that occur elsewhere in human experience?

Design is an intrinsically complex task that is not fully specified at the outset of the design process. Initial investigations in the design process lead to the discovery of new information and that new information leads, in turn, to new questions that require subsequent investigation. Because of that, ship design is an engineering task that involves discovery, like science. To use the words of a former Secretary of Defense, it involves both known-unknowns and unknown-unknowns.[8] However, just as we don't rely on an optimization algorithm to generate new forms of scientific understanding, we should not expect optimization algorithms to design an entire system. Like science, design is fundamentally a richer form of investigation than can be captured in a completely, a priori, well-defined mathematical optimization algorithm.

Though complex system design is not a mathematical problem, it does fall

[1]This chapter is based on an upcoming paper coauthored with D.J. Singer and G. van Anders.

into a more general class of “Wicked Problems”.[9, 10] Wicked problems contrast with “tame”, or well-defined and well-behaved problems. A problem is classified as “wicked” by the presence of one or several of the following factors:

1. A wicked problem is never completely specified at the outset. The formulation gets continuously refined as a certain form of solution is considered and investigated.
2. A wicked problem is driven by multiple stakeholders, who often have competing goals and constraints. These goals and constraints might be political in nature, or economical, or technological.
3. A wicked problem does not have a natural stopping criterion, i.e. it is unclear when it has been solved correctly.
4. The scale of the problem often prevents conducting experiments and trial runs, or producing prototypes. This creates a pressure to understand solution as well as possible before implementing it in order to get it right on the first try.

The intrinsic wickedness of a design problem is compounded by the sheer physical scale of the designed object.[11] Multifunctional devices such as consumer electronics or home appliances might be manufactured in the thousands or millions of units. Even if it is expensive to produce the first trial unit, after the product design is finalized, it can be mass-produced reaching economies of scale. This is not the case for products that are important but only occur in small copy numbers. An example of such a product is a naval battleship, which would serve as an important case study further in the dissertation.[12] Typically no more than 2-3 ($\mathcal{O}(10^0)$) copies of a battleship would be produced, but unit cost reaches up to a billion dollars ($\mathcal{O}(10^9)$ USD).[13] This seemingly leaves no margin of error, but of course large-scale mistakes occur that can render the battleship unusable.[14]

The characterization of a problem as wicked has to do with several limitations of the agents trying to solve it – i.e. human beings.[2] Human beings are *finite* in their individual and collective capacity of processing information.[3] At the same time, the design problems are essentially *complex* in terms of scale, cascading effects, and feedback loops. The obtained solutions have different *normative* evaluation from the point of view of different stakeholders. The trio of finitude, complexity, and normativity limitations drives the wicked nature of design problems.[2]

Importantly, the above limitations apply to the *human* participants of the design process and seemingly cannot be fully lifted. However, humans are aided in design process by *inanimate entities*, such as general-purpose computing hardware and specialized software and computational methods. The limitations of these inanimate companions are continuously relaxed, first by Moore's law of the scale of a single chip,[15] more recently by the advent of massive parallel computing,[16] and in the near future perhaps by the availability of quantum computing.[17] However, relaxing the computational constraints just allows pushing the boundary of brute-force approaches to the wicked problems further, but would never eliminate the essential, human-limited wickedness.

If the wickedness is insurmountable, why at all can we achieve quantitative solutions to the wicked problems? The resolution lies in *taming* the problem, or effectively ignoring its essential complexity in order to reduce it to a mathematically tractable form (see Fig. 2.1).[10] The taming process itself is not a mathematically rigorous procedure since it relies on the human judgment. However, once a problem is tamed, it is amenable to any number of mathematical treatments, such as constraint satisfaction,[18] genetic algorithms,[19] mathematical optimization,[20] or heuristics such as simulated annealing.[21] The solution of the tame problem is

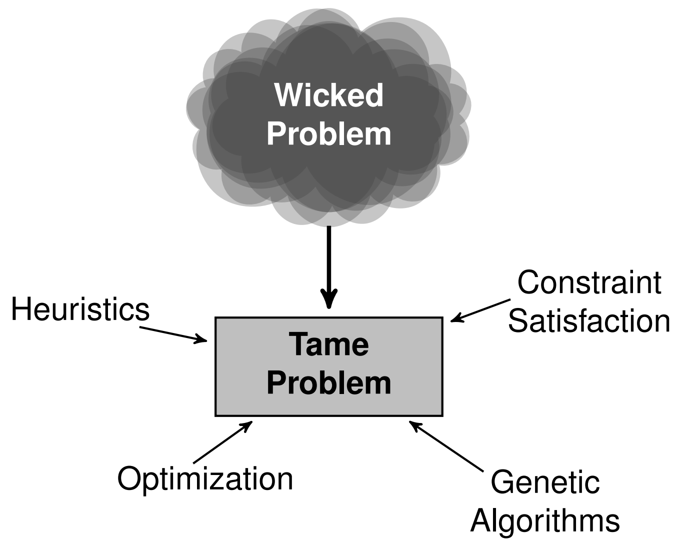


Figure 2.1: Schematic illustration of the taming process, in which a fuzzy and ill-defined wicked problem is represented as a clearly stated tame problem. The tame problem can then be addressed by any of the mathematical techniques available.

typically identified with the solution to the original wicked problem.

However, such an identification is problematic, since sone essential complexity of the wicked problem has been ignored. The goal of the design process was to create a product, not solve equations. The result of solving equations needs to be re-contextualized in the original wicked problem. In other words, we need to complement the taming process with its reverse and thus form a loop (see Fig. 2.2). The essential elements of this loop are *model* (taming), *compute* (solving the tame problem), and *learn* (relating the tame solution to the wicked problem). Upon each iteration of this loop, we gain some knowledge about the original problem that allows us to refine the model and go more prepared into the next loop.

This dissertation introduces a framework of "Systems Physics" that addresses all three stages of the loop (Fig. 2.3). In *modeling*, I focus on the quantitative components of the tame problem and vary them in order to explore the space of possible design problems. In *computing*, I provide qualitative concepts and develop

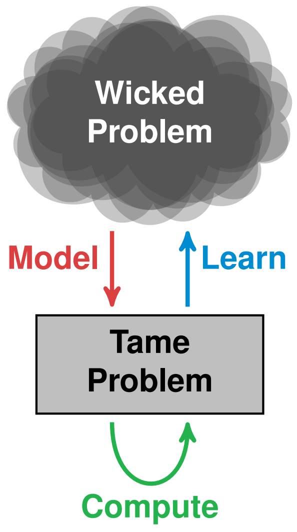


Figure 2.2: Schematic illustration of the loop design process. The fuzzy and ill-defined wicked problem is *modeled* as a tame problem, upon which *computation* can be performed. The results of the computation allow us to *learn* something about the wicked problem.

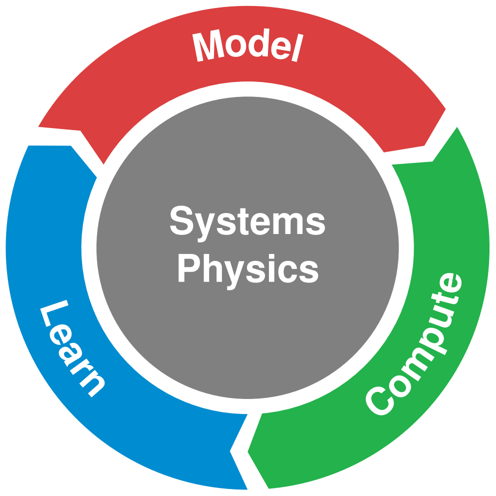


Figure 2.3: Systems Physics implements the loop design process through the stages of Model, Compute, and Learn.

analytical methods and software packages to extract quantitative conclusions for the tame problem. In *learning*, I show novel visualizations of the analysis that have not been used in industrial design problems before. Throughout the whole loop, I rely on physics-based causal modeling that aims to answer not just *what* solutions should be chosen, but *why* they emerge when the solution space and the design objectives interplay on multiple scales. While some of the techniques allow combing through extremely large design spaces quickly and efficiently, I want to emphasize that the novel parts of this dissertation cannot remove the essential wickedness of the design problem. All that I can aim for is making the design loop tighter and execute it more efficiently.

While we previously established design as having the opposite goals to science, the two rely on principally similar cognitive processes, such as exploration and discovery.[2] Since the cognitive processes are not too different, they are subject to the same human limitations of finiteness, complexity, and normativity. A single research paper may address a particular tame problem via theory, computation, or empirical data. However, science at large does not form a logically closed system; instead, it was argued by Popper to resemble an ecosystem of theories under the selective pressure of empirical data.[1] This view does not rely on posing fundamental laws, and instead refines the theory as the empirical base grows.

A researcher interested in addressing a big open science problem needs to first narrow down the scope to a feasible level, similar to a funnel. At the feasible level, analysis can be performed. The conclusions of the analysis need to be related back to the broader problem in a reverse funnel, so that the results are useful for a broader range of other researchers. On one side, this algorithm is remarkably similar to a single iteration of the wicked problem loop (Fig. 2.2). On the other side, it is very similar to the structure of effectively written research papers.[22] In this way, the practice of science on a large scale is quite similar to the practice of design. In other words, while the goals are somewhat different, both science and design belong to the class of wicked problems.

2.2 Design approaches

How are the wicked design problems addressed in practice? They are not a new phenomenon, and the design of complex systems is not a brand new challenge. The abstract study of design has long been a discipline in its own right. Of course, the way one would characterize design approaches depends on the design problem space and one's professional training. In order to illustrate how the problem space affects our discussion of design, this dissertation explores two different design spaces: that of Naval Engineering and that of self-assembly.

Naval Engineering is centered on design of naval ships: huge, complex, expensive objects with high impact per unit and high potential for loss of human life. The full life cycle of design, construction, and usage of a ship involves thousands of people and billions of dollars. Consequently, ship design is a long multi-stage process that can be described on multiple scales. To account for those scales, Ref. [23] proposed the following hierarchical taxonomy of Naval Engineering design practices:

1. Design Approach: the overarching principle of a design effort.
2. Design Process: a particular sequence of steps to implement the principle.
3. Design Method: the way in which design alternatives are created, analyzed, and selected.
4. Design Tools: the way to get information that supports design decisions.

This taxonomy allows us to more precisely trace the history of design studies, and more precisely position the innovation of the present dissertation.

The first consistently formulated design approach in Naval Engineering is the *Design Spiral*, formulated in the 1950s.[24] A Design Spiral prescribes considering different subsystems of the ship in a specific sequence; at each stage, the subsystem needs to be optimized to the limit of available information. After the sequence of subsystems finishes, it can restart for the next loop, building on the information generated earlier. After enough iterations the design is expected to converge on the optimal solution, though not guaranteed.[25] The converging trajectory looks like a spiral, hence the name. The design spiral model was originally developed at a similar time and under similar academic influences as the *waterfall model* of software development.[26] In both cases it is presumed that the necessary design steps and their sequence of actions are known *a priori*, and they just need to be executed.

A Design Spiral specifies both the design *approach* (local optimization) and the design *process* (order in which the design elements are considered). The most common *method* of such design is mathematical optimization.[20] The core idea of optimization is to select several design variables, introduce equality or inequality constraints that limit the range of those variables, and search for the global minimum of some cost function in the space of design variables. Importantly, the global minimum is implicitly identified with the final design solution; the output of the optimization algorithm is typically the end of the design process.

Both the broader design spiral approach and the specific mathematical optimization method focus on a single solution, the current vision of the product that is iterated upon. This is called the *product-centric* perspective, which is sufficient for simple cases. However, in design of more complex products issues start to crop up: the global optimum of the system might not coincide with local optima of separate subsystems;[27] backtracking any changes leads to expensive rework;[28] lastly, the integration of all systems might be completely impossible due to emergent knowledge failures.[29]

The failures of product-centric approaches can be mitigated in *knowledge-centric* approaches, such as Concurrent Engineering.[30] Concurrent Engineering prescribes pursuing several solutions in parallel from the outset. While there are several processes that implement it for different problems, a particularly important method is Set-Based Design (SBD).[31] Under SBD, each subsystem is developed by a team that considers a set of feasible solutions. Instead of focusing on promoting the good solutions, SBD focuses on eliminating the bad ones that are either strictly Pareto-dominated by others, or incompatible with other subsystems. Since the subsystems are typically coupled, the design teams need to periodically convene and negotiate the joint feasibility of their solutions; that is, search for the intersection of solution sets. This iterative negotiation produces knowledge about subsystem couplings, thus delaying the decisions on design elements and reducing the likelihood of integration failures.

SBD was arguably invented at Toyota, a major Japanese automobile manufacturer, in the 1980s, and reported in the seminal works by Liker et al.[32, 33] Significant evidence of SBD practices was found in a survey of aerospace industry.[34] SBD has also become an established method in the Naval Engineering community.[31] However, while the name of *Set-Based* Design suggests that the definition and accounting of solution sets should be at the focus, a recent meta-analysis suggests that the implementation of SBD is quite inconsistent.[35] We note that the inconsistency of implementation is not an issue of SBD *per se*; instead, it is an artifact of the design pipeline. SBD is a design *method* within the Concurrent Engineering *approach*, but for a complete design cycle it needs to be proceed according to a proper design *process* and using various *tools*. The tools are analysis methods, frequently embodied in software packages, that are the primary sources of *knowledge*.

However, generating knowledge in isolated pieces is not enough. For knowledge pieces to aid in taking design decisions, they need to be cognitively linked into a *knowledge structure*.[36] Such links are missing in many design software tools that act as databases of analysis results that can be filtered by the user to distill relationships.[37] Other approaches, such as the Knowledge-Action-Decision framework, focus on tracking the development of a knowledge structure throughout the design process to help in backtracking the reasons for design decisions.[7]. The Knowledge-Information framework focuses on organizing the knowledge structure to be conceptually robust: that is, to ensure that knowledge pieces obtained later does not invalidate those obtained earlier.[38]

One important knowledge structure recently introduced in Naval Engineering is a framework that distinguishes several kinds of ship architecture, such as Physical, Logical, and Operational.[39] The Physical Architecture accounts for the spatial location of ship's subsystems. The Logical Architecture describes the pattern of interconnections and interdependencies of the subsystems in form of networks. The Operational Architecture describes the temporal dynamics of ship systems during different routine or emergency operations. All of these architectures need to be considered in the design of a complete ship, but using such a framework allows decomposing the problem into parts and communicating about their interactions more efficiently.

The knowledge structures described above give an account of *what* solutions were selected and how they can be described from different perspectives. However, we need to know *why* certain solutions are preferred before they are selected, and before the objective landscape is even fully specified. In other words, we need to find a *leading indicator* of the solutions.[40] In this dissertation, I show how to compute such indicators in form of physical causal relationships. These relationships function as knowledge structures not previously used for design problems, but ubiquitous in physics: phase diagrams, stress–strain curves, and free energy landscapes to give a few examples. All of these are computed for large design spaces in accordance with the SBD *method* and rely on the computational *tools* of statistical physics. Still, as we shall see later, the physics-inspired analyses provide substantial commentary not only on the lower part of design practices, but also on the design *approach* and *process*.

Before we proceed to building the quantitative Systems Physics framework, we still need to explore the broader phenomenology of design problems, the view of design in self-assembly, and conceptual connections between design process and physics investigation.

2.3 Design as a collective phenomenon

The discussion so far mostly followed the terms used in the design community. However, there are significant analogies that can be drawn by someone trained in collective phenomena, statistical physics, and complex systems. In further chapters I build up these analogies quantitatively, but first they need to be introduced conceptually.

We start with introducing the notion of *solution space*, or the set of all feasible design solutions that we need to choose between. This set is spanned by the values of *design degrees of freedom*, or *variables*. For example, let a design solution α be fully described by specifying three variables $\alpha = \{a, b, c\}$. Let's assume that the variables are discrete and a can take 2 different values, b can take 10 different values, and c 6 different values. In this case the solution space consists of all possible $\{a, b, c\}$ combinations and thus has $2 \times 10 \times 6 = 120$ different options. This number is very conservative since we did not consider many variables or many options for each variable. For some problems it might be possible to compare 120 options by hand, but if the analysis of each option were more complicated, even 120 is a prohibitively large number.

In order to speed up the evaluation, we introduce a quantitative design objective $\mathcal{O}(\alpha)$. The same objective function would have been required for mathematical optimization, so this is not a requirement of specifically the SBD method. If we were following the design spiral approach, we could use one of the optimization tools to find the solution α out of the 120 options that gives the lowest objective function value.

Note that the objective function directly encodes the *interactions* between the degrees of freedom. A non-interacting objective function can be decomposed into a sum of independent functions of each variable:

$$\mathcal{O}(\alpha) = \mathcal{O}_1(a) + \mathcal{O}_2(b) + \mathcal{O}_3(c). \tag{2.1}$$

Note that varying the choice of the variable a does not affect the other terms, same for the other variables. We say that the solution space is then factorized into three separate parts. In this case the search for the minimum is much simpler: one merely needs to consider all options for each of the variables separately, for a total of $2 + 10 + 6 = 18 \ll 120$ options. This enumeration can be done much faster and, importantly, the options for each variable can be considered by different designers or different decoupled tools.

Unfortunately, all useful, interesting, and challenging design problems involve interacting design objectives and thus force us to deal with combinatorially large design spaces that do not factorize. I discuss the specific design degrees of freedom, solution spaces, and objective functions in much more detail in further chapters that develop quantitative design tools. While the design objective includes interaction, it is typically not dense, i.e. not all-to-all coupling. Instead, the interactions are usually pairwise, that is, the terms of the objective function include two variables at a time. The whole pattern of pairwise interactions frequently forms a complex network,[41] which becomes a major theme later.

Most commonly, the objective function is treated as a precursor to the optimization algorithm. The minimum of the objective function is identified with the desired solution, often somewhat uncritically.[20] This identification comes in tension with other needs of the design process. On one side, in order to turn a conceived design into a physical product, one needs to use a manufacturing process that always has finite tolerance. In other words, the manufactured product is never exactly the same as designed, and whether it still performs the intended function is an open question. It is possible to incorporate the finite tolerance into the optimization algorithm by searching not for a point solution but a ball of predefined size.[42, 43] These so-called robust optimization techniques account for manufacturing imperfections, but don't give us the crucial information about the objective landscape.

The objective landscape is the other side of the tension between design and optimization. The design of complex systems occurs in stages, and it is important to ensure that the early decisions do not over-constrain the later ones.[27, 11] To

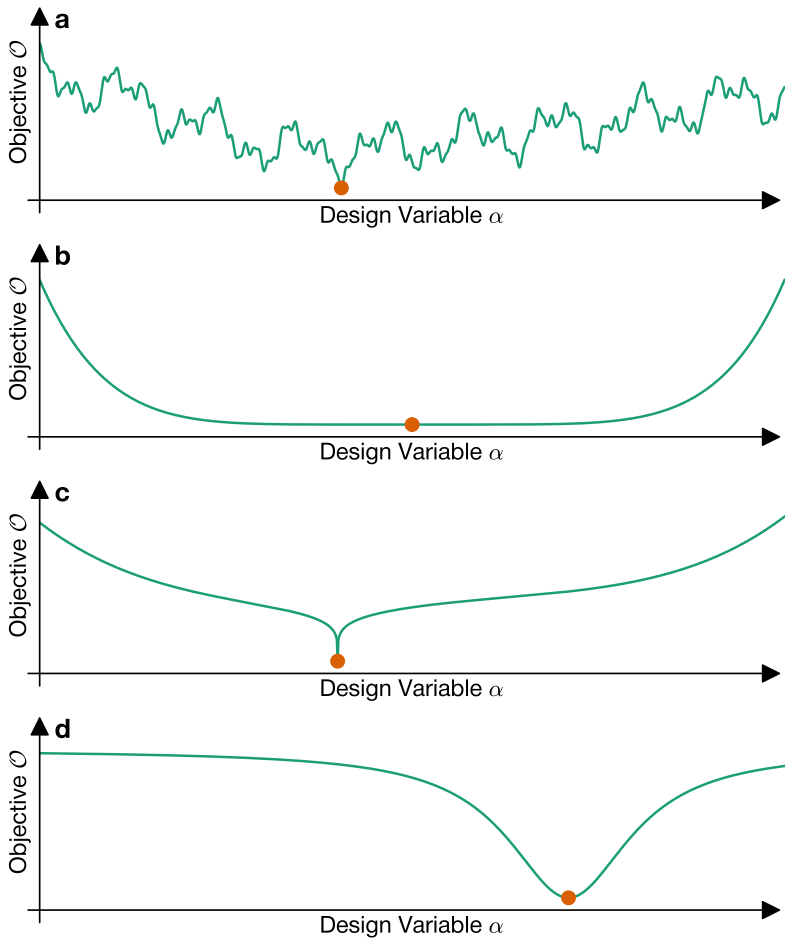


Figure 2.4: Qualitative types of objective function landscapes. Brown marker indicated the absolute minimum. (a) Rough landscape with many local minima. (b) Flat landscape with low variation of the objective around the minimum. (c) Smooth landscape with a single very narrow minimum. (d) Smooth landscape with a single finite-width minimum.

achieve that, we need to build in the need for flexibility and redundancy of design into our tools early on. In order to quantify this design freedom we need to look at the whole shape of objective landscapes and not only at their absolute minima.

Objective landscapes are widely discussed in the optimization literature in terms of how difficult it is to find the *global* minimum (or maximum).[44] The typical villains of these stories are *rough landscapes* that appear in over-constrained problems, with lots of *local* minima that make gradient descent optimization impractical. Rough landscapes routinely occur in physical systems such as structural and spin glasses,[45, 46] as well as problems such as the layout of integrated circuits.[21] This abundant discussion of rough landscapes might occlude the difficulties associated with other kinds of landscapes.

Fig. 2.4 illustrates several qualitatively different kinds of optimization landscapes. A rough landscape (panel a) has many local minima that compete with the single global minimum. A special optimization algorithm might be required to find the true minimum of this landscape, such as simulated annealing.[21] At the other extreme, the minimum of the flat landscape (panel b) is trivially easy to find. However, the minimum is barely different from the surrounding area, so choosing the solution at exactly the minimum is not particularly meaningful. The minimum can also be narrowly isolated within an otherwise smooth landscape (panel c). This minimum is also easy to locate, but it is so narrow that any manufacturing imperfections might lead to a radically different objective function value. By this logic, the ideal landscape is then a minimum that is fairly easy to find and has a reasonably wide basin of attraction (panel d). Across these four examples, it is already evident that we need some tools to characterize them.

The discussion of landscapes such as in Fig. 2.4 is seductive, but in certain ways quite oversimplified. The landscapes attempt to classify the optimal points of a complex system, but any such discussion needs to pay attention to typical caveats, such as the following:

1. A landscape might have multiple "well-behaved" minima of comparable value, and the way to decide between them is unclear.
2. Frequently design needs to follow several objectives at the same time, which can be represented as terms in the function $\mathcal{O}(\alpha)$. Depending on the balance of these terms, the locations and basins of the minima might shift, or completely

disappear or appear.

3. The designed systems have high-dimensional solution spaces, but Fig. 2.4 presents one-dimensional landscapes. The high-dimensional spaces have a lot of structure on multiple scales, which might be represented by a landscape in terms of some collective coordinates.

4. Since the shape of the landscape is dominated by the interactions between design variables, the network topology of such interactions becomes extremely important.

These properties are quite reminiscent of collective behaviors in complex systems: meta- or multi-stability,[47] criticality,[48] multiscale structure,[49] and complex network topology.[50] Such an analogy can be carried further in a quantitative fashion, but that requires abandoning the optimization perspective. The minima of the objective function would still play an important role, but the absolute minimum is not immediately taken as the correct design solution. In accordance with the knowledge-centric design paradigm, the design tools we develop would draw on the techniques of collective phenomena and would serve to provide information to inform the designer decisions, rather than supplant those decisions.

2.4 Design for self-assembly

In order to complement the intuitions of Naval Engineering design of large structures, we turn attention to something much smaller by linear scale: self-assembly. Self-assembly is the phenomenon of microscopic particles self-organizing from an initially disordered state without changing their identity.[51] It is driven by weak, non-covalent interactions that compete with thermal noise. The importance of thermal noise closely relates self-assembly to soft matter, or the study of systems large enough for quantum effects to be negligible, but small enough for thermal fluctuations to be still significant.[52] The soft matter umbrella accounts for many bulk materials we encounter on daily basis, such as gels, polymers, paints, and glasses, which are all disordered or semi-ordered.

Highly ordered structures are, however, unique to self-assembly. Self-assembly might be the only method that reliably fills the gap between nanometer scale assembly with chemical methods and millimeter scale assembly with conventional machinery.[51] A notable exception is, of course, the deterministic top-down photolithography process for manufacturing integrated circuits,[53] but it is strictly limited to planar, two-dimensional structures, whereas self-assembly is more flexible. Self-assembly can produce not only static, but also dynamic structures such as collectively acting robots.[54, 55]

The most common dynamic self-assembling system is any biological organism at sub-cellular level. Many cellular structures find their proper configurations by themselves, since top-down guidance is unavailable. Top-down guidance is not required to correctly produce even the most complicated macromolecular machine, the ribosome, which reliably self-assembles from its constitutive parts *in vitro*.[56] This remarkable robustness of biological self-assembly inspired studies of synthetic systems with life-like behaviors, such as self-replication,[57, 58] time keeping,[59] and formation of membranes.[60] Self-assembly studies are partially responsible of expanding the reputation of DNA from merely a genetic information repository in biological organisms to a programmable building block in its own right.[61]

Since self-assembly is a collective phenomenon in a system of classical (non-quantum) particles, it is naturally treated by classical statistical mechanics. Statistical mechanics was originally developed to account for collective behavior of atomic and molecular systems in which individual units are both governed by quantum mechanics and too small to be seen directly. However, we can directly observe micron-scale self-assembling building blocks with optical microscopes. This observation of individual particles raises direct theoretical questions about their distinguishability,[62] the role of solvent and particle masses,[63] and symmetry numbers of "colloidal molecules".[64] Self-assembling systems are efficient test beds for questions about less accessible atomic systems, such as the nature of solid-solid phase transitions [65], the nature of topological order long thought to be an exclusively quantum phenomenon,[66] and the distribution of entropy across the system.[67] More broadly, using self-assembling systems as a case study allows us to refine our understanding of statistical mechanics more broadly.

The fundamental advances in statistical mechanical theory,[63, 68] computer simulations,[69] and particle synthesis and imaging [70, 71] allow us to shift from merely understanding existing self-assembling systems to *designing* new ones. The design of atomic systems is limited by the discreteness of chemical elements, a direct consequence of their quantum nature.[72] However, quantum mechanics does not constrain larger building blocks, which then can have continuously adjustable parameters, such as shape and size,[73] interaction patterns,[74] and specific bindings.[75] The latter have already been implemented on both biologically-inspired platforms such as DNA and proteins,[76, 77] and synthetic platforms such as magnetic dipole patterning.[78] Together, specific interactions and particle geometry open up a wide *design space*, in which lots of self-assembly behaviors are possible.[79]

One avenue of designing self-assembling behaviors is focusing on the space of particle shapes. Particle shape is the dominant parameter that governs the behavior of hard particles that don't have any interaction apart from prohibition of overlap. With lack of energetic interactions, their collective behavior is fully driven by entropy maximization. However, maximizing entropy in dense systems generically causes the particles to form ordered structures.[80] The formed structures depend sensitively on particle shape and span crystals,[81] glasses,[82] liquid crystals,[83] and quasicrystals.[84] The effective entropic interactions governing these structures can not only be quantified, but directly designed.[73] Treating particle shape as a design parameter, one can formulate and solve the *inverse design* problem: find the shape that forms a target desired self-assembled structure.[72, 85] Similar arguments allow designing not only structural but functional behaviors, such as tunable photonic crystals.[86]

The other avenue of self-assembly design is via specific, heterogeneous binding interactions. These interactions can be tuned to form diverse crystal structures,[75] or control the transition between crystal and gas phases.[87] Introducing many species of particles with controlled interactions allows self-assembling finite structures of complex geometry, with dozens of particle types finding their proper place.[88] In practice, the yield of complete complex structures would be limited by multiple factors. An important limitation is the weak binding between building blocks that should not be interacting, known as cross-talk or off-target binding.[89] If the contrast between on-target and off-target binding is too low, heterogeneous building blocks collectively behave as homogeneous ones, limiting structural control.[90] The yield of heterogeneous self-assembly might be improved by manipulating the concentrations of different types of monomers.[91] Different concentration profiles also allow extracting several structures encoded in the interaction matrix.[92]

The studies of heterogeneous self-assembly mostly focused on dense colloidal clusters or bulk crystal structures. However, it might be possible to encode a wider range of structures distinguished by topology. What is the shape of the design space of the specifically interacting building blocks? To what degree can we control the topology of self-assembled structures via interaction energies and concentrations? How is this assembly limited by cross-talk? What kind of mathematical tools are required to account for these structures? How do we design complete sets of self-assembling building blocks with collective properties? Answering these questions would both inform important aspects of design at large and widen our control of self-assembly. Self-assembly offers the possibility of creating dynamic, evolving, adaptable, controllable and programmable materials, and the ultimate limits of what is possible are currently only imagined in science fiction.[93]

As we see, self-assembly is a burgeoning field that rests on an ever firmer theoretical foundation but still presents an abundance of open questions, both fundamental and practical. In the next section I show that both self-assembly and Naval Engineering design can be seen through the same lens of statistical mechanics theory.

2.5 Why is design physics?

2.5.1 Model–Compute–Learn loop

The overview of self-assembly research shows that a lot of important behaviors are explained by physics theory, especially statistical physics. But why is the design process itself physical? And how can it encompass the case studies as disparate as self-assembly and Naval Engineering? To answer this question, we return to our previous definition of design, with the same key terms highlighted:

*Design is the **act** of generating **knowledge** for **decision-making** through **time**.*

I would argue that the highlighted terms are directly comparable to concepts in physics. The *act* is a deliberate computation performed by a physicist or a designer in the interests of learning something about the system. The outcome of that computation contributes to our *knowledge* of the system by clarifying relationships between its parts. We can act upon that knowledge by making *decisions* that narrow the design space down to its preferable part; this can be done for instance by freezing certain design variables at desired values. Lastly, the *time* concept is key in study of any dynamical or non-equilibrium systems.

In many collective physical systems time runs forward; that is, the dynamics of the system are irreversible.[94] In design we proceed irreversibly from states of low knowledge and high freedom to states with larger amount of knowledge and less freedom due to the decisions made. The specific trajectory along which design knowledge and freedom evolve depends significantly on the chosen design process.[7] The design process might lead to a well-constrained and well-understood design, or an ill-specified and poorly characterized design, also known as a design failure.[29]

These variability of design outcomes has to do with the nature of design as a wicked problem. As I argued previously, to make any progress with a wicked problem, we inevitably need to turn it into a tame problem. Of course, a tame problem is not selected once and for all. In order to understand the wicked problem via quantitative analysis, we need to turn it into a tame one many times over. More formally, within the scope of Systems Physics, we need to perform the three-stage loop of Model–Compute–Learn. Each stage of this loop serves as a milestone that asks questions about the designed system. All of these questions can be asked (and often answered) in a uniquely physical way.

2.5.2 Model: the spaces of design

Intuitively, design investigations should care about the connection between our design decisions and their results. There are many decisions that could be made and many results that could come out, both of those form *spaces*. However, I want to set up two important dichotomies in the characterization of these spaces.

The first dichotomy is between what I would call the *solution space* and the *design space*. The solution space is the set of states in which the designed system can be (“state space” or “phase space” in statistical physics language [62]). The design space is the set of states among which we, external designers, can make our choices. The difference between the two spaces is akin to the difference between tame and wicked problems. The solution space is defined exactly and fully; picking the solution space is an essential modeling choice in moving from a wicked problem to a tame one. In contrast, the shape of design space is often not known from the outset. We start with an idea of a few design options, but as the design process proceeds, we understand the space of design choices better and better.

The difference between the solution and design spaces is best illustrated for our case studies. In the case of Naval Engineering, we are going to be primarily concerned with arrangement problems that ask questions about locations of functional units and connections. Our solution space consists of all possible unit locations; at the same time, our decisions are also directly concerned with unit locations. For this case study, the design space and the solution space essentially coincide.

The picture is different for case of self-assembly. The solution space spans all possible spatial configurations and binding combinations of building blocks. The building blocks explore the solution space driven by thermal fluctuations. Because of fluctuations, we don’t have direct control over the exact position of each building block as in the case of *directed* assembly. In *self*-assembly we control the specific interactions and concentrations of the building blocks, which in turn bias them to preferentially bind into the structures we desire. The self-assembly design space is thus much smaller than the solution space.

The second dichotomy is between the *objective space* and the *outcome space*. A design objective is what drives us to choose one point in solution space over another. Typically, for every solution α we would define one or several objective functions $\mathcal{O}(\alpha)$, which in statistical mechanics directly affect the probability of choosing that solution. In contrast, the design outcome is a quantitative metric of the resulting design, typically an average over the solution space. The distinction of objectives and outcomes again has to do with the wicked and tame aspects of the problem: while we pick the objective exactly, finding the right outcome metrics requires more exploration.

It might be counter-intuitive why the outcome metric is not exactly the same as the objective. After all, isn't the point of design to reach the stated objective? As I argued above, the point of design is instead to generate knowledge. It might be informative to consider different outcome metrics to find whether the pursuit of the stated objectives also leads to unintended side consequences. The difference between outcomes will become apparent across the Naval Engineering studies in the dissertation.

The design objective of self-assembly is even more restricted. The probability of observing different configurations of building blocks is directly driven by the energy of such configurations. While we can manipulate the energy landscape, we cannot change the selection factor from energy to something else. At the same time, the interesting outcome of self-assembly is not the minimal possible energy, but maximal possible yield of the target structure. There are still several possible ways to define the “yield” that have different merits, discussed in more detail in the corresponding chapter.

With these dichotomies of solution vs design space and objective vs outcome space, the Model stage of the Systems Physics loop becomes more clear. In order to move from the original wicked problem to a well-defined tame problem, we need to make choices about each of these spaces. What objectives are we pursuing and how much do we care about each of them? What is the resolution of considered spatial arrangements? What is the shape and size of the box in which all functional units or building blocks are to be placed? What constraints do we have on the on-target and off-target binding? These are the kinds of questions that need to be answered to get a specific formulation of a tame problem. We then need to discuss how to compute something useful about this problem, and how to learn from the results of the computations.

2.5.3 Compute: forward, inverse, and in between

In this “physical” picture, in order to connect the design space with the outcome space, we need to account for all of solutions appropriately weighted by the objectives. I will flesh out this connection mathematically in specific case studies, but here it can be qualitatively illustrated by quoting self-assembly examples. A traditional way, which we would call *forward design*, is asking what outcomes correspond to a given point of design space. For instance, what structure (outcome) would be self-assembled by hard particles of a given shape (a point in the design space)?

Answering this question requires special computational techniques, but is possible on a case-by-case basis. The solution space has to be set *a priori*: the set of all positions and orientations of particles in a 3D box. The design objective is also set *a priori* in form of an algorithm that detects the overlap of particles and prohibits all configurations with such an overlap (e.g. Ref [95]). The study by Damasceno et al. picked heuristically a set of shapes out of the design space of all possible shapes, and showed that these shapes can self-assemble a wide variety of structures, including very exotic ones.[81] This result is significant and gives us a lot of knowledge about the shape–structure relationship, even though neither the design space of all shapes nor the outcome space of all structures were mapped out before running the algorithm.

If one wants instead to assemble a particular structure, it might be impractical to keep guessing the proper shape and running the forward design computation repeatedly. Instead, a better way is to turn the shape–structure relationship around and perform *inverse design*. For hard particle self-assembly such method was provided in form of the "digital alchemy" method, which allows for smooth changes not only in the solution space of particle positions and orientations, but also in the space of particle shapes.[72, 85] This method also provides knowledge about the shape–structure relationship, but inverts the relationship of the known and the unknown.

Forward and inverse design are thus two complementary perspectives on the relationship between design and outcome spaces. While one of them might be more useful in solving a given practical (tame) problem, neither is inherently superior to the other one in understanding the whole wicked problem. For most cases of interest, neither forward nor reverse map can be performed analytically, but rather requires expensive computations. As computations get cheaper and algorithms for a particular problem improve, more and more parameter–outcome pairs can be reported per study, from one,[84] to a hundred,[81] to a two-parameter shape family,[96] to millions of shapes in a hundred-dimensional space.[85] Each such study expands the boundaries of the charted region of either design or outcome space.

However, knowledge about a design problem does not boil down to the correspondence of points in the design and outcome spaces (essentially the *what* question). A typical surprising phenomenon in this mapping is a small change in the design leading to a large change in the outcome. For example, in context of self-assembly, the particles of slightly different shape might assemble into a very different crystal structure [65], or not assemble at all.[82] To understand *why* this happens, and whether such behavior occurs elsewhere, a more detailed investigation is required.

In order to understand the reasons for strange collective behaviors, one needs to look at an intermediate scale structure of the solution space, between the individual points and the average over the whole space.[73] In statistical physics, such intermediate scale structure can be accessed via lumping together many solutions, known as *coarse-graining.* Coarse-graining allows characterizing the groups of solutions, specific motifs, or mesoscale design solutions with an effective design objective, known as *free energy.* The patterns of free energy give an intuitive understanding of the intermediate-scale precursors of large-scale changes in behavior. Both the mathematical description of free energy and many examples of its usage are found further in the dissertation. In Naval Engineering I group together solutions that share the same design feature, while in self-assembly I group together structures that share the same length or the same topology, in a hierarchical fashion.

The computations of free energy and the averages across the whole solution space rely on a set of mathematical techniques described elsewhere. However, in those computations we will encounter specific pieces that encode large amounts of information in a structured form, often mirroring the statement of the design problem. All across statistical physics, the partition function encodes information about conjugate averages over the solution space, but the partition function is not the only useful object. In condensed matter physics, the object of central attention is the correlation function which contains a lot of signatures of system's behavior, phase transitions, and universality.[97] Similarly, in this dissertation tensor networks will play a similar role, and manipulations with tensor networks will help us generate knowledge about the intermediate scale behaviors in the system.

I would refer to such pieces of computations as *information structures.* Information structures technically contain answers about all, or at least a majority, of interesting system behaviors. However, extracting a specific answer requires constructing a specific "query" to the information structure, such as a specific derivative of the partition function, or a modification of tensor network topology. These queries are the deliberate acts mentioned in the definition of design, and it is precisely such acts of the designer that convert raw information into knowledge.

2.5.4 Learn: physics knowledge structures

Information structures store extensive descriptions of both forward and inverse relationships in the designed system that can be extracted via computations. How do the results of these computations turn into useful knowledge? These results form a set of related facts, i.e. the knowledge structure of the behaviors of the system. In Naval Engineering-driven studies, such structures have been represented in the language of networks that record the results of analyses in form of relations of parameters and outcomes.[7, 38] Understanding the parameter–outcome relationship for the broadest range is a goal of physical science. In studies of nonlinear dynamical systems, the goal is not in precisely computing the trajectory, but delineating the regimes dominated by attracting fixed points, limit cycles, or chaos.[98] In statistical physics, the goal is often in producing a phase diagram of the possible macroscopic regimes.[62] These phase or bifurcation diagrams do not have the character of foundational, broadly valid physical laws. Instead, they describe a specific system; they are the shape of an answer. I present such answers in each of the case studies of the dissertation, most frequently with graphical diagrams.

Phase diagrams presenting the macroscopic relationships between design space and outcome space. Knowledge of this relationship answers the *what* questions of design. In order to answer the *why* questions, I use the information structures to analyze the free energy landscapes. If free energy is the effective potential energy for the system, then its gradient, which I call *design stress*, is the effective force. The idea of force as a cause for motion, as the “why” of motion, is one of the first ideas taught in intro physics classes. Per Newton’s laws, to determine whether and how a body would move one needs to add up all forces of different origin that drive this body. I show that the same intuition of competing forces, even if the forces have uncommon origins, answers why certain design solutions are preferable over others.

The understanding about *what* and *why* solutions are preferred under a given formulation of the tame problem allows us to return and revise this formulation, i.e. the Model part of the loop. The design space and outcome space maps are expanded precisely by executing the Learn part. In learning, we discover the unexpected outcomes, or new design forms that fulfill the objective. The new iteration of the loop, starting with Model, is possible precisely because the loop is completed and the last stages feeds into the first.

The discussion of the Model–Compute–Learn loop highlights what we have to assume and what we get to learn in modeling the wicked problems as traceable tame problems. While the solution and objective spaces are parts of the tame problem and have to be precisely assumed *a priori*, the design and outcome spaces are part of the wicked problem and can be gradually mapped out by executing the loop multiple times. The forward and inverse design paradigms are then two sides of the same coin as they inform the same parameter–outcome relationship. The reasons for large-scale shifts in this relationship are typically hidden in the intermediate-scale behaviors that are all accessible to the statistical physics methods.

In order to show the execution of this loop in example problems and address the main questions of the dissertation, we need to first convert the qualitative discussion above into quantitative form in the language of statistical mechanics.

CHAPTER III

Methods

3.1 Statistical Mechanics

In this section I review the central tenets of statistical mechanics from two axiomatic perspectives, based on maximal entropy and bath exchange. I show how to compute generic and conjugate observables within the ensemble. I introduce the concept of coarse graining and the associated computations and interpretations. Further in the chapter I introduce tensor networks as an information structure for statistical mechanics and discuss their conceptual and computational properties, as well as the associated software. Finally, I discuss certain issues of classical statistical mechanics that are relevant for self-assembly studies but are not adequately covered by existing textbooks, and lay out a study program to formulate a fully consistent classical statistical mechanical theory.

3.1.1 Maximal entropy perspective

Here I review the maximal entropy perspective on statistical mechanics. The maximal entropy principle was first proposed to regularize discussion and inference in large systems under very sparse measurements.[99] We consider an ensemble of many mutually exclusive system microstates that can be indexed with a variable α. The number of microstates has to be large, typically exponential in the number of particles in the system. We can pick a random state from the ensemble with a probability distribution p_α, but *a priori* we don't know anything about the properties of this probability distribution apart from it being normalized $\sum_\alpha p_\alpha = 1$.

The inference problem is stated as following: given a few measured observables $\langle \mathcal{O}_i \rangle$, predict the average value of other observables $\langle X_j \rangle$. Making this prediction requires learning something about the underlying probability distribution p_α. We can attempt to write down the equations that constrain the probability distribution by the known observables:

$$\langle \mathcal{O}_i \rangle = \sum_\alpha p_\alpha \mathcal{O}_i(\alpha), \quad \forall i. \tag{3.1}$$

However, the number of observables is typically much smaller than the number of microstates, so this system of equations is severely underconstrained. In order to make it well-behaved, one needs to add a regularizing principle, some global constraint that would make the probabilities well-defined, but still subject to the constraints (3.1).

As evident from the name, we choose the principle of Maximal Entropy. The "entropy" refers, of course, to Shannon's entropy,[100] which is a measure of uncertainty of a probability distribution in the ensemble that fulfills certain desirable axioms, such as non-negativity, extensivity, and additivity. Shannon showed that there is a unique functional form that satisfies these properties:

$$S = -\sum_\alpha p_\alpha \ln p_\alpha. \tag{3.2}$$

With this principle, we can formulate the inference task as finding a probability distribution that maximizes the Shannon entropy (3.2) while satisfying the observational constraint (3.1). In absence of any constraints, Shannon entropy is maximized by the uniform distribution, which is not biased to any particular state. In the presence of constraints, the distribution would be only biased by the observables that we put in by hand. In this way, the maximal-entropy probability distribution is the one least biased by any undesirable influences. Of course, maximal-entropy probability distributions occasionally display mathematically pathological behaviors in certain parameter regimes.[101] However, such behaviors are often model-specific and thus should be discussed together with the corresponding models.

Let's derive the maximal-entropy probability distribution. We shall enforce each of the constraints (3.1) with a corresponding Lagrange multiplier λ_i, thus getting the bulky functional:

$$S[p_\alpha] = -\sum_\alpha p_\alpha \ln p_\alpha - \sum_i \lambda_i \left(\sum_\alpha p_\alpha \mathcal{O}_i(\alpha) - \langle \mathcal{O}_i \rangle \right) - \gamma \left(\sum_\alpha p_\alpha - 1 \right), \tag{3.3}$$

where we chose to write the Lagrange multipliers with a minus sign for convention. The Lagrange multiplier γ is necessary to ensure that the probability distribution is normalized, i.e. we are guaranteed to randomly pick one of the microstates.

Maximizing this functional is straightforward via functional differentiation:

$$\frac{\delta S}{\delta p_{\alpha'}} = 0 \quad \Rightarrow \quad p_\alpha = \frac{1}{\mathcal{Z}} e^{-\sum\limits_i \lambda_i \mathcal{O}_i(\alpha)}. \tag{3.4}$$

This form reveals the optimal probability distribution to be a nearly *local* transform of the observables $\mathcal{O}_i$, i.e. only one state α is present. Because of this locality, we got the probability expression without making any explicit reference to model parameters, such as the ensemble of the possible states α or the functional form of the observable functions $\mathcal{O}$. The prefactor $\mathcal{Z}$ is the normalization condition that is computed as sum of all exponential factors:

$$\mathcal{Z} \equiv \sum_\alpha e^{-\sum\limits_i \lambda_i \mathcal{O}_i(\alpha)}, \tag{3.5}$$

and is known as the *partition function.* The partition function turns out to play a central role in statistical mechanics, but we will discuss that below. The name partition *function* implies that it is a function of some arguments, even if we suppress them in notation, and we will address that below too.

3.1.2 Statistical mechanics and statistical physics

Jaynes used the above maximal-entropy argument to argue about the difference between statistical *mechanics* and statistical *physics.*[99] His view, now also known as subjective statistical mechanics, or statistical inference, is one of multiple ways to derive quantitatively the same form of expressions. The expressions (3.4)-(3.5) are the centerpiece of *statistical mechanics*, which according to Jaynes is nothing else than glorified accounting of probabilities and a set of mathematical tricks, agnostic of what the probabilities represent. This agnosticism is what makes statistical mechanics a generic tool, which is valid whenever the maximal entropy assumptions are justified. In order to turn it into *statistical physics*, we need to imbue the concepts of $\alpha, \mathcal{O}, \lambda$ with physical meaning and interpretation. This freedom of interpretation is what would allow us to turn statistical mechanics either into Systems Physics that describes the arrangement problems in design (Chapters IV-VI), or more conventional statistical physics of self-assembly (Chapter VII).

Let's start with the more conventional interpretation. In this case, the microstates α refer to particular configurations of particles. Each particle has a certain position, orientation, momentum, and angular momentum which can be specified independently of other particles. Depending on these variables, we can compute the energy E of the system. The function that evaluates energy of any particular state is called the system's *Hamiltonian*:

$$E = \mathcal{H}(\alpha), \tag{3.6}$$

which is the first example of a particular observable $\mathcal{O}$. Note that here we keep all of the variables of individual particles bundled up inside the Hamiltonian to preserve its maximally generic form. Typically, energy breaks up into the kinetic energy dependent only on momenta and angular momenta, and potential energy dependent only on particle positions and orientations. The potential energy usually accounts for the interactions between the particles, such as prohibiting them to overlap. Since the momenta describe movement of particles in space, it is sometimes possible to connect the statistical description of the system with a kinetic one.[102]

The Lagrange multiplier λ associated with the Hamiltonian is typically identified with the *inverse temperature* $\beta = 1/T$. With these parameterizations, the probability takes the familiar Boltzmann form:

$$p_\alpha = \frac{1}{\mathcal{Z}} e^{-\beta \mathcal{H}(\alpha)}. \tag{3.7}$$

Typically the Hamiltonian $\mathcal{H}$ describes the inner interactions within the system, while the inverse temperature β describes the system's coupling with an external bath. I discuss this interpretation below. However, from the perspective of statistical mechanics, the value of β specifies how strongly the probability distribution is driven by temperature. At low T (high β) this drive is strong and the microstates with lowest energy have significantly higher probability than ones with high energy. On the contrary, at high T (low β) the drive gets weaker and the probability is distributed more uniformly. In the limit $T \to \infty$ ($\beta \to 0$), the probability distribution becomes fully uniform and all microstates are equivalent.

Let's look at the contrary case of design. The microstates α are the considered design solutions, such as spatial arrangements of some functional units. Units can have different orientations, but we typically don't track their movement. In this way, the ensemble description is still statistical but not kinetic.

The observables $\mathcal{O}_i$ are identified with *design objectives* and the Lagrange multipliers λ_i with *design pressures.* The relationship between them is analogous to the Hamiltonian and inverse temperature. The higher the design pressure, the more the corresponding design objective drives the probability distribution. It is tempting to make comparisons between the design pressures, such as $\lambda_1 \overset{?}{>} \lambda_2$. However, such comparisons cannot be made due to the difference of measurement units. All design objectives are quantitative properties of design solutions, but they don't necessarily have the same units. One design objective might be measured in dollars, another in seconds, and the third in miles per hour, which are all valid units of measurement. The corresponding design pressures need to have the inverse dimensionality so that the product $\lambda_i \mathcal{O}_i$ is dimensionless. In this way, we can talk about the design pressure space where each axis has different units.

It might be confusing that we maintain many design objectives $\mathcal{O}_i$ for the design case, but only one Hamiltonian $\mathcal{H}$ (without an index) for the conventional case. However, this has to do with the physics convention. In statistical physics systems, the probability of different states is usually driven by a Hamiltonian that is a combination of charges and fields:

$$\mathcal{H}(\alpha) = \phi_1 Q_1(\alpha) + \phi_2 Q_2(\alpha) + \phi_3 Q_3(\alpha) + \dots \tag{3.8}$$

Here, each of the *charges* Q_i (e.g. electric, magnetic, color) typically still depends on all of the design variables α, unlike the separable objective function in Eqn. 2.1. The parameters ϕ_i are the *fields* acting on each type of charge (e.g.

electric, magnetic, color) that determine the relative magnitude of each term in the Hamiltonian. The separation of the Hamiltonian terms is the same thing as separation of design objectives, up to the semantics of conversion between design pressures and fields $\lambda_i = \beta\phi_i$. I use both separate conventions for consistency with the different domains of literature within different chapters of this dissertation.

3.1.3 Extensive and intensive parameters

There is a certain sleight of hand committed in proceeding from the entropy functional (3.3) to the probability distribution (3.4). The entropy functional was formulated with the expected observable averages $\langle \mathcal{O}_i \rangle$ and auxiliary Lagrange multipliers λ_i. However, the expected observable averages disappeared from the probability expression, and λ_i were elevated to the important parameters.

In order to explain this discrepancy, we review the distinction of *extensive and intensive thermodynamic parameters*. Here the word "thermodynamic" is meant in the sense of applying to the whole ensemble, rather than some parts of it. The distinction of extensive and intensive parameters can be illustrated by system doubling.[102] Next to the system we place a copy of it and treat both as a single system of two decoupled parts. We then compare the different thermodynamic descriptors of the combined system. If the descriptor value doubled (like energy or volume), it is called extensive. If the descriptor value stayed the same (like temperature or pressure), it is called intensive.

This definition is strictly valid when the two identical systems are decoupled. It is valid for very large systems that only have short-ranged interactions on the boundary. For systems with long-range interactions different notions of extensivity are required.[103] In my Naval Engineering case studies, I always only consider finite size systems where finite size effects are typically important. In the self-assembly case study I only consider systems with short-range interactions which are extensive in the sense described above.

The parameters $\{\lambda_i, \mathcal{O}_i\}$ are known as a *conjugate pair*, and the index i refers to different pairs. The design pressure λ_i is an intensive parameter, the design objective $\mathcal{O}_i$ is an extensive parameter. Within each conjugate pair, we need to pick only one of the two parameters that we would use as an independent thermodynamic parameter. However, in each pair i we can make that choice independently. The choice needs to be made because the two parameters are related to each other via an equation:

$$\langle \mathcal{O}_i \rangle = \frac{\partial}{\partial \lambda_i} \ln \mathcal{Z} \bigg|_{\{\lambda_i\}}, \tag{3.9}$$

which can be solved for unknown values of λ_i. I derive this expression below in the section on observables. For now, note that it depends on the partition function $\mathcal{Z}$, which requires performing complicated summations that depend on the specifics of the system model. Note that the partition function depends on all of design pressures λ_i in a way that is not typically separable, so solving for their values is especially hard. Because of these difficulties, while the derivation of maximal entropy probability distribution requires introducing the constrained averages, we are not actually going to use them as real values.

On the other hand, working in terms of intensive parameters λ_i is typically simple. This is possible because using these parameters, we can write down the probability distribution (3.4) as a local function of the microstates α and design pressures λ_i. It appears that I just introduced the choice within the conjugate pair $\{\lambda_i, \mathcal{O}_i\}$ and immediately took that choice away by calling one of the options "too difficult". If we define all the design objectives manually but always choose to work in terms of intensive design pressures, are there any extensive parameters remaining?

We first need to make a distinction between fixing $\langle \mathcal{O}_i \rangle$ on average, as opposed to fixing $\mathcal{O}_i$ as an exact value. For large systems in thermodynamic limit there is typically no difference, as the mean and the median of $\mathcal{O}_i$ become arbitrarily close and the relative fluctuations in $\mathcal{O}_i$ become arbitrarily small. However, some aspects of the problem, such as the size of the box, are defined exactly, not up to fluctuation. Because of the convenience reasons outlined above, we describe fluctuating design objectives primarily via intensive design pressures λ_i.

However, explicitly fixed extensive parameters enter our calculations from a different side, not in the definition of design objectives $\mathcal{O}_i$ but in definition of design ensemble $\{\alpha\}$. Every time we consider some system within a finite box, the size of that box is an extensive parameter. Derivatives with respect to that parameter would give us the conjugate intensive parameters, or design pressures. For example, for normal thermal systems the volume derivative is the pressure:

$$P \equiv \frac{1}{\beta} \frac{\partial}{\partial V} \ln \mathcal{Z}. \tag{3.10}$$

In some cases we are interested not just in the cost of a global change of volume, but in the cost of a specific localized change of the box geometry. In that case, more specific derivatives or finite differences can be taken. A specific example would be discussed in context of space premium and void free energy in Chapter VI. It is typically easier to evaluate the cost of reduction of the available ensemble microstates than extension, since it is easier to deal with subsets than supersets.

3.1.4 Bath perspective

An alternative way to interpret the extensive and intensive thermodynamic parameters is via the notion of a bath, or a repository external to the system of interest with which an extensive parameter $\mathcal{O}$ can be exchanged at a unit cost of the intensive parameter λ.[62]

For instance, a "thermal bath" is a repository of energy that can be exchanged with the system. If we consider the system+bath ensemble to be closed, then the total energy has to be conserved. In a system of constant total energy, every state has the same probability (this either follows from the maximal entropy principle or can be taken as an axiom). However, every state of the system of interest can be realized via many bath states. In order to infer the probability of a specific state of the system of interest, we need to figure out the number of bath states that correspond to it.

The number of bath states depends on assuming some sort of a bath model. In the simplest case, the bath can be assumed to be infinitely large and have an exponential density of states Ω_{bath} in the vicinity of total energy. In other words, the bath entropy (logarithm of density of states) should be linear in energy. If we observe the system of interest in the microstate α with energy $E(\alpha)$ (which might be positive of negative), that energy had to be taken from the bath. As the energy remaining in the bath changes, so does its entropy, approximately linearly:

$$\ln \Omega_{bath}(\alpha) \approx \ln \Omega_0 + \beta(-E(\alpha)). \tag{3.11}$$

The linear coefficient β is the derivative of bath entropy with respect to its energy:

$$\beta \equiv \frac{d \ln \Omega_{bath}}{dE}. \tag{3.12}$$

The probability of observing a given system state is then proportional to the number of bath states once system's energy has been taken away. With proper normalization, this probability becomes:

$$\Omega_{bath}(\alpha) \propto p_\alpha = \frac{1}{\mathcal{Z}} e^{-\beta E(\alpha)}, \tag{3.13}$$

where we have recovered the same probability exponential in energy, also known as the *Boltzmann distribution.*

In principle, the system of interest can be coupled to different baths and exchange different extensive quantities, such as energy, volume, magnetization, number of particles etc. If these bath couplings are independent, then each would just contribute a linear term within the exponential factor, as in Eqn. (3.4). Since the resulting functional form for probability is the same, we have two interpretations of what it means: either as a subjective inference problem, or as an equilibrated exchange of some extensive quantity with a bath. To make an example relevant to design, one can imagine the exchange of money between the budget of a particular project and the broader enterprise. The conjugate design pressure, cost tolerance, regulates how important is the cost of a particular design solution in determining whether that solution would be picked.

Throughout this dissertation we assume that the system and the bath are always in *equilibrium.* This assumption was required to derive the probability formula, and consequently the rest of the mathematical formalism. In general there are many ways to break equilibrium of collective systems, such as transient processes, explicit external drive or energy injection, and coupling to different thermal baths. While all these effects constitute a fascinating study of *non-equilibrium* statistical mechanics,[104] I am not going to refer to those phenomena within this dissertation.

3.1.5 Computing observables

So far we detailed how to infer the probability distribution p_α from the observables $\langle \mathcal{O}_i \rangle$, but to complete the agenda of the inference problem we need to also specify how to compute the desired new observables $\langle X \rangle$. We presume that the value of observable X_α can be straightforwardly computed for any given microstate α. In this case, the average of the observable is just as sum over the ensemble weighted by the probability:

$$\langle X \rangle = \sum_\alpha X_\alpha p_\alpha = \frac{1}{\mathcal{Z}} \sum_\alpha X_\alpha e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)}. \tag{3.14}$$

For a generic observable X, this is the extent we can do. Some clever tricks might help in computing the sum in α, and I talk below about several such tricks. Still, before $\langle X \rangle$ is computed, one needs to compute the value of the partition function $\mathcal{Z}$, itself a tricky sum. Given our previous discussion, the partition function is an explicit function of the intensive thermodynamic variables λ_i, and also an implicit function of the extensive variables that restrict the domain of α. Given that we inevitably need to do the hard work of computing the function to serve as a probability normalization factor, is there more benefit that can be extracted from it?

Turns out, the partition function contains a wealth of information on certain kinds of statistical averages. Specifically, let's focus on the *conjugate averages*: given the design pressures λ_i, let's extract the averages of design objectives $\langle \mathcal{O}_i \rangle$. Consider the following construct with the partition function log-derivative:

$$\begin{aligned} -\frac{\partial}{\partial \lambda_i} \ln \mathcal{Z} &= -\frac{1}{\mathcal{Z}} \sum_\alpha \frac{\partial}{\partial \lambda_i} e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)} = \frac{1}{\mathcal{Z}} \sum_\alpha \mathcal{O}_i(\alpha) e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)} \\ &= \sum_\alpha \mathcal{O}_i(\alpha) p_\alpha \equiv \langle \mathcal{O}_i \rangle \end{aligned} \tag{3.15}$$

Turns out, statistical averages are encoded in the derivatives of the partition function! This is very profound. I am not sure if this is an astute observation or an incredible coincidence, but popular statistical mechanics textbooks use it instrumentally without additional commentary.[62, 102] This property definitely has to do with the exponential functional form of probability, and would not necessarily hold in more exotic versions of statistical mechanics that don't use Shannon entropy.[103]

Additional derivatives reveal additional properties of the averages. For example, a repeated derivative in the same design objective λ_i gives the variance of the observable, and a mixed second derivative gives covariance between two conjugate observables:

$$\left(-\frac{\partial}{\partial\lambda_i}\right)^2 \ln\mathcal{Z} = \left\langle\mathcal{O}_i^2\right\rangle - \left\langle\mathcal{O}_i\right\rangle^2 = \left\langle\mathcal{O}_i^2\right\rangle_c ; \tag{3.16}$$

$$\left(-\frac{\partial}{\partial\lambda_i}\right)\left(-\frac{\partial}{\partial\lambda_j}\right) \ln\mathcal{Z} = \left\langle\mathcal{O}_i\mathcal{O}_j\right\rangle - \left\langle\mathcal{O}_i\right\rangle\left\langle\mathcal{O}_j\right\rangle = \left\langle\mathcal{O}_i\mathcal{O}_j\right\rangle_c . \tag{3.17}$$

The covariance relationship can be used to quantify the correlation between two design pressures and determine whether they drive the ensemble to similar, different, or independent solutions. The repeated derivatives also give rise to an important relationship::

$$\left(-\frac{\partial}{\partial\lambda_i}\right)^2 \ln\mathcal{Z} = -\frac{\partial}{\partial\lambda_i}\left\langle\mathcal{O}_i\right\rangle = \left\langle\mathcal{O}_i^2\right\rangle_c . \tag{3.18}$$

This implies that the *susceptibility* (or sensitivity) of the average of a design objective is closely related to the *fluctuation* of that objective. In other words, if in a certain regime of design pressure the preferred solutions change rapidly, then they would also fluctuate a lot, which is a generally undesirable trait. I use this relationship to discuss the cost phase transition in Chapter IV.

A similar relationship applies to the mixed derivative, given that the order of derivatives with respect to the two design pressures λ_i, λ_j can be interchanged:

$$\left(-\frac{\partial}{\partial \lambda_i}\right)\left(-\frac{\partial}{\partial \lambda_j}\right)\ln \mathcal{Z} = -\frac{\partial}{\partial \lambda_i}\langle \mathcal{O}_j\rangle = -\frac{\partial}{\partial \lambda_j}\langle \mathcal{O}_i\rangle . \tag{3.19}$$

This is a generalized *Maxwell relation*, which is useful in cases when computing certain kinds of derivatives is more convenient than other kinds, especially when computations involve numerical methods or experimental data.[105, 66]

We derived the probability expression by maximizing its entropy, but in the process shifted attention away from the entropy itself. We can now compute the value of the entropy by plugging in the probability expression:

$$S[p_\alpha] = -\sum_\alpha p_\alpha \ln p_\alpha = \ln \mathcal{Z} - \sum_i \lambda_i \langle \mathcal{O}_i\rangle = \ln \mathcal{Z} + \sum_i \lambda_i \frac{\partial}{\partial \lambda_i}\ln \mathcal{Z}, \tag{3.20}$$

so that the value of entropy is also encoded within the partition function.

Lastly, the computation of non-conjugate derivatives X can sometimes be streamlined by using the observable X as a *source term*, or another design objective. We compute a modified partition function with a generic value of associated design pressure λ_X, extract the derivative and set the design pressure to zero:

$$\mathcal{Z}_X(\{\lambda_i\}, \lambda_X) \equiv \sum_\alpha e^{-\lambda_X X(\alpha) - \sum_\alpha \lambda_i \mathcal{O}_i(\alpha)} \tag{3.21}$$

$$\langle X\rangle = \frac{\partial}{\partial \lambda_X}\ln \mathcal{Z}_X \bigg|_{\lambda_X = 0} . \tag{3.22}$$

The above properties of the partition function allow us to interpret it as a basic information structure of the design space. The value of the partition function is usually hard to obtain and not directly useful. However, knowing the function allows us to take derivatives that directly provide knowledge about macroscopic relationships in the designed system. While the partition function is certainly useful, it is not the only information structure that we shall use, the other one being tensor networks introduced later.

3.1.6 Coarse graining

The previous discussion concerned two disparate scales: an individual microstate (or detailed design solution) α or a sum over all possible α. The latter sum allows us to evaluate various observables that characterize the ensemble as a whole. However, it proves to be useful to study the ensemble on an intermediate scale, somewhere between a single microstate and the whole ensemble. These kinds of questions will arise in all of the main chapters of the dissertation. Here we provide a mathematical backdrop required for such computations, and explore the philosophical implications of this backdrop.

The microstate index α typically lives in a high-dimensional space. The objective functions $\mathcal{O}_i(\alpha)$ are hard to visualize in that space, and they obscure the entropic drive (presence of many states at the same objective value). In order to characterize the system at an intermediate scale, we introduce a function $\vec{x}(\alpha)$, which in conventional statistical physics is called an *order parameter* and in design context I shall call it a *design feature.* The function $\vec{x}(\alpha)$ is *surjective*, i.e. for every possible value in the domain of $\{\vec{x}\}$ there is one or more values in $\{\alpha\}$, but each α maps to exactly one $\vec{x}$. This function therefore compresses the design ensemble, sometimes dramatically. Instead of operating with microstates, we operate with lumps of microstates that share the same value of the feature.

Why is this lumping useful? It is useful because we can compute the *effective objective function*, which for consistency with statistical mechanics usage we would call *Landau free energy.* Often in earlier statistical physics texts a lot of attention is paid to a phenomenologically proposed free energy based on system symmetries and actually introduced by Lev Landau in the 1930s.[106] However, several texts such as [107] give an explicit formula and others mention this formula in passing [62]: [1]

$$e^{-F(\vec{x})} = \sum_{\alpha} e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)} \delta(\vec{x} - \vec{x}(\alpha)). \tag{3.23}$$

The delta function corresponds to restricting the summation domain to only those values of α that map to the same design feature $\vec{x}$. The resulting function $F(\vec{x})$ defines the Landau free energy landscape over the domain of $\vec{x}$. In conventional statistical physics the free energy is defined with a prefactor of β in order for it to have the same units as energy. However, since we are going to deal with several design objectives that have different units, we set the free energy to also be unitless. In this way, free energy is a sort of "common currency" of different design objectives.

Apart from balancing the design objectives, free energy also accounts for the entropic drive. In the simplest case, imagine that all $n(\vec{x})$ states α in the same lump $\vec{x}$ have the same value of design objectives. In that case the entropic contribution to the free energy would be $(-\ln n(\vec{x}))$. The more microstates are lumped together, the lower the Landau free energy becomes, and the more likely such a mesoscopic state becomes. Chapter IV demonstrates an example of a trade-off between the design objective drive and the entropic drive.

The procedure of computing the Landau free energy via Eqn. (3.23) is known as *coarse graining*, following the procedure introduced by Leo Kadanoff for lattice spin models.[115] In order for the coarse-grained description to be consistent with

[1] It is unclear whether the expression of Landau free energy as a restricted sum is due to Lev D. Landau himself. It appears in the 5th volume of the Landau and Lifshitz *Theoretical Physics* textbook starting with the 3rd edition in Russian (1976)[108] which was the base for the 3rd edition in English (1980)[109]. The previous Russian 2nd edition (1964),[110] the last one published before Landau's death, does not mention the formula. The 3rd edition, prepared by Landau's colleagues and coauthors, quotes the origins of the formula as "This formulation of the problem for the phase transitions of the second kind is due to L.D. Landau (1958)", but does not provide a bibliographic reference. Complete collections of Landau's works show no relevant papers from 1958 or any time later.[111, 112] The first verifiable appearance of this formula in a peer-reviewed paper that I managed to track down is from J.S. Langer in 1974,[113] though similar ideas appeared before in context of long-wavelength hydrodynamic approximations.[114] The formula was popularized in N. Goldenfeld's textbook,[107] who in private communication with me confirmed learning it from Langer.

the microscopic one, the statistical weights of the mesoscale states need to add up to the original partition function:

$$\sum_{\vec{x}} e^{-F(\vec{x})} = \mathcal{Z}. \tag{3.24}$$

The coarse-graining procedure can be repeated, each time leading to a coarser description. In some cases the functional form of the free energy remains the same and only the coefficients change. The study of those coefficients with this *renormalization group* procedure gives important insights in the theory of critical phenomena.[107] However, in context of design the functional form would usually have a different functional form at each scale, so the renormalization group treatment is not useful for my case.

Computing the full partition function is done via the same formula as an intermediate scale coarse graining. Previously we showed that a lot of important observables are done in terms of $\ln \mathcal{Z}$, which can be identified with the free energy of the whole system.[102] If connection of microscopic to intermediate to macroscopic scale is done via a single coarse graining formalism, then none of these scales have a fundamental meaning. The description at each scale is *effective* in the sense that it ignores detail finer than that scale.

Even the microscopic scale description then is effective, since it depends on the modeling choices of the observer. This is most apparent in the statistical physics theories of colloidal particle systems. Statistical physics was first developed to account for systems of moving atoms and molecules, which are at least chemically indivisible and indistinguishable.[116] The indistinguishable nature of the atoms leads to Gibbs' paradox that is most commonly explained by appealing to an underlying fundamental quantum principle.[63] On the other side, each colloidal particle consists of thousands or millions of atoms. It is practically impossible to synthesize the particles to be exactly identical, and they can be distinguished with a good electron microscope. Still, we write extremely similar statistical physics expressions for both atoms and colloids. The resolution of this apparent paradox lies in recognizing the observer's *modeling choice* to treat the colloidal particles as *undistinguished* and rigid particles.[62, 63]. This and other conceptual issues of statistical mechanics of *classical* particles in context of self-assembly are discussed in the last section of this chapter. As for the Naval Engineering case study, I make a choice of what is the most detailed scale of design solution description that I want to resolve, specified in each chapter.

3.1.7 Formal definitions of design space

The mathematical techniques developed in the previous pages allow establishing an explicit equivalence between the concepts in design and the concepts in statistical physics theory.

The ensemble of microstates α is the solution space, which may or may not coincide with the design space as discussed in Chapter I. The goal of the design process is not in selecting an optimal solution from the solution space, but in understanding the structure of this space. The structure is encoded in the probability distribution p_α, but the whole space is too large and too high-dimensional to visualize p_α directly. Because of that, we need to supply additional tools to characterize the ensemble.

The probability distribution is shaped by the design pressures λ_i imposed on it. In the Naval Engineering Chapters IV–VI I assume that the design objectives stay constant and we only vary their relative importance by changing the design pressures. The objective space of possible problems is then spanned by the variation of design pressures $\{\lambda_i\}$, with each λ_i contributing a space dimension. We explore two dimensions of design space in Chapters IV–V and one dimension in Chapter VI.

In chapter VII that discusses the self-assembly problem, there are multiple design parameters considered. The chief design pressure is the thermodynamic temperature T which I consider fixed. Instead, I vary the concentrations of different self-assembly building blocks and the binding energies between the blocks, which jointly form the design space.

In order to evaluate the design outcomes, I compute both conjugate $\langle \mathcal{O}_i \rangle$ and non-conjugate $\langle X \rangle$ observables averaged over the solution space. The conjugate observables explain the direct correspondence between the desired design objectives and the capacity of achieving them. However, the conjugate observables are not necessarily sufficient. In chapter IV I show that the pursuit of two design objectives might induce unexpected correlation of design variables, thus reducing the effective number of design choices that fully determine the solution. In chapter VI I show how different couplings within the solution space manifest when our observables are focused on particular degrees of freedom. In chapter VII I show how subtle modifications in the definition of "yield", the key observable outcome of self-assembly, can make theory much more relatable to experimental measurements.

But why do we need to consider non-conjugate observables at all? Why is it not sufficient to declare a design objective and directly evaluate our capacity to reach it? The reasons for this are twofold. On one side, in practical design situations we do not always get an opportunity to regulate the design pressures. All self-assembly behaviors result from an interplay between the energy landscape and the thermal fluctuations of the building blocks. As designers, we don't get to control the thermal fluctuations, and our control of the energy landscape is limited by the substrate technology (see Chapter VII for more discussion). On the other side, having a narrow view of only goals and outcomes deprives us of the crucial knowledge about the design space behavior, and gaining that knowledge is a key feature of design.

Following this knowledge-oriented design philosophy is the main reason to bring in statistical physics methods. The goal of these methods is not in providing solutions for well-posed design problems (optimization is fully capable of that), but in providing an arsenal of tools to study ill-posed, wicked design problems. Pursuing this study throughout the main chapters of the dissertation focuses our attention not only on the macroscopic relationship of the design space and design outcomes, but also on the topography and structure of the solution space on multiple scales. The ability of statistical physics to investigate collective phenomena on multiple scales is its most powerful, and arguably its defining feature.

3.1.8 Statistical mechanics computations

The previous sections set up the philosophical principles and the mathematical backbone of the statistical physics approach, but what is it that we want to compute? Which system-specific computations would actually yield insights into the design problem structure? The main quantities we are interested in are the partition function $\mathcal{Z}$, observable averages $\langle X \rangle$, and free energy landscapes $F(\vec{x})$ (defined in Eqns. 3.5,3.14,3.23 respectively). The complicated part that has been glossed over so far is how to actually perform the sums $\sum_\alpha$ of terms and over domains that depend on the specific system considered.

The "gold standard" of statistical mechanics calculations is to extract the quantities of interest as analytical functions of the relevant arguments, so-called exact solutions.[117] Before the large-scale availability of computers this was the main way to make progress. A lot of the exactly solved models are defined on periodic lattices and have highly symmetric Hamiltonians. Because of the universality property, analytical results for these highly idealized models turn out to be closely relevant for certain experimentally measurable condensed matter systems, especially close to *phase transitions*.[97, 107] Phase transitions are large-scale rearrangements of the system state caused by minor variation of the thermodynamic parameters close to a critical point. While we will see phase transitions in design problems, especially in chapter IV, but they will not be as sharp.

The systems considered in this dissertation, even when defined on periodic lattices, have much less symmetric Hamiltonians, complicated boundary conditions, and strong finite-size effects. If we can't look up answers in the book, how do we actually perform computations of $\mathcal{Z}$, $\langle X \rangle$, $F(\vec{x})$? We rely on a combination of established and new methods, as detailed below.

Direct and hierarchical summation

The simplest an perhaps naïve way to perform a summation $\sum_\alpha$ is to just directly add terms in a program loop. This is the main computational method in Chapters IV-V. The brute force nature of this method is alleviated with two slight tricks. First, the computation there relies on adding up many routings that have the same cost. In this case, adding up many identical terms can be replaced by a multiplication by the number of such terms, which can be computed analytically. Second, several parts of that sum occur multiple times in a partition function. In that case, the results can be intermediately stored in a look-up table and later substituted from there instead of doing the calculation from scratch again.

The obvious disadvantage of such computational tricks is their *ad hoc* nature that highly relies on the specific problem considered and is not generalizable.

Stochastic simulation

A common approach to statistical mechanics systems is stochastic simulation, such as Markov Chain Monte Carlo methods. Instead of counting all of solutions $\{\alpha\}$, such methods count a random representative subset of solutions and use it to compute approximate averages across the solution space. Such methods were used in early investigations that led to Chapter IV, but were ultimately superseded with exact enumeration methods.

Saddle point approximation

The saddle point approximation method is closely tied to coarse-graining of the solution space. In terms of the free energy landscape, the partition function is given by:

$$\mathcal{Z} = \sum_{\{\vec{x}\}} e^{-F(x)} \simeq e^{-F(x^*)}, \tag{3.25}$$

where we approximate the whole sum with its largest term, achieved at the global minimum of the landscape F in the space of design feature $\vec{x}$. This approximation typically becomes better for large system sizes where the fluctuations around the free energy minimum are negligible. We use this approach in spirit rather than explicitly in Chapter V. There we compute the free energy landscape $F(\vec{x})$ and find all of its local minima, not just the global one. We then analyze the shape of the landscape around these minima to compute the design robustness metrics.

Perturbation theory

Perturbation theory is the idea of separating the Hamiltonian (design objective) into two parts. One part defines the dominant behavior of the system and is easily computable. The other part defines a small correction to the dominant behavior and can be computed as a perturbative expansion, usually to a low order. We use a similar idea to model the response of subsystem design to external design stress in Chapter V. The effect of external design stress is assumed to have an extremely simple, linear form which nevertheless allows us to get a lot of insight of system design robustness.

Tensor networks

The major computational innovation of this dissertation is the usage of tensor networks as statistical mechanics information structures in Chapter VI. Tensor networks allow us to not only compute the partition function, but also a variety of marginal and conditional probability distributions via a simple high-level interface. This makes them a highly adaptable and generalizable tools, unlike the earlier direct enumeration methods. The next section of this Chapter provides more details about tensor networks.

Series summation

In some cases statistical mechanics computations involve infinite but structured series summations. Several of such expressions occur in self-assembly computations of Chapter VII. The infinite series can then be summed analytically by using special (or not very special) functions. Since the summed series are expressed in terms of not scalar, but matrix arguments, I present suitable matrix generalizations of the expressions. The divergence of such series summations has special physical meaning discussed there.

Eigenvalues and singular values

While the convergence of scalar series is determined by the value of the argument scalar, the convergence of a matrix series is determined by the lead eigenvalue of the argument matrix. This naturally leads me to consider the spectra of the matrices, where important information is contained in both the leading and subleading eigenvalues. The implications of this spectral analysis are explored in Chapter VII.

Spectra also enter the computations in a different, implicit way in Chapter VI. The coupling tensors I use in tensor network methods are essentially large square matrices. The spectrum of these matrices spans many orders of magnitude, with the smallest eigenvalues contributing very little to the results of the computation. Dropping such small eigenvalues, formally known as Singular Value Decomposition (SVD), drastically simplifies the tensor network contraction. SVD is performed automatically in the tensor network contractor I use.

Mean field theory and self-consistent expressions

The last method I need to mention is mean field theory. Mean field theory expresses a statistical mechanics quantity via an auxiliary variable; this variable itself is computed by averaging over the ensemble. This results in self-consistent expressions, usually implicit equations in terms of mean field. This technique is not explicitly used in the dissertation, but can be used to reproduce and generalize certain combinatorial results of Chapter VII.

3.2 Tensor Networks

3.2.1 Basic notions

In this section I review the central ideas and existing applications of tensor networks. As the name suggests, a tensor network is a network where all nodes are tensors and all edges are contractions of tensors. As any other network, it can be easily drawn to illustrate the topology. It is this graphic property that motivated the introduction of tensor networks as a new form of mathematical notation by

Penrose in the 1970s.[118] This form of notation can be used in General Relativity computations where upward and downward pointing edges directly represent covariant and contravariant indices and can be connected via metric tensors.[119]

Each tensor $A_{ijk\ldots}$ is essentially a multidimensional array that can be addressed with indices $ijk\ldots$. The number of indices is known as *tensor rank* (due to historical abuses of terminology it is distinct from matrix rank). A scalar number, a vector, and a matrix are all special cases of a tensor with ranks 0,1,2, respectively. The dimensions of different indices don't have to be the same, as it is easy to imagine an array of dimensions $2 \times 10 \times 6$, though one then needs to keep track of the meaning of each index. The fundamental operation of two tensors is *contraction*, i.e. a summation of elements over a shared index, which results in a new tensor:

$$A_{ijk}B_{kl} = C_{ijl}, \tag{3.26}$$

where the k summation is implied. Contraction is only possible between two indices of same dimension. In some cases it is necessary to keep track of covariant or contravariant nature of indices and only contract two of opposite type, but such a distinction is not needed or used in this dissertation. The two indices can belong to the same tensor, in which case such a contraction is known as *trace*.

A tensor network is essentially an instruction of the pattern of contractions between tensors in an arbitrary order that respects the above rules. Some tensors can have remaining free indices that are not summed over and would remain on the resulting tensor, just like in Eqn. 3.26. In the tensor network, such indices are shown with edges that leave from a tensor but don't connect to another tensor, also known as *external legs*. Since all contractions are just linear summations of elements, the result of a network contraction does not depend on the order in which the pairwise contractions are performed, so we don't indicate such an order. The computational aspects of tensor network contraction are described below.

It is important to distinguish a tensor network from the single tensor to which it evaluates. Such evaluation requires computational resources, often prohibitive, and also frequently loses the special structure built into the network. Delaying the contraction allows us to perform lots of complex operations on the tensor network, thus treating it as an *information structure*. We review below the existing applications of tensor networks as information structures in different domains and synthesize the best practices that make tensor networks useful to us.

3.2.2 Applications

In order to motivate why contracting a network is not always useful, consider a network of many tensors indexed with i, each having rank d_i and typical index dimension K. The memory required to store all elements of such a network scales as following:

$$M \propto \sum_i K^{d_i}. \tag{3.27}$$

Note that the scaling is very steep with respect to the dimension and rank of tensors. On the contrary, the scaling is essentially linear in the number of tensors of the same type. If in constructing a model one can split a single large tensor into several smaller ones, it is possible to achieve exponentially large savings in memory. If storing the large tensor would overwhelm the largest computers available, storing an assortment of smaller tensors might just about make the problem tractable.

This idea of memory saving underlies all of the modern applications of tensor networks, including ones in this dissertation. A large multi-linear object of high rank is split up into many smaller objects that collectively perform the same function, exactly or as a controlled approximation. The meaning of the individual tensors and their contraction patterns needs to be negotiated in each problem on a case-by-case basis.

Quantum condensed matter

The most common tensor network application is the study of quantum condensed matter systems such as correlated electrons.[120, 121, 122] This usage is so common that some sources perceive tensor networks as an inherently “quantum” technique rather than a general mathematical tool. A typical problem in quantum systems is representing an entangled multi-particle wavefunction from a large Hilbert space. Given a fixed basis, this wavefunction becomes a tensor with rank equal to the number of particles considered – i.e. quickly becoming prohibitively large.

Tensor networks in this case serve as a wavefunction *ansatz*, i.e. a prescribed functional form that attempts to fulfill the properties expected of the exact wavefunction. Fortunately, such approximation ansatz is very good at representing the “interesting” corners of Hilbert space, such as the ground state or low-lying excited states. In order to probe the properties of such states, many optimized algorithms have been developed that exploit the simple tensor network topologies: the Matrix Product State or Tensor Train (MPS/TT) for 1D lattices, Projected Entangled Pair State (PEPS) for 2D lattices, and Multiscale Entanglement Renormalization Ansatz (MERA) for hierarchical wavefunctions. For these topologies, there are plenty of off-the-shelf algorithms that find ground states, evaluate observables, compute entanglement entropy, and perform real or imaginary time propagation.

Quantum chemistry

Another important area of application of tensor networks is precision quantum chemistry. While the quantum condensed matter studies are interested in the general properties of entangled wavefunctions and properties of quantum phase transitions, quantum chemistry instead focuses on the specific spectra and energy levels of atoms, molecules, and bulk structures. While popular computational techniques such as the Density Functional Theory (DFT) are able to account for a lot of systems of interest, in some cases it is essential to keep track of electron correlations to get accurate answers, and tensor network based techniques were developed for that.[123]

Simulation of quantum computing

Tensor networks can also be used to simulate the behavior of quantum computing circuits on classical computers.[124] A typical circuit consists of preparing several qubits in desired states and, acting on them with unitary logic gates, and performing measurements. While on a real quantum computer such a gate is an indivisible operation, it can be represented as a linear operator acting on the wavefunction vectors, at increased computational cost. A combination of such linear operators acting in a specific order on specific single qubits or pairs of qubits is then isomorphic to a tensor network. Leaving the remaining external legs uncontracted allows reading out the result of the computation. An additional benefit of such a notation is that the direction of legs on the tensors simplifies reasoning about bra and ket vectors, similar to the covariant and contravariant vectors in General Relativity.

Renormalization calculations

Back in classical physics, tensor networks allow formalizing the heuristic block-spin renormalization procedure of Kadanoff.[115] In tensor network language, renormalization amounts to summing the internal degrees of freedom of a block of spin, dropping the small singular values, and relabeling the coarsened tensors to get back to the original topology. With some extra precautions, such a procedure allows reconstructing the renormalization group flows and critical exponents of original studies, while creating a more flexible, interpretable, and computationally efficient framework.[125, 126]

Machine learning

Tensor networks have also found applications in storing the trained weights of multi-linear machine learning classifiers.[127] While it is not in general expected that the correlations between the weights have MPS topology, in practice such a representation of high-rank data gives good classifier results, while being economical in memory. Tensor-based structures, though different from our discussion here, are also used in the linear layers of neural networks to store the trained weights, thus giving the name to the `TensorFlow` package.[128] The need to rapidly perform low-precision tensor arithmetic led to the development of dedicated Tensor Processing Units, or TPUs.[129]

Constraint satisfaction counting

The inherent combinatorial nature of tensor networks allows using them in solving #SAT problems, a subset of constraint satisfaction problems where the goal is not in *finding* a solution that satisfies a set of constraints, but in *counting* the number of all such solutions. The original paper by Penrose shows that the problem of counting the 3-colorings of a given graph can be solved by contracting Levi-Civita tensors along the topology of that graph, but did not have a computational method to actually perform that contraction.[118] More recently, such methods were proposed for general constraint satisfaction problems.[130, 131] These methods rely on

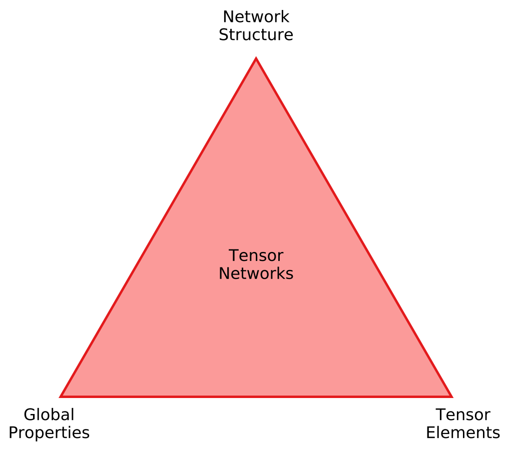


Figure 3.1: Tensor networks express the relationship between three factors: tensor elements, network topology, and global properties.

bipartite tensor networks, where some tensors encode the variables, others encode constraints, and variables can only be contracted with constraints. In my own work outside of this dissertation I used tensor networks to count the number of tilings of lattice dimers, i.e. the entropy of a discrete hard particle system.[80]

Linear algebra

Tensor train networks have been used to accelerate the numerical linear algebra computations of large multilinear objects.[132, 133, 134]

Common features

The common feature of the examples above is attempting to express some large-scale property of the whole tensor network via the entries of separate tensors. I illustrate this relationship via a triangular diagram in Fig. 3.1. This diagram is not an expression of a particular algorithm, but rather a framing device to ask questions about the tensor network. In a particular problem, typically two corners of the diagram are known or assumed via a modeling choice, and computation is required to find the third corner.

The first corner is the global properties encoded by the whole network. Sometimes they are already known, such as the need to find a wavefunction minimizing a given Hamiltonian, or find a set of weights that best fit a given dataset. In other cases the global properties are the target, for instance the partition function or coupling flows in the classical renormalization calculations, or the number of solutions of an #SAT problem.

The second corner is the elements of the tensors within the network. The goal of quantum condensed matter computations is frequently to find the elements of the wavefunction that would minimize energy. The goal of machine learning is to find (or in the jargon, train) the weights in a classifier or a neural network that best fits the dataset. In other cases the weights are known *a priori*, such as the variable and constraint tensors for constraint satisfaction, or the coupling tensors for classical statistical mechanics.

The third corner is the topology of the tensor network, which is the least discussed. Most frequently, the topology of the tensor network mimics the actual or presumed topology of couplings in the system's Hamiltonian. For a 1D lattice system where each element in coupled to the next, it is natural to consider a 1D tensor network (MPS/TT) to encode the wavefunction, even though it is only an approximation of the real wavefunction. For similar reasons, the 2D systems are usually described with 2D lattice networks (PEPS). Both of those topologies have issues with capturing long-range correlations characteristic of near-critical systems, which led to the development of a hierarchical tensor network (MERA).

However, the tensor network literature is virtually disconnected from *network science* literature.[50, 135] The central idea of network science is that the topology of connections in real-world systems is neither regular like a lattice, nor fully random like an Erdös-Rényi graph, nor densely connected like in mean-field or well-mixed models. Instead, the topology is a complex network that can be described on a low level with an adjacency matrix, or on a high level via community structure. Network science aims to explain how the topology of networks originates, and how it affects processes on networks. However, for our purposes network science gives us a license, or a task of accounting for systems with complex irregular topology that we cannot freely approximate.

This ethos drove me to develop the tensor network methods and software to study emergent couplings of Logical and Physical Architecture, described in detail in Chapter VI. I treat tensor networks as information structures that encode the whole design space. I ask questions about this design space via a specific graphical and computational language. For my system, the tensor elements and the network topology are known *a priori*, and the goal is to find the global properties of the network, i.e. the design outcomes.

However, the macroscopic relationships between design pressures and outcomes answer the less interesting *what* question. The more interesting *why* question is answered by considering in a lot of detail the emergent structures on the intermediate scales of the design space. This need drove the development of the `TenZ` package described below, and the specific set of techniques and questions described in Chapter VI.

3.2.3 Software solutions

The rising popularity of tensor network methods lead to proliferation of dedicated software across many programming languages, see `http://tensornetwork.org/software/` for a list. The most common low-level operation to be performed on the network is a pairwise contraction of two nodes. A sequence of these contractions can have different high-level aims. Most commonly it aims to reduce the whole network to a single tensor by contracting all internal links. Occasionally, however, different operations are called upon by the algorithms, such as variational optimization.

While tensor networks give us a useful graphical language and a high-level abstract concept to manipulate, deep down the contraction of a tensor network amounts to a long sequence of multiplications and additions of real (or complex) numbers. The result of this sequence of operations does not depend on the order in which they are performed, so long as the computations are all exact.

However, the time complexity of the whole contraction is known to depend sharply on the sequence in which the elementary contractions are performed. Depending on the application, there exist not only different technical ways to deal with the contraction sequence, but also different schools of thought towards it.

One school of thought, materialized in the `NCON` package, comes from the ethos of the condensed matter community.[136] The central idea of `NCON` is in finding the optimal contraction sequence that minimizes the number of operations by searching the vast but finite space of possible contraction sequences. Finding the optimal contraction sequence is an NP-hard problem that scales exponentially with system size. However, the contraction sequence does not depend on the contents of each tensor, thus the base of time scaling is relatively mild. Finding the full contraction sequence before the contraction begins is useful if the same network topology is used many times, perhaps sweeping over a system parameter range. `NCON` has several additional features to it, such as an ability to efficiently recycle partial results and quickly compute the environments of all nodes in the network at the same time, a useful trick in variational optimization. NCON is originally implemented in Matlab, though ports to other languages exist.

An alternative way to deal with the contraction sequence is designing it on the fly by using some heuristic method, as exemplified by the PyTNR package for Python.[137, 138] A sequence created this way is not necessarily optimal, but likely better than random. Additionally, PyTNR dynamically performs approximations at user-specified accuracy using Singular Value Decomposition (SVD). Thus the results obtained are not exact but the computation can proceed much faster and scale much better for large systems. Since the number of singular values preserved by SVD at given accuracy depends on the tensor contents, the runtime of PyTNR on a given network actually depends on the tensor entries and not just topology.

PyTNR promotes the ethos of being creative with the network constructions and performs different calculations by modifying the network topology between contractions. Though a modified topology might lead to a significantly different contraction sequence, especially if loops in the network are formed or broken, there are no resources invested into finding the true optimal sequence, and not using it does not constitute a large efficiency loss.

3.2.4 TenZ package

In order to facilitate the conversion between physics questions about a particular system and the tensor network contractions that address those questions, I developed the TenZ package, short for "Tensor $\mathcal{Z}$", where $\mathcal{Z}$ stands for the partition function. The package consists of several modules as depicted in Fig. 3.2. The goal behind TenZ is to encode the basic structure of the tensor network as an information structure in a persistent object, and then rapidly create tensor networks with different small topology modifications to ask specific questions of this information

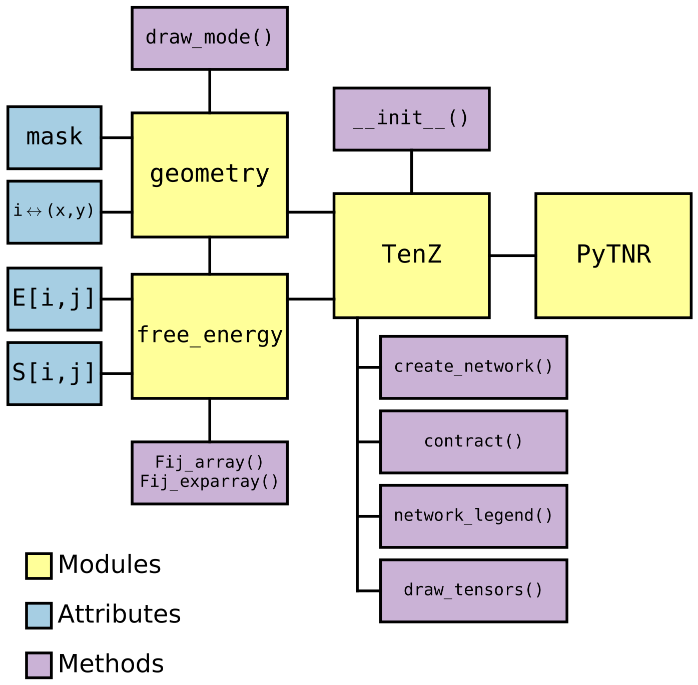


Figure 3.2: The `TenZ` package interfaces physics models with `PyTNR` contractions. The `free_energy` and `geometry` modules on the left encode the properties of the specific modeled physical system. The `TenZ` module (center) encodes the basic topology of the tensor network and generates the networks with desired on-demand modifications and passes it to `PyTNR` for contraction.

structure in order to generate knowledge.

The topology of the tensor network in form of the adjacency matrix A_{ij} is put into a `TenZ` object upon initialization. The `TenZ` object interfaces with two other objects: `free_energy` and `geometry` modules. Each of these modules has standard interfaces, but the internals can be changed to model a specific system. The `geometry` package encodes the space in which each of the functional units in the system can be placed, and allows for such high-level commands as conversion between 2D coordinates in space and 1D index of coupling tensors, as well as drawing arbitrary scalar fields within the space. The `free_energy` package computes the coupling tensor between any pair of functional units. The "free energy" name was chosen because the coupling tensor already depends on the coarse-grained design space variables, weighted by effective rather than raw design objective. The `create_network()` command allows the user to create a tensor network that describes the coupling of functional units along the edges of the network, while the keyword arguments of the command implement the desired modifications of the topology. Chapter VI gives a more detailed description of modeling a design problem via tensor networks, the language of basic moves to modify topology, and the questions asked of this information structure.

We anticipate releasing the source code of the `TenZ` module along with the paper based on Chapter VI.

3.3 Statistical mechanics for self-assembly

3.3.1 Scope of theory needed

The subject of this dissertation, as much as design, is statistical mechanics. In the earlier part of this Chapter I primarily focused on revising the definitions of state space, design pressures and outcomes. I use a variety of conceptual and computational tools such as coarse-graining and Landau free energy, and introduce new ones, such as the tensor networks. The motivation behind them, and the idea behind Systems Physics is that the structures that appear in the design space can be mapped to physical objects, whose interactions can be accounted for with familiar mental models such as forces and displacements. This is a novel application, and thus requires a lot of discussion.

Surprisingly, lengthy qualitative discussions are necessary not only for the “novel” application of statistical mechanics to arrangement problems, but also for the “traditional” physical systems of interacting colloidal particles exhibiting self-assembly. A theory of such particles, of completely classical nature, is not readily available in a complete form. For the theory to be completely interpretable and experimentally testable, we need to introduce an accounting of building block microstates and use this accounting to predict yields of desired self-assembled structures. Bits and pieces of such accounting are available, and the later part of this section lays out a research program towards a unified theory.

However, before developing such a theory, we need to make sure that we are not reinventing a standard “textbook” approach. Surprisingly, there is no relevant textbook to cover the subtle conceptual issues. There are plenty of good textbooks that emphasize different parts of statistical mechanics and allow quickly looking up useful formulas.[107, 102, 139, 140, 109] However, in recent years only Sethna’s textbook [62] radically modernized the order of presentation to emphasize more modern, interdisciplinary, unconventional applications while skipping the conventional thermodynamics almost entirely. While Sethna’s textbook coins a key notion of *undistinguished* particles, it does not pursue building a logically complete classical statistical mechanics theory.

3.3.2 Why don't we have a theory yet?

This lack of a comprehensive theory has to do with the history of statistical mechanics as a subject. The history starts with thermodynamics, a precursor to statistical mechanics that was originally developed to quantify the limits on efficiency of heat engines, culminating in the Carnot treatise in the 1820s,[141] and ultimately developed into a closed theory based on four laws. The need to justify the four laws from a more microscopic perspective led to a series of works in the second half of the 19th century that introduced the notions of entropy,[142, 143] derived the velocity probability distributions,[144] and culminated in the first textbook with a systematic treatment of statistical ensembles.[116] The main object of those investigations was atomic and molecular systems of "indivisible" particles. Investigations into the nature of such particles led to a pileup of problems of early 20th century physics and precipitated the development of quantum mechanics.

Some physical problems indeed only have quantum mechanical explanation, such as the indistinguishable nature of electrons, discreteness of light quanta, or freezing out of molecular degrees of freedom at low temperature. Other problems, however, fall back onto quantum concepts unnecessarily, such as the explanation of the Gibbs paradox,[145] or the usage of Planck's constant h as the universal measure of phase space volume. These intuitions can be traced to Bohr's correspondence principle saying that quantum systems ought to behave classically in the large-size, large-energy limit.[146] If classical behavior is only observed in certain limits, then the quantum theory is more fundamental and thus more worth pursuing to find explanations.

This pursuit occupied a large part of the 20th century, especially within the discipline of *solid state physics*, later renamed *condensed matter physics*, and now even more appropriately named *hard (quantum) condensed matter physics*. It is hard to underestimate the impact of those studies, from the tremendous technological impact of enabling all of modern electronics, to quantitatively explaining complex phenomena such as superconductivity, to developing sophisticated mathematical tools such as the renormalization group.[107] The need to train people in solving those kinds of problems led to a demand for textbooks teaching statistical physics via the prism of quantum systems.[147] More recent textbooks also include modern and practical computer simulation methods but don't necessarily refine the philosophical foundations.[148]

Of course, a century of research resulted in important theoretical ideas. The study of phase transitions, symmetry breaking, and universality resulted in the widely-used notions of *effective theory* and *emergence*.[149] The notion of *Landau free energy* as an aggregate description of a bundle of microstates allows discussing effective theories quantitatively, as I do in this dissertation.[107] The ideas about maximal entropy distributions connected statistical mechanics with information theory.[100, 99] A lot of arguments were proposed in the debate about the nature of macroscopic irreversibility and the arrow of time, though my interest here is exclusively in equilibrium.[94] Even in isolation from philosophy, many other condensed matter techniques allow calculations for models of complex systems such as networks.[101]

Historically, all these advances in hard condensed matter are hugely important, inspirational, and cannot be ignored. However, from the strict logic point of view, they do not contribute to the axiomatic, closed theory of *classical* statistical mechanics. The intervening century distracted us from pursuing the program started by Gibbs to the degree that Gibbs' own resolution of the paradox named after him is replaced with a deference to quantum mechanics.[63] The paradox has to do with adding the entropy of two types of "colored" particles and then continuously merging their colors into one, resulting in an apparent discontinuity of entropy. While in quantum mechanics it is impossible to continuously turn one atom into another, in the space of colloidal particles lots of properties can be adjusted continuously.[74] The different physical mechanisms of distinguishing particles were discussed by Jaynes via a thought experiment involving yet undiscovered elements whifnium and whafnium,[150] and again by Frenkel in context of colloidal particles.[151] The paradox is resolved by noting that distinguishing the particles is the observer's *choice*, recognized by Gibbs [145] but promptly forgotten. This choice was revived only a hundred years later by Sethna who coined the notion of *undistinguished* particles in his textbook.[62]

The unresolved questions were finally risen again by the advances in experimental *soft* condensed matter systems. The experiments by Meng et al. highlight the key role of rotational entropy and symmetry numbers in determining the equilibrium yields of colloidal clusters of different shapes.[64] Klein et al. closely interpret the subtleties of colloidal partition functions in canonical ensemble (what I called statistical weights in Chapter VII).[68] Cates and Manoharan comment on an assortment of apparent paradoxes and contradictions that appear in statistical mechanics of colloids when classical and quantum ideas are indiscriminately mixed up.[63] A framework by van Anders et al. explicitly treats the particles as continuously variable in shape,[72], while Antonaglia et al. show how to map out the distribution of entropy in the system.[67] These works do not yet form a comprehensive and unified theory, but they provide a lot of pieces that seem to fit together. With some more work, classical statistical mechanics can be a closed theory that allows for precise numerical computations without reference and deference to quantum mechanics to plug up conceptual holes.

This discussion was not meant as a comprehensive historical review, but as evidence for the conceptual holes and the reasons behind them. Below, I lay out two sets of questions that would lay out a research program towards building a complete theory. The first set of questions has more to do with conceptual questions and interpretations, while the second set has more to do with practical implementation and calculation. Answering these questions in full in the most general form will take more space than a Methods section would allow. Instead, I address them in a more specific form, with an eye towards self-assembly, in Chapter VII, but there is plenty of other phenomena in classical thermal systems that could benefit from a systematic framework.

3.3.3 Conceptual questions

In order to build a systematic framework for a classical self-assembly theory, we need to connect the dots by answering several specific questions. A lot of these answers are essentially modeling choices, owing to the subjective interpretation of statistical mechanics originally proposed by Jaynes,[99] and most recently articulated by Cates and Manoharan in application to colloidal particles.[63]

1. Why can we treat colloidal particles of micrometer or nanometer size as indistinguishable? Why can we treat clusters of particles as indistinguishable?
2. Do we need any references to quantum mechanical concepts such as the Planck's constant or the Pauli principle to make a consistent classical statistical mechanical theory? What is a good phase space measure h and when does it matter? What is then the thermal wavelength λ_{th}?

3. In classical mechanics, the rotations and vibrations of particle clusters are always active and never get frozen out. How do we account for the classical bond entropy and bond free energy? When and why are the interaction ranges and the shapes of interaction potentials important?
4. How do we separate the "integral" and the "combinatorial" parts of the theory? That is, when do we care about the continuous space of particle positions and orientations, and when do we care about the discrete space of particle membership in clusters?
5. Self-assembly statistics is obviously dependent on particle concentrations: at low concentrations building blocks don't encounter their binding partners often enough to form structures, whereas at high concentrations approaching close packing the building blocks either vitrify or form bulk structures as opposed to finite and discrete clusters. What range of concentrations is physically reasonable for synthetic systems and for biological systems? When is it more reasonable to talk about concentrations of discrete clusters as opposed to bulk descriptions?
6. Should the theory be built in the canonical or grand canonical ensemble, that is, fixed particle number or fixed chemical potential? Fixed particle number corresponds more closely to the setup of experiments and computer simulations, while fixed chemical potential makes theory easier. The physical difference can be attributed to the capability of the available chemostat to keep constant density of building blocks.[91]
7. What is the proper counting of terms that enter the grand canonical partition function? Should one count the unique cluster types or track their copy numbers as well?

8. What is the chemical potential? Is it merely a calculational device, or does it have a physical meaning? Why do the partition functions diverge at some value of chemical potential? What do the divergence and its exponent physically mean?

9. What are the good definitions of target structure yield? How do they relate to the desired applications of self-assembly, e.g. forming maximally precise structures or having the highest conversion of raw building blocks into approximately correct structures?

10. What are the degrees of freedom of the self-assembling system that we can reasonably control or design to set up an experiment or simulation, such as binding energies and concentrations? What are the observables that we can reasonably measure in experiment or simulation? The theory we construct should in the end relate the experimental controls to the experimental observables in a testable manner.

3.3.4 Implementation questions

This second set of questions is aimed more specifically at the implementation of a specific self-assembly theory building on Refs. [91, 90]. This theory considers a specific limit where the building blocks form discrete, finite clusters that have negligible interactions with each other.

1. An important part of statistical weight of self-assembled cluster is its internal entropy. How do we compute the rotational and vibrational entropies of clusters? How do they depend on the shape of the interaction potential that binds the particles?

2. For heterogeneous self-assembly, we need to find not just one optimal building

block, but a whole set of building blocks. What are the good collective metrics of a set of heterogeneous building blocks? How do we optimize these collective metrics?

3. Self-assembled clusters differ by topology, size, and specific sequence of building blocks. How do we make a unified accounting over these hierarchical aspects of structure? How do we compute the sum over all possible sequences of self-assembled clusters?

4. Apart from the attractive interactions at the binding patches, the building blocks have repulsive, steric interactions away from the patches. How do we account for steric interactions of the building blocks? How does the topology of the cluster affects its entropic weight? How do we account for the loop entropy penalty?

5. Building blocks with more than two binding sites allow the assembled chains to branch off, forming tree-like or looped structures. How do we enumerate all possible tree topologies that can form from a given set of brancher particles? How do we account for loops? How about denser topologies such as close packed 2D and 3D structures?

6. We primarily consider the lock-key specific binding and non-specific cross-talk. How does the cross-talk limit the size and accuracy of self-assembling structures? Do we need to also account for the key-key and lock-lock cross-talk? If so, how do we do that?

7. How do we account for the possible allostery of the building blocks, or many-point binding interactions between different binding sites?

There are probably other questions as well that arise out of the confusions over practical computational concerns. It takes a little bit of questioning and looking up the "textbook" answers to lay these questions base. While I address most of these questions once in Chapter VII, they still open up a fascinating field of work to get into.

CHAPTER IV

Phase Transitions in Design

4.1 Introduction

[1]Designing products with an emergent, overall function that is more than the sum of their parts is a crucial challenge in science and engineering.[153] Meeting this challenge is complicated by the fact that, for many complex products,[32, 33, 34, 31, 27] different subsystems employ diverse technologies and are designed using a variety of methodologies. Moreover, meeting the overall design goal for a specific product is seldom achieved by optimal performance for every individual subsystem.[24] The need to design subsystems that achieve target performance and contribute to overall system outcomes is becoming more pressing.[5, 6] The increased pressure arises because engineered products in a wide variety of industries now incorporate several distinct, but interconnected types of functionality.[5] As a result, for many modern engineered products more economic value is added in designing a product than in manufacturing it.[6] Making design more effective requires the ability to understand and quantify how the design of a subsystem is affected by overall design objectives, and how deviations from optimal performance are affected by interaction with other subsystems.

Here, we use techniques from information theory and statistical mechanics to

[1]This chapter is based on the paper [152] coauthored with C.P.F. Shields, D.J. Singer, and G. van Anders.

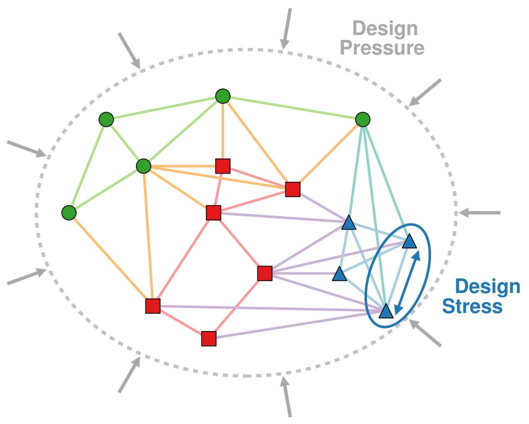


Figure 4.1: Schematic of the relationship between global design pressure and local design stress in a generic design problem. A complex system (whole network) is divided into three subsystems (represented by green, red, and blue nodes). Design pressure is represented by the inward pointing grey arrows and applies to all parts of the system. Locally, the global design pressure manifests itself as design stress, here between two blue nodes. If design parameters are e.g. the spatial locations of the nodes, then after taking into account the interactions among all other elements, the design stress expresses the marginal cost of incrementally changing the relative position of the two nodes. Figs. 4.4, 4.6 demonstrate the design stress patterns for the specific system studied in the Chapter.

show that subsystem performance and interactions can be cast in terms of "stress" and "strain" from materials physics. We illustrate this behavior in design problems that can be cast as arrangement problems. Arrangement problems arise in design in a wide range of disciplines, including at several scales in electronics,[154] as well as in distribution logistics [155] and facility layout.[156] Here, we focus on arrangement problems that arise in naval architecture.[157, 41] Naval architecture, specifically that of warships or other multi-use vessels, provides an ideal case for understanding the role of subsystem behavior in complex engineering design. Ships incorporate several competing design pressures,[158, 31] they require design specifications at several levels of detail,[11] and costs frequently prevent prototype production.[13] Additionally, ship design has a need for design freedom, i.e. it requires the consideration of nearly-redundant designs of comparable "cost" of the overall design objective. This type of design cannot be done via approaches that focus on finding individual designs, e.g. simulated annealing [21], that don't capture entropic drives in design. We show that situating design problems in a more generic statistical physics framework facilitates the computation of *local* "design stress" that arises in subsystems from different competing *global* design pressures (see Fig. 4.1 for illustration). We demonstrate how global design pressures from the remainder of a system induce sub-optimal subsystem performance, which we quantify through Pareto frontiers computed using effective, or Landau,[107] free energies. The design stress is the marginal cost of moving one of the system elements in design space that results from all other element interactions. Similarly, spring tension is the marginal energy cost of an infinitesimal change in length (regardless of the intrinsic nature and linearity of the spring).

Our approach draws on work on effective interactions in soft matter systems without a clear separation of scales [159, 73, 79] and on statistical mechanics based approaches for materials design,[72, 160, 85] which we apply here at the level of systems. Using this "systems physics" approach, we compute free energies for sample systems and show how the effects of competition between design pressures drive subsystem designs into distinct classes. We also use the same method to show that it is possible to determine likely arrangements of functional units, and routings between them, independently. Our approach gives new concrete, quantitative understanding of how competing design pressures affect subsystem design in complex naval systems. Our approach can be straightforwardly generalized to other classes of design problems involving complex couplings between interconnected systems.

4.2 Systems Physics Framework

We seek a framework for studying tradeoffs in design problems. To do so, we begin from the fact that many classes of design problems can be cast in the form of a network of functional components.[157, 161] Different candidate design realizations arise from different intrinsic properties of the functional units, the topology of the network of functional connections and, possibly, the spatial embedding of the functional network. For many real-world design problems this results in a combinatorially large space of feasible design solutions.[162, 161, 41] The structure of design space determines the form of tradeoffs between design considerations.

To study how the structure of design space encodes tradeoffs, we consider a combinatorially large set of feasible designs ($\{\sigma\}$) and a set of design objectives ($\{\mathcal{O}_i\}$). A powerful approach to the design of complex engineering systems, known as Set-Based Design,[32, 33, 34, 31] involves finding candidate sets of feasible designs, as opposed to focusing on a singular optimal design.[24] Different design objectives select different corners of the full design space into the candidate set. Given the full design space and a set of specified average outcomes for the design objectives ($\{\langle\mathcal{O}_i\rangle\}$), an important task is to determine the probability (p_σ) that a given design σ would be selected for inclusion in the set of candidate designs.

To construct a set of candidate designs with average outcomes $\{\langle\mathcal{O}_i\rangle\}$ for the design objectives, information theory [100, 99] indicates that the least-biased estimate of p_σ is given by maximizing the functional

$$S = -\sum_\sigma p_\sigma \ln p_\sigma - \sum_i \lambda_i \left(\sum_\sigma p_\sigma \mathcal{O}_i(\sigma) - \langle\mathcal{O}_i\rangle \right) , \tag{4.1}$$

with respect to p_σ, where λ_i are Lagrange multipliers enforcing the constraint on candidate designs. Carrying out the maximization gives

$$p_\sigma = \frac{1}{\mathcal{Z}} e^{-\sum_i \lambda_i \mathcal{O}_i(\sigma)} \ , \tag{4.2}$$

where $\mathcal{Z}$ is a normalization constant. In principle, further algebraic manipulation could determine the λ_i and yield a precise form for p_σ. That form of p_σ would answer the question of *what* designs are likely to be selected. *Why* certain design classes are likely to be selected, however, presents an equally important question. Answering this question is important in untangling the dependence of specific design solutions on overall design priorities. To answer the "why?" question, we note that p_σ has the form of Boltzmann weight in statistical physics. Using the statistical physics approach takes us from Eq. (4.2) to the so-called partition function

$$\mathcal{Z} = \sum_\sigma e^{-\sum_i \lambda_i \mathcal{O}_i(\sigma)} \ , \tag{4.3}$$

in which each λ_i quantifies the "design pressure" of meeting corresponding design objective $\mathcal{O}_i$. By specifying how the variable design pressure affects the determination of candidate designs, the partition function provides a means to determine why candidate designs are candidates. To concretely demonstrate the power of this approach for general design problems, we use a specific problem from naval architecture. However, this approach generalizes straightforwardly to other problem classes by appropriate selection of candidate designs (σ) and design objectives ($\mathcal{O}_i$).

4.3 Arrangement Problem Model

We consider the spatial embedding of a subsystem of the overall functional network that contains only two units and a single functional connection. In both cases we choose a subsystem at random among two possible cases that differ by

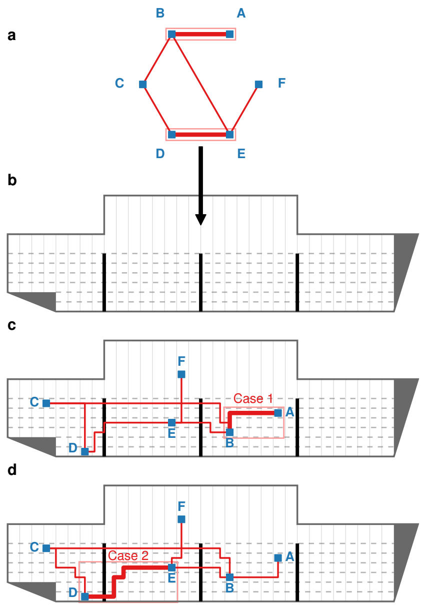


Figure 4.2: Illustration of the model for arrangement problems. The functional network (a) is embedded into an inhomogeneous space (b), here a ship hull. Spatial embeddings (c-d) require routings between connections, with two generic cases. Case 1 (c), routings that are not affected by features of the embedding space, is described in Figs. 2 and 3. Case 2 (d), routings that are affected by features of the embedding space, here bulkheads, is described in Figs. 4-6.

whether the embedding of the remainder of the functional network localizes the subsystem in a homogeneous space (Case 1), or a space that is structured by the remaining ship design (Case 2). See Fig. 4.2 for an illustration. We show below that Case 1 exhibits behavior that results from trade-offs between considerations of cost and design freedom, and Case 2 exhibits behavior that results from trade-offs between considerations of cost, design freedom, and performance.

For Case 1 we introduce a single explicit design objective $\mathcal{O}_1$ together with the corresponding design pressure λ_1, rigorously defined below. For Case 2, we additionally introduce a second design objective and pressure pair $\mathcal{O}_2, \lambda_2$ that acts concurrently with the first. The design pressure for so-called *design freedom* is not put in by hand but rather emerges organically from careful consideration of redundancy of similar design solutions through the Landau free energy technique. It is important to note that we are not trying to find specific preferred or optimal values for design pressures λ_1, λ_2. Instead, we are interested in exploring a wide range for both of them and detecting the statistical, macroscopic changes in the ensemble of solutions. Two main techniques are used for this exploration: computing the ensemble-wide statistical averages $\langle \mathcal{O} \rangle$ and computing the Landau free energy landscapes $F(S)$ to find the preferred values of some mesoscopic design feature S.

In both cases, the monetary cost expended on routing a connection between units (E) is given by the "Manhattan" distance (the sum of horizontal and vertical steps) of a minimal path between the units at some cost per unit length C. The objective for units separated by some relative Δx and Δy is

$$\mathcal{O}_1 \equiv E = C(\Delta x + \Delta y) \;, \tag{4.4}$$

and we quantify the design pressure for cost through $\lambda_1 \equiv 1/T$ where T is interpreted as a "cost tolerance". Low cost tolerance means that the design pressure to minimize costs is strong, which should lead to a preference for low cost designs. Increasing cost tolerance weakens the design pressure to minimize costs. Note that the limit of $T \to \infty$ represents complete indifference to cost as a design decision factor, rather than a preference for high cost. In statistical physics terms, E plays the role of energy, T plays the role of temperature. In addition, distinct routings and overall displacements of the units contribute entropy, a measure of the design freedom to realize distinct designs at fixed cost. We theoretically predict the critical cost tolerance $T_{\mathsf{crit}} = C/\ln 2$ that separates the cost-dominated and design freedom-dominated regimes (see SM for T_{crit} computation).

In addition, Case 2 models the performance penalty associated with routing functional connections through the bulkhead. We do so with the objective

$$\mathcal{O}_2 \equiv B \ , \tag{4.5}$$

which takes the value 1 if a routing penetrates the bulkhead and 0 if it does not. We represent the penalty for bulkhead penetration by $\lambda_2 \equiv \gamma$.

In both cases we use statistical physics to extract design information. Constitutive relations or "equations of state", evaluated via the expression

$$\langle O_i \rangle = -\frac{\partial \ln \mathcal{Z}}{\partial \lambda_i} \ , \tag{4.6}$$

quantify how outcomes for design objectives are determined by design pressure. In the specific case we consider here, fixing the design pressures through T and γ yields expected outcomes for $\langle E \rangle$ and $\langle B \rangle$, which indicate expected costs and likelihood of bulkhead penetration, respectively. Likewise, the sensitivity of design outcomes to changes in design pressure is described by "susceptibilities" that can be evaluated by further differentiation. The magnitude of susceptibility is directly related to the magnitude of fluctuations about the average design objective (see

SM for more information). We also evaluate the likely design outcomes for specific design features S_j

$$\langle S_j \rangle = \frac{1}{\mathcal{Z}} \sum_{\sigma} S_j(\sigma) e^{-\sum_i \lambda_i \mathcal{O}_i(\sigma)} \,. \tag{4.7}$$

Finally, effective, or Landau, free energies F for different system elements (e.g. unit locations, routing locations) can be computed as

$$e^{-F(S_j)} \propto \sum_{\sigma} \delta(S_j(\sigma) - S_j) e^{-\sum_i \lambda_i \mathcal{O}_i(\sigma)} \,, \tag{4.8}$$

and represent the change in the overall design objective resulting from the competition between the design pressures. Note that, because of the summation in Eq. (4.8), each value of a design feature S_j corresponds to a *bundle* of detailed design solutions σ rather than a specific one.[107] Minimal free energy corresponds to the value of design feature S of the "optimal" design bundle, whereas free energy isosurfaces represent non-optimal Pareto frontiers. Differentiating the free energy $(-\nabla F)$ yields a "design stress", which quantifies how overall, global design pressure is distributed locally among design elements in the subsystem. Similarly, "design strain" in a subsystem expresses the displacement of subsystem units or routings from optimality due to stress between subsystem and whole system design pressure. We use "displacement" to denote any deviation from the subsystem free energy minimum in the space of design feature S. For the case considered here, the design criterion itself is the spatial location (x, y) of one of the functional units, thus displacement has the usual spatial meaning as well. Details of analytic and numerical computations that yield these quantities for our model systems are described in SM.

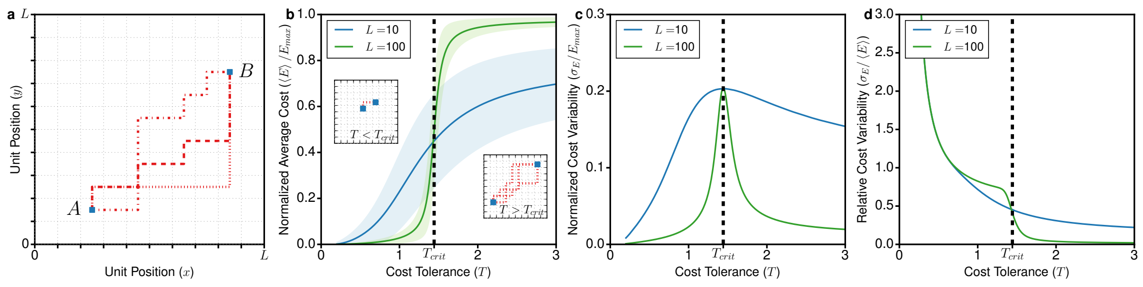


Figure 4.3: (a) Example unit positions and routings for spatially homogeneous subsystem embeddings (Case 1, see Fig. 4.2). Blue markers indicate unit positions, red lines indicate possible routings. (b) Equation of state relating cost tolerance (T) and average cost ($\langle E\rangle$, in currency) normalized by maximum possible cost expended (E_{max}) for subsystems localized in an $L \times L$ region of a ship ($L = 10$ blue curve; $L = 100$ green curve). Shaded areas indicate cost variability. Inset images illustrate typical design realizations below (condensed) and above (separated) $T_{\mathsf{crit}} = 1/\ln 2$. (c) Cost variability ($\sigma_E$, a susceptibility, see SM for mathematical details) normalized by maximum possible expenditure as a function of cost tolerance. The peak at T_{crit} for a finite sized system ($L = 100$) would correspond to a phase transition in the thermodynamic limit. (d) Cost variability normalized by average expenditure as a function of cost tolerance. Data indicate that for both large and small systems relative cost variability is large for low average cost designs.

4.4 Results

We consider the spatial embedding of a subsystem of the overall functional network that contains only two units and a single functional connection. In both cases we choose a subsystem at random among two possible cases that differ by whether the embedding of the remainder of the functional network localizes the subsystem in a homogeneous space (Case 1), or a space that is structured by the remaining ship design (Case 2). See Fig. 4.2 for an illustration. We will show below that Case 1 exhibits behavior that results from trade-offs between considerations of cost and design freedom, and Case 2 exhibits behavior that results from trade-offs between considerations of cost, design freedom, and performance.

4.4.1 Case 1, Homogeneous Embeddings: Cost/Design Freedom Trade-off

We consider the homogeneous embedding of a subsystem with two units, labeled A and B, within a homogeneous region of space, here a single watertight compartment (illustrated schematically in Fig. 4.2c). The location of A and B within the compartment, and the routing of a functional connection between them, leads, in our model system, to a trade-off between cost expenditure, E, and design freedom, measured by the routing entropy. The preferred design of this subsystem is driven by the relative importance of cost and design freedom, which we parametrize through the cost tolerance T. In Fig. 4.3a we illustrate example schematic embeddings of the subsystem of interest into a region of space of size $L \times L$. We study examples in which the subsystem is highly localized ($L = 10$) and delocalized ($L = 100$) in Fig. 4.3b-d. For both values of L we study ensembles of design solutions at a series of values for cost tolerance.

For $L = 10$, we find that there is a slowly varying, monotonic increase in average cost with increasing cost tolerance (Fig. 4.3b, blue curve). However, for $L = 100$, where the subsystem embedding is less constrained by the remainder of the network, we find a sharp increase in cost around $T_{\mathsf{crit}} = C/\ln 2$ (Fig. 4.3b, green curve). This sharp increase in cost is reminiscent of a phase transition in physical systems, and we find that the amount of absolute cost uncertainty across feasible solutions (Fig. 4.3c; akin to a susceptibility for cost) has a peak at T_{crit}. For $L = 100$, when the subsystem is less constrained, the absolute cost uncertainty is low at both low and high cost tolerance, indicating that in those regimes routings between unit pairs are almost always cheap, or almost always expensive relative to possible maximum cost. For $L = 10$, when the subsystem is more tightly constrained, the absolute cost uncertainty is large over a broad range of cost tolerances.

However, when measured relative to average cost, we find that cost uncertainty is large for both $L = 10$ and $L = 100$ in the limit of low cost tolerance. Fig. 4.3d shows that relative cost uncertainty diverges as cost tolerance goes to zero. This result means that even though, as expected, low cost tolerance leads to low cost designs for the subsystem of interest, possible design outcomes show uncertainty of 100% or more in terms of average cost. Though this effect might not be a large design concern if it occurred only in the subsystem of interest, we note that our choice of subsystem was arbitrary, so that every subsystem in the network should exhibit this effect. A cascade of such occurrences throughout a large functional network in a complex product, such as a ship, would lead to large macroscopic fluctuations in cost of the overall design.

For $L = 100$, Fig. 4.3d indicates that as the cost tolerance increases across the critical value, there is a sharp drop in the cost uncertainty relative to average cost, that is driven by the sharp increase in average cost seen in Fig. 4.3b. This indicates that above the critical cost tolerance candidate designs are high cost, but show relatively small cost uncertainty. Taken together, the features of the relative cost uncertainty curve indicate a fundamental trade-off: tight cost constraints lead to wild relative cost uncertainty, whereas low relative cost uncertainty can only be achieved at large cost.

To make the origin of these behaviors more concrete, in Fig. 4.4 we fix the position of one of the units to be the origin, and examine how the design pressures from cost (Fig. 4.4a) and design freedom (Fig. 4.4b) influence the (x, y) location of the second unit. We use the (x, y) location of the second unit as a design feature S to compute the free energy and Pareto frontiers (see Eqn. (4.8) and SM). For the case of $L = 30$, we plot one quadrant, the other quadrants being

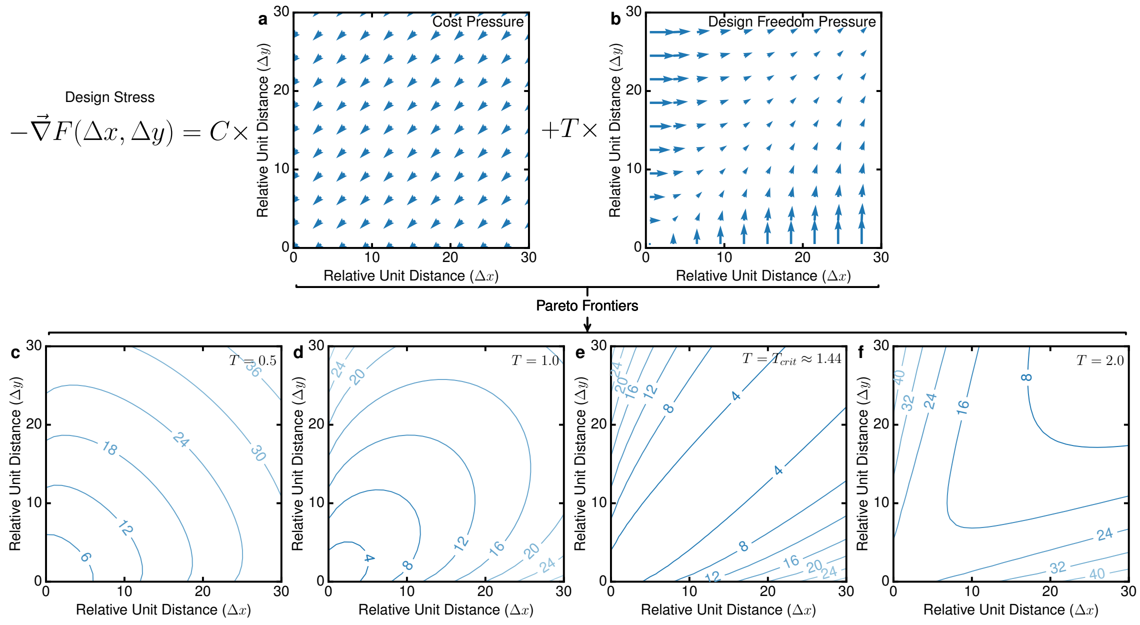


Figure 4.4: Pressure from overall design objectives induces stress on subsystem design elements. For spatially homogeneous subsystem embeddings (Case 1, see Fig. 4.2) design stress (measured in units of cost tolerance T) can be decomposed into contributions from cost pressure (panel a) and design freedom pressure (panel b). Depending on the relative strength of the design pressures, the different phase behaviors in Fig. 4.3 originate from underlying subsystem effects, illustrated in panels c-f. Panels c-f plot Pareto frontiers (Landau free energy isosurfaces) that indicate equivalent, suboptimal subsystem designs that could arise if the subsystem design was forced to sacrifice performance to the remainder of the system. At low cost tolerance ($T = 0.5$ c; $T = 1.0$ d) units are preferentially condensed. At high cost tolerance ($T = 2.0$ f) units are preferentially separated. At the critical cost tolerance ($T = T_{\mathsf{crit}} = 1/\ln 2$ there is no preferred separation distance.

related by symmetry. Arrows indicate the relative magnitude and direction of stress that each different form of design pressure induces on the location of the second unit. Comparing Fig. 4.4 panels a and b shows that cost and design freedom pressures act in different directions with cost driving the units closer together and design freedom driving them further apart. The balance between these forces is determined by the cost tolerance, and leads to qualitatively different outcomes depending on this value, which can be seen in the Pareto frontiers plotted in Fig. 4.4c-f. For physics readers, we note that Pareto frontiers correspond to isosurfaces of the Landau free energy (see, e.g., Ref. [107]) for unit locations. We plot Pareto frontiers describing the deviation in design feature space S from the optimal overall objective at a series of cost tolerances. We stress that the "optimal" value of design feature is merely a local minimum of free energy and is not necessarily the value to be chosen. The reason for considering non-optimal solutions is that any subsystem is only part of the overall design, and we do not expect that, in general, overall optimal designs will correspond to optimal outcomes for all subsystems. Non-optimal Pareto frontiers provide a means of communicating how design pressure from the rest of the functional network could be expected to influence the behavior of a subsystem.

When we compute the corresponding Pareto frontiers, we find that at low cost tolerance ($T = 0.5$; Fig. 4.4c), units are condensed, since the behavior is dominated by cost minimization, which is characterized by Pareto frontiers with constant $x+y$ in the limit of $T = 0$. Increasing cost tolerance alters the balance between cost and design freedom. Even below the critical tolerance ($T = 1.0$; Fig. 4.4d), this causes a change in shape in the Pareto frontiers. At the critical cost tolerance ($T = T_{\mathsf{crit}}$; Fig. 4.4e) Pareto frontiers more closely resemble surfaces with constant $x-y$ rather than

$x + y$ as we found at low cost tolerance. Above the critical cost tolerance ($T = 2.0$; Fig. 4.4f), Pareto frontiers reverse their order with low free energy locations for the location of the second unit forced to the boundary.

4.4.2 Case 2, Inhomogeneous Embeddings: Cost/Design Freedom/Performance Trade-offs

We next consider the additional design pressure that arises from an inhomogeneous embedding space. For concreteness, we represent this as a bulkhead within the ship hull. Bulkheads are features designed to prevent water that enters the hull through a breach from filling all parts of hull and sinking the ship. Routings through a bulkhead are expensive and also can reduce a distributed system's effectiveness, and thus overall ship performance. Hence, additional performance pressure arises in the case that elements of a subsystem are located in different bulkhead compartments (schematic illustration in Fig. 4.2c). We parametrize it with bulkhead penalty γ, acting as the second design pressure in the system. Again, from a large functional network we randomly choose a subsystem comprised by a pair of units with a single functional connection. However, we assume that the connections between the subsystem of interest and the remainder of the functional network drive the location of one unit to be on one side of the bulkhead and the other unit to be on the opposite side. Both units are vertically constrained to be below the top of the bulkhead. We allow two types of routings between the units to study their trade-off: one routes along the shortest path through the bulkhead and suffers the penalty γ; the other routes along the shortest path around the bulkhead, with no penalty. For concreteness we give results for systems of fixed size (20×20 with a vertical bulkhead in the middle) which are representative of the general behaviors we observe. See SM for results for other system sizes.

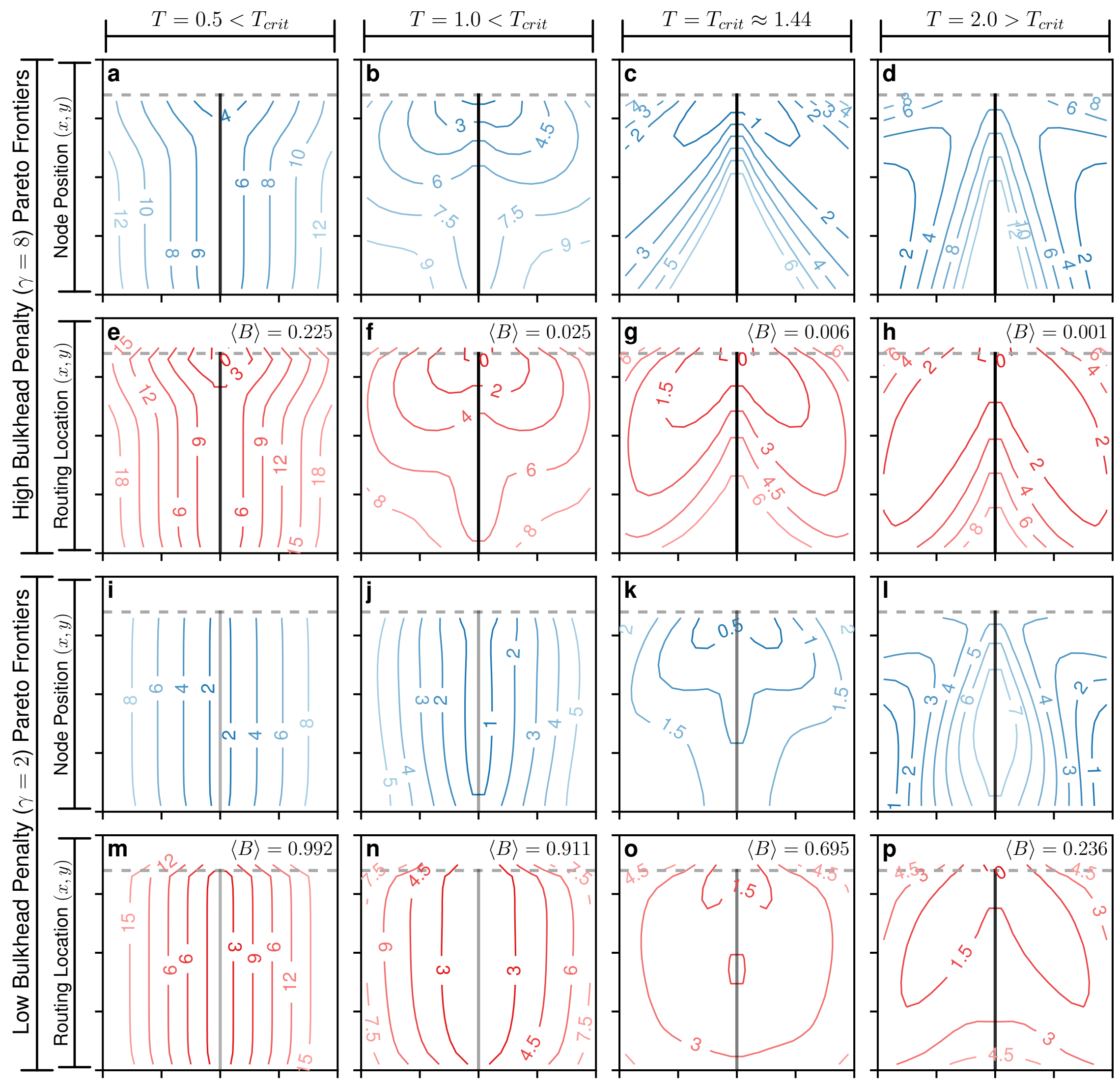


Figure 4.5: Pareto frontiers (Landau free energy isosurfaces) for unit and routing locations for spatially inhomogeneous subsystem embeddings (Case 2, see Fig. 4.2) for different cost tolerances T ($T = 0.5$ first column, $T = 1.0$ second column, $T = T_{\text{crit}}$ third column, $T = 2.0$ fourth column) and performance penalties for bulkhead penetration ($\gamma = 8$ top two rows, $\gamma = 2$ bottom two rows). Each panel shows the 20×20 cell domain in which the units can be placed, split in the middle by a bulkhead of height 17. The (x, y) coordinates correspond to the possible positions of functional units and routings. Blue curves indicate unit positions, normalized so that most favorable unit locations have value 0, with increasing values indicating the loss in subsystem objective in units of the cost tolerance. Red curves indicate routing locations, normalized so that locations through which connections route with absolute certainty have value 0, and increasing values indicate the reduction in subsystem objective of routing through a given location in units of cost tolerance. Dashed horizontal lines indicate the upper boundary of the domain where the units may be placed. Thick black vertical lines indicate the position of the bulkhead, their transparency is color-coded by $\langle B \rangle$. Solid bulkhead indicates that it serves as a significant obstacle and routings would run around it (low $\langle B \rangle$). Transparent bulkhead indicates that it is relatively easy and likely to route through (high $\langle B \rangle$).

Compared with Case 1, breaking spatial homogeneity makes the relationship between route paths and unit locations more complicated. This complication arises because routings now couple to both unit positions and geometric features. Because of this, we study unit positioning and routing separately. As in Case 1, we compute Pareto frontiers via Landau free energies, but in this case we do so by integrating out the degrees of freedom of units and routings separately. Fig. 4.5 shows Pareto frontiers for unit routing positions as a function of cost tolerance for bulkheads with representative high ($\gamma = 8$, panels a-h) and low ($\gamma = 2$, panels i-p) bulkhead penalty. The difference of $\Delta\gamma = 6$ between the two values implies that the relative statistical weight of routing through the bulkhead changes roughly by a factor of $e^6 \sim 400$, and the effects on node positioning are immediately visually apparent. Also apparent is the effect of $\Delta\gamma$ on design performance, characterized by the $\langle B \rangle$, i.e. the fraction of all designs that route through the bulkhead. See SM for computation details.

At high bulkhead penalty ($\gamma = 8$), and low cost tolerance ($T = 0.5$) Pareto frontiers for unit locations (Fig. 4.5a) and routing (Fig. 4.5e) both indicate strong coupling to the top of the bulkhead. Results for increased cost tolerance ($T = 1.0$) that is still below T_{crit} indicate that unit locations are less strongly coupled to the bulkhead (Fig. 4.5b). Comparison with results for routing (Fig. 4.5f) indicate that this coincides with a drop in the fraction of designs that route through the bulkhead by nearly an order of magnitude ($\langle B \rangle = 0.025$ at $T = 1.0$, c.f. $\langle B \rangle = 0.225$ at $T = 0.5$), and though routes remain strongly localized at the top of the barrier, Pareto frontiers at equivalent objective cost (free energy) are further from the bulkhead. These trends continue through T_{crit} (Fig. 4.5c,g). However, above T_{crit} ($T = 2.0$) Fig. 4.5d we observe that although the units delocalize from the bulkhead

(Fig. 4.5d), the routings remain strongly coupled to the top of the bulkhead, and the probability that a design routes through the bulkhead drops to $\langle B \rangle = 0.001$. Comparing unit locations (Fig. 4.5a-c) and routing locations (Fig. 4.5e-g) indicates that at or below T_{crit} unit locations are correlated with routing locations. However, above T_{crit} (Fig. 4.5d,h) the most probable unit locations do not correspond to most probable routing locations.

We contrast the above results at high bulkhead penalty ($\gamma = 8$, Fig. 4.5a-h) with low bulkhead penalty ($\gamma = 2$, Fig. 4.5i-p). At low cost tolerance ($T = 0.5$) we see that relaxing the bulkhead penalty still causes the unit positions to localize near the bulkhead (Fig. 4.5e) but the units no longer localize near the top of the bulkhead as they did at high bulkhead penalty (Fig. 4.5a). Likewise, routings no longer localize near the top of the bulkhead (Fig. 4.5m), but follow the unit locations and pierce the bulkhead with high probability ($\langle B \rangle = 0.992$). At increased cost tolerance ($T = 1.0, T_{\mathsf{crit}}$) the localization at the top of the bulkhead appears again (Fig. 4.5j-k,n-o). At high cost tolerance $T = 2.0$ the units again delocalize from the bulkhead (Fig. 4.5l,p) and the cases $\gamma = 2$ and $\gamma = 8$ start looking very similar.

To further understand the competing design pressures of cost, design freedom, and performance, we compute design stress in unit positioning (see Fig. 4.6). At a given unit position (corresponding to "strain" in the language of materials science) design stress indicates the magnitude and direction in which changing the placement of the unit would lead to the greatest decrease in the overall objective cost for the subsystem. We find that at low cost tolerance ($T = 0.5$, Fig. 4.6a,e), design stress is directed primarily toward the bulkhead, with discernible stress toward the top of the compartment for high cost penalty. An increase in cost tolerance ($T = 1.0$, Fig. 4.6b,f) leads to similar design stress at low bulkhead penalty (Fig.

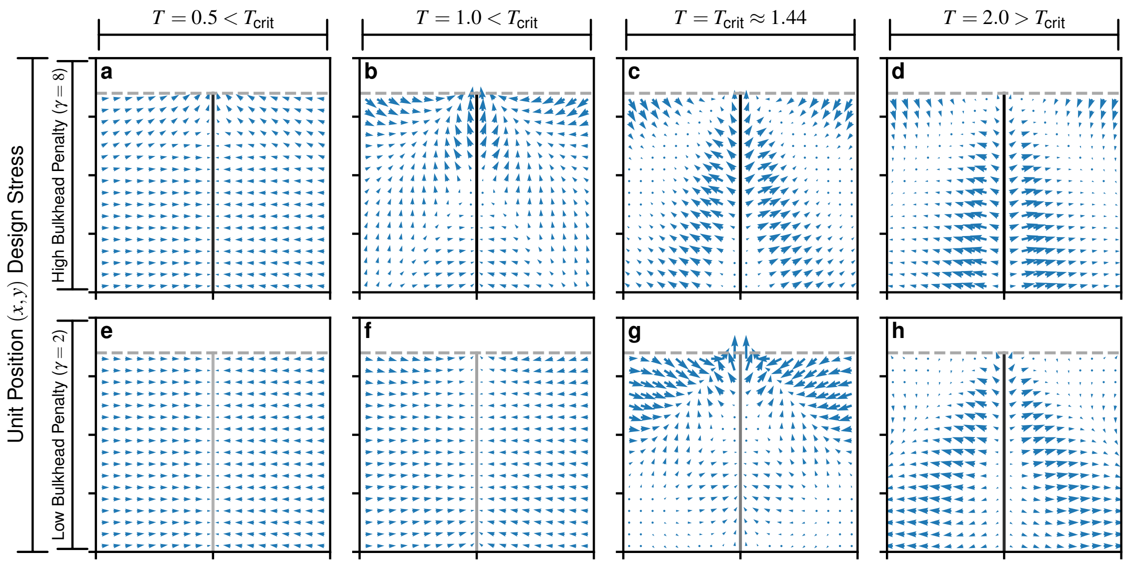


Figure 4.6: Design stress for unit locations in spatially inhomogeneous subsystem embeddings (Case 2, see Fig. 4.2) for different cost tolerances T ($T = 0.5$ first column, $T = 1.0$ second column, $T = T_{\text{crit}}$ third column, $T = 2.0$ fourth column) and performance penalties for bulkhead penetration ($\gamma = 8$ top row, $\gamma = 2$ bottom row). Plots indicate that if a unit was sited at the origin of an arrow in response to whole system design pressure, design pressure acting on the subsystem alone would drive the unit in the direction of the arrow, with a strength proportional to the length of the arrow.

4.6f) but a more intricate pattern of stress at high bulkhead penalty (Fig. 4.6b) that includes regions with stress toward and away from the both the bulkhead and the top of the compartment. Similarly, complex patterns of stress occur at both low and high bulkhead penalty at T_{crit} (Fig. 4.6c,g). At high cost tolerance ($T = 2.0$, Fig. 4.6d,h), the pattern of design stress is predominantly away from the bulkhead.

The behaviors we find that arise from the competition between cost, design freedom, and performance design pressures can be classified qualitatively according to the phase diagram in Fig. 4.7. In Fig. 4.7 we show, schematically, the effects of bulkhead penalty γ and cost tolerance T on bulkhead penetration (a) and relative unit distance (b). The combination of these effects also results in a complicated emergent relationship between the vertical positions of the units (c). To provide a more concrete and quantitative example, panels (d) and (e) show respectively the

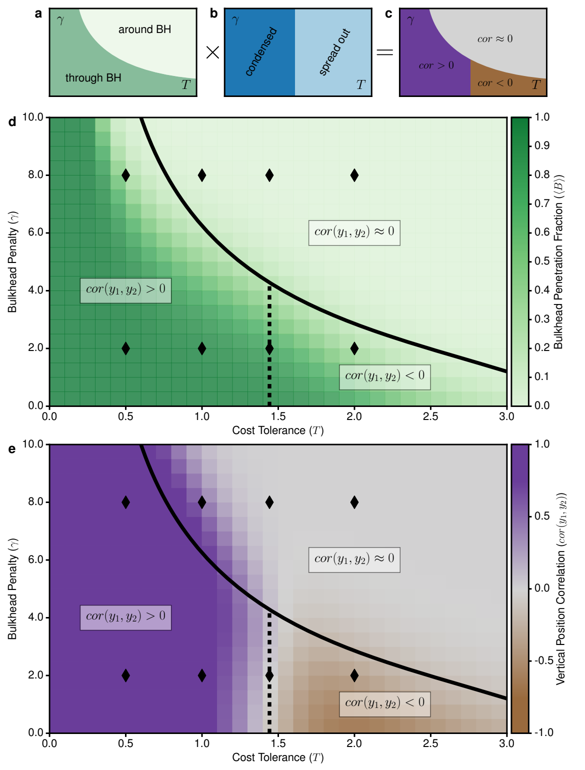


Figure 4.7: Phase diagram for spatially inhomogeneous subsystem embeddings (Case 2, see Fig. 4.2), summarizing effects of performance penalties associated with bulkhead penetration (γ) and cost tolerance (T). (a) Schematic illustration of effects of T and γ on bulkhead penetration (performance proxy) for arbitrary subsystem localization. (b) Schematic illustration of effects of T and γ on unit separation (cost proxy) for arbitrary subsystem localization. (c) Schematic illustration of combination of bulkhead penetration (performance proxy) and unit separation (cost proxy) on vertical correlation in unit layout (architecture-class proxy) for arbitrary subsystem localization. (d) Quantitative phase plot for subsystem localization of fixed size $L = 20$. Green shade indicates average system performance (bulkhead penetration probability). (e) Quantitative phase plot for vertical correlation in unit layout (architecture-class proxy). Markers in (d-e) indicate T, γ values corresponding to plots in Figs. 4.5 and 4.6.

bulkhead penetration fraction and the correlation in vertical node positions for the same system of size $L = 20$.

4.5 Conclusion

We developed a general, statistical physics framework for analyzing complex design problems. We demonstrated the application of this framework to characterizing tradeoffs between competing design presures. For concreteness, we studied trade-offs between competing design pressures of cost, design freedom, and performance in arrangement problems from naval architecture design. We analyzed ship models by applying physics principles at the systems-level and found a rich pattern of behavior. We gave an explicit formulation of Pareto frontiers in terms of isosurfaces of Landau free energy, and computed "design stress" induced by sub-optimal subsystem embedding. Our framework recasts common design challenges in terms of the well-understood concepts of pressure, stress and strain. We find that these concepts, which are typically used to characterize the behavior of materials, also provide a means of characterizing system-level behavior.

Our approach opens new avenues for addressing design challenges that arise in complex systems. Our framing of system design in terms of statistical mechanics has some technical overlap with optimization approaches based on simulated annealing.[21] Simulated annealing invokes thermodynamics by using a fictitious Hamiltonian cooled *in silico* to zero temperature to find the global minimum of an objective function. Our approach with minimally biased probability distributions, though derived from information theory, is mathematically equivalent to a fictitious Hamiltonian held at a constant *finite* temperature. Maintaining finite temperature highlights the role of design pressures that arise from design freedom and become relevant in combinatorially large optimization spaces, and in early stage design.[27] We believe this approach can give important information about the systems of interest that could enable human designer choices. The separation of subsystem designs into different architecture classes can enable designers to communicate about qualitative style choices. Knowledge about where the paths between the units are likely to route, even if the unit locations are not specified, and vice versa, could facilitate the control of ship outfit density. Knowledge about tradeoffs between cost constraints and cost variability could inform aspects of the design project around predictable and consistent expenses. Understanding how different design objectives create design stress on subsystems could facilitate educated choices of sub-optimal designs for individual subsystems in the service of optimizing the system as a whole. All of these forms of knowledge are crucial in the early design stages of a broad class of complex design problems.

Finally, physics concepts and principles are typically used to understand the behavior of a part of a larger system. E.g. for a ship it is common to: use the physics of electromagnetism to understand the function of a radar; use materials physics to understand the properties of a hull; use solid state physics to understand the properties of electronics; use hydrodynamics to understand the interaction of a hull with water; use thermodynamics to understand the function of an engine. Here, without explicit reference to the underlying physical nature of component subsystems, we show that the principles of statistical mechanics give rise to direct analogs of familiar macroscopic physics concepts, such as pressure, stress, and strain, as well as provide new insight into the architecture of the ship as a whole. Our focus on an established,[157] minimal model of ship design was motivated both by pressing challenges in naval architecture, and by the goal of providing a concrete, self-contained example of our approach. However, our "systems physics" approach generalizes straightforwardly in several respects: to more detailed models of naval architecture, to subsystems with more units, and more complex functional connections, and, most importantly, to other classes of systems-level design problems. Systems-level applications of physics have led to constructive engagements between physics and economics,[163, 164] network science,[165, 50] and epidemiology.[166, 167] We believe the present systems-level application of physics will lead to a similar constructive engagement with design problems in a wide variety of domains.

4.6 Supplementary Methods

4.6.1 Evaluation of Outcomes and Variability

The importance of the partition function $\mathcal{Z}$ is in that it contains all information about statistical averages of design objectives. The average outcome for a design objective is given by

$$\begin{aligned}\langle \mathcal{O}_i \rangle &= \sum_\sigma \mathcal{O}_i(\sigma) p_\sigma = \frac{1}{\mathcal{Z}} \sum_\sigma \mathcal{O}_i(\sigma) e^{-\sum_j \lambda_j \mathcal{O}_j(\sigma)} \\ &= -\frac{\partial}{\partial \lambda_i} \ln \mathcal{Z} \ . \end{aligned} \tag{4.9}$$

The variability in an outcome can be evaluated by further differentiation

$$\left\langle \mathcal{O}_i^2 \right\rangle_c = \left\langle \mathcal{O}_i^2 \right\rangle - \left\langle \mathcal{O}_i^2 \right\rangle = \left(-\frac{\partial}{\partial \lambda_i} \right)^2 \ln \mathcal{Z} \ . \tag{4.10}$$

The variability in an outcome is directly related to the sensitivity of the design objective average to design pressures:

$$\left\langle \mathcal{O}_i^2 \right\rangle_c = -\frac{\partial}{\partial \lambda_i} \left\langle \mathcal{O}_i \right\rangle \ . \tag{4.11}$$

4.6.2 Case 1: Computation Details

Two units at separated by distances $\Delta x, \Delta y$ along the two axes can be joined with multiple routings of the same Manhattan length $\Delta x + \Delta y$. Since the routing consists of a fixed number of vertical and horizontal steps which can be taken in any order, the number of possible routings is given by the binomial coefficient

$$n(\Delta x, \Delta y) = \begin{pmatrix} \Delta x + \Delta y \\ \Delta x \end{pmatrix} = \frac{(\Delta x + \Delta y)!}{\Delta x! \Delta y!} \ . \tag{4.12}$$

The number of paths $n(\Delta x, \Delta y)$ grows rapidly with path length, thus creating the entropic stress pushing the units apart. The partition function (Eq. 2 in main text) is a sum over all candidate designs. We separate that sum into summing over possible unit positions and possible paths. For two units this becomes

$$\mathcal{Z} = \sum_\sigma e^{-E_\sigma/T} = \sum_{x_1, y_1, x_2, y_2} e^{-E_\sigma/T} n_\sigma(\Delta x, \Delta y) \ . \tag{4.13}$$

To understand the origin of T_{crit} more intuitively, we can use an approximation. If we assume large separations $\Delta x \sim \Delta y \ll 1$, Stirling's approximation for the binomial coefficient gives

$$\ln n(x, y) \approx x \ln\left(1 + \frac{y}{x}\right) + y \ln\left(1 + \frac{x}{y}\right) \approx (x + y) \ln 2 \tag{4.14}$$

Substituting this into the partition function gives

$$\mathcal{Z} = \sum_{x_1, y_1} \sum_{\Delta x, \Delta y} \exp\left(\left(\ln 2 - \frac{C}{T}\right)(\Delta x + \Delta y)\right) , \tag{4.15}$$

where the unit separation $\Delta x, \Delta y \in [0, L]$. Because the contributions that correspond to energy and entropy have the same form, depending on the sign of $(\ln 2 - C/T)$, this is either a descending or ascending finite geometric series. In either case it evaluates to a finite value that changes rapidly near $T = T_{\mathsf{crit}} = \frac{C}{\ln 2}$. The average cost and the cost variance/susceptibility are evaluated with straightforward derivatives with respect to $\lambda_1 = 1/T$.

4.6.3 Case 2: Computation Details

We consider two types of routings between the two units: through the bulkhead and around the bulkhead. Since they are mutually exclusive, the partition function can be computed as

$$\mathcal{Z} = \mathcal{Z}_{\mathsf{t}} e^{-\gamma} + \mathcal{Z}_{\mathsf{r}} \; , \tag{4.16}$$

with the two component partition functions being

$$\mathcal{Z}_t = \sum_{\sigma \text{ through}} \exp(-E(\sigma)/T) \tag{4.17}$$

$$\mathcal{Z}_r = \sum_{\sigma \text{ around}} \exp(-E(\sigma)/T) \; . \tag{4.18}$$

The fraction of bulkhead penetrations, $\langle B \rangle$, is a design objective conjugate to bulkhead penalty, thus it can be evaluated via a partial derivative

$$\langle B \rangle = -\frac{\partial}{\partial \gamma} \ln \mathcal{Z} = \frac{Z_t e^{-\gamma}}{Z_t e^{-\gamma} + Z_r} \; . \tag{4.19}$$

Free energy landscapes $F(x, y)$ are computed via Eq. 8 of main text with the “design feature” $S(x, y)$ evaluated as follows, respectively for unit and routing free energies:

$$S_{\mathsf{unit}}(x, y) = \begin{cases} 1, & \text{there is a unit at } (x, y) \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

$$S_{\mathsf{route}}(x, y) = \begin{cases} 1, & \text{there is a routing through } (x, y) \\ 0, & \text{otherwise} \end{cases} \tag{4.21}$$

The vertical node correlation is defined the usual way with averages taken in the sense of Eq. 7 of main text:

$$cor(y_1, y_2) = \frac{\langle y_1 y_2 \rangle_c}{\sqrt{\langle y_1^2 \rangle_c \langle y_2^2 \rangle_c}} \tag{4.22}$$

4.7 Supplementary Results

For the inhomogeneous embedding (Case 2) unit positions explicitly couple to the geometric features of the embedding space, as shown in Fig. 4 (main text) for a system size of 20×20. SI Figs. 4.8, 4.9, and 4.10 show identical computations performed for a series of other system sizes. Fig. 4.8 depicts a system size of 20×40. Fig. 4.9 depicts a system size of 40×12. Fig. 4.10 depicts a system size of 20×20, with the bulkhead off-center.

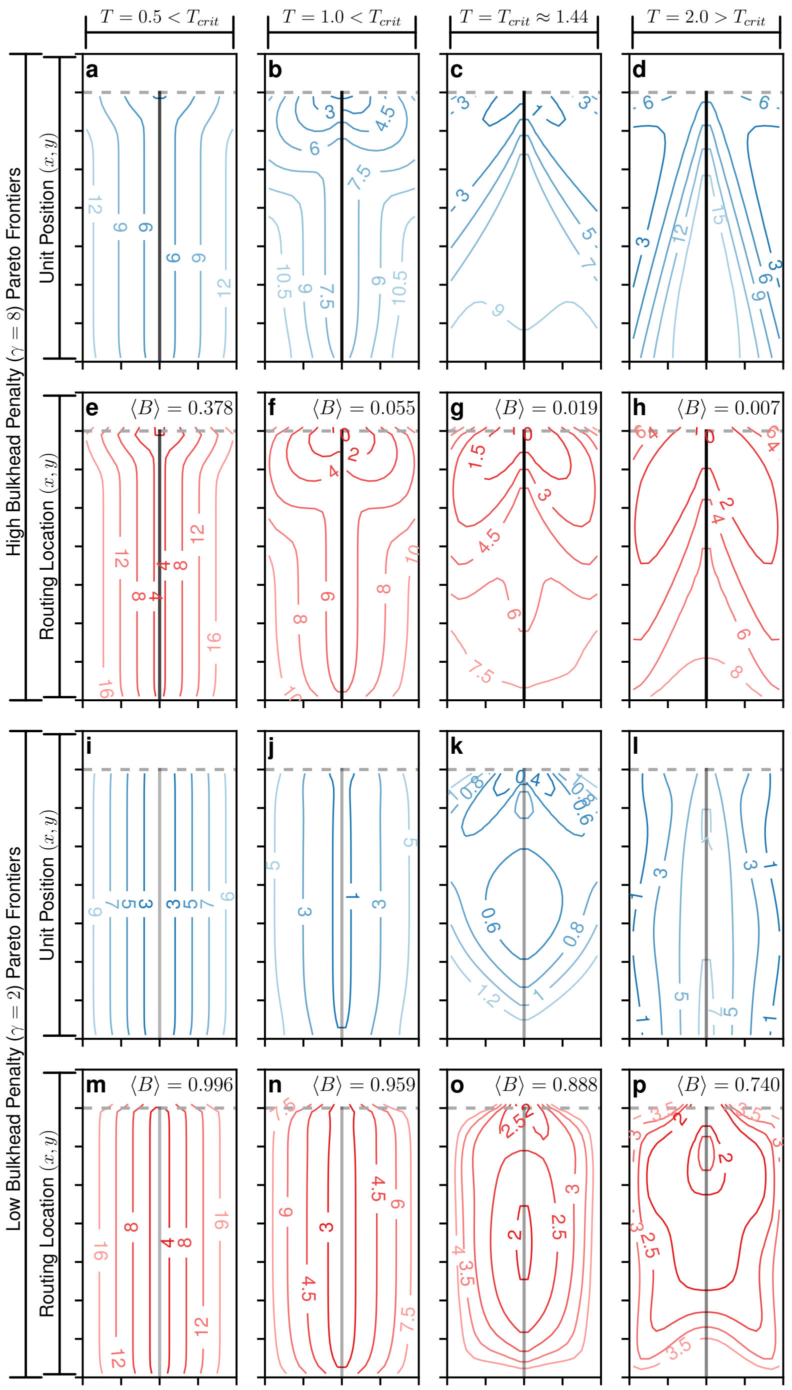


Figure 4.8: Pareto frontiers (Landau free energy isosurfaces) for unit and routing locations for spatially inhomogeneous subsystem embeddings (Case 2). System size is 20×40 (width and height), bulkhead is positioned horizontally in the center and extends up to height $h = 35$. Normalization for free energies is identical to that of Fig. 4 of main text. Organization of rows and columns of (T, γ) parameters is also identical to that of Fig. 4 of main text.

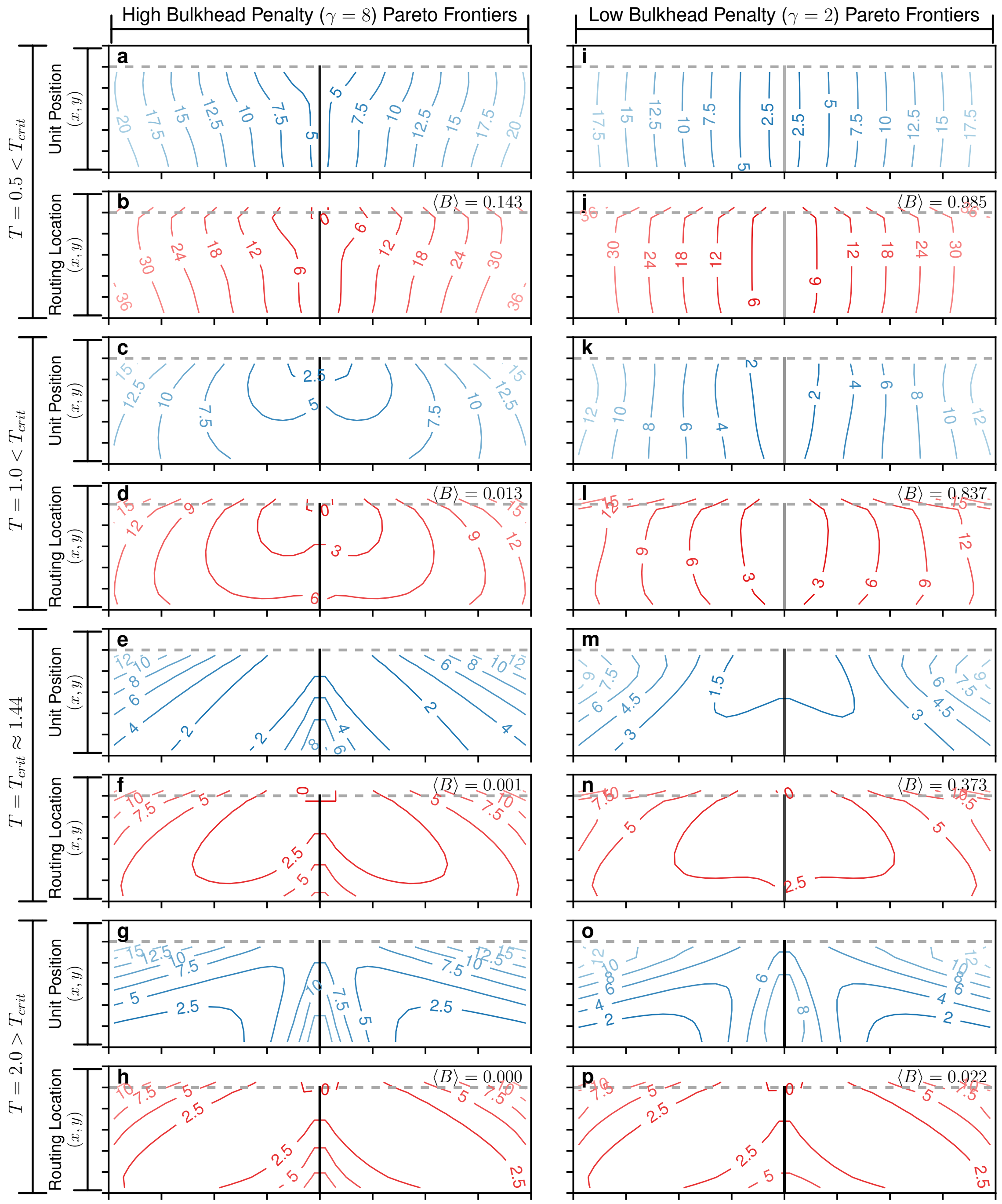


Figure 4.9: Pareto frontiers (Landau free energy isosurfaces) for unit and routing locations for spatially inhomogeneous subsystem embeddings (Case 2). System size is 40×12 (width and height), bulkhead is positioned horizontally in the center and extends up to height $h = 10$. Normalization for free energies is identical to that of Fig. 4 of main text. First column corresponds to $\gamma = 8$, second to $\gamma = 2$. Four rows of graph pairs correspond to cost tolerances of $T = 0.5, 1.0, T_{\mathsf{crit}}, 2.0$, respectively.

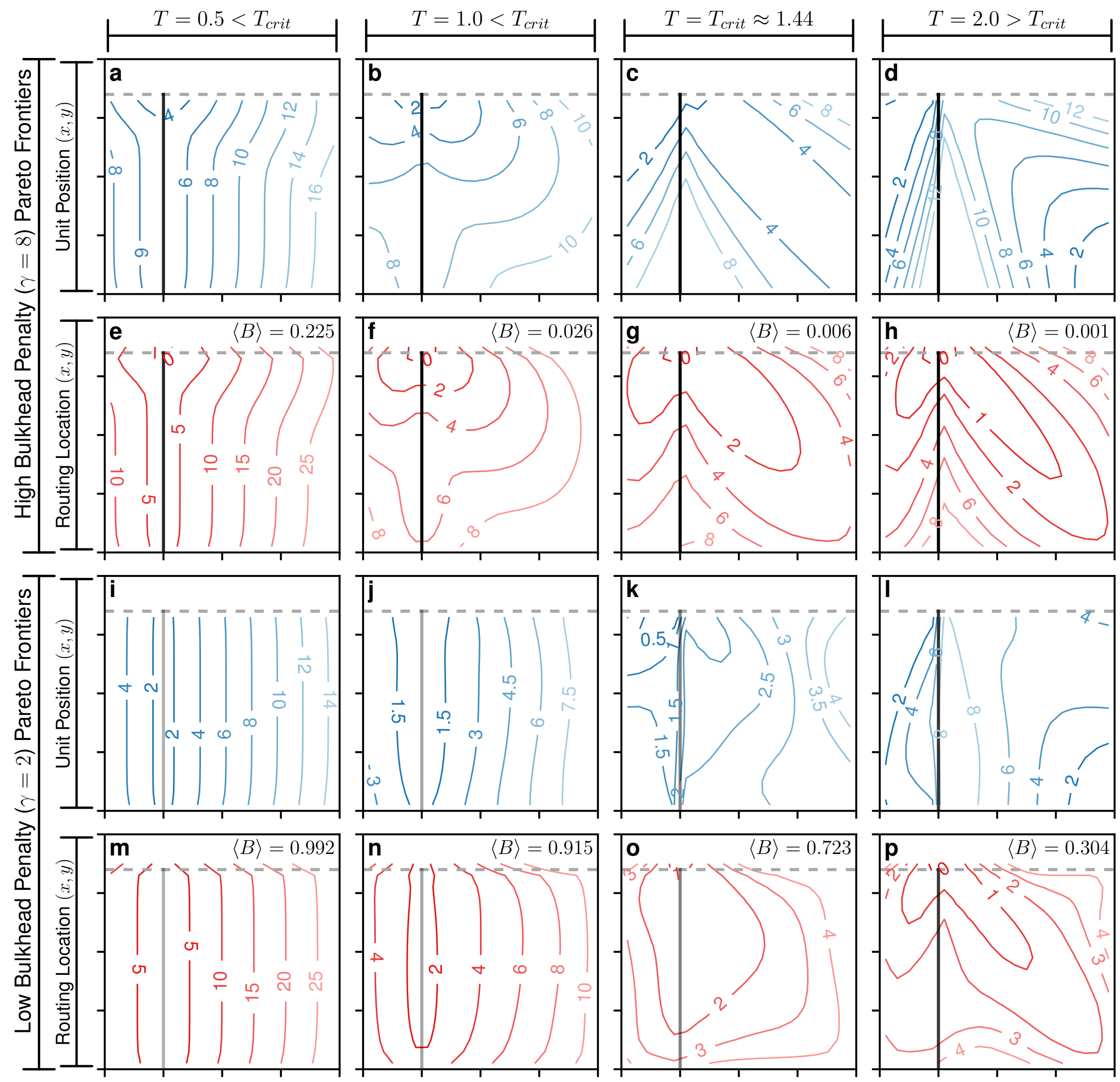


Figure 4.10: Pareto frontiers (Landau free energy isosurfaces) for unit and routing locations for spatially inhomogeneous subsystem embeddings (Case 2). System size is identical to that in Fig. 4 of main text at 20×20 (width and height), however bulkhead is positioned horizontally off center at $x_{\mathsf{bh}} = 5$ and extends to the same height $h = 17$ as in Fig. 4 of main text. Normalization for free energies is identical to that of Fig. 4 of main text. Organization of rows and columns of (T, γ) parameters is also identical to that of Fig. 4 of main text.

CHAPTER V

Robust Design

5.1 Introduction

[1]Modern manufacturing and industrial development demand both robust products and robust designs. Whereas robust products exhibit similar, predictable behavior in a variety of operating conditions, robust designs preserve design elements under uncertainty in problem statements (see Fig. 5.1).[169, 170] Achieving robust design enhances supply chain stability, avoids rework, and thus reduces downstream cost and performance uncertainty.[171, 28] Minimizing these uncertainties through robust design has become both increasingly important, and increasingly difficult to achieve, as products coming to market incorporate broader arrays of functionality that rely on the integration of heterogeneous subsystems.[5] The coupling of heterogeneous subsystems restricts subsystem component specifications, and small changes in the design of one subsystem can trigger avalanches of change in connected subsystems.[157] Preventing or controlling avalanches requires developing the ability to not only describe subsystem interdependencies, but also how interdependencies affect the robustness of subsystem and overall design. (see Fig. 5.2).

Throughout engineering, the design of system elements often exploits known physical phenomena, leveraging knowledge developed through decades or centuries

[1]This chapter is based on the paper [168] coauthored with A. Kirkley, D.J. Singer, and G. van Anders.

of investigation of basic physical science principles. For example, the principles of robustness of engineering materials have a long history and a rich language and mathematical apparatus. In this language, intuitive contrasts such as “brittle” vs “ductile” or “strong” vs “weak” achieve precise meaning in terms of performance thresholds on stress, or localized force, and strain, or localized displacement.[172] Unlike the study of materials, the study of the basic physical phenomena that underlie the behavior of systems integration is in its relative infancy.[173, 174, 175] Because of this relative infancy, what it means to be robust, and how to quantify robustness are open questions.

Here, we operationalize questions about the robustness of subsystem design via the “Systems Physics” approach of Chapter IV. Via Systems Physics we draw quantitative comparisons between *design* classes, or architectures, at intermediate, “mesoscale” levels of analysis. We find that at the mesoscale, the robustness of architecture classes can be rigorously discussed in precisely the same terms that are used to quantify the robustness of materials, i.e., in terms of stress–strain relationships. The design stress–strain curves, in the distributed systems we study, generically exhibit strain softening, i.e. a decrease in design stress with increasing design strain. By, focusing on stress and strain thresholds, we classify the mesoscale designs by their response to external subsystem coupling as “brittle” or “ductile” and “strong” or “weak”. We show that the stress–strain analysis can be concisely summarized in two-factor robustness plots that directly compare system architectures. For concreteness, we show explicit examples of brittle, ductile, strong, and weak designs that arise in the context of a naval architecture-inspired arrangement problem. We show that local architecture classes can change between brittle and ductile behavior depending on the form of global design pressure. This analysis of

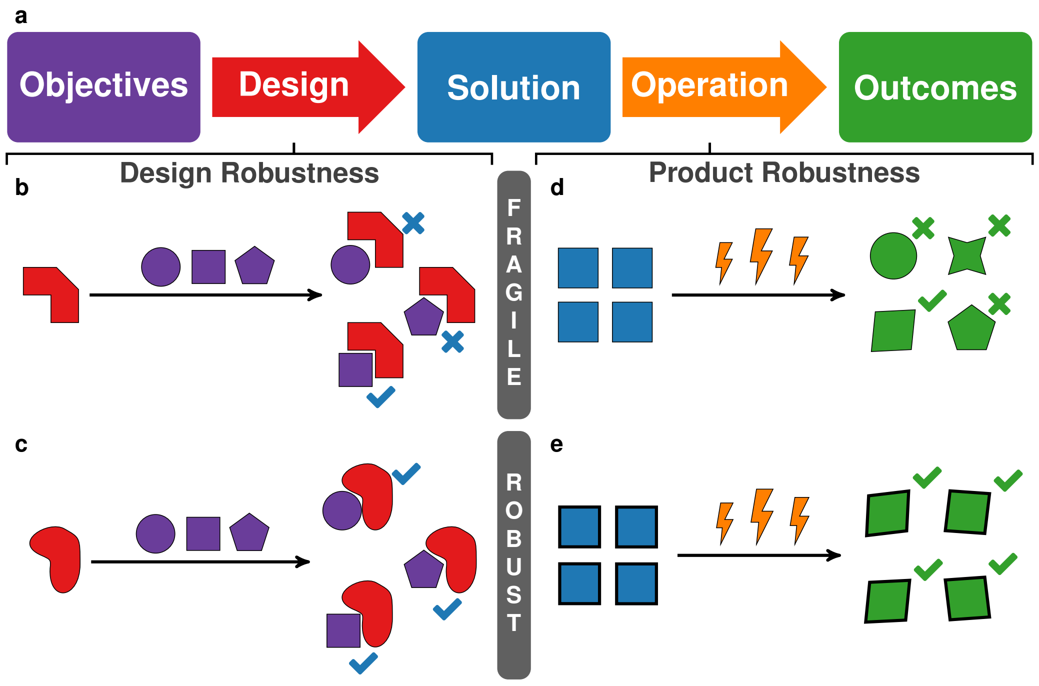


Figure 5.1: Schematic representation of the difference between a robust design and a robust product. (a) The pathway of product design and operation can be represented as a flow from a set of objectives, to a design process that produces a solution or product, followed by the testing and operation of the product results in one or more outcomes. The robustness of the product describes the product's performance under different operating conditions. Robust products (panel e, blue squares) perform well under different operating conditions (lightning bolts), whereas "fragile" products (panel d) perform poorly. An analogous classification can also be applied to the design process. A robust design (panel c) is one in which the same solution (red shape) would be produced to meet different sets of objectives (purple shapes), whereas a fragile design (panel b) would not stand up to changes in the objectives.

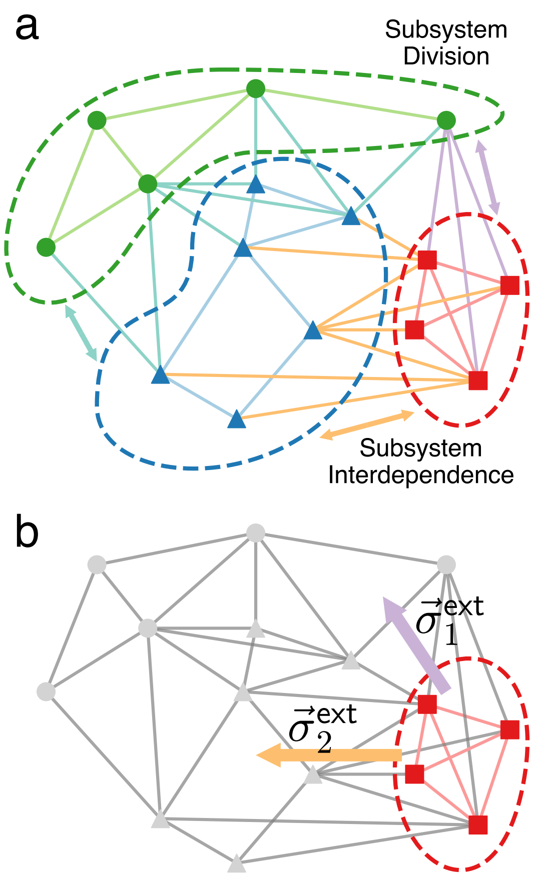

a
Subsystem Division
Subsystem Interdependence
b
$\vec{\sigma}_1^{\text{ext}}$
$\vec{\sigma}_2^{\text{ext}}$

Figure 5.2: Schematic representation of interdependencies in a complex, integrated system. (a) Complex systems can be comprised of a set of connected, interdependent subsystems (distinguished by color). (b) A common design problem is to determine the relationship of a component subsystem (in red) to the other subsystems (gray). External subsystems induce "design stress" ($\vec{\sigma}_{1,2}^{\textbf{ext}}$) on a subsystem that we use to characterize robustness.

local manifestations of global design drivers provides a novel form of insight into a ubiquitous set of challenges faced in industrial design, as well as a new means of communicating about and achieving robust design.

5.2 Robust Design from Statistical Physics

Engineering complex systems is a difficult, longstanding problem. Early systematic design paradigms prescribed optimizing subsystems sequentially and independently, in the hope of forming a design "spiral" that narrows to a final, single solution.[24] These "point-based" design approaches rely heavily on mathematical optimization.[20] Optimal solutions, however, are only as good as the underlying models that produce them, and a key source of model uncertainty is the interaction of the model with external subsystems. The difficulty of design lies not in the individual subsystems but in their integration.[27] Large systems further compound the potential for integration failure.[11] The failure potential can be mitigated by focusing not on finding "good" solutions, but by focusing on avoiding "bad" ones. Broad-based, so-called "Set-Based Design", paradigms have become influential across automotive,[32, 33] aerospace,[34] and naval design.[31] Regardless of the domain, a key challenge in Set-Based Design is to comb through a space of potential design solutions and eliminate ones that have elements that are likely to lead to future problems. Those future design problems are likely to arise when design elements are not robust.

However, describing robustness in set-based and other flexible design paradigms requires new approaches. Robustness approaches in narrowing, convergent design [42, 43] describe single-design solutions, living at a point in design space. In contrast, set-based design paradigms require determining the robustness not of *a* design, but of *sets* of designs. The quantitative, collective treatment of sets is precisely the subject of statistical physics.

Statistical physics has a long history of describing collective behaviors that range from the long-known thermodynamics of gases,[116] to more recent investigations of entropy-driven order,[73, 176, 85, 177] and a host of non-thermal collective phenomena, including flocking behaviors,[178] the collective motion of human crowds,[179] traffic jams,[180] the synchronization of agricultural yields,[181] and primate social dynamics.[48] A central advantage of statistical physics is its ability to group together microscopic system states and investigate the properties of and transitions between those groups in the language of free energy (see the conceptual and mathematical discussion below). Free energy ideas have been used to classify transitions between collective behavior regimes in complex systems [182] and cognition.[183] Chapter IV has also shown that statistical physics approaches can be applied to systems design problems, under the guise of Systems Physics. Here, we use Systems Physics to study the intermediate-scale structure of design spaces to study design robustness.

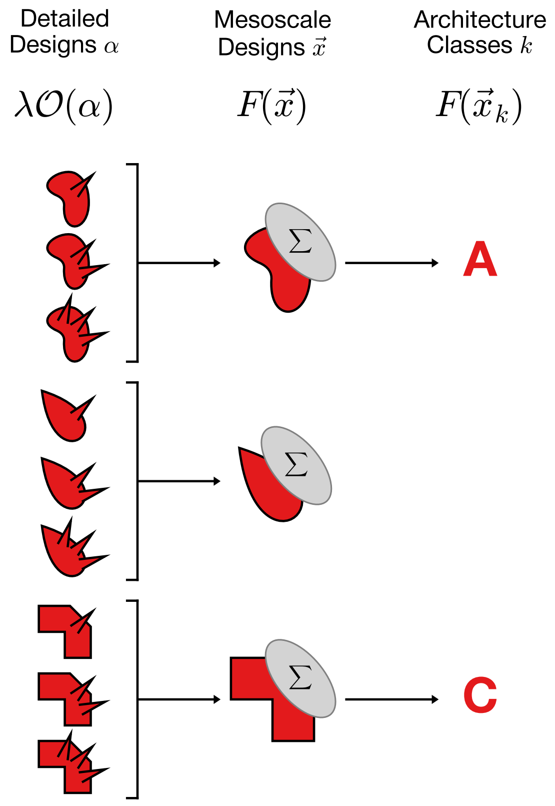
5.2.1 General Approach

To establish a physics approach for understanding robust systems design, we take cues from the physics of materials. In materials an instructive example is a steel rod under mechanical load. Under load the rod can take one of two qualitatively different states, intact or broken. Before it breaks, the response of a rod to external forcing can be quantified using material-dependent relationships between force and deformation, i.e. stress–strain relationships. The nature of the stress–strain response of the rod can be used to concretely describe its material along the independent axes of *weak–strong* and *brittle–ductile*.[172] Brittle and ductile materials show qualitative differences in behavior. Both brittle and ductile behavior can, in different industrial contexts, find appropriate uses. But in either case, determining which material to use requires knowing how it behaves.

Adapting the materials analogy to systems design requires identifying the key behaviors and what drives them. In systems design a key factor in robustness is how design elements behave as they integrate with other subsystems (see schematic illustration in Fig. 5.2a). Their behavior, in terms of how specifications, locations, etc., respond to integration is driven by multiple factors. Different subsystems are generally designed by different designers to satisfy different *design objectives.* Also generally, each design objective has different relative importance, which we term *design pressure.* Design pressures act on all the elements at the same time and thus represent an externally imposed, whole-system level, or *global* drive.

Global design pressures manifest themselves *locally* by driving specific design elements in different directions, e.g. in terms of their physical locations or specifications. For example, competing design pressures of cost and performance can produce discord in the specification of design elements. This discord at the level of elements or subsystems, is analogous to local mechanical stresses and strains that occur in materials under external load. This suggests there should be an analogous local measures of force and deformation, i.e. *design stress* and *design strain*, that express "locally" how design elements respond to the "load" of global design objectives.

If global design pressures are connected to local design stress and strain, how can this be quantified? The challenge in quantifying the global–local connection is that meeting global design objectives is the collective result of all component elements. Moreover, when design features could be placed in a number of possible locations



Detailed Designs α
Mesoscale Designs $\vec{x}$
Architecture Classes k
$\lambda\mathcal{O}(\alpha)$
$F(\vec{x})$
$F(\vec{x}_k)$
Σ
A
C

Figure 5.3: Detailed designs α group into mesoscale designs $\vec{x}$, and locally optimal mesoscale designs form architecture classes k. Each detailed design is characterized by multiple features (here system shape and spike pattern) and quantified by the design objective $\lambda\mathcal{O}(\alpha)$. We use the system shape as the feature $\vec{x}$ to define mesoscale designs by summing over all spike patterns to get the Landau free energy $F(\vec{x})$. Top and bottom shapes are each better than the middle one and thus form architecture classes, here A and C, in the local free energy minima $F(\vec{x}_k)$.

and/or could meet different specifications, this produces a combinatorial explosion of possible design states. The challenge of describing the collective behaviors of combinatorially large numbers of states has an analogue in the thermodynamics of atomic systems, a problem that prompted the development of statistical physics. Here, instead of using it to group states of atoms, we will use statistical physics to group designs.

Our statistical physics approach to quantify design stress and strain relies on grouping designs that share at least one feature. (see Fig. 5.3 for an illustration). We term a group of designs with a shared feature as a *mesoscale* design. Each mesoscale design needs to be described by a quantity that encodes its properties. The first quantity that needs to be encoded is the number of designs the mesoscale design has grouped by common feature. Mesoscale designs that group many detailed designs need to be distinguished from ones that group few. The second quantity that needs to be encoded is how well the grouped designs meet global design objectives. These two factors can be lumped together into a mathematical function, referred to as a *free energy* in statistical physics. The free energy of a mesoscale design decreases as the number of detailed designs that comprise it increases, or as the detailed designs better satisfy the design objectives, or both. In contrast, changes that reduce the number of detailed designs or their suitability for the objectives increase the free energy.

By grouping designs and computing their free energy, mesoscale designs reduce the complexity of a large number of detailed designs to be reduced to the consideration of a small number of features of interest. Describing designs in terms of features of interest has two advantages. First, features of interest that sit at local minima of the free energy correspond to locally optimal mesoscale designs. The deviation from a locally optimal feature set gives a definition of *design strain* and the free energy change this deviation induces can be used to define *design stress.* Second, The set of designs that receive a "pull" in design stress toward the same feature set, akin to a watershed, constitute a "basin" in design space that can define an*architecture class.*

Grouping designs by feature sets provides each architecture class with a definition of design stress and strain. Appropriate stress and strain definitions, in turn, inform a two-factor determination of robustness. To see why, the materials analogy is again instructive. In materials, robustness is determined by response to mechanical stress and strain. Mechanical stress and strain give two key performance indicators of material performance under different kinds of external influence: maximal loading, or "ultimate stress", and maximum deformation, or "ultimate strain". In everyday language, materials with high ultimate stress are strong and low ultimate stress are weak; materials with low ultimate strain are brittle and high ultimate strain are ductile. Given design analogues of material stress and strain, computing the analogous robustness stress and strain thresholds will provide weak/strong and brittle/ductile classifications for designs.

Knowing how weak/strong or brittle/ductile particular design architecture is is useful. However, it is also important to compare the robustness of designs. To make this comparison, the existence of weak/strong and brittle/ ductile contrasts suggests plotting architectures on two axes that run weak–strong and brittle–ductile. A sketch of this is given in Fig. 5.4a. An architecture X can be located on a pair of axes representing the two measures of robustness. The region around X can be divided into quadrants. Additional architectures would fall into one of those quadrants, permitting a direct comparison of robustness. We refer to the lower left quadrant as the shadow of X because architectures falling in that region would be inferior to X in both strength and ductility. In contrast X would be in the shadow of any architecture that falls in the upper right quadrant, because that architecture would have greater strength and ductility. We refer to that region as the eclipsing region of X. The other two quadrants, in the upper left and lower right, are regions where architectures would involve trade-offs with X, either greater strength but reduced ductility, or greater ductility but reduced strength.

An example robustness comparison between is sketched in Fig. 5.4b. Fig. 5.4b gives an example of a two-factor robustness plot (R^2-plot) for four architectures, which we label W, X, Y, and Z. In the scenario depicted in the R^2-plot 5.4b, W has the same ductility as X, but W is stronger, so all else being equal the architecture W would represent a more robust choice. Similarly, architectures W and Y have the same strength, but W is more ductile, so all else being equal W would represent a more robust choice. Similar reasoning also indicates that W is more robust than Z in both strength and ductility. If architecture W was unavailable, similar considerations would make Z a less robust choice than either X or Y. Comparing X and Y shows that the two architectures have different forms of robustness, X is more ductile, but Y is stronger. In this case, all else being equal, the designer would need additional information to determine whether strength or ductility is likely to be the more important measure of robustness.

Fig. 5.4 illustrates what could occur in a design process under fixed external design pressure. However, changes in design pressure can change the robustness of architectures, echoing e.g. the brittle–ductile transitions that occur in industrial materials [184] and geology.[185] Similar to the varied industrial uses of both brittle and ductile materials, we anticipate the usefulness of architecture classes manifest-

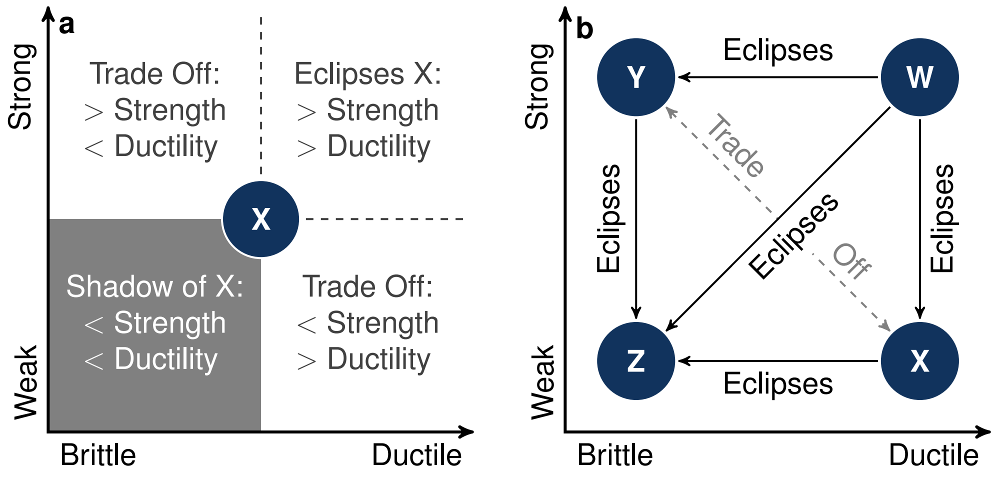


Figure 5.4: Two-factor robustness (R^2) comparison of design architectures. Design architectures can be described by their response to external design stress (e.g., changes in cost) or design strain (e.g., changes in specification limits). The stress–strain response can be used to determine the robustness of an architecture, and to compare architectures. Panel (a) shows that locating an architecture on robustness axes of weak–strong and brittle–ductile facilitates the comparison of that architecture X with other potential architectures. The existence of architecture X casts a “shadow” (lower left quadrant) over other potential architectures that would be less robust by both measures (lower strength, lower ductility). Conversely situating X also identifies (upper right quadrant) forms of robustness, that if they were found in other potential architectures would “eclipse” the robustness of X (greater strength, greater ductility). The other two quadrants (upper left, lower right) describe regions where potential architectures would involve a trade-off in forms of robustness (either in strength or ductility) between architectures. Panel (b) illustrates how this could inform comparison of a set of architectures W, X, Y, and Z. In this illustration W eclipses all other architectures either in terms of strength (X) ductility (Y) or both (Z). If architecture W did not exist, Z is eclipsed by both X (in ductility) and Y (in strength) but a choice between X and Y means a trade-off between the two forms of robustness.

ing different forms of robustness. In evaluating robustness, our goal is to generate knowledge about the possible emergent behaviors in the design space to inform the human designer, who would make the final design choice.

5.2.2 Systems Physics

In this section we cast the foregoing general approach into a concrete mathematical form. Our mathematical model expands upon Systems Physics, a statistical mechanics framework for design problems introduced in Chapter IV. We consider a design problem with a combinatorially large ensemble of candidate *detailed design* solutions $\{\alpha\}$. For each detailed design α we compute several quantitative *design objectives* $\mathcal{O}_i$, where i is an index. An example of a numerical design objective would be the cost of routing a cable between two functional units; we explain the design objectives for our model system in the next section. Given a set of design objectives, a standard calculational device from statistical mechanics that is applied in analogous problems is to associate an expected average outcome $\langle \mathcal{O}_i \rangle$ with each objective. Given only the above data, information theory implies that the minimally biased (maximal entropy [99]) probability distribution for choosing a detailed design α that matches the objectives with their outcomes is given by

$$p(\alpha) = \frac{1}{\mathcal{Z}} e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)} , \tag{5.1}$$

where $\mathcal{Z}$ is a normalization constant. High probability designs are the ones that best fulfill the competing design objectives. Whereas the design objectives $\mathcal{O}_i$ assumed to be known *a priori* and represent *what* criteria need to be considered by the designer, the design pressures λ_i represent *how much* each criterion matters, and the choice of these pressures could differ between stakeholders with different concerns, e.g. cost versus performance. Selecting particular values of λ_i significantly shapes the probability distribution $p(\alpha)$ and can radically and abruptly change the types of the preferred designs α. Predicting and quantifying the preferred designs within the ensemble is a goal of Systems Physics.

The properties of the whole ensemble are contained in the normalization of Eq. (5.1) that can be computed as

$$\mathcal{Z} = \sum_{\alpha} e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)} \, , \tag{5.2}$$

and has the familiar form of a partition function from statistical physics. In statistical physics, the partition function encodes the statistical averages of design objectives across the whole ensemble. However, summing over the whole ensemble masks the fact the many detailed designs α achieve the same design objective value $\mathcal{O}_i$, and thus obscures the intermediate scale design drivers. Discovering these design drivers requires studying designs at a higher level of granularity. This granularity is given by the architecture classes we discussed in general terms above, and which we will now define more precisely.

5.2.3 Architecture Classes

We achieve a higher level of granularity by selecting a design feature $\vec{x}_\alpha$ that multiple detailed designs α share. An example of a shared feature could be the spatial location of a particular functional unit or one of its internal operational parameters (e.g. pressure or voltage). Regardless of the specific feature chosen, a set of designs sharing a common feature $\vec{x}$ can be described by a *mesoscale design* (see Fig. 5.3). The feature space $\{\vec{x}\}$ is typically much smaller than the set of detailed designs $\{\alpha\}$, but can be used to recover the statistical information of the full design ensemble via the expression

$$\mathcal{Z} = \sum_{\vec{x}} e^{-F(\vec{x})} \, , \tag{5.3}$$

where $F(\vec{x})$ is the *free energy.* The free energy quantifies the effective design objective of the mesoscale design $\vec{x}$, and is determined by the expression

$$e^{-F(\vec{x})} = \sum_{\alpha} \delta(\vec{x} - \vec{x}_\alpha) e^{-\sum_i \lambda_i \mathcal{O}_i(\alpha)} . \tag{5.4}$$

Here $\delta(\vec{x} - \vec{x}_\alpha)$ is an indicator function, equal to 1 when the detailed design α belongs to the mesoscale design $\vec{x}$ and 0 otherwise. $F(\vec{x})$ is an example of a so-called Landau free energy [107], and provides a mesoscale characterization of classes of designs that share characteristics specified by $\vec{x}$. The procedure of going from a large ensemble of detailed designs α to mesoscale designs $\vec{x}$ is known as *coarse graining* in statistical physics.[107] Coarse graining applied to a model system and the resulting free energy landscape are illustrated in Fig. 5.5.

The shape of the free energy landscape F indicates the relative preference between different mesoscale designs. Designs that accord better with the balance of design pressures λ_i have smaller F, and vice versa. Local minima of $F(\vec{x})$ are "best-in-class" designs. We denote these designs as $\vec{x}_k$, and index them as $k \in \{A, B, C, \dots\}$ (colored circles in Fig. 5.5b). However, although best-in-class designs are defined to be those that best meet a fixed set of design objectives, changes in the specification of the objectives can alter the classification, so understanding the robustness of designs is crucial.

5.2.4 Quantifying Robustness

A deviation from a local minimum $\vec{x}_k$ within the feature space gives a design strain $\vec{\epsilon}_k = \vec{x} - \vec{x}_k$. In design strain coordinates, design stress is given by the free energy gradient $\vec{\sigma}(\vec{\epsilon}_k) = -\vec{\nabla} F(\vec{x}_k + \vec{\epsilon}_k)$. Sufficiently close to the local minimum, design stress pulls the design back to the minimum, i.e. $\vec{\sigma} \cdot \vec{\epsilon}_k < 0$. However, at larger strains in a particular direction, the design can reach a threshold, or saddle point, in free energy and get pulled by the local design stress to a different minimum. We call that point the *ultimate strain* and formally define it as

$$\vec{\epsilon}_k^{\,\mathsf{ult}} = \arg\min_{\vec{\epsilon}_k} |\vec{\epsilon}_k| : \ \vec{\sigma} \cdot \hat{\epsilon}_k > 0 \ , \tag{5.5}$$

where $|\cdot|$ denotes a suitable vector norm (here we use standard Euclidean norm) and $\hat{\epsilon}_k$ is a unit vector pointing along the strain direction. The operator arg min finds the closest saddle point but still returns the vector $\vec{\epsilon}_k^{\,\mathsf{ult}}$ rather than just its norm. We illustrate the path from local free energy minima to the saddle points in an example system in Fig. 5.6a,c,e.

As a mesoscale design is strained from $\vec{0}$ to $\vec{\epsilon}_k^{\,\mathsf{ult}}$, it will develop design stress. To analyze the stress response, it is convenient to compute the projection of the stress along the strain direction, $\sigma = |\vec{\sigma}(\vec{\epsilon}_k) \cdot \hat{\epsilon}_k|$. From this projection it is possible to compute the *ultimate stress*, i.e. the magnitude of externally exerted stress that causes designs to switch between classes. Formally, this is given by

$$\sigma_k^{\mathsf{ult}} = \max_{a \in [0,1]} \left| \vec{\sigma}(\vec{x}_k + a\vec{\epsilon}_k^{\,\mathsf{ult}}) \cdot \hat{\epsilon}_k^{\,\mathsf{ult}} \right| \ , \tag{5.6}$$

where a is an auxiliary variable parametrizing a straight line from $\vec{0}$ to $\vec{\epsilon}_k^{\,\mathsf{ult}}$. While the free energy minimum $\vec{x}_k$ defines the "best-in-class" mesoscale design, the basin of all mesoscale designs that design stress brings back to the best-in-class design defines an *architecture class.*

We have defined architecture classes thus far for isolated subsystems. When subsystem designers incorporate effects that arise from coupling to other subsystems, other subsystems exert *external* design stress or strain on the subsystem of interest. External stress and strain correspond, respectively, to what are referred to in statistical mechanics as "intensive" (size independent) or "extensive" (size dependent) modifications of the specification of the system.

Going back to the analogy of a metal rod, this nomenclature reflects that a mechanical load can be applied to the rod with two protocols. One is to subject the rod to a fixed external stress force, or intensive modification, and measure the resulting strain. The other is to subject the rod to a fixed linear strain, or extensive modification, in form of stretching or compression and measure the resulting stress.

Protocols for materials response have direct analogues in systems design. In systems design, an example of external design stress would arise from the need to route a connection from a functional unit to an external subsystem, with the direction and cost per unit length specified for the connection. This scenario creates a uniform design stress $\vec{\sigma}^{\mathsf{ext}}$ on the subsystem (Fig. 5.5a), and the new local optimum would be found at the location where the internal design stress balances the external $\vec{\sigma}^{\mathsf{ext}} + \vec{\sigma} = 0$. An example of external strain would be the need to position an additional object in the location $\vec{x}_k$, thus requiring the shift of subsystem design features by design strain of $\vec{\epsilon}^{\mathsf{ext}}$ away from the minimum. In either case, external stress or strain may or may not push the mesoscale design into a different architecture class basin. Resisting the architecture class shift is the property that we call *robustness*.

We determine the robustness of each architecture class by computing the design stress–strain curves. From these curves, we extract the ultimate stress and strain for each architecture class and plot them together without averaging in Fig. 5.8. These stress–strain relationships facilitate the characterization of each design architecture class as weak or strong by comparing the respective σ_k^{ult} among different architecture classes k. Weak designs have small σ_k^{ult}, whereas strong designs have large σ_k^{ult}. We also characterize designs as brittle or ductile by comparing the relative $|\vec{\epsilon}_k^{\mathsf{ult}}|$. Brittle designs have small $|\vec{\epsilon}_k^{\mathsf{ult}}|$, whereas ductile designs have large $|\vec{\epsilon}_k^{\mathsf{ult}}|$. Considering both strength and ductility gives us the two-factor robustness, R^2, of the architecture classes, presented in Fig. 5.9.

5.2.5 Beyond Ultimate Stress

Whereas the determination of ultimate strain can characterize the robustness of an architecture class, it is also important to understand what happens once architecture classes are pushed beyond their viability limit. Doing so requires understanding the configuration of architecture classes under large external stress.

To model the external stress, we consider not only the free energy of the subsystem of interest $F(\vec{x})$ that depends on the design feature $\vec{x}$, but also the free energy of an external subsystem $F'(\vec{x}\,')$ that depends only on the design features $\vec{x}\,'$ of the external subsystem. The interdependence of the two subsystems is captured by the interaction free energy $F_{int}(\vec{x},\vec{x}\,')$. The goal is to understanding what happens in the system of interest, under the assumption the external subsystem can take any configuration it prefers. We carry this out by integrating out the external subsystem, leaving a description of the remaining subsystem of interest. Mathematically, the procedure is similar to to the earlier coarse-graining procedure in Eq. (5.4),

$$e^{-\tilde{F}(\vec{x})} = \sum_{\vec{x}\,'} e^{-F(\vec{x})-F'(\vec{x}\,')-F_{int}(\vec{x},\vec{x}\,')}. \tag{5.7}$$

In general, performing the computation in Eq. 5.7 is challenging, since it requires a detailed model of the external subsystem. However, we note that free energy is only defined up to an additive constant which does not affect the locations or properties of local minima. Thus we can get insight into the effect of external couplings by adopting a simplified form of the interaction free energy:

$$F_{int}(\vec{x},\vec{x}\,') \approx (\vec{x}\,' - \vec{x}) \cdot \vec{\sigma}^{\mathsf{ext}} + \text{const}. \tag{5.8}$$

This form of interaction free energy describes a uniform external design stress $\vec{\sigma}^{\mathsf{ext}}$ applied to each feasible design in the domain of design feature $\vec{x}$, caused for example by the addition of a cable of fixed direction and cost per unit length. Computationally, this form of interaction makes the summation in Eq. 5.7 separable and gives the effective free energy landscape as:

$$\tilde{F}(\vec{x}) = F(\vec{x}) - \vec{x} \cdot \vec{\sigma}^{\mathsf{ext}} + \text{const.} \tag{5.9}$$

In general, $\tilde{F}$ and F will have different sets of local minima, i.e. design configurations that are best-in-class will change in the presence of external stress. We investigate the effect of variable external stress by considering different vectors $\vec{\sigma}^{\mathsf{ext}}$ that span the space $\{\vec{\sigma}^{\mathsf{ext}}\}$. If the external stress $\vec{\sigma}^{\mathsf{ext}}$ is much larger than any internal design stresses $-\vec{\nabla}F(\vec{x})$ naturally arising in a given architecture class, the subsystem is completely dominated by external stress that eliminates candidate architecture classes, reducing design richness.

We characterize the loss of design richness under external stress by finding the domain in $\{\vec{\sigma}^{\mathsf{ext}}\}$ space in which a minimum of the same type k exists. Here, by "same" we mean a minimum that moved less than some threshold Δx^{th} under a small change of stress $\delta\vec{\sigma}^{\mathsf{ext}}$

$$|\vec{x}_k(\vec{\sigma}^{\mathsf{ext}} + \delta\vec{\sigma}^{\mathsf{ext}}) - \vec{x}_k(\vec{\sigma}^{\mathsf{ext}})| < \Delta x^{\mathsf{th}} \ . \tag{5.10}$$

We illustrate how external stress affects the viability ranges of subsystem design classes in Fig. 5.7 using Venn diagrams in the $\{\vec{\sigma}^{\mathsf{ext}}\}$ space. Together, the viability ranges for architecture classes and the analysis of their ultimate design stress and strain constitute the quantitative knowledge required for robust system design.

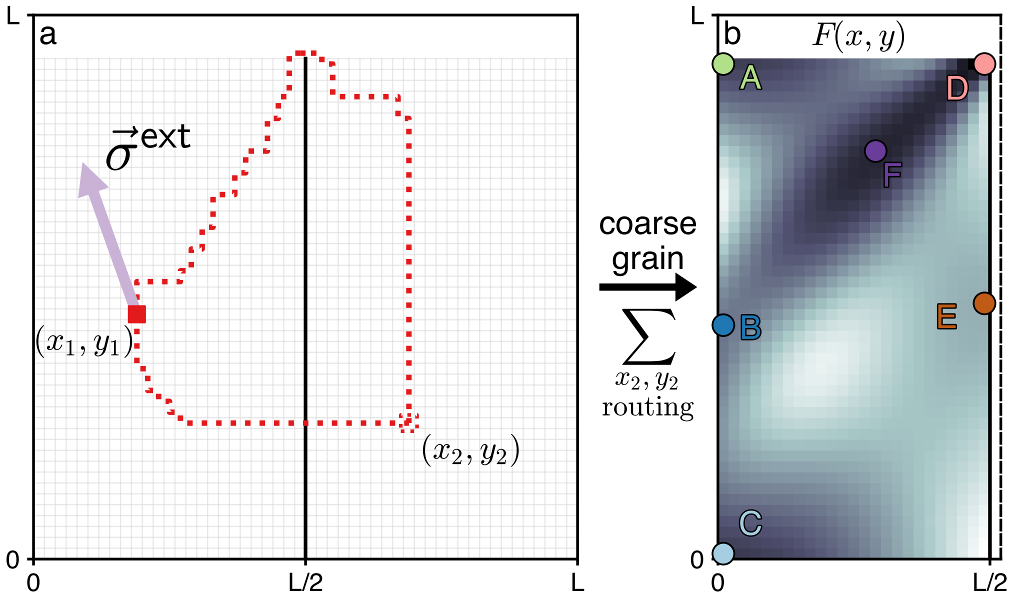


Figure 5.5: Schematic representation of the specific subsystem investigated for robustness. (a) Two connected functional units are placed in positions (x_1, y_1) and (x_2, y_2). There are two qualitative ways to connect them along the shortest Manhattan route: either directly by drilling a hole through the bulkhead, or by first routing up to the bulkhead, over it and back down. The position of the first unit is taken as the design feature $\vec{x}$, while the position of the second one and the possible routings are "integrated out" by computing the free energy via Eqn. 5.13. The external design stress on the system has the form of a constant force $\vec{\sigma}^{\text{ext}}$ shown with a purple arrow. (b) Coarse graining procedure leads to the free energy landscape $F(x, y)$ for the possible positions of the first unit in the part of the domain left of the bulkhead. Local free energy minima are identified with architecture classes labelled with capital letters A through F and distinct colors.

5.3 Example System: Model, Results, and Discussion

5.3.1 Model System

The Systems Physics based robustness analysis developed in the previous section can be applied to a broad range of design problems. For concreteness, we will illustrate its use in arrangement problems that arise in Naval Architecture, taking as a specific example an established model of early stage ship design.[157] Our ship design model involves embedding a network of many functional units into a ship hull of fixed geometry, and routing connections between the units. Large-scale effects on unit routing induced by changes in design pressure were studied in Chapter IV. Here, we study this model at the mesoscale to understand the robustness of system design. To determine robustness, we focus on a subsystem with two connected functional units that is externally connected to other subsystems, modelled via external design stress (see Fig. 5.5a).

The subsystem is situated in a square domain of $L \times L$ discrete cells. The domain is separated along the middle line into two compartments housing one functional unit each, divided up to height h_{bh} by a watertight bulkhead. The bulkhead serves to prevent simultaneous water flooding of both compartments in case one of them is breached. It is possible to drill a hole through the bulkhead, reinforce that hole, and route a connection through it. However, such a hole bears risks that affect ship survivability. To assess how survivability considerations affect the routing problem, we consider two distinct types of routing: either along the shortest possible route through the bulkhead, or along the shortest possible route around the top of the bulkhead. The connections are only routed horizontally and vertically, so there is a large but finite number of possible routings for each choice of positions of the two units. A particular realization of unit positions and routings of two types is shown in Fig. 5.5a. The possible positions of the two units (x_1, y_1) and (x_2, y_2) along with the choices of particular routing form the detailed design space $\{\alpha\} = \{(x_1, y_1, x_2, y_2, \text{routing})\}$.

Within the design space, detailed designs are evaluated with respect to two design objectives, $\mathcal{O}_1, \mathcal{O}_2$, and corresponding design pressures λ_1, λ_2

$$\begin{aligned} \mathcal{O}_1 \equiv & E = C\left(|\Delta x| + |\Delta y|\right), & \lambda_1 \equiv 1/T; \\ \mathcal{O}_2 \equiv & B, & \lambda_2 \equiv \gamma\ . \end{aligned} \tag{5.11}$$

The first design objective E represents the cost of the routing, linearly proportional to the Manhattan (taxicab/grid) length of the routing used. The corresponding design pressure is *inverse cost tolerance* T, similar to the thermodynamic temperature. Low cost tolerance means that designs with shorter routings are strongly preferred, whereas high cost tolerance means that cost is not a strong factor in choosing a design. The second design objective $B \in \{0, 1\}$ is a binary variable indicating whether a given design routes through the bulkhead (1) or goes around (0). The corresponding design pressure is the bulkhead penalty γ that quantifies the survivability penalty associated with routing through the bulkhead. Low, near-zero, values of γ mean that routing through or around the bulkhead are equally preferable, whereas high values of γ strongly suppress routing through the bulkhead.

In terms of these specific design objectives and design pressures, the partition function (5.2) takes the form

$$\mathcal{Z} = \sum_{\alpha} e^{-\frac{E(\alpha)}{T} - \gamma B(\alpha)}\ , \tag{5.12}$$

where the sum over α runs over *the whole set* of possible detailed designs. To group the detailed designs into mesoscale designs, we use the position of the *left* functional unit $\vec{x} = (x_1, y_1)$ as the design feature of interest. The position of the right functional unit (x_2, y_2) and the routings are integrated out. The position of the left unit then has the associated free energy $F(x, y)$

$$e^{-F(x,y)} = \sum_{\alpha} \delta(\vec{x} - \vec{x}_1(\alpha)) e^{-\frac{E(\alpha)}{T} - \gamma B(\alpha)} \; . \tag{5.13}$$

An example of the free energy landscape is shown in Fig. 5.5b. Local minima of the free energy are associated with the architecture classes A through F. We find that the free energy landscape and the pattern of architecture classes vary greatly with the design pressures T, γ, and that in all cases the design robustness is given directly by Eqns. 5.5-5.6. There are several large-scale reorganizations between architecture classes as the design pressures T and γ are varied. These reorganizations are analogous to phase transitions in thermodynamic systems, but are not sharp transitions because the design problems we study have finite size. Despite the finite sizes, it is possible to approximately determine where the reorganization of architectures occurs, as shown in Chapter IV. In this model reorganization occurs around a *cost phase transition*: at low $T < T_{\mathsf{crit}} \sim C/\ln 2$ units prefer short connections to minimize the routing cost, whereas at large $T > T_{\mathsf{crit}}$ they prefer long connections to maximize the flexibility in carrying out the routing. Another large-scale reorganization is the transition of the average bulkhead penetration $\langle B \rangle$, or fraction of designs routing through the bulkhead: it approaches 0 for simultaneously large T and γ (preferring flexibility in routing and suppressing bulkhead penetration), and approaches 1 when either T or γ is small (preferring low-cost routing and allowing bulkhead penetration, or a combination of both). As we will find below, the origin of these large-scale reorganizations can be traced in the mesoscale through the appearance and disappearance of architecture classes and changes in their robustness.

To capture the reorganization of architecture classes, we study the routing problem for a range of design pressures $\{T, \gamma\}$. We fix our system of units by setting $C = 1.0$, which fixes the units of cost tolerance T. To maximize illustrative power, we seek a set of choices of γ and T that allows for the study of all the possible architectural organizations with the fewest number of state points. For this purpose, a suitable choice is to scan along the line of $\gamma = 2.0$ with $T \in [0.5, 2]$ because that choice crosses both the cost- and bulkhead penetration-based reorganizations. We also make specific choices of system geometry parameters. Since the positions of the functional units are discrete, we cannot reliably resolve the stress–strain curves on length scales less than 1 cell. Moreover, the comparison of robustness between architecture classes requires the system domain to be sufficiently large to support multiple architecture classes. To this end, we set the domain size $L = 50$, with the bulkhead going up to $h_{\mathsf{bh}} = 46$, though our analysis and results are similar for different system sizes. We find that ultimate strains vary from 1.5 to 18 cells, allowing us to reliably distinguish the architecture classes along both weak–strong and brittle–robust axes.

5.3.2 Results and Discussion

Design Stress and Strain

Fig. 5.6 demonstrates the range of architecture classes and their respective stress–strain curves that appear in the model subsystem. For this subsystem, for each fixed T in the range of $[0.5, 2.0]$ we identify as many as 6 qualitatively different architecture classes that we label A through F. Figs. 5.6 and 5.7 illustrate the architecture classes at three representative values of $T = 1.20, 1.50, 1.70$. $T = 1.20$ corresponds to the low-cost regime. $T = 1.70$ corresponds to the high flexibility regime. $T = 1.50$ corresponds to the regime in which there is a near

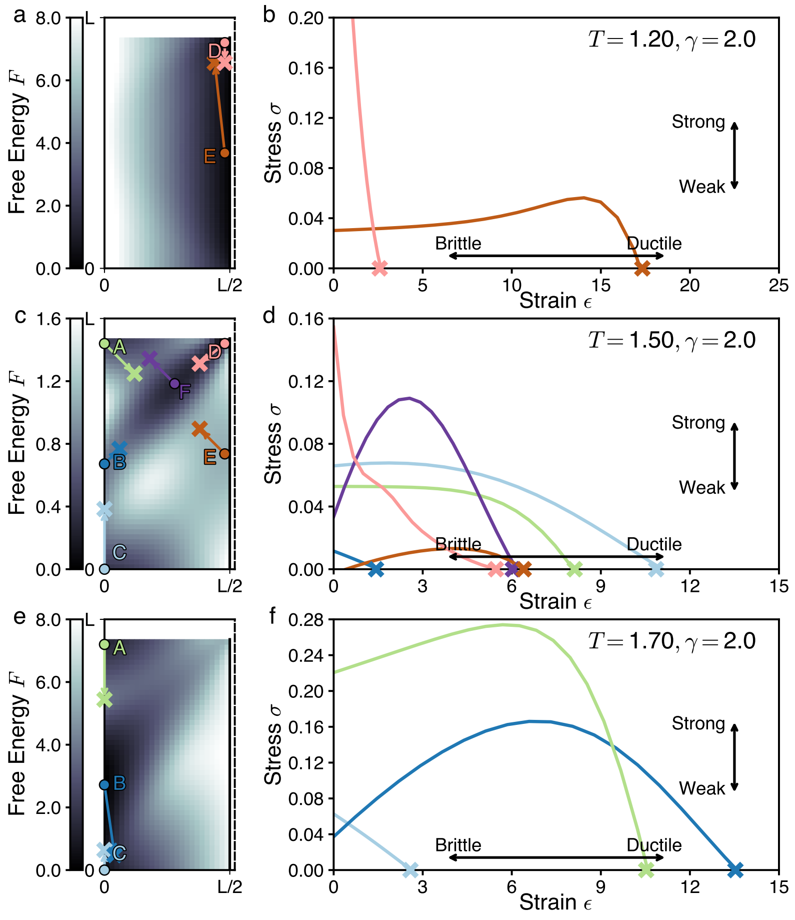


Figure 5.6: Statistical physics approach quantifies stress–strain relationships in design problems. Plots show free energy landscapes and stress–strain curves at three different cost tolerances $T = 1.20, 1.50, 1.70$ and constant bulkhead penalty $\gamma = 2.0$. (a,c,e) Free energy landscape for the position of the left functional unit. The solid vertical line on the right denotes the position of the bulkhead. The dashed vertical line cuts off the domain of the second functional unit that has been integrated out. Note the different colormap scales at different T. Each colored circle indicates a local minimum that forms an architecture class, indexed with a unique letter A through F and a unique color (green, blue, purple etc.). The cross marks and lines connecting them to circles indicate the ultimate strain locations for each minima, as determined by condition (5.5). (b,d,f) Stress-strain curves for each of the local minima at given T, with stress measured along the ultimate strain direction via spline interpolation of the free energy landscape. The cross marks indicate the ultimate strain ϵ_k^{ult} for each minimum. The maximum of each curve indicates the ultimate stress for each minimum σ_k^{ult}.

balance between cost and flexibility.

At each cost tolerance, for each architecture class we compute the stress–strain response (Fig. 5.6b,d,f). As described above, for materials the stress–strain response can be measured via two principal protocols. In the first, the material is deformed by a fixed strain ϵ^{ext} and equilibrates at some corresponding stress $\sigma(\epsilon^{\mathsf{ext}})$. In the second, the material is affected by a fixed stress σ^{ext} and equilibrates at some corresponding strain $\epsilon(\sigma^{\mathsf{ext}})$. Graphically, this is equivalent to picking first a point on the vertical σ axis and finding the corresponding curve point on the horizontal ϵ axis, or vice versa.

Stress–Strain: Comparison with Materials

For common materials, the difference in protocols is not very noticeable since at low stress and strain their relationship is linear, at larger deformations it is weakly nonlinear but still monotonic, up until the breaking point at finite stress and strain. However, this textbook materials science picture breaks down for the design stress–strain relationship in our example design problem in three important ways, at both low and high strain.

The differences between stress–strain relationships for designs and materials can is facilitated by expanding the relationships at low strain as a power series

$$\sigma = \sigma_0 + Y\epsilon + O(\epsilon^2) \ . \tag{5.14}$$

The first deviation from common material behavior is that for design stress the constant term is nonzero $\sigma_0 \neq 0$ for all of the curves in Fig. 5.6b,d,f. Similar effects are common in manufactured engineering components that exhibit residual internal stress, usually resulting from plastic deformations in manufacturing, thermal expansion, boundary effects, or phase change of materials.[186] The main implication of residual stress is that applying small external design stress $\sigma^{\mathsf{ext}} < \sigma_0$ does not result in measurable design strain, as opposed to the conventional linear response.

The second deviation is that the linear part of stress response can be both positive ($Y > 0$, as in Fig. 5.6f, architecture classes A, B, green and dark blue curves) and negative ($Y < 0$, same figure, architecture class C, light blue). A positive linear response means that the architecture class can support at least small design stress above σ_0 level via a small deformation. A negative linear response means that the ultimate stress $\sigma^{\mathsf{ult}} = \sigma_0$ and is already reached for $\epsilon = 0$, or no strain. For any higher, supercritical external stress, there is no corresponding point $\epsilon(\sigma^{\mathsf{ext}})$ and thus the architecture class immediately "breaks", transitioning the design to a different class.

Whereas the first two deviations from textbook materials response are observed at low strain, the third one appears at high strain, right before the breaking point. Many conventional materials (e.g. steel) break at finite stress. However, some materials (e.g. fiber-reinforced brittle concrete [187]) exhibit a different phenomenon known as "tension softening",[188] whereby they support decreasing amounts of stress as they are strained, and ultimately fail at zero stress. We observe this phenomenon in all of the architecture classes we find in the present model system.

Together, these three deviations describe the unconventional pathways in which architecture classes can break. Via strain: If the external subsystem coupling provides a fixed design strain, the design stress remains positive and finite for a wide strain range, ensuring that the chosen architecture class remains viable. Via stress: Conversely, if the external subsystem coupling provides a fixed stress, the architecture class only responds noticeably beyond a certain stress threshold, but often responds with an abrupt architecture class change. Via stress and strain: a combination of external design stress and strain can push the architecture class into the tension softening regime, making it unviable.

Beyond Ultimate Stress

Fig. 5.6 illustrates what happens to an architecture class as it is strained up to its limit of viability by showing the design stress along the direction of least ultimate strain. This is useful for assessing the further viability of an architecture following the effects of an external design strain. However, it can also be important to determine the effects on the number of architecture classes under the influence of external design stress that can push one or more classes beyond their limit of viability.

To assess robustness in this form, Fig. 5.7 illustrates the effect of the external design stress taking any values in the plane $(\sigma_x^{\mathsf{ext}}, \sigma_y^{\mathsf{ext}})$. In this plane, each architecture class A through F has a viable domain. We find that the shapes of these domains are complex, implying that the viability of an architecture class is highly sensitive to both direction and magnitude of external stress. We show the overlap of the domains of all six architecture classes in the center of each panel in Fig. 5.7 in form of a computed Venn diagram.

We start analysis with the richest Venn diagram at the near-critical $T = 1.50$ (Fig. 5.7b). In that diagram, the classes E, F (pink and purple) are viable in very narrow and specific ranges of external stress $(\sigma_x^{\mathsf{ext}}, \sigma_y^{\mathsf{ext}})$. Compared with the other architectures, small amounts of uncertainty in external design stress would be be sufficient to render architecture classes E, F unviable. At the same time, the architecture classes A through D, in which the functional unit is localized either in one of the three corners or the middle of one side of the allowed domain, are viable given almost any amount of external design stress outwards toward the domain

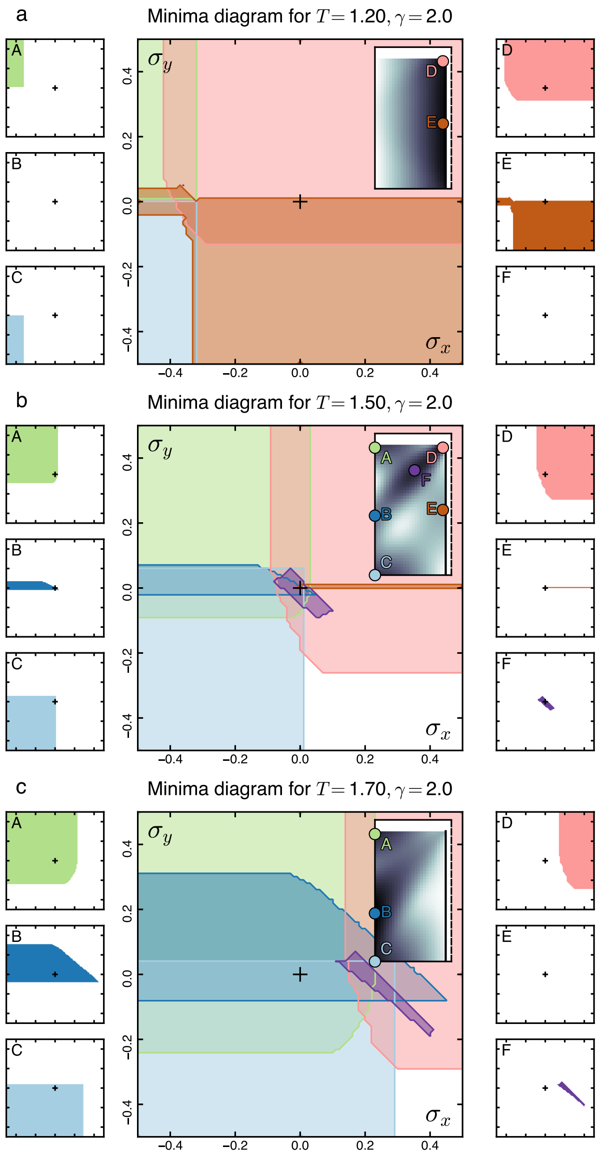


Figure 5.7: The response to external stress gives viability limits on architecture classes and shows external stress can become viable under external stress. Plots shows regions of existence of architecture classes in the $(\sigma_x^{\text{ext}}, \sigma_y^{\text{ext}})$ (external stress) plane. (a) $T = 1.20$, (b) $T = 1.50$, (c) $T = 1.70$. For each panel, (center) Venn diagram of regions in the $(\sigma_x^{\text{ext}}, \sigma_y^{\text{ext}})$ plane where each of 6 architecture classes exists. Black cross indicates the origin of the plane $\vec{\sigma}^{\text{ext}} = 0$. (inset) Free energy landscape with the architecture classes A through F labelled in color. (sides A–F) Regions in the $(\sigma_x^{\text{ext}}, \sigma_y^{\text{ext}})$ plane where the corresponding individual minima exist, in the same stress plane as the central diagram.

boundaries, as well as moderate stress directed inwards toward the middle of the domain. This form of analysis gives a more detailed understanding variations in the robustness of architecture classes when the effects of external design stress are not uniform in all directions.

From the depiction of the response to anisotropic stress in Fig. 5.7 it can be seen that external stress can, indeed, push some architectures beyond their viability limits while leaving others viable. However, Fig. 5.7 also shows that in situations where there is low cost tolerance (i.e., a strong preference for low cost, $T = 1.20$, panel a) or high cost tolerance (i.e., a weak preference for low cost, $T = 1.70$, panel c), the external stress can make viable the architecture classes that would not be viable without the external stress. Architectures A and C are examples of this at low cost tolerance (panel a), whereas architectures D and F are examples of this at high cost tolerance (panel c).

Robustness and Design Pressure Changes

Figs. 5.6-5.7 showed a detailed analysis of how particular architecture classes respond to external design forcing at three representative values of cost tolerance T. To validate that these choices of T are representative, and to better understand how robustness changes in response to changes in design pressure, Fig. 5.8 aggregates the specific results shown in Figs. 5.6-5.7 with the results of similar analysis over a fine grid of $T \in [0.5, 2]$. As we scan the T range, the robustness of architecture classes experiences a clear shift in the region of T where the architectures reorganize from low-cost to high flexibility. In the low-cost regime, architecture classes D, E are viable, and their properties are typified by the prior analysis at $T = 1.20$. In the high-flexibility regime, architecture classes A, B, C are viable, and their properties typified by the prior analysis at $T = 1.70$. At intermediate values of T where there

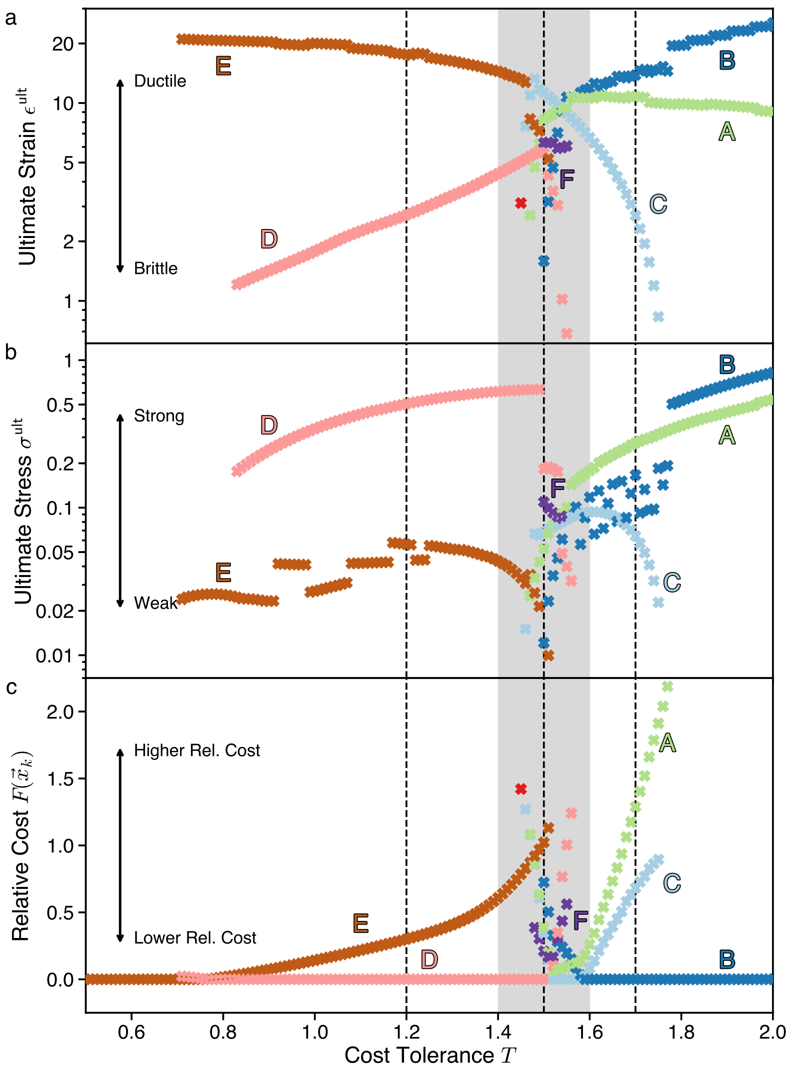


Figure 5.8: Architecture classes can transition strong to weak, or from brittle to ductile if design pressures change. Plots give robustness and optimality measures for all architecture classes (local minima of Landau free energy $F(\vec{x})$) existing at each value of cost tolerance T. The gray-shaded area on all three graphs indicates the region near $T_{\text{crit}} = 1/\ln 2 \approx 1.44$, which exhibits almost all of the architecture classes. The three dotted vertical lines indicate the values of T for a detailed analysis is given in Figs. 5.6, 5.7. Color and letter coding remain the same as in those Figures. (a) Ultimate strain values, corresponding to the *brittle–ductile* robustness characterization of architecture classes. (b) Ultimate stress values, corresponding to the *weak–strong* robustness characterization of architecture classes. (c) Landau free energy $F(\vec{x}_k)$ values at all local minima, normalized so that for any T the lowest F value is zero. Vertical position of the points corresponds to *higher-lower relative cost* characterization of architecture classes.

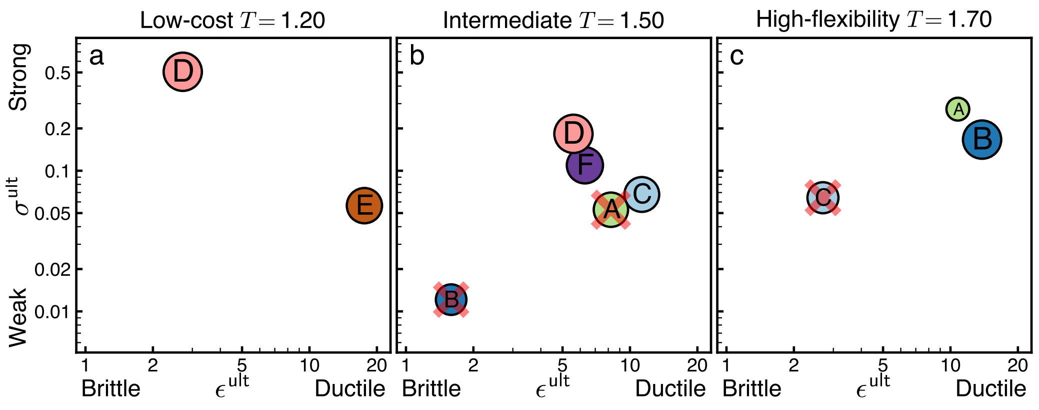


Figure 5.9: Two-factor robustness (R^2) comparison for model-system architecture classes facilitates the elimination of non-robust (weak/brittle) architectures. The analysis of the model system implements the scheme sketched in Fig. 5.4, using the two robustness factors of ultimate stress (ϵ^{ult}, vertical axis) and ultimate strain (σ^{ult}, horizontal axis). The three panels depict the architectures and robustness relations among them at different cost tolerances T ($T = 1.2$-a, $T = 1.5$-b, $T = 1.7$-c). Architectures are represented by circles with size proportional to global design objectives. At high cost tolerance (c), where design pressure favors flexibility in realizing designs, the architecture C (also marked with a red $\times$) falls in the shadow of both architectures A and B. Between architectures A and B there is a trade-off in robustness between strength and ductility. A similar trade-off exists at low cost tolerance (a) between architectures D and E. At intermediate tolerance (b) where there is a balance of concern between cost and flexibility, more architectures are possible. Comparing their robustness, the architecture A is eclipsed by C, the architecture B falls in the shadow of all other architectures, and a trade-off exists between the three architectures C, D, and F.

is a reorganization between these regimes (shaded area in Fig. 5.8) we observe the highest diversity of viable architectures at zero stress. However, we also observe sharp changes in the robustness of architectures in response to external stress or strain. These change in robustness of the mesoscale design precisely in the near-critical T region suggests a causal relationship: the shift between viable architecture classes is the primary mechanism to drive the large-scale phase transitions in the whole design space.

Two-Factor Robustness Comparison of Architecture Classes

The analysis above gave detailed information about the structure of the design space. Working from detailed information about the space, it is possible to distill essential pieces of information that can inform design decisions. Decisions involving comparing the robustness of architectures can be cast in the form of R^2-plots. Whereas the Fig. 5.4 diagrams were schematic illustrations, applying the analysis framework illustrated in Fig. 5.4 to our model system yields the R^2-plots shown in Fig. 5.9.

Fig. 5.9 gives a proof-of-principle that two-factor comparisons of architectures can be successfully computed in model systems. However, for a given model system the comparisons themselves are interesting. First, in the model systems we study, we find no case in which a single architecture class eclipses all others in robustness in both strength and ductility in a R^2-plot. We find that, in the low-cost regime $T = 1.20$ (Fig. 5.9a), the trade-off is between the only two existing architecture classes: architecture D is stronger, but more brittle, whereas architecture E is weaker but more ductile. In the regime where design pressure favors designs that can be realized with more flexibility, $T = 1.70$ (Fig. 5.9c), there is a strength/ductility trade-off between architectures A, B, both of which eclipse architecture C. In the intermediate regime, where design pressures for cost and flexibility are nearly balanced, $T = 1.50$ (Fig. 5.9b), there is a three-way strength/ductility trade-off between architectures C, D, F, but both of their robustness factors are much smaller than at either higher or lower T. These results indicate that the comparison of robustness between architectures is complex. It depends both on the system being designed and on the design pressures, and that even simple models can exhibit trade-offs.

5.4 Conclusion

In this Chapter we described a method to engineer systems that are robust to variation in the design problem statement. We situated robustness in design as a necessary complement to the robustness of a product under changes in the operating environment (Fig. 5.1). We showed that the robustness of a design is not a unitary concept, but instead it takes multiple forms including being described on the axes brittle–ductile and weak–strong (Figs. 5.4,5.9). We found that these axes describe system design in the same mathematical terms that are used to describe the robustness of materials. The approach developed in this Chapter sets the stage for the further investigation of robust design.

First, to give a concrete demonstration of the application of our approach we used an example model system drawn from arrangement problems in Naval Engineering.[157] In that model system we found that the design architectures generically manifested forms of residual stress and tension softening that are observed in materials systems. Though both phenomena are well-characterized in materials, they are phenomena that are associated with special forms of preparation that introduce boundary effects (residual stress) or by unconventional micromechanics in the material (tension softening).[187] Our observation of these effects in a model of system design raises the question of whether that observation is specific to our model system, or whether system designs behave, in general, like unconventional materials. The fact that the microscopic interactions in systems design can easily have more varied forms than the micromechanics of materials suggests that conventional behavior for systems may resemble unconventional behavior in materials, but further investigation is required.

Second, though we needed to show the approach on a concrete model system, it can be extended straightforwardly to other classes of design problems. We believe that for a broad class of problems, the two-factor classification of robustness (R^2-plot; Fig. 5.4) will provide a straightforward means for designers to compare the robustness of different design architecture classes. The comparison of classes in our example showed that the R^2 analysis provides a straightforward means to eliminate weak/brittle architecture classes. We expect this to be useful in complex system design where it has been argued that eliminating bad choices is more important than selecting good ones.[31] The nontrivial trade-offs in R^2 we found here suggest that a similar multifaceted characterization will be useful in broader classes of complex systems design.

CHAPTER VI

Interplay of Logical and Physical Architecture

6.1 Introduction

[1]A fundamental question in the design of complex, multicomponent systems is how the components of the system are arranged.[155, 154, 156] Arrangement problems, generically, present the challenge of anticipating or identifying "prime real estate",[189, 157] i.e. sectors of the system's architecture that have premium or priority because of the mutual avoidance, adjacency, or association between system components (see Fig. 6.1). Determining or anticipating components' patterns of avoidance, adjacency, and association is important in so-called "greenfield" settings, i.e. before design aspects have been specified, and in "brownfield" settings, i.e. when one or more system design aspects have been determined.[190, 191, 192] In both greenfield and brownfield settings, determining the patterns of arrangement and identifying system factors driving those behaviors is crucial for managing, mitigating, or adapting to likely design outcomes.[27]

Managing likely design outcomes by identifying patterns of arrangement depends crucially on both a system's logical architecture, i.e. the set of functional connections between components, and on the system's physical architecture, i.e. the physical properties of the components and their arrangement in space.[39] A

[1]This chapter is based on an upcoming paper coauthored with D.J. Singer and G. van Anders.

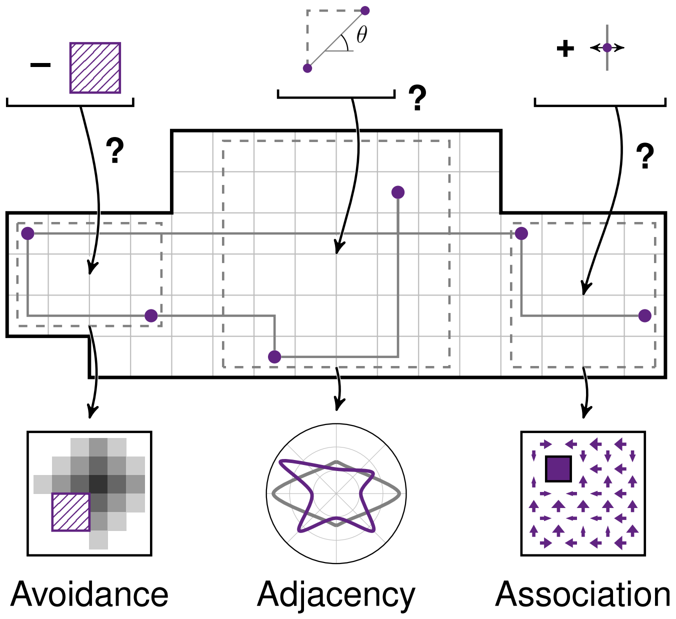


Figure 6.1: Complex system design raises the question of identifying arrangement patterns of avoidance, adjacency, and association. Avoidance patterns (left) can be probed by testing the "cost" of creating a void in the design. Adjacency patterns (center) describe arrangement motifs found in the design, e.g. angles between the placement of design elements. Association patterns (right) relate to the preference for proximity between design elements, e.g. measures "preferred" locations in adding design elements.

system's logical architecture is essentially topological, and can be treated using network theory techniques.[135] In contrast, describing the physical architecture of a system is typically done by disciplinary engineering approaches that rest on known physical principles. Treating problems in arrangements that arise from an interplay of a system's logical and physical architecture requires a framework that bridges a system's network-theory-level description and its physical/spatial description. Whereas approaches exist at the network theory and at the physical spatial levels, how they can be bridged is an open question.

Here, we show that topological and physical descriptions of complex systems design can be bridged using statistical physics. Using statistical physics we demonstrate a framework that reveals patterns of avoidance, adjacency, and association in arrangement problems. We use an example arrangement problem to concretely demonstrate how our framework can identify patterns of arrangement and how those patterns are driven by the design's logical and physical architecture, in both greenfield and brownfield settings.

6.2 Systems Physics Framework

6.2.1 Motivation: Design Challenges Lurk Between Logical and Physical Architectures

Complex systems are typically comprised by several interacting entities. The interactions among the entities are often described at two different levels: the description of what-is-connected-to-what, which is mathematically a graph-theoretic description, and by the description of how the entities physically interact with one another in space, which is described by physics. Taken separately, both levels of description give useful but incomplete insights into design.

The logical architecture's graph-theoretic description of a complex system design is valuable because it isolates the connections between system elements that underly functionality.[39] Functionality in the logical architecture is reduced to the topology of connections, and this connection topology can be analyzed with network theory techniques.[174, 173, 41] Network theory approaches to analyzing logical architecture are powerful because they abstract out the system's physical realization.[193] However, realizing the logical architecture physically can produce emergent functional connections that are lost when logical architecture is analyzed alone.

The physical architecture describes the realization of a complex system design in terms of physical entities with physical properties. Whereas entities in the logical architecture have abstract interactions that are encoded topologically, in the physical architecture interactions interact mechanically, thermodynamically, electromagnetically, etc., depending on physical factors such as energy consumption and proximity in space. By retaining this level of detail, the physical architecture provides an intimate picture of the performance of design elements. However, this intimate portrait of performance typically describes a single physical architecture instance. What that single instance means for the space of possible designs more generally is often unclear.

Though they can provide key insight into single design instances, both physical and logical architecture descriptions restrict our ability to understand general characteristics of design. This restriction exists because general design characteristics are properties of design problem spaces rather than of design instances. The focus on design instances has been described previously as design organized around "product structures", i.e., around a particular outcome of the design process.[29] Contrasting with product structures are "knowledge structures" that organize the

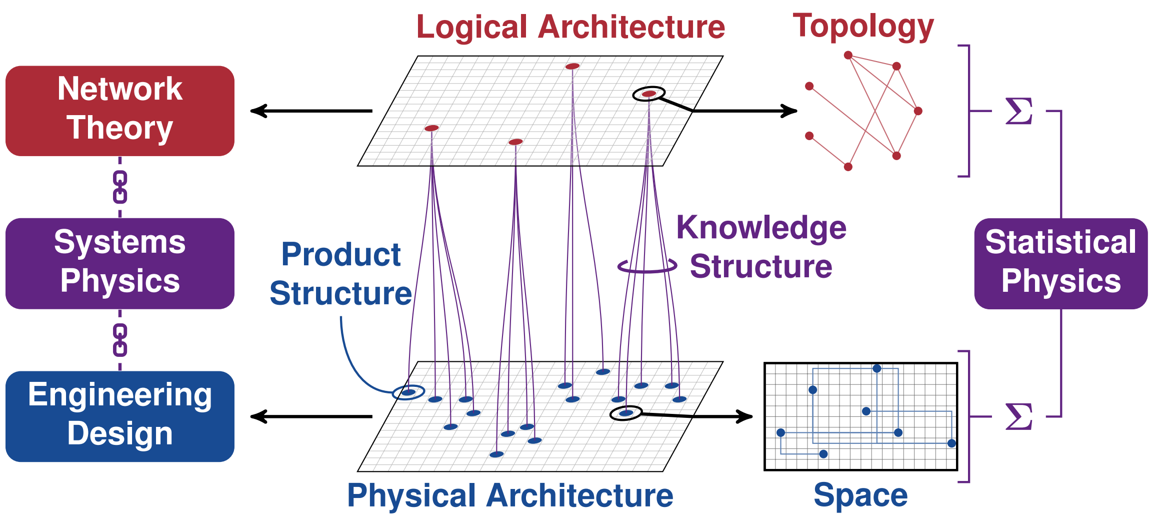


Figure 6.2: The need to grow from "product structure" approaches, the focus on single design instances, to "knowledge structure" approaches, the patterns of design outcomes or challenges that persist across collections of instances, suggests the applying the framework statistical physics. Statistical physics collectively sums the topological-level description of the logical architecture that expresses the system's underlying functionality and the physical/spatial description of design instances. Connecting the logical and physical descriptions of design in this way, the resulting "systems physics" picture that emerges bridges between traditional network theory and engineering design approaches.

design process around relationships between design elements that persist across instances.[7]

Searching across instances is key for identifying patterns of avoidance, adjacency, and association that are generic features of design problem spaces. Achieving this requires a different approach. To formulate this approach, the key challenge is in framing knowledge that emerges from collections of possible instances of a system. The problem of many instances that give rise to collective behavior is the underlying principle that motivated the development of statistical mechanics.[116] The fact that an analogous problem emerges in design, i.e. the need to formulate knowledge structures to identify patterns in design space, suggests that statistical physics could serve as the foundation for a similar approach. Fig. 6.2 illustrates this strategy of attack.

6.2.2 Statistical Physics Approach

The need to address the problem of identifying patterns of avoidance, adjacency, and association that persist across spaces of designs points to statistical physics as a framework. To construct this framework there are two key challenges: formulating the design problem as a statistical mechanics model, and extracting from the model the knowledge structures that encode design space properties.

To construct statistical physics models of design, we need two things: the space of states and some metric on this space. For design problems that are studied with optimization techniques like simulated annealing, these two things are already at hand. A generic approach for constructing a statistical physics design framework given a space of possible designs and a set of design objectives was developed in Chapters II–III.

We denote the *design space* as the set $\{\alpha\}$ consisting of individual detailed designs α. Each design α can be quantitatively evaluated with a *design objective* $\mathcal{O}(\alpha)$. Instead of exclusively focusing on the design that minimizes $\mathcal{O}$, we consider a probability distribution over designs. The maximal entropy distribution driven exclusively by the design objective is given by:[99]

$$p_\alpha = \frac{1}{\mathcal{Z}} e^{-\lambda \mathcal{O}(\alpha)}; \quad \mathcal{Z} = \sum_\alpha e^{-\lambda \mathcal{O}(\alpha)}, \tag{6.1}$$

where λ is the *design pressure*, or the relative importance of the corresponding design objective in driving the distribution. The normalization $\mathcal{Z}$ is known as the *partition function* and contains a wealth of information on the properties of the *whole* design space.

With this formulation of design problems as statistical mechanics models, the next challenge is to extract collective properties that encode information about the structure of the design problem and the space it lives in. Doing this typically requires computing sums over a combinatorially large set of α, which relies on various problem-specific mathematical techniques. For simple scenarios involving only one or two nodes with integrable interactions, Chapters III–V have shown that this can be done by coarse-graining to extract effective, so-called Landau, free energies. We expect that effective free energy approaches will, as they do in the ordinary statistical mechanics of particles, provide the means to gain insight into the collective properties of more complex design spaces. However for more complex design spaces, again as is the case in the ordinary statistical mechanics of particles, some mathematical techniques are required to study systems that lack closed-form, integrable interactions.

To meet the challenge of extracting information about design spaces with complex forms of interaction it is useful to take cues from the structure of the problem. For complex problems the advantage of the logical architecture is that it reduces the complexity of interactions among elements to simple, binary, yes/no connections. The disadvantage of this simplicity is that it loses the richness and specificity of the underlying design problem. This suggests a more complete treatment of the design problem would be to "decorate" the topological description of the logical architecture with information about the "topography" of the underlying design space and its physical architecture. This topographic decoration can be carried out by encoding the design space as a tensor network.

Tensor networks were originally introduced as a graphic notation for geometric tensors,[118] but over the last 25 years have grown into powerful computational tools for storing and manipulating high-rank data. The tensor network computations are especially efficient when the connections are sparse. This property spurred the popularity of tensor networks in a broad range of applications, from encoding entangled wavefunctions in quantum condensed matter systems,[120, 121, 122] to performing precision quantum chemistry calculations,[123] renormalizing lattice models,[125, 126] solving constraint counting problems,[130, 131] accelerating numerical linear algebra,[132, 133, 134] and learning multilinear classifiers in machine learning.[127] Across these applications, tensor networks serve as an *information structure* that contains an exhaustive but raw description of the system.

We use tensor networks to bridge the logical and physical descriptions of a design problem space. Network nodes encode the design elements' topography in the physical space of their placement and their properties. Network connections encode the functional connection topology and topography of the physical interaction of design elements based on their spatial location and physical properties. The topography–topology connection that the tensor network encodes has the useful side effects that it provides a simple graphical representation of the interaction of design elements, and that well-developed methods exist for extracting information from tensor networks.

Tensor networks encode information about the design space, knowledge about patterns of avoidance, adjacency, and association among design elements, has to be extracted. Extracting this information requires adding specifically formulated pieces to it that represent key design questions (in physics language, i.e., observables or order parameters). We formulate design questions about avoidance, adjacency, and association among design elements by acting on the tensor network with a combination of elementary "moves". The moves yield patterns of the placement over sets of solutions in design space, that are computed via contraction of tensor networks. See Methods for a detailed description of moves and contraction.

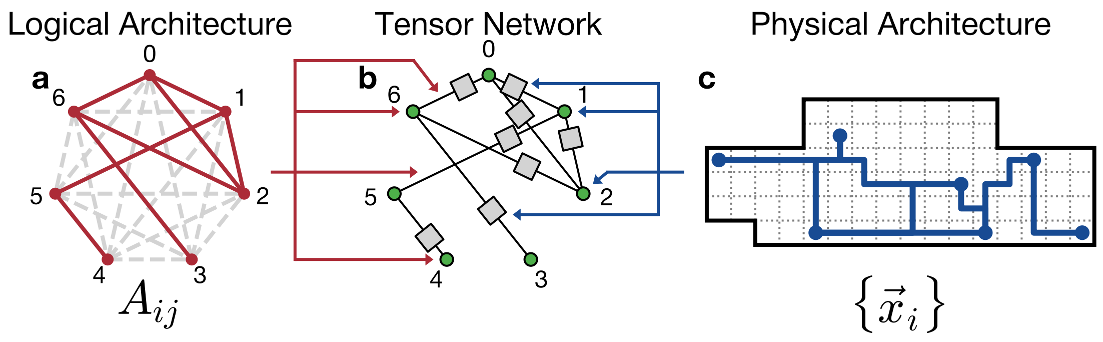


Figure 6.3: Tensor network bridges the logical and physical descriptions of the design space for an example Naval Engineering arrangement problem. (a) The logical architecture is represented graphically by a network of seven functional units (red circles) and their specific pattern of functional connections (red lines), and algebraically with the adjacency matrix A_{ij}. The dashed gray lines represent the non-links in the network, which do not directly drive the arrangement but can be investigated. (b) The structure of the whole design space is contained in an information structure in form of a tensor network. Logical architecture determines the network pattern in which the tensors are connected, while physical architecture determines the contents of both site and coupling tensors. (c) The physical architecture is represented graphically by a square grid within a complex hull shape, and algebraically by the set of possible unit locations $\{\vec{x}_i\}$. A particular arrangement consists of the placement of all seven functional units within the hull and the routing of all functional connections between them (blue circles and lines).

6.2.3 Example Model System

Functional Units and Connections—To demonstrate the Systems Physics analysis and tensor network computations on a concrete example of a design problem, we first define the specific logical and physical architectures for the problem. We use a problem from Naval Engineering,[157] in which the functional units of a shipboard system need to be arranged within the hull of a naval battleship, while respecting their functional connections, such as pipes or cables. The network pattern of connections constitutes the logical architecture, also represented algebraically as an adjacency matrix A_{ij}. We find a wide variety of network motifs arise in networks of as few as $n = 7$ functional units without any graph symmetries (Fig. 6.3a), which we will study in the remainder of this work.

Ship Hull—We position units and connections within a ship hull that we represent, following [157], by a 2D square grid with a complex, but fixed boundary (Fig. 6.3c). We constrain all connections within the ship hull to always run along a shortest path between the two functional units; we choose our hull to be L_1-convex to ensure that at least one shortest path exists between any pair of cells. We find that hull models with a few tens of cells are sufficient to establish placement patterns; computations reported below are for hulls with $Y_0 = 78$ distinct cells for unit placement each labelled as $\vec{x}_i$.

Design Objectives—Picking the locations of all units and the routings of functional connections between them defines a detailed design solution $\alpha = (\{\vec{x}_i\}, \mathsf{routing})$. In early-stage design, design architectures are typically not fixed, therefore the full combinatorial design space needs to be considered. Each detailed design is quantitatively evaluated with a *design objective* $\mathcal{O}(\alpha)$, here we model routing cost:

$$\lambda \mathcal{O}(\alpha) = \lambda \sum_{ij} A_{ij} C L_1(\vec{x}_i, \vec{x}_j) \ , \tag{6.2}$$

where L_1 is the "Manhattan" distance between the two cells, C is the cost per unit distance, and λ is the design pressure. Given the placement of units, we consider all allowed shortest paths between them. By definition, all shortest paths have the same length, so the value of $\mathcal{O}$ doesn't depend on the particular routing chosen, yet the number of routings is important. To account for the redundancy of routings, we introduce an *effective design objective*:

$$\lambda \mathcal{O}_{\mathsf{eff}}(\{\vec{x}_i\}) = \sum_{ij} A_{ij} f(\vec{x}_i, \vec{x}_j; T) \ ; \tag{6.3}$$

$$f(\vec{x}_i, \vec{x}_j; T) = \frac{C}{T} L_1(\vec{x}_i, \vec{x}_j) - \ln n_{\mathsf{rout}}(\vec{x}_i, \vec{x}_j) \ , \tag{6.4}$$

where n_{rout} is the number of shortest routings between $\vec{x}_i$ and $\vec{x}_j$ within the ship hull, typically growing with distance. The routing lengths $L_1(\vec{x}_i, \vec{x}_j)$ and the number routings $n_{\mathsf{rout}}(\vec{x}_i, \vec{x}_j)$ are fully determined by the shape of the hull and can be precomputed, stored as matrices, and scaled by the design pressure as needed.

Tensor Network—The representation of the design space in form of a tensor network depends on both logical and physical architectures (Fig. 6.3b). Logical architecture in form of the network A_{ij} determines the pattern in which the site and coupling tensors are connected. Physical architecture determines the set of available locations for all units $\{\vec{x}_i\}$ that is used as index for all tensors. The effective design objective $f(\vec{x}_i, \vec{x}_j; T)$ determines the entries of the coupling tensor. See Methods for a detailed mathematical discussion.

Greenfield/Brownfield Settings—In the above formulation, the design space $\{\alpha\}$ of the problem is the space of all possible arrangements of each functional unit $\{\vec{x}_i\}$, and is it necessary to establish a means of distinguishing units with fixed and variable position. This distinction is necessary because the formulation needs to address arrangement before or after some of the units have been placed. We refer to situations in which no units have fixed placement as greenfield settings. Greenfield settings are generically associated with green color-coding in results figures that follow. Also, we refer to situations in which one or more units have fixed locations as brownfield settings. Brownfield settings are generically associated with brown color-coding in results figures that follow. In Results figures that describe brownfield settings that combine placed units with yet-to-be-placed units we make a visual distinction between the two by color-coding placed units and their effects brown and yet-to-be-placed units green.

Low Cost, High Flexibility, and Crossover Regimes—The formulation of design problems in terms of spaces of solutions weighted by objectives of the form of Eq. (6.4) has been studied in Chapter IV. As in that Chapter we expect that the choice of the design pressure associated with each objective (general case: Eq. (6.1); this model: Eq. (6.2)) will have qualitatively distinct effects on design outcomes. To maximize generality, we study design pressures that correspond to multiple behavioral regimes. We do this by first expressing the design pressure via its inverse $\lambda = 1/T$, where T is the *cost tolerance.* Low cost tolerance means that minimizing the routing cost $\mathcal{O}$ is a strong driver of a detailed design choice, whereas high cost tolerance means that the choice among the detailed designs is not driven by cost. Chapter IV showed that the system driven by this design objective undergoes a large-scale rearrangement (akin to a phase transition, but at finite-size) around $T_{\mathsf{crit}} = C/\ln 2 \approx 1.44C$. We pick $C = 1$ to fix the measurement units for T. $T < T_{\mathsf{crit}}$ favors low cost and we therefore expect units to organize into motifs that facilitate short (cheap) connecting paths. We expect this setting to be characterized by effective attraction. $T > T_{\mathsf{crit}}$ favors maximal flexibility and we expect units to organize into motifs that facilitate maximizing routing degeneracy. We expect this setting to be characterized by effective repulsion. We expect that for $T \approx T_{\mathsf{crit}}$ where cost and flexibility drivers are competing on near-equal footing there will be a crossover in behavior.

6.3 Results

6.3.1 Avoidance

Void Premium—The interplay of logical and physical constraints among design elements induces a complex landscape for element placement. Intervening in that landscape by reserving space for future use could induce functional units to make a complex, collective rearrangement to avoid the reserved space. We characterize the cost of avoidance by computing the *void premium* that must be paid to forbid any units to be placed in the reserved space.

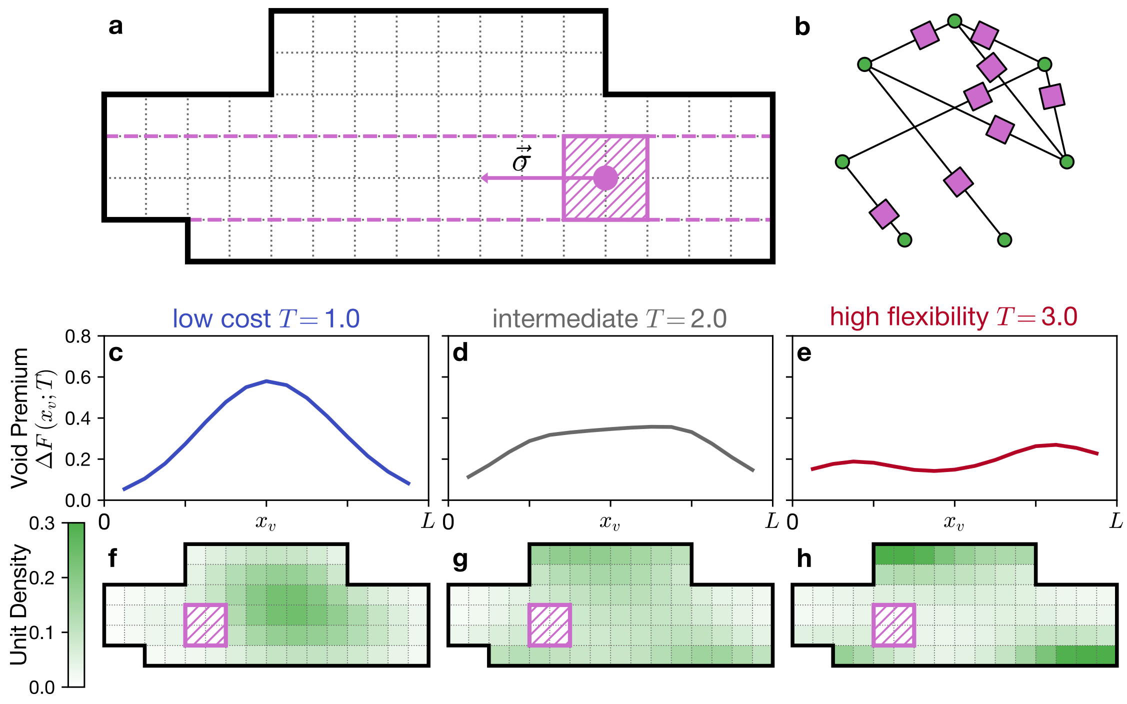


Figure 6.4: Void premium quantifies the cost of avoidance of reserved space across the whole design space in a greenfield scenario. (a) Schematic of the ship hull and square cells within (Physical Architecture). Pink square represents a void where unit placement is prohibited, driven by the void design stress $\vec{\sigma}$ along the center line of the hull (pink dashed lines). (b) Tensor network used to compute the void premium, with each coupling tensor modified. (c-e) Graphs of void premium (void free energy $\Delta F(x_v; T)$) against the void coordinate x_v for three values of T (color coded). (f-g) Functional unit density in presence of the void.

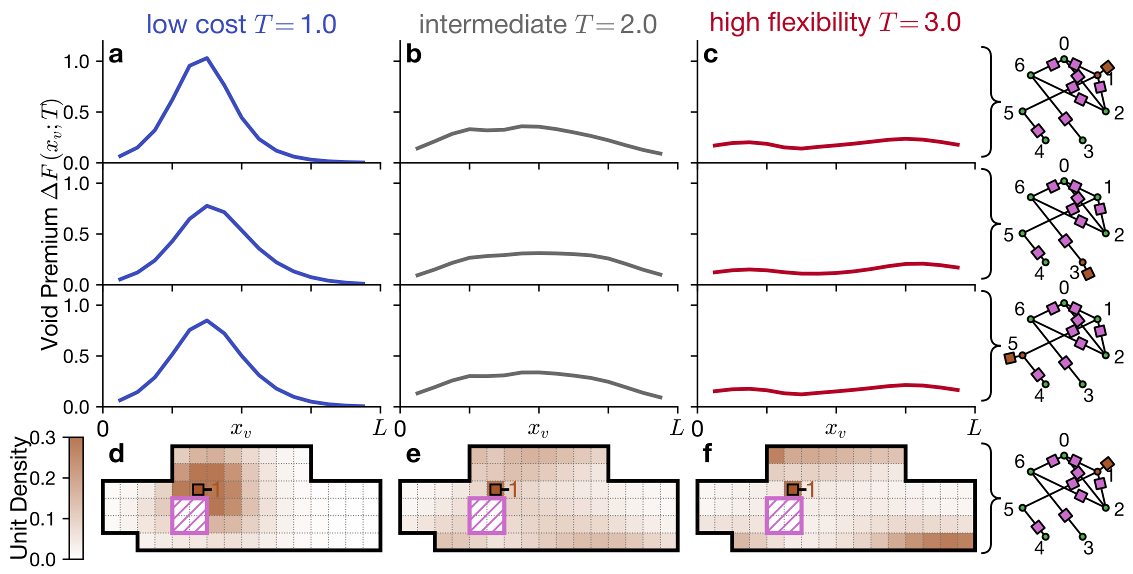


Figure 6.5: A brownfield scenario, such as anchoring one functional unit, sharpens the void premium curve. (a-c) Graphs of void premium (void free energy $\Delta F(x_v;T)$) against the void coordinate x_v for three values of T (columns, color coded) and different choice of the anchored unit (rows, anchor shown in the tensor network on the right). (d-e) Functional unit density in presence of the void and unit 1 anchored in the indicated cell (brown square).

We implement reserved space mathematically by creating a *void* 2×2 cells in size. In the results below we introduce the void on the hull midline, though in general it could be placed anywhere. We vary the horizontal location of the void, x_v, from zero to the hull length L (see Fig. 6.4a). We compute the cost of the void via the tensor network approach by suppressing several rows and columns of the coupling tensor (see Methods). Contracting the modified tensor network results in a modified partition function, which is smaller or equal to the original one $\mathcal{Z}(x_v;T) \leq \mathcal{Z}(T)$. The ratio of the two partition functions defines the non-negative free energy:

$$\Delta F(x_v;T) = -\ln \frac{\mathcal{Z}(x_v;T)}{\mathcal{Z}(T)}. \tag{6.5}$$

We take this void free energy as a measure of the void premium, the effective "cost" of the avoidance of a specified region in space.

Void Design Stress—The magnitude of the void premium ΔF corresponds to the placement opportunity cost for the functional units. To understand this opportunity cost, note that functional units that are not yet placed form a greenfield "cloud" of possibilities within the hull, the location and density of which depends on the cost tolerance T. Cutting a void from a dense part of the cloud costs a lot of free energy, whereas cutting a cloud from a sparse part of the cloud costs almost nothing. In this way, scanning the void free energy along the void coordinate x_v gives a direct probe of the morphology of the cloud. Conversely, if we regard the unit positions as fixed, and the void as moveable, the cloud of units drives the void with an effective force $\sigma = -\Delta F/\Delta x_v$, which we call *void design stress*. The void free energy and design stress are then a concise description of the collective effect of avoidance in functional unit placement.

The void premium and void design stress give a description of collective avoidance effects in placement. These effects can be examined in greenfield and brownfield settings.

Greenfield—We studied avoidance metrics in greenfield settings, i.e. before any unit data have been fixed, in three design regimes specified by cost tolerance T. We plot results in Fig. 6.4c-e. At subcritical $T = 1.0$ (low cost priority), the void free energy curve shows a clear single maximum in the middle of the hull, and two minima on the ends of the hull. At near-critical $T = 2.0$ (cost-flexibility tradeoff) the curve maintains the same qualitative shape, but the maximum gets flatter. At supercritical $T = 3.0$ (high flexibility priority), the curve shape flips to have a local minimum at the center of the hull, surrounded by local maxima on two sides. In other words, at low T the void prefers to be at either of the two ends of the ship (but a choice needs to be made in favor of one of them). In contrast, at high T

the void prefers to be in the center of the ship. Thus the change from designing for flexibility (high T) to designing for cost (low T) induces a change from one architecture class (central-void) to two architecture classes (bow-void and stern-void). This collective effect is analogous to symmetry-breaking phase transitions in conventional physical systems.[107]

To understand the origin of the symmetry breaking we note that void free energy is a proxy for the morphology of the unit cloud. To illustrate the shape of the cloud in a different way, we approximate the cloud density as a sum of one-unit densities $\rho(\vec{x}) = \sum_i p_i(\vec{x})$ and plot densities as heatmaps in Fig. 6.5f-h. These heatmaps are approximate because, unlike the void free energy curves, they ignore the correlations in unit placement. At low T (panel f), units attract each other and thus preferentially form a cloud in the center of the hull and push the void to either side of the hull. At near-critical T (panel g), the distribution becomes more homogeneous throughout the hull, flattening the curve. At high T (panel h), functional units strongly repel one another, concentrating near the edges of the hull. This leaves the center nearly empty, resulting in a single void free energy minimum.

Brownfield—Both the void free energy curve and the unit cloud morphology can, however, change dramatically in brownfield settings, e.g. if even one unit is fixed to a specific location in space. We pick the location indicated with the brown square in Fig. 6.5d-f and fix one unit there. We choose three different units to fix: unit 3 (which has 1 functional connection), unit 5 (2 functional connections) and unit 1 (3 functional connections). We plot the resulting void free energy curves in panels a-c, and unit clouds in panels d-f.

Consider first the low-cost regime $T = 1.0$ (Fig. 6.5a,d). In the greenfield setting

(Fig. 6.4f) units positions were determined solely by ship geometry, and formed a dense cloud in the middle of the ship (Fig. 6.4f). Fixing a unit position places an additional constraint on unit positions, and forces the unit cloud to condense around it (Fig. 6.5d). Because of this condensation, the void free energy curve becomes simultaneously steeper and more focused around the fixed unit point (panel a), but decays faster close to the edges of the hull. The void free energy cost also depends on the topological position of the anchored unit: it is highest for the most-connected unit 1 (bottom curve) and lowest for the least-connected unit 3 (top curve).

At the near-critical and supercritical $T = 2.0, 3.0$ the anchored unit similarly creates a reference point for the cloud, but the units in the cloud *repel* from that point. When repulsion and attraction are nearly balanced at $T = 2.0$, the cloud profile becomes nearly uniform and is not strongly affected by the fixed unit (compare Fig. 6.4g and Fig. 6.5e). Similarly, fixing a unit at supercritical $T = 3.0$ creates a point of strong repulsion, forcing the unit cloud to the opposite corners of the ship hull (Fig. 6.4h and Fig. 6.5f). At both values of $T = 2.0, 3.0$, the cloud morphology is not affected strongly by the single fixed unit position, and thus the brownfield void premiums (Fig. 6.5b-c) closely resemble their greenfield couterparts (Fig. 6.4b-c). Discussion of the effects of avoidance on unit positions, i.e. backreactions on the cloud, can be found in Supplementary Materials (SM).

Avoidance: Logical–Physical Architecture Interplay—The above avoidance analysis gives a case study of basic phenomenology of the interplay between design pressure (favoring low-cost vs high-flexibility) and the logical and physical architecture. Shifting the design priority from low-cost to high-flexibility changed the interaction between pairs of functional units. However, unit interactions were modulated by connection topology (i.e., logical architecture) and by the spatial domain

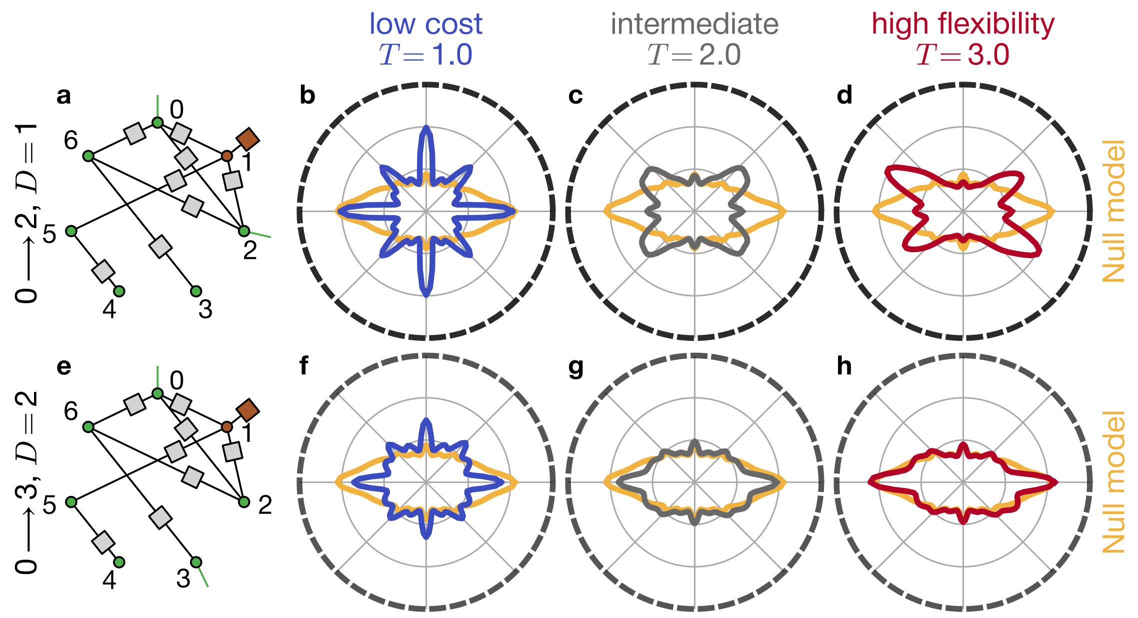


Figure 6.6: Bond diagrams quantify the adjacency patterns for both directly and indirectly connected functional units. (a-b) Tensor networks used for computations of bond diagrams for two pairs of units, one connected directly ($0 \to 2$, $D = 1$), the other indirectly ($0 \to 3$, $D = 2$). The site tensors in the measured pair have external legs (green). The unit 1 is anchored in the center of the hull (brown square). (c-h) Bond diagrams for the angle between the two units of the pair, for two different pairs (rows) and three values of T (columns, color coded). Yellow curve shows the null model of the bond diagram, identical for all graphs. Black rim of bond diagram axes indicated the topological distance $D = 1$, gray rim indicates $D = 2$.

(i.e., the physical architecture). We captured the effect of these complex interactions on spatial avoidance the unit clouds in Figs. 6.4,6.5. However, within the unit clouds, the interplay of design pressure with the logical and physical architectures also induces emergent coupling. This emergent coupling within the cloud induces patterns of adjacency and association between units, which we turn to next.

6.3.2 Adjacency

Whereas our avoidance analysis derives from and illustrates the basic morphology of the unit cloud within the ship hull, questions about unit adjacency derive from correlations within the cloud, and the emergent coupling between units that arise.

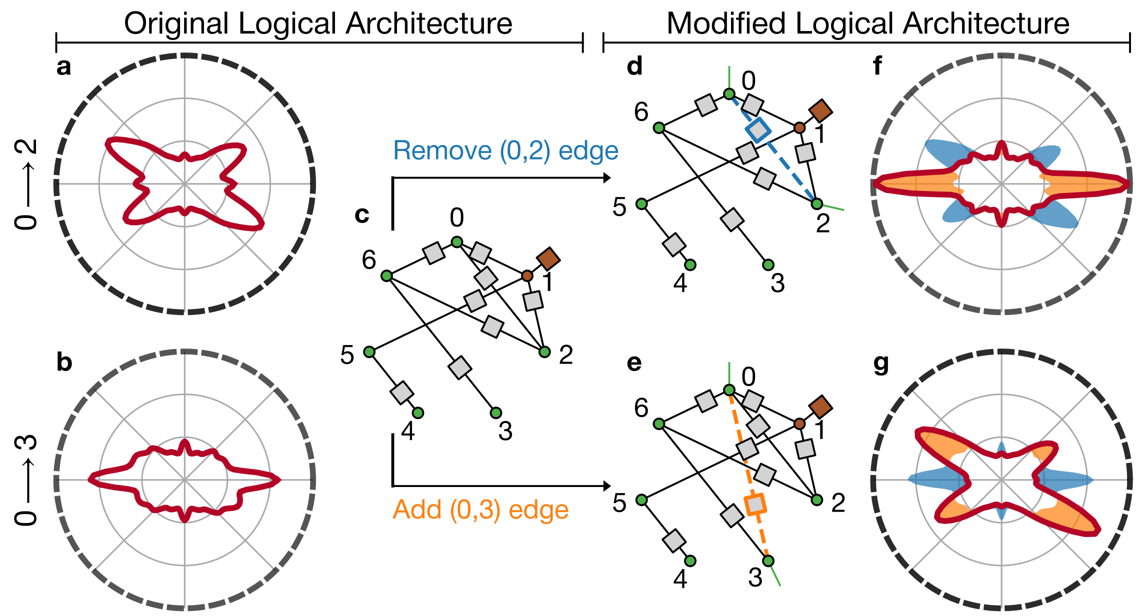


Figure 6.7: Small changes of Logical Architecture lead to large changes of adjacency patterns, visible in the difference of bond diagrams. Panels a-c correspond to the original Logical Architecture, while panels d-g illustrate two different changes of Logical Architecture. (a-b) Bond diagrams at $T = 3.0$ for two pairs of units, $0 \rightarrow 2$ ($D = 1$) and $0 \rightarrow 3$ ($D = 2$). (c) Tensor network used for computations of bond diagrams with original Logical Architecture, external legs not shown. (d) Tensor network used for computation of the $0 \rightarrow 2$ bond diagram when the functional connection (0,2) is removed (blue dashed line). (e) Tensor network used for computation of the $0 \rightarrow 3$ bond diagram when the functional connection (0,3) is added (orange dashed line). (f-g) Resulting bond diagrams (dark red curve) with highlighted positive difference (orange shading) and negative difference (blue shading) with respect to the bond diagrams at the original Logical Architecture (panels a-b). Black rim of bond diagram axes indicated the topological distance $D = 1$, gray rim indicates $D = 2$.

Bond-Diagram Measures of Adjacency—To determine how emergent coupling between units leads to arrangment motifs, we consider pairs of units and their relative positions. We examine pairs of units in 2D space, and express motifs as the polar angles $\theta(i \to j)$, which vary along with the positions of the units in the cloud. Across the cloud, the angle takes form of a probability distribution $p_{i\to j}(\theta)$. This distribution is analogous to the bond order that is used to describe structure in condensed matter.[194, 195] In condensed matter, bond angles are whole-system aggregate measures of adjacency. Here, the heterogeneous connectivity of design elements yields "bond" diagrams that are specific to each pair of units i, j. Depending on whether the units i and j are directly connected or not ($A_{ij} = 1$ or 0), the bond diagrams illuminate the strength of direct or emergent adjacency patterns.

Computational Approach—To compute the bond diagrams mathematically, we use tensor networks to compute the raw 2-unit marginal distributions $p(\vec{x}_i, \vec{x}_j)$ (Fig. 6.6a,e), and convert them into the angular distributions $p_{i\to j}(\theta)$ using Kernel Density Estimation to reduce the numerical artifacts (see Methods). In order to demonstrate more sharply defined bond diagrams, we assume that one unit has already been placed (anchored) in the center of the hull and all other units need to be placed with respect to it. The bond diagram $p_{i\to j}(\theta)$ of any pair of units is not uniform with respect to the angle θ even if the units are not connected at all, directly or indirectly. This non-uniformity is driven by the shape of the hull, a manifestation of the physical architecture, and we account for it by computing *null model* $p_0(\theta)$ of the bond diagram(see Methods). Differences between the null model and computed bond diagrams are indicators of interaction-driven adjacency. This interaction driven adjacency depends on unit connectivity; we use a *topological distance*, $D(i, j)$ metric. $D(i, j)$ is the minimal number of network hops to get from unit i to unit j. In our example problem, the minimal number of hops varies from 1 (e.g. units 0→1) to 5 (e.g. units 3→4).

Direct Adjacency—We show topological distance, bond diagrams, and the null model for our model in Fig. 6.6 in form of polar plots for two example unit pairs (corresponding plots for all unit pairs are given in SM). We start discussion with the bond diagrams for direct adjacency (0→2, $D = 1$). At subcritical $T = 1.0$ (panel b) most units are located very close to each other, either in cardinal or intercardinal directions (orthogonally or diagonally), resulting in a bond diagram with a strong eightfold signal. At near-critical $T = 2.0$ (panel c) the orthogonal attraction is balanced with diagonal repulsion, resulting in a bond diagram with smaller peaks. At supercritical $T = 3.0$ (panel d) the units are located relatively far from each other and prefer diagonal relative location (since diagonal location allows them to maximize their routing entropy), resulting in a fourfold, X-shaped signal. The symmetry of the fourfold signal is further broken by anchoring the unit 1. This additional symmetry breaking is driven by the high density of units in top-left and bottom-right corners of the hull (see Fig. 6.4).

Emergent Adjacency—Across the whole T range, the bond diagrams for direct adjacency are significantly different from the null model. However, adjacency can also be induced for indirectly connected unit pairs. The indirectly connected unit pair 0→3 shows emergent adjacency, since its bond diagram is different from both the direct adjacency and the null model (Fig. 6.6f-h). The bond diagrams for unit pairs with even larger topological distance the bond diagrams gradually approach the null model (see SM), following the intuition of decay of correlation functions with distance in condensed matter systems. This observation suggests the general takeaway: at large topological distance bond diagrams always approach the null model, and thus are fully determined by the Physical Architecture; at small topological distance the bond diagrams depend strongly on both the design pressure T and the explicit details of Logical Architecture. However, our results reveal that for emergent adjacency topological distance is a good predictor of *strength* but is not a good predictor of *shape.*

Logical Architecture Modifications—To further test the interplay between Logical Archiecture and Adjacency, we investigate what happens to bond diagrams when we modify the Logical Architecture. We consider two types of modifications: removing an existing functional connection (Fig. 6.7d), or adding a new one between two units (Fig. 6.7e). Instead of comparing the resulting bond diagrams to the null model again, we focus on the difference between bond diagrams before and after modification (panels f-g). *Addition:* The effect of adding the (0,3) connection is strong. We can anticipate that with the added direct adjacency, the adjacency pattern should approach that of other directly adjacent units. The fourfold signal that results (panel g) is a signal of this. Since neither of the units 0 or 3 is explicitly anchored in space, at high $T = 3.0$ they want to be positioned at the opposite ends of the longest diagonal available within the hull, in this case the diagonal from top-left to bottom-right corners, similarly to the original adjacency $0 \rightarrow 3$ (Fig. 6.6d). *Removal:* Like for addition the effect of removing the (0,2) connection is dramatic, however the result is unexpected. Instead of fourfold diagonal direct adjacency, the two units now have twofold horizontal emergent adjacency (panel f). The reason for this is that with the (0,2) connection removed, the units 1-2-6-0 now form a rhombus. Unit 1 is fixed in space, and because of high $T = 3.0$ all unit pairs prefer to have diagonal adjacency. In this case the units 0 and 2 on the opposite corners of the rhombus will have an orthogonal adjacency, of which only horizontal adjacency manifests because the ship hull is larger in length than height.

Adjacency: Changes and Constraints Drive Patterns—We showed that the irregular, complex-network nature of a system's Logical Architecture drives the patterns of direct and emergent adjacency. We showed adjacency patterns can change significantly with changes in Logical Architecture. One outcome of this approach was the ability to detect emergent adjacency. The emergent effects we observed were with a single fixed unit. Though having having few fixed units is a characteristic of early-stage design, later-stage design situations will result in more fixed units. Fixing more units will induce more constraints, and further constraints will complicate the interplay of the logical and physical architecture. A more complicated logical–physical architecture interplay should induce more complex patterns of association between units, which we will examine next.

6.3.3 Association

Analyzing association patterns extends our avoidance and adjacency investigations to situations in which multiple, pre-existing constraints restrict functional units, i.e. in brownfield settings. These settings model either of two situations: (i) actual late-stage design in which multiple functional units have been fixed during preceding design stages, or (ii) an early-stage design investigation of hypothetical late-stage situations under different decision scenarios.

Constraints and Localization—In either case, the expectation that multiple active constraints will drive complex forms of interaction suggests that identifying patterns of association that result will require different techniques than identifying patterns of avoidance and adjacency. In general we expect that patterns of association arising from multiple constraints will localize those patterns relative to fixed design elements. This suggests that metrics of association patterns should

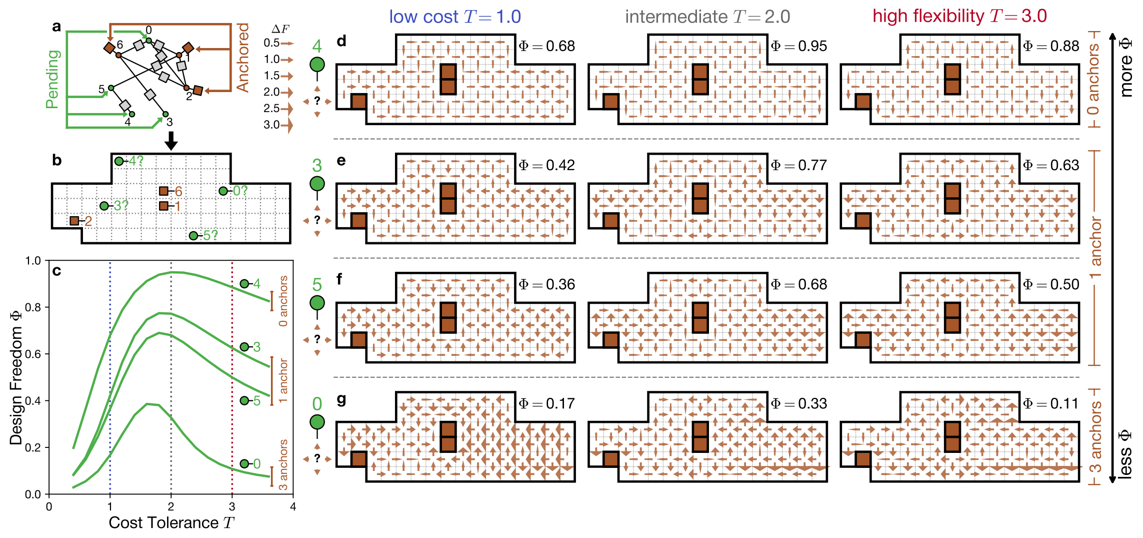


Figure 6.8: Early stage design decisions determine the association patterns and design freedom for subsequent ones. (a) Tensor network used for association computations. Three out of seven units have already been anchored to specific locations (brown), other four are pending placement (green). External legs not shown. (b) The units 1,2,6 are anchored at the indicated locations within the ship hull (brown squares). The units 0,3,4,5 can still be placed in many locations, some demonstrated for an example (green circles). (c) Graph of the design freedom Φ for the four units without anchors across a range of cost tolerance T. Brown brackets on the right indicate that units with more adjacent anchors have less remaining design freedom. Vertical dotted lines indicate the T values investigated in more detail in panels (d-g), as well as in avoidance and adjacency patterns. (d-g) Design stress ΔF patterns for the placement of each pending unit (rows in order of decreasing design freedom Φ) at three values of T (columns, color coded). Legend for design stress magnitude ΔF is shown to the left of panel (d).

signal a tendency toward (or away from) placement proximity relative to fixed elements, either globally or locally. Here, for a global signal we adapt measures of emergent localization to compute a scalar design freedom for unit locations. For a local signal we compute the design stress associated with specified, hypothetical unit placement.

To study the effect of added constraints, their interplay with the logical and physical architecture and the resulting localization, we employ the same model system as in the avoidance and adjacency investigations. However we introduce constraints that fix units 1,2, and 6 to specific locations. We investigate the emergent localization of the other 4 units with two metrics via the global design freedom Φ and local design stress ΔF. Results are shown in Fig. 6.8 and broken down below.

Global Signal of Association: Design Freedom Mathematically, both global and local metrics of localization use a tensor network computation of the marginal probability distribution $p(\vec{x}_i)$. The conversion of the distribution into design freedom is inspired by the metric of existence area, commonly used in studies of the Anderson localization of wavefunctions in disordered media and the localization of vibrational eigenmodes.[196, 197] We define design freedom as:

$$\Phi = \frac{1}{Y_0} \frac{\left(\sum_{\vec{x}} p(\vec{x}) \right)^2}{\sum_{\vec{x}} p(\vec{x})^2} \ , \tag{6.6}$$

where the normalization Y_0 is the total number of cells within Physical Architecture; in this example $Y_0 = 78$. Given this normalization, Φ takes a value between 0 and 1 and has the meaning the effective fraction of the total area available for unit placement, if the distribution was uniform. For a unit with uniform distribution $p(\vec{x}_i) = const$, Φ would be 1, whereas for an anchored unit Φ would be $1/Y_0 \to 0$.

Because of the heterogeneous connectivity of the logical architecture, design freedom Φ varies between units. The variation between units is in addition to variation with design pressure, via changing cost tolerance T. Fig. 6.8c plots $\Phi(T)$ by unit, and shows that all units have design freedom peaks near $T \approx 2.0$. In the range of this near-critical T, cost (effective attraction) and flexibility (effective repulsion) drivers of unit interactions balance and allow the units to explore the largest range of placement. As well, we observe $\Phi(T)$ to fall into three groups according to how constrained each unit is. Unit 4 is not directly connected to any of the anchored units and thus enjoys the largest design freedom, almost approaching $\Phi = 1$. Units 3 and 5 are each connected to one anchored unit and thus have intermediate Φ. Unit 0 is connected to all three anchored units and thus has the lowest Φ which quickly decays at both low and high T.

Local Signal of Association: Design Stress—Whereas Φ serves as an global scalar metric of design freedom, it is also important to understand how global design freedom is distributed locally. This local distribution is captured by design stress. Design stress is closely related to an effective (Landau) free energy (LFE), defined as follows:

$$F(\vec{x}_i) = -\ln p(\vec{x}_i) + C \ , \tag{6.7}$$

where C is an arbitrary additive constant. We chose a convention where C is such that the minimal value of F is zero. The LFE can be interpreted as an effective design objective for the chosen degree of freedom, given that other degrees of freedom have been fixed or integrated out. This interpretation is analogous to the void premium (Eq. 6.5) in our avoidance investigation, but instead of the effects of unit placement on voids, here we examine the effects of unit placements on one another. Similar to the definition of void design stress via a spatial difference, the difference of LFE between two horizontally or vertically adjacent cells is the design stress ΔF. Design stress is then an effective "force" that pushes individual functional units towards their preferred locations (Chapters IV–V). Compared with global design freedom, design stress patterns give a more detailed picture of effective localization.

Fig. 6.8d-g presents the design stress patterns for all four pending units at three values of T. Design stress is represented by brown arrows drawn across the boundary of two adjacent cells and pointing from higher to lower LFE. In other words, to decrease LFE and reach lower values of its effective design objective, a unit needs to follow the arrows towards a basin. As the basin gets smaller and its walls get steeper, the pending units become more closely associated with the anchored ones and thus exhibit stronger emergent localization. The localization effect is strongest at lowest T, where all of pending units are strongly attracted to the two anchored units 1,6 in a single basin. The basin is steepest for the most constrained unit 0, less steep for the units 3,5, and the shallowest for the least constrained unit 4, consistent with our expectation based on design freedom Φ. At higher $T = 2.0$, 3.0, the constrained unit 0 develops complex LFE and design stress landscapes with multiple local minima, maxima, and ridges (panel g). Units 3,5, connected respectively to the anchored units 6 and 1 in the center of the hull, show an X-shaped pattern of LFE (panels e,f), similar to the bond diagrams for directly connected unit pairs, e.g. Fig. 6.6d. Lastly, the unit 4 is not connected to any of the anchored units, instead it is "dangling off" unit 5 and thus shows almost nonexistent design stress across the whole hull (panel d).

Association: Interaction and Decision Drivers—Both the design stress and design freedom metrics show that the association patterns and the emergent localization phenomenon strongly depend on the position of both the fixed and the pending units within the logical architecture. The logical architecture alone gives an interpretation of the emergent localization result by counting the anchored neighbors. However, fully predicting localization requires examining the logical–physical architecture interplay that arises from the systems physics analysis. Unlike the simplified unidirectional design stress discussed in Chapter V, in this system the design stress pattern is emergent both from unit interactions and from previous design decisions. Chaining design decisions into sequences and achieving optimal control of emergent localization stands out as an important question for further study.

6.4 Discussion

In this Chapter we showed that questions of avoidance, adjacency, and association among the elements of complex, distributed systems hinge on the interaction between logical- and physical-architecture description planes. We bridged these descriptive planes with statistical physics techniques and showed that patterns of avoidance, adjacency, and association can be mapped for an example system.

Design Phenomena: Symmetry-Breaking, Emergent Adjacency, Localization— Our mapping of avoidance gave a space premium landscape. We found this landscape to undergoe a symmetry-breaking transition with a change from design pressure that prioritizes high flexibility to pressure that prioritizes low cost (Figs. 6.4,6.5). Our mapping of adjacency gave a description analogous to "bond" directions in matter systems. From this bonding description we observed that indirectly connected design elements developed emergent adjacency (Fig. 6.6). We also found large downstream changes in adjacency from small changes in underlying connectivity (Fig. 6.7). Our mapping of association patterns quantified changes in global design freedom driven by fixing design elements and changes in design pressure. Mapping

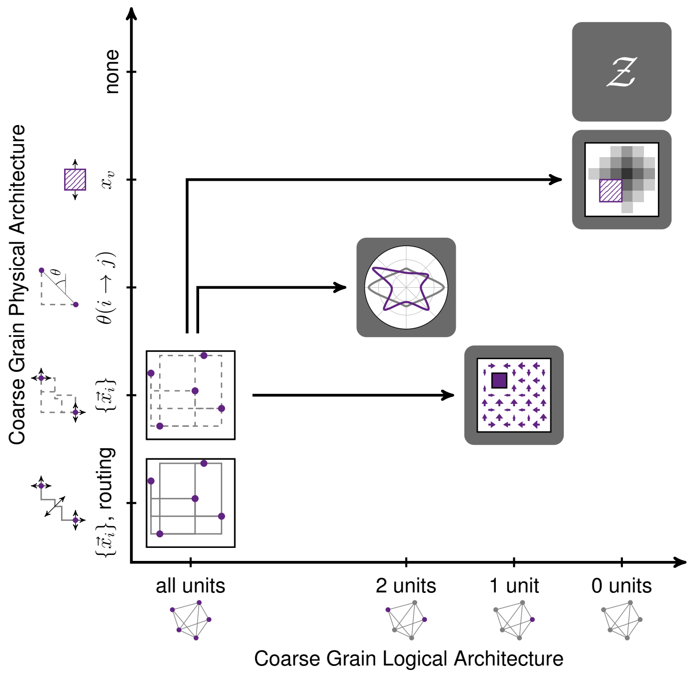


Figure 6.9: System exploration across levels of detail in physical and logical architectures via two orthogonal directions of coarse-graining. Horizontal axis represents the reduction in the number of functional units explicitly considered. Vertical axis represents the reduction in spatial detail considered. Black arrows represent the computational pathway across the study of avoidance, adjacency, and association patterns. The full design objective $\mathcal{O}$ (Eqn. 6.2) depends on all unit locations and routings. The effective design objective $\mathcal{O}_{\text{eff}}$ (Eqn. 6.4) depends only on locations of all units. The association pattern considers detailed location $\vec{x}_i$, but only for a single unit. The adjacency pattern reduces the spatial detail to just the relative direction $\theta(i \to j)$ between two units. The avoidance pattern further reduces the spatial detail to just the void position x_v as a single collective coordinate of the unit cloud. Finally, the partition function $\mathcal{Z}$ loses all detail of physical and logical architecture and summarizes the properties of the whole design space.

these effects locally showed the emergent localization of design elements (Fig. 6.8).

Coarse Graining for Other Design Contexts—Our mappings of avoidance, adjacency, and association patterns were done for a model system motivated by problems in Naval Engineering. However, for other design contexts where questions of avoidance, adjacency, and association patterns arise, our statistical physics approach opens new lines of attack. In particular, our approach can be summarized in two steps. First we "decorated" the logical architecture with detail from the physical architecture. Then, we systematically chose two sets of system details, one to examine in detail, and the other to treat in aggregate, in a coarse-grained way. The aggregated details induce effective patterns of interaction among the remaining elements, that reveal underlying patterns of arrangement. We illustrate this strategy in Fig. 6.9. Fig. 6.9 casts the strategy into two orthogonal forms of coarse-graining: one in the physical architecture, the other in the logical architecture. In this representation, in statistical physics language, microstates that retain complete detail of both the physical and logical architecture sit in one corner, whereas the partition function, which aggregates microstates into a single scalar sits in the opposite corner. Though the specific locations that correspond to our investigations are given at specific points on these axes, regardless of design context, answers to questions about patterns of avoidance, adjacency, and association lie at intermediate levels of detail between those extremes.

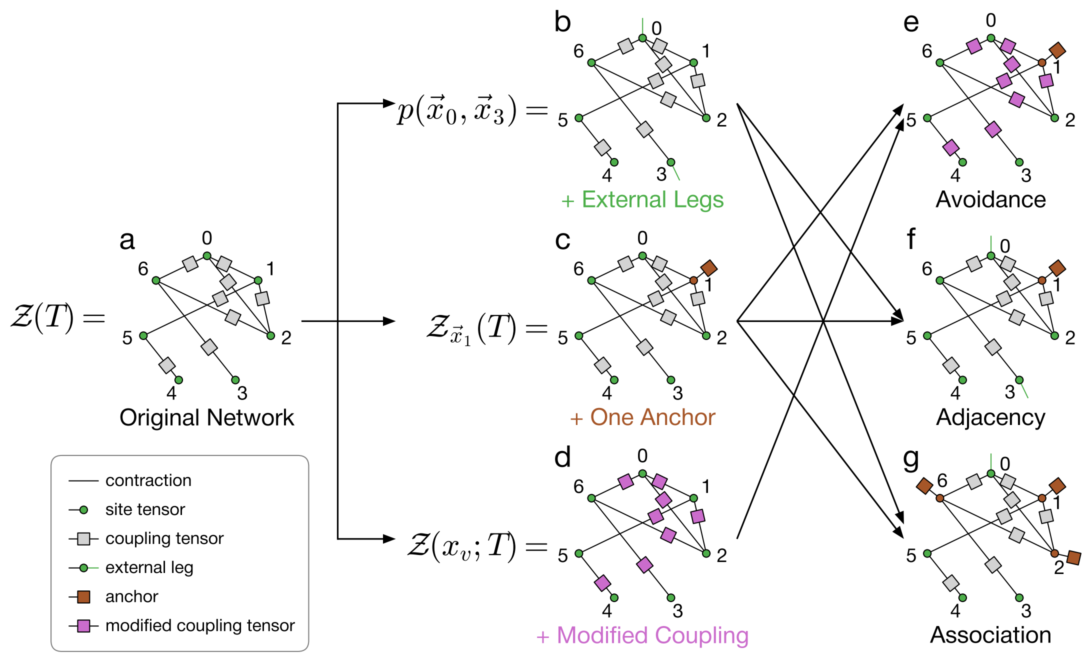


Figure 6.10: Tensor networks can be used as information structures as combinations of basic moves express complex questions about the design space. Panel (a) shows the original network, panels (b-d) describe the three basic moves that modify its topology, panels (e-g) show how the elementary moves are recombined to study the emergent patterns. (bottom-left corner) Legend of tensor network elements. (a) The original network connects $n = 7$ site tensors (green circles) with coupling tensors (gray squares) in the same pattern as the Logical Architecture (Fig. 6.2 top). Since this network has no outgoing legs, it contracts into a single scalar number equal to the partition function $\mathcal{Z}(T)$. (b) Move 1 adds extra outgoing legs on site tensors 0 and 3 (green lines), making the contraction result in the rank-2 tensor containing the joint marginal probability distribution on the spatial positions of the two units $p(\vec{x}_0, \vec{x}_3)$. (d) Move 2 attaches an additional rank-1 anchor tensor (brown square) to site 1, fixing it to a specific spatial location. This network contracts to the conditional partition function $\mathcal{Z}_{\vec{x}_1}$. (d) Move 3 modifies all of the coupling tensors (shown as pink squares), for example to account for the voids. This network contracts to the modified partition function $\mathcal{Z}(x_v; T)$. (e) For the avoidance pattern, we use both anchors and modified couplings to compute the placement opportunity cost via Eqn. 6.5. (f) For the adjacency pattern, we fix node 1 with an anchor and compute the 2-unit marginal distributions $p(\vec{x}_i, \vec{x}_j)$ for all possible pairs of external legs i, j, and further convert them into bond order diagrams. (g) For the association pattern, we encode the past design decisions with anchors on units 1, 2, 6 and study the 1-unit marginal distributions on each of the other units.

6.5 Methods

6.5.1 Tensor Network Construction

The partition function of the system can be expressed in terms of the effective design objective in the following factorized form:

$$\mathcal{Z}(T) = \sum_{\{\vec{x}\}} \prod_{\substack{i<j: \\ A_{ij} \neq 0}} e^{-f(\vec{x}_i, \vec{x}_j; T)}. \tag{6.8}$$

We interpret the factorized partition function (6.8) graphically in the form of a tensor network (Fig. 6.10a). Like other networks, a tensor network consists of nodes and links. Each node is a tensor with rank equal to node degree, and the network links represent the pattern of tensor contractions. We construct tensor networks following a recent prescription of Ref. [137], in which the tensor networks are bipartite: they consist of *coupling tensors* and *site tensors*, with each type only connected to the other type as shown in Fig. 6.10a. Logical architecture A_{ij} is contained in the pattern of the coupling tensors, whereas physical architecture domain $\{\vec{x}\}$ serves as the index of all the tensors, and the design objective determines the contents of the tensors.

The basic tensor network (Fig. 6.10a) represents the partition function $\mathcal{Z}(T)$ and is merely an alternative, graphical way to express Eq. 6.8. Each multiplicative term in the partition function becomes a rank-2 coupling tensor with elements defined as $M_{\vec{x}_i \vec{x}_j} \equiv e^{f(\vec{x}_i \vec{x}_j; T)}$ (note elementwise rather than matrix exponentiation). The site tensors have the form of a multi-dimensional Kronecker delta with the rank corresponding to site's degree in the logical network A_{ij}. For example, the unit 0 in Fig. 6.10a has three network neighbors, and thus corresponds to a rank-3 tensor $\delta_{\vec{x}_{01} \vec{x}_{02} \vec{x}_{06}}$. The indices mean the value of unit 0 coordinate $\vec{x}_0$ that is "presented" to each network neighbor, in this case units 1, 2, and 6. The Kronecker delta ensures that each neighbor perceives the unit 0 at the same location, while index summation ensures that all possible locations are considered. The rank of site tensors can be adjusted for other computations, as shown below. For brevity of notation, we suppress indices on edges because the contraction pattern is dictated by the graphical notation.

Contracting all the tensors along the network links corresponds to performing the sum in Eq. 6.8. Since that sum has no free indices, the network of Fig. 6.10a has no external (unpaired) legs. Contraction of the network preserves the number of external legs, resulting in a rank-0 tensor, a scalar number. Since each coupling tensor implicitly depends on cost tolerance T, the result of contraction is the T-dependent partition function $\mathcal{Z}(T)$.

In order to compute quantities other than the partition function, we add minor modifications of the tensor network. These modifications are described in the graphical language of moves. Here we define three moves: adding external legs, adding anchors, and modifying the coupling tensors (Fig. 6.10b-d). These moves are recombined to create networks that address the patterns of avoidance, adjacency, and association (Fig. 6.10e-g).

6.5.2 Move 1: External Legs

The first move adds extra legs to specific site tensors to control whether specific design degrees of freedom are marginalized or not. If none of the degrees of freedom are marginalized, then carrying out the multiplication but not the summation in the sum (6.8) would result in an un-normalized joint probability distribution $p(\{\vec{x}_i\})$ over all the units, which is a rank-N tensor of prohibitive size. However, following the usual probability theory calculus, in a joint probability distribution each of the entering variables can be in three states: joint, marginalized, or conditional. In the tensor network representation of Fig. 6.10a, *every* variable $\vec{x}_0, \vec{x}_1, \ldots$ is marginalized, resulting in the distribution normalization, i.e. the partition function $\mathcal{Z}(T)$.

In this perspective, a special action needs to be taken to *not* marginalize some of the variables. We do this by adding external legs to the corresponding site tensors (green lines in Fig. 6.10b). External legs are the site degrees of freedom that are *not* summed over, functioning as free indices for the sum (6.8). The result of contracting the network in Fig. 6.10b is a rank-2 tensor that represents the un-normalized joint probability distribution $p(\vec{x}_0, \vec{x}_3)$. Since the original network contracted to yield the full $\mathcal{Z}$, the normalized probability distribution can be expressed as $\tilde{p} = p/\mathcal{Z}$.

6.5.3 Move 2: Anchors

The second move adds anchor tensors to the network. The anchors represent the design decisions already taken and woven into the information structure, thus encoding the brownfield aspects of design. In tensor network language, this is equivalent to fixing some of the local degrees of freedom $\vec{x}_i$ and thus summing over a restricted ensemble, *conditional* on the fixed $\vec{x}_i$. We do this by creating an additional tensor that we call an "anchor". An anchor is a rank-1 tensor (vector) that is coupled to a site tensor and is illustrated as purple square in Fig. 6.10c. The elements of an anchor vector are given by the Kronecker delta $\delta(\vec{x}_i, \vec{x}_a)$, where $\vec{x}_i$ is the index connected to the site and $\vec{x}_a$ is the specific location to which the functional unit is pinned as result of a design decision. Since we didn't create any external legs, the tensor network in Fig. 6.10c also contracts to a scalar number of *conditional* partition function $\mathcal{Z}_{x_6}$ that functions as a similar statistical summary of the system as the original partition function $\mathcal{Z}$.

6.5.4 Move 3: Modified Coupling

The third move modifies the coupling tensors $M(\vec{x}_i, \vec{x}_j)$ and traces the effect of this modification on the partition function. In our study, we use the modification to account for the void where no units can be placed. The modification consists of suppressing the statistical weight of the void cells in the coupling tensor:

$$M^*(\vec{x}_i, \vec{x}_j) = M(\vec{x}_i, \vec{x}_j) \prod_{\vec{x}_v} (1 - \delta(\vec{x}_i, \vec{x}_v))(1 - \delta(\vec{x}_j, \vec{x}_v)), \tag{6.9}$$

where $\vec{x}_v$ denotes the positions falling into the excluded void. In the network on Fig. 6.10d we modified each coupling tensor in this way (marked in pink). The network results in the modified partition function $Z(x_v; T)$, from which we compute the void free energy via Eq. 6.5.

6.5.5 Bond Diagrams

To compute bond diagrams, the raw two-unit distributions $p(\vec{x}_i, \vec{x}_j)$ need to be converted into the angular distributions $p_{i \to j}(\theta)$ in post-processing. Since all functional units are placed in a discrete, finite, and fixed set of cells $\vec{x}_i$, we pre-compute the directions between any pair of cells $\theta(\vec{x}_i, \vec{x}_j)$, measured in radians from 0 to 2π, ahead of time and store them.

Within the design ensemble, the locations of units $\vec{x}_i$ are random, drawn from the joint distribution encoded in the tensor network. We compute a series of marginal two-units distributions $p(\vec{x}_i, \vec{x}_j)$ for all pairs $i < j$ (see the tensor network in Fig. 6.10f). Since the possible unit locations are discrete, the possible directions $\theta(\vec{x}_i, \vec{x}_j)$ form an artificially irregular discrete set. This numerical artifact would result in a jagged direction distribution $p_{i \to j}(\theta)$. We smooth the distribution by using a version of non-parameteric Kernel Density Estimation (KDE) [198, 199] with periodic boundary conditions in which higher angular harmonics are suppressed:

$$p_{i\to j}(\theta) = \frac{1}{\mathcal{N}} \sum_{k} \sum_{\vec{x}_i,\vec{x}_j} p(\vec{x}_i,\vec{x}_j) e^{-\frac{1}{2}(hk)^2} \cos\left(k(\theta - \theta(\vec{x}_i,\vec{x}_j))\right) \tag{6.10}$$

Here $\mathcal{N}$ is a normalization factor, h is the KDE smoothing factor (bandwidth), $k \in \{0, 1, 2, \dots\}$ is the angular mode index. We find that using smoothing factor of $h = 0.1$ radians and angular modes up to $k_{\mathsf{max}} = 30$ gives good results.

The resulting distributions $p_{i\to j}(\theta)$ need to be compared with the null distribution induced by the Physical Architecture (ship hull shape). We compute the null distribution by evaluating the formula (6.10) for $p(\vec{x}_i, \vec{x}_j) = const$. The identical null distribution is shown in every panel of Fig. 6.6.

6.5.6 Numerical Aspects of Computation

Implementing the tensor networks described above on a computer requires two different kinds of computational work: constructing the networks from Logical and Physical Architecture and possible modifications with the three moves, and contracting said networks numerically. We perform these two tasks in Python.

Using code we developed, we create tensor networks by specifying the network topology, the spatial domain geometry, the design objective, and additional moves. These specifications are done via high-level commands, allowing for the rapid generation of diverse networks.

Tensor network contraction is handled by a Python package. Existing tensor network packages use different methods of executing a sequence of pairwise tensor contractions. The contraction result does not depend on the contraction sequence, but the computational time and memory requirements rise by orders of magnitude for suboptimal sequences. Optimal sequences are known for certain frequently used networks, wherease for others one can use exhaustive enumeration algorithms to find the optimal sequence and then execute it repeatedly for the same network topology.[200] However, almost all networks that we contract in the present work are subtly different, and therefore might require different contraction sequences. We perform all contractions with the `PyTNR` package, an open-source general purpose tensor network contractor.[137, 138] The features of `PyTNR` include using heuristics to automatically generate the contraction sequences on the fly, and performing SVD approximations of controlled precision to reduce the dimension of stored tensors.

The features of `PyTNR` define the computational constraints on the size of systems that our approach can handle. The size and structure of the Logical Architecture directly change the number of units n in the network and the number of tensors n_t (counting both site and coupling tensors). The size or resolution of the Physical Architecture domain directly affect the tensor bond dimension D. A rigorous, though pessimistic upper bound on the time complexity of contraction stands at $\mathcal{O}\left(D^{2\sqrt{\Delta n_t}}\right)$, where Δ is the maximal tensor rank.[131] In contrast, `PyTNR` relies on a heuristic and stochatic generation of contraction sequences, which complicates even the empirical investigations of numerical scaling of complexity. For highly structured networks, such as d-dimensional hypercubic lattices, the time complexity scales a power law $\mathcal{O}(N^\gamma)$, where the exponent γ is a bit larger than the space dimension d and depends on the nature of system's boundary conditions (periodic or closed).[137]

To provide more concrete numbers, each tensor network contraction in this Chapter takes less than 10 seconds on a laptop computer (Intel Core i5-3360M @ 2.8GHz CPU, 8Gb RAM) for our example system ($n = 7$ units, $n_t = 15$ tensors, $D = 78$, total number of combinatorial states $\mathcal{O}(10^{13})$). The example system size was chosen to best illustrate the physical phenomena at single unit resolution. In other investigations we reliably contracted lattice networks of up to $n_t = \mathcal{O}(10^3)$ tensors accounting for $\mathcal{O}(10^{167})$ combinatorial states using `PyTNR`.[80] This result suggests that the current tensor network methods would remain tractable for systems even one order of magnitude larger, or perhaps even larger as the tensor network methods develop.

6.6 Supplementary Results

6.6.1 Avoidance: Excess Density

In the main text of the Chapter we show that void premium quantifies the cost of reserving space within the ship hull, with the effect being further amplified by the presence of an anchor (Figs. 4-5). We associated high void premium with a large rearrangement of the functional units. We can quantify the degree of rearrangement by computing the unit density profile without void $\rho_{\textsf{no void}}(\vec{x})$, the unit density profile with void $\rho_{\textsf{void}}(\vec{x})$, and their difference:

$$\Delta\rho(\vec{x}) = \rho_{\textsf{void}}(\vec{x}) - \rho_{\textsf{no void}}(\vec{x}), \tag{6.11}$$

which we term *excess density* that can be both positive and negative. Since the total number of functional units does not change upon addition of void, the positive and negative regions of excess density have to cancel each other. In this case large rearrangements of the unit cloud are characterized by large *contrast* between the positive and negative regions, graphically visible in saturation of colors.

We plot all three densities in Fig. 6.11, both without and with an anchor on unit 1. In low-cost case $T = 1.0$ (panel a) the units want to form a compact cloud, which can be located anywhere within the hull. Upon creation of a void slightly to the left of center, the unit cloud relocates to the right of the void, as seen by large negative $\Delta\rho$ in the left half of the hull and positive in the right half (visible as

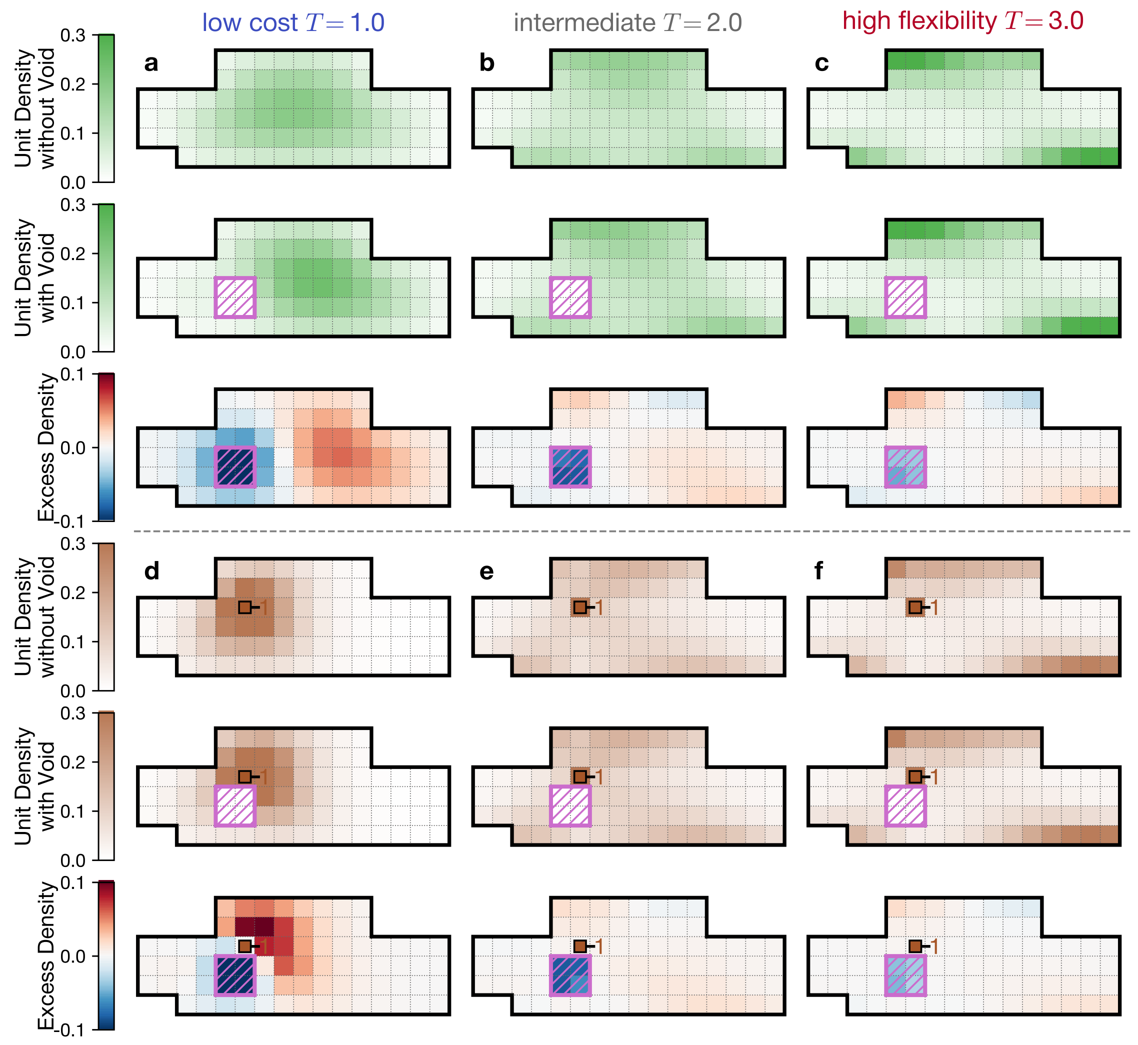


Figure 6.11: Reserving space in locations with large void premium causes large rearrangements of the unit cloud morphology. The rearrangement is shown via the density profile without the void, with the void, and their pointwise difference (rows) for three values of T (columns, color coded). Panels (a-c) show the rearrangement in a fully greenfield scenario (no anchors). Panels (d-f) show the rearrangement in a brownfield scenario (one anchor indicated with a brown square).

red and blue clouds). When unit 1 is anchored (panel d), this effect becomes even more pronounced since the unit cloud condenses around a reference point. Creation of a void pushes the unit cloud to the right and above the anchor, but it cannot move far from the anchor, resulting in high contrast of excess density (strong color saturation in the figure) and thus high void premium. At intermediate and high values of T, both with and without an anchor (panels b,c,e,f) the rearrangements are much smaller, visible in much paler colors on excess density heatmaps.

6.6.2 Adjacency: All Bond Diagrams

In the main text of the Chapter we show how to compute the bond diagram $p_{i\to j}(\theta)$ for any pair of units i, j. Since the directions $\theta(i \to j)$ and $\theta(j \to i)$ only differ by a trivial rotation by angle π, and the direction from a node to itself is not defined, a system of n units would have $n(n-1)/2$ independent bond diagrams. The Logical Architecture of the example problem was deliberately chosen to not have any graph symmetries, therefore the bond diagrams are not related to each other via any symmetries.

While in the main text we only show several representative bond diagrams (Figs. 6-7), Fig. 6.12 shows all of the bond diagrams for the low-cost regime $T = 1.0$ and the high-flexibility regime $T = 3.0$. The diagrams are arranged as a lower-triangular and an upper-triangular matrix of polar plots so that the diagrams in positions symmetric with respect to the diagonal refer to the same pair of functional units and are thus directly comparable. The full set of diagrams shows all adjacency features highlighted in the main text. For units that are directly connected, most of the diagrams at $T = 1.0$ show eightfold signal (e.g. $2\to 6$), while diagrams at $T = 3.0$ show X-shaped fourfold signal (e.g. $4\to 5$). All diagrams to or from unit 1 show further symmetry breaking because unit 1 is anchored in space.

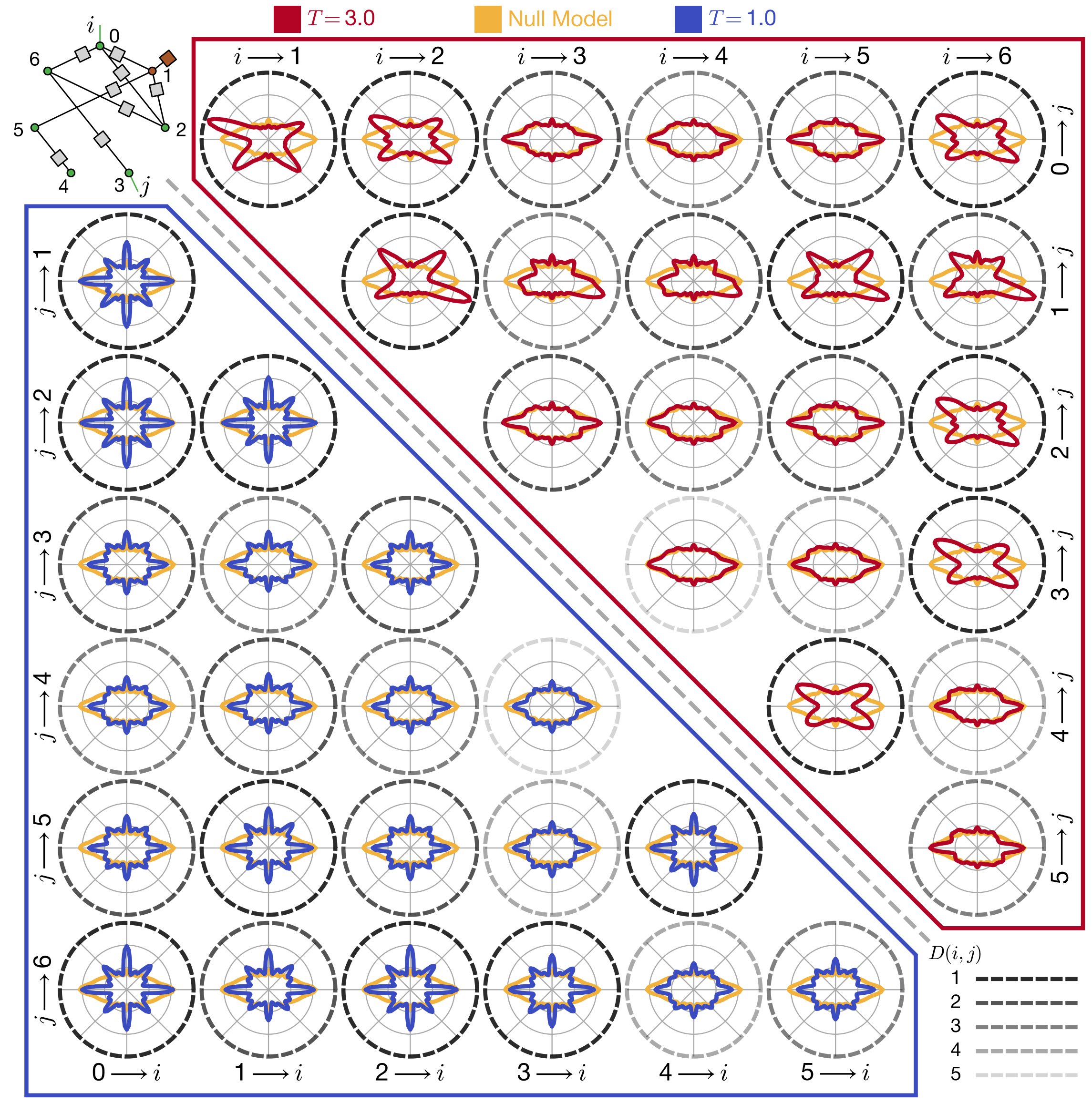


Figure 6.12: Adjacency patterns between the 7 functional units shown via bond diagrams $p_{i\to j}(\theta)$. (top-left corner) Typical tensor network used for bond diagram computations, with a brown anchor for unit 1 and green external legs at units i and j for each origin-destination pair i, j. (bottom-right corner) Legend for the opacity of axes boundaries representing the topological distance $D(i, j)$ varying from 1 to 5 hops. (top-right triangle, red) Bond diagrams for $T = 3.0$. (bottom-left triangle, blue) Bond diagrams for $T = 1.0$. The origin and destination of each bond diagram are indicated on the outside boundaries of the triangle. Axes in positions symmetric with respects to the diagonal refer to the same origin-destination pair and thus can be directly compared. The yellow curve in each axes shows the null model bond diagram $p_0(\theta)$.

For units that are connected only indirectly with large topological distance, the bond diagram approaches the null model at both $T = 1.0$ and $T = 3.0$ (e.g. $3 \rightarrow 4$).

CHAPTER VII

Topological Design in Heterogeneous Self-Assembly

7.1 Introduction

[1]Self-assembly is an incredibly robust, and often unique method of organizing loose building blocks to adopt a certain structure.[51] A wide range of structures have been observed in synthetic self-assembly in both experiments and numerical simulations, from strictly periodic crystals [81] to quasicrystals,[84] liquid crystals,[83] polymers,[201] nets,[78] and finite clusters.[64] While the initial investigations were guided by anecdotal evidence and limited availability of building blocks, later works deliberately discuss optimizing the building block properties for assembly of a desired target structure.[72] The emergent connection between the building block properties and the structures they form makes self-assembly a perfect model system for discussion of more general design principles.

Self-assembling building blocks are typically particles of nanometer or micrometer size, small enough to experience thermal fluctuations but too large for the quantum effects to be significant.[51] Quantum mechanics usually imposes discreteness on the atomic parameters, thus responsible for the discreteness of the chemical elements in the periodic table.[202, 203] The absence of quantum effects then frees the building blocks to have continuously adjustable parameters such as size, shape,

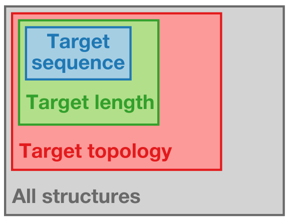
[1]This chapter is based on an upcoming paper coauthored with M.P. Brenner

roughness, or interaction patches.[74, 73] These parameters span combinatorially large design spaces that sometimes can be efficiently searched with inverse design methods.[72, 85]

The design spaces are best characterized for *homogeneous* self-assembly, where all building blocks are identical. On the other side, in *heterogeneous* self-assembly several types of building blocks are used, primarily to allow encoding the target structure in specific interactions. Specific interactions can be realized with several different substrates, the most popular being DNA origami [76] and DNA-functionalized colloidal particles,[75] while newer frameworks focus on *de novo* proteins [77] or magnetic panels.[78]

The shared goal of all these substrates is to make the desired ("on-target") binding much stronger than the undesired ("off-target"), but some amount of off-target binding, also known as cross-talk, is inevitable.[89, 204] Because of cross-talk, the object of heterogeneous self-assembly design is not a single building block, but a set of all building blocks used together. The performance of the whole set of building blocks can be characterized by several metrics,[90, 91] but they have limited capacity of predicting quantitatively the yield of target structures.

The "target structure" in heterogeneous self-assembly can be defined at several levels of detail by aggregating different number of system microstates. The self-assembly target is never a single microstate, since under the effect of thermal noise the system continuously explores microstates that are statistically similar, but distinct. In order to distinguish larger-scale differences, the microstates are aggregated, for example with order parameters that distinguish bulk structures.[107, 65] However, here we are interested instead in finite structures, which we distinguish hierarchically: by topology, by structure length, and by specific monomer sequence a



b

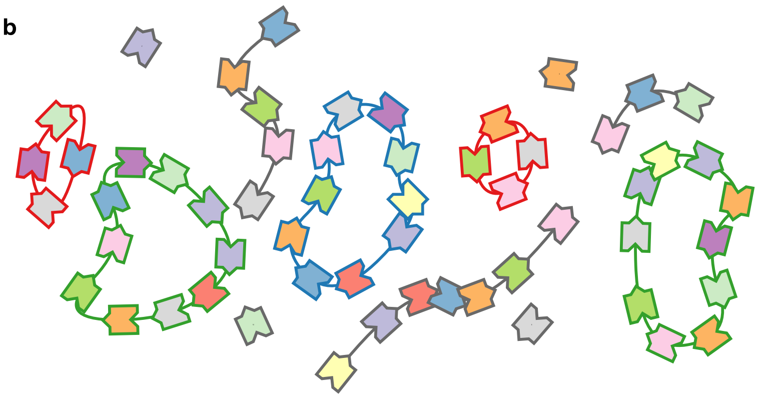


Figure 7.1: The nested hierarchy of detail of self-assembled clusters. (a) Nested Venn diagram of clusters: all clusters of a specific sequence are a subset of all clusters of a specific length, which are a subset of all clusters of a specific topology, which in turn are a subset of all possible structures. (b) Example microstate of a self-assembling system with different target structures identified. The central cluster blue rim is a ring of 10 monomers in a specific non-repeating sequence (blue rim). Two other clusters are rings of 10 monomers in a wrong sequence with repetitions (green rim). Another two clusters are much smaller rings (red rim). All other clusters are either linear chains or free monomers (gray rim).

(see Fig. 7.1). The more narrow is the definition of the target, the more stringent the design of the building blocks becomes.

In this Chapter, I show how to characterize the target structures at different levels of detail and design the set of building blocks that will self-assemble them. The argument consists of three sections. First, I focus on the philosophical questions laid out in Chapter III, lay out the general statistical mechanics framework, and show how the theoretical quantities relate to measurable self-assembly "yield". Second, I address the practical questions and show how to compute the partition functions and yields for two topologies of self-assembled structures: linear chains and looped rings. Third, I show which features of the design space govern the outcome of self-assembly and discuss how cross-talk and entropy limit the precision of self-assembly. In conclusion, I discuss the generalization of the framework of the chapter to wider classes of branching and looped structures.

7.2 Statistical mechanics of self-assembly

This section sets up the statistical mechanics framework in application to fully classical self-assembling particles. I start with the characterization of design space via binding energies, concentrations, and bending rigidity. Then I introduce the statistical weights of clusters Q_s and show how they can be summed into two different, but related types of partition functions. I show how our theoretical control parameter of chemical potential relates to the experimentally measurable control of building block concentrations. Lastly, I discuss the two effects of polymer chain bending entropy: one distributed along the polymer length, and the other penalizing loop formation.

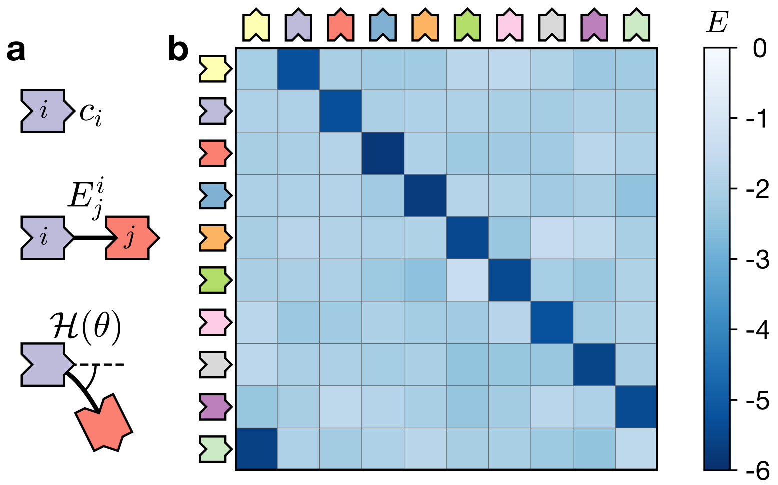


Figure 7.2: The design space of the self-assembly problem. (a) Each particle type i is present at a specific concentration c_i. The key site of particle i binds to the lock side of particle j with energy E_j^i. The bending rigidity of the bond is given by the function $\mathcal{H}(\theta)$, independent of particle types. (b) Example of a binding energy matrix $\mathbf{E}$, with rows corresponding to keys and columns corresponding to locks. The on-target binding energies are very negative (dark blue), while the off-target binding energies are only slightly negative (light blue).

7.2.1 Self-assembly design space

In this chapter we consider a set of N building blocks, jointly designed (see Fig. 7.2a). Each building block has two binding sites on the opposite sides. The binding sites are of two types: "lock" and "key", and binding is only possible between the lock of one particle and the key of another, that is, lock-lock and key-key interactions are negligible. Lock and key interactions have been experimentally realized with several substrate systems,[205, 206, 207] but for the present investigation we abstract out the specific realization.

Regardless of realization, the central description of the interaction is the binding energy. We denote the binding energy between the key side of particle i and the lock side of particle j as $E^i_{\ j}$. As the indices i, j span the range $[1, N]$, the entries form a matrix $\mathbf{E}$ (see Fig. 7.2b). Since all particles exhibit attractive interactions, all entries of this matrix are negative. In the process of *designing* the matrix **E**, we designate certain (i, j) pairs as on-target, and all other ones as off-target. In order for the particles to preferentially bind on-target, the corresponding binding energies need to be stronger (more negative):

$$|E_{\textsf{on-target}}| \gg |E_{\textsf{off-target}}|. \tag{7.1}$$

Ideally, the off-target binding energy should be zero or even positive (repulsive). However, it is still negative for off-target interactions in all existing substrates. Because of this limit in contrast between on- and off-target interactions, we typically cannot control all of the entries of the matrix **E** independently. How much does this limited control over the binding energies allow us to control the self-assembled structures is an open question.

The target structure includes the building blocks in a specific proportion, known as the stoichiometric ratio. However, the concentrations of the building blocks in the system c_i don't need to follow the stoichiometric ratio. Non-stoichiometric, carefully optimized concentrations have been shown to inhibit the assembly of incorrect structures and thus boost the yield of the correct ones.[91] Since for each target structure the optimal ratio is different and we want to explore the space of possible target structures, we treat the concentrations c_i as a free parameter in the design space.

The last important part of the design space is bond entropy. The length and angle of the bond is subject to thermal fluctuations since the bonds between classical particles cannot be frozen out.[63] We divide these fluctuations into two parts: longitudinal and transverse. The longitudinal fluctuations correspond to the vibrations of bond length around the potential energy minimum. These fluctuations can be integrated out and provide a small entropic contribution to the bond (free) energy. For simplicity, we absorb this contribution into the bond energies $E^i_{\ j}$. On the other side, the transverse fluctuations correspond to the fluctuations of the bond angle, or bending of the chain of monomers. We assume that the bending has finite rigidity given by the angle-dependent potential $\mathcal{H}(\theta)$, the rigidity of which we denote with P.

The design space of the heterogeneous self-assembly problem thus consists of the binding energies $\mathbf{E}$, concentrations c_i, and bending rigidity P. The binding energies and bond rigidity are limited by the particle synthesis techniques on a specific substrate, but once the particles are synthesized, the concentrations can varied freely. As we show below, the concentrations are not the most convenient variable to use for theoretical calculations, but the ultimate yield results need to be expressed via concentration to be interpretable.

7.2.2 Partition functions

In order to make predictions of self-assembly of the building blocks into various clusters, we are going to use statistical mechanics. Statistical mechanics is a theory that uses the energy and entropy of different system configurations to make statements about their relative probability. This notion of relative probability will come to be important to interpret seemingly divergent or mathematically nonsensical expressions, and to connect them to the experimental observables or the design space parameters.

Macroscopic parameters that specify a statistical mechanical system typically form thermodynamic conjugate pairs, i.e. only one parameter from a pair can be used as an independent variable for theoretical description. The conjugate pair important here is that of chemical potentials $\vec{\mu}$ and building block concentrations $\underset{\rightarrow}{c}$. Mathematically, we use the vector arrow above $\vec{\mu}$ to indicate that it is a column vector, and the vector arrow below $\underset{\rightarrow}{c}$ to indicate that it is a row vector. Physically, a description in terms of concentrations forms a canonical ensemble, while description in terms of chemical potentials forms a grand canonical ensemble. In the limit of large system size, as considered here, the two ensembles are equivalent, but converting between them is cumbersome. Typically self-assembly experiments directly control the concentrations, but chemical potentials make theoretical calculations easier and we shall use them in the present Chapter.

By choosing the chemical potentials to be the independent variable, we are creating a system description in terms of a grand canonical ensemble. In the grand canonical ensemble, we don't need to track the distribution of a finite number of building blocks between different clusters. Instead, within each cluster, we "create" building blocks by paying their chemical potential price $\vec{\mu}$. Since the total number of building blocks in the system is not fixed, it needs to be evaluated as an observable as discussed later.

The building blocks form distinguishable self-assembled clusters. In principle the clusters can get arbitrarily large, so there is infinite number of cluster types, but since the types are discrete we can enumerate them with an index s. We assume that two clusters of types s and s' can be distinguished from each other if and only if $s \neq s'$. Each cluster is characterized by the integer number of particles of each type in it $\underset{\rightarrow}{n}_s$, the energy of all the bonds E_s, and the internal entropy S_s. The internal entropy includes rotations and vibrations, but not the translations in the assembly volume which we will take care of separately. Together these characterizations of the cluster form it *statistical weight*:

$$Q_s = \exp(\beta\,(\underset{\rightarrow}{n}_s \cdot \vec{\mu} - E_s) + S_s), \tag{7.2}$$

where $(\cdot)$ denotes a dot product between a row vector of particle numbers and a column vector of chemical potentials. The expression (7.2) is very similar to the colloidal partition function discussed and interpreted in Ref. [68] for the canonical ensemble case. In order to get from the individual statistical weights, we need to sum over all possible system states. The standard statistical mechanics textbooks are often vague on how exactly this sum is performed (Kardar [102], Sethna [62], Huang [139, 140], Landau [109]). We show below two variants of this summation and demonstrate that they are closely related to each other but have slightly different interpretations.

The first form of summation is what we call a *cluster partition function*, and is a direct sum of statistical weights of all unique clusters, without taking their copy numbers into account:

$$\mathcal{Z}_{cl} = \sum_s Q_s. \tag{7.3}$$

Note that the cluster partition function is somehow ignorant of the box in which the clusters can move. Accounting for the volume of the box is the same thing as accounting for the translational entropy of the cluster. Taking into account the volume of the box adds a factor of V, while integrating out the kinetic energy of motion of the whole cluster adds a factor of thermal wavelength λ_{th}^{-3}. Note that the thermal wavelength only appears once, regardless of the number of particles in the cluster, since all other factors of thermal wavelength are absorbed in the calculation of vibrational entropy.[63]. We denote the factor of thermal wavelength as the reference concentration $c_0 \equiv \lambda_{th}^{-3}$.[91] The reference concentration is very high and typically unattainable, but serves as a convenient measurement unit.

Apart from the translational entropy, we need to account for the copy number m_s of each type of cluster: while clusters of the same type s are indistinguishable from each other, they can be counted. Each cluster type appears between zero and infinity times within the box, regulated by the corresponding statistical weight. Performing the sum over all cluster types and their copy numbers, we get the *box partition function*:

$$\mathcal{Z}_{box} = \sum_{\{m_s\}} \prod_s \frac{1}{m_s!} \left(\frac{VQ_s}{\lambda_{th}^3} \right)^{m_s} = \prod_s \exp(c_0 V Q_s) = \exp(c_0 V \mathcal{Z}_{cl}). \tag{7.4}$$

The box partition function turns out to be directly related to the cluster partition function. The box partition function is more similar to the textbook definition, but the textbooks typically don't include different types of "super-particles" in the sum. Since in the box partition function the clusters of different types do not interact with each other, they are treated as independent and coexistent ideal gases of "super-particles", or clusters. The statistical weight of each cluster plays the role of fugacity of the corresponding "super-particle". In order to find the relationship between this effective fugacity and the cluster concentration, we can perform the usual derivative:

$$c_s = \frac{1}{V} \frac{\partial \ln \mathcal{Z}_{box}}{\partial \ln Q_s} = c_0 Q_s. \tag{7.5}$$

The reference concentration c_0 is the same for all cluster types, therefore the statistical weight Q_s of every cluster is directly related to the number concentration of the corresponding clusters. A special case of a cluster type s is a free monomer: a cluster i which consists of only one particle of type i and no bonds so that $E_i = 0$ and $S_i = 0$. The statistical weight of such a cluster is then $Q_i = e^{\beta \mu^i} \equiv z_i$, known as fugacity. The concentration of such free monomers is then:

$$c_i^{\text{free}} = c_0 z_i = c_0 e^{\beta \mu^i}. \tag{7.6}$$

Note that the concentration of *free monomers* can be substantially different from the *total concentration* of all monomers, free and bound. We discuss this distinction below. The relationship (7.6) can be interpreted as a simple nonlinear unit conversion between the chemical potential and the free monomer concentration. It is agnostic of what else is happening in the system and is thus not an equation of state as might be implied by some theories.[91]

The simplest example of how the chemical potential could be misleading is the generic divergence of the partition function (7.3). The sum over cluster types s typically has an infinite number of terms with ever increasing particle counts $\underset{\rightarrow}{n}_s$. Frequently this infinite sum will be convergent and give sensible results. However, since the chemical potentials $\vec{\mu}$ are free parameters in our theory, we can always make them arbitrarily high, thus making the later terms in the sum grow and the whole sum diverge. At the same time, the conversion 7.6 from chemical potential μ^i to free monomer concentration c_i^{free} is smooth and analytic, thus it will never diverge.

While the existence of such a divergence is generic for all partition functions with an infinite number of terms, the exact value of μ^{i*} at which it happens, and the behavior of the partition function close to that divergence are model-specific. We analyze those specific behaviors and their physical implications in several models below. However, before we discuss the specifics, we need to make sense of this seemingly unphysical divergence by discussing how it relates to the experimentally controllable and observable quantities.

7.2.3 Experimental control and yields

The apparent unphysicality of the partition function divergence is resolved by carefully considering the experimentally controllable variables. If we treat the chemical potential $\vec{\mu}$, or equivalently, the free monomer density $\underset{\rightarrow}{c}^{\mathsf{free}}$ as a free parameter, we encounter that for certain values the partition function diverges. This means that given the building block properties and interactions, there don't exist any *equilibrium* configurations of the system with that many free monomers. We can certainly prepare a system configuration with an arbitrarily high concentration of free monomers (for example the moment when the building blocks are just placed into the box), but that configuration will be far from equilibrium. As it equilibrates, many monomers will bind up into various structures and become not free anymore.

The quantity conserved throughout the equilibration process is the total monomer concentration of each type $\underset{\rightarrow}{c}^{\mathsf{total}}$, since an experimentalist can put an arbitrary proportion of monomers into the box. In order to compute the total concentration, we need to add up the numbers of monomers bound up in each type of cluster. A cluster type s has the number $\underset{\rightarrow}{n}_s$ of each monomer type, and thus contributes $c_0 Q_s \underset{\rightarrow}{n}_s$ to the total particle concentrations. The number of particles can be pulled out of the exponential in Q_s by taking a derivative with respect to $\vec{\mu}$ (remember that a derivative of a scalar with respect to a column vector is a row vector). Fortunately, the partition function sum has already been performed, so the total concentration is found by a straightforward derivative:

$$\underset{\rightarrow}{c}^{\mathsf{total}} = c_0 \frac{1}{\beta} \frac{\partial \mathcal{Z}_{cl}}{\partial \vec{\mu}}. \tag{7.7}$$

The expression 7.7 is the equation of state for the self-assembling system, since it relates two conjugate thermodynamic variables. If the total concentrations $\underset{\rightarrow}{c}^{\mathsf{total}}$ are known, the chemical potentials $\vec{\mu}$ can be found. The two vectors have the same dimension N, so the expression (7.7) is a map $\mathbb{R}^N \to \mathbb{R}^N$, or a system of N coupled equations. Since the particles of different types can interact with each other (that is the whole point of heterogeneous self-assembly), the equations are not separable and need to be solved jointly.

The partition function $\mathcal{Z}_{cl}$ diverges as $\vec{\mu}$ approaches a critical surface in $\mathbb{R}^N$. The derivative of the partition function then diverges even faster. This fast divergence ensures that any arbitrarily high values of total concentration are reached close to the critical $\vec{\mu}$. Conversely, if in experiment we control $\underset{\rightarrow}{c}^{\mathsf{total}}$ and can make it arbitrarily high, then we can approach the critical $\vec{\mu}$ surface arbitrarily closely but never cross it. If the crossing never happens and we always stay in the region where the partition function is analytic, there will be no observable equilibrium phase transition as the total concentrations are varied.

The divergence occurs because of the growth of the magnitude of terms with large $\underset{\rightarrow}{n}_s$. These terms correspond to large, complex structures of many monomers – but those are precisely the structures we want to assemble! Therefore, in order to observe high yields of the complex structures, we need to be in the near-divergent regime of the parameter space, and understanding the nature of the divergence for a given self-assembly model is incredibly important.

How can we then evaluate the yield of a particular structure? The simplest metric is the *relative yield*, introduced in the Ref. [91]. We can either designate a specific structure s as the target, or aggregate the weights of several structures, for instance all that share the same topology, into a larger target. The relative yield is then given by:

$$Y^{\mathsf{rel}}_{target} = \frac{Q_{target}}{\mathcal{Z}_{cl}}, \tag{7.8}$$

where Q_{target} is either a specific Q_s or sum of all Q_s designated as the target. Physically, the relative yield answers the question: given a randomly-picked cluster, what is the probability that this cluster is of target type? Notably, the picking of clusters is done without taking their size into account, thus equally comparing free monomers with most complex structures.

Whether or not the relative yield is an appropriate metric depends on the design goals. If we interpret self-assembly as a manufacturing technique, we can evaluate it by the amount of target structures it successfully produces, or by the conversion of raw monomers into the target structure. We can compute such metrics by combining the notions of total concentration and the relative yield. We define the *absolute yield* as the concentration of monomers bound up in the target structure:

$$\underset{\rightarrow}{Y}^{\mathsf{abs}}_{target} = \frac{c_0}{\beta}\frac{\partial Q_{target}}{\partial \vec{\mu}}. \tag{7.9}$$

Similarly, we define the *conversion ratio* as the percentage of all monomers that are bound into the target structure:

$$\underset{\rightarrow}{R}_{target} = \underset{\rightarrow}{Y}^{\mathsf{abs}}_{target} / \underset{\rightarrow}{c}^{\mathsf{total}} = \left(\frac{\partial Q_{target}}{\partial \vec{\mu}}\right) / \left(\frac{\partial \mathcal{Z}_{cl}}{\partial \vec{\mu}}\right), \tag{7.10}$$

where the division of vectors is carried out elementwise.

We still need to relate these yield metrics to the experimentally controlled parameters, i.e. the total concentrations $\underset{\rightarrow}{c}^{\mathsf{total}}$. However, solving the equations that relate $\underset{\rightarrow}{c}^{\mathsf{total}}$ to $\vec{\mu}$ is in general hard and becomes even harder and more numerically unstable close to a divergence point, so it is an undesirable strategy. Instead, we will treat the chemical potential $\vec{\mu}$ as an implicit parameter and plot parametric yield curves. It is important to determine the exact point of divergence as precisely as possible and have a fine grid of $\vec{\mu}$ close to it since small variations in the chemical potentials there correspond to great changes of the concentrations. As well, to avoid sampling high-dimensional spaces, we can consider the situations where all of the chemical potentials are the same, or follow a simple pattern parameterized with a single number.

7.2.4 Polymer entropy and loop penalty

Apart from binding energies and chemical potentials, the statistical weights of different clusters depend on their entropy. Entropy is a shorthand for the number of microstates that are identified as the same cluster. While we can tell apart two clusters based on the sequence of their monomers and the presence of specific bonds, we choose to treat two clusters with slightly different bond angles to be identical. However, the number of states that results from such bending depends on the size of the structure and its topology.

All structures considered in this Chapter are one-dimensional sequences of monomers: the key site of one is bound to the lock site of the next. The last monomer in the sequence may or may not be bound to the first one, thus the whole structure looks either like a linear chain, or a ring. Here we discuss the entropy of this chain or ring, while the next section is devoted to accounting for the sequence-dependent energy and chemical potential. To account for the bending entropy, we adopt the worm-like chain model [208] described by the following Hamiltonian:

$$\mathcal{H} = -P\sum_{k=1}^{n-1} \hat{t}_k \cdot \hat{t}_{k+1} = -P\sum_{k=1}^{n-1} \cos(\theta_k), \tag{7.11}$$

where P is the bending rigidity, n is the length of the chain, $\hat{t}_k$ is the unit vector in the direction of each monomer, and θ_k is the angle between two subsequent monomers. The Hamiltonian is minimized when each monomer has the same direction as the previous one, and deviations from this minimum are penalized more for larger P. The wormlike chain model is equivalent to the Heisenberg model of ferromagnetism where the unit vectors are mapped to spins. For a one-dimensional chain with open boundary conditions each of the angles θ_k can be treated as an independent integration variable, thus finding the partition function is straightforward:

$$\mathcal{Z}^{(n)}_{\mathsf{wl\ chain}} = \int \left(\prod_{k=1}^{n-1} d\Omega_k \right) e^{-\beta\mathcal{H}} = \left(4\pi \frac{\sinh(\beta P)}{\beta P} \right)^{n-1} = e^{S_b \cdot (n-1)}, \tag{7.12}$$

where $d\Omega_k$ is integration over a solid angle and S_b is the bending entropy of each of the $n-1$ bonds. From the partition function we can easily extract the correlation in direction of the two adjacent monomer directions:

$$\langle \cos(\theta) \rangle_2 = -\frac{1}{\beta} \frac{\partial \ln \mathcal{Z}^{(2)}_{wl}}{\partial P} = \left[\coth(\beta P) - \frac{1}{\beta P} \right] \equiv c(\beta P) \tag{7.13}$$

$$\langle \cos(\theta) \rangle_n = c^{n-1} = e^{-(n-1)/\xi}, \tag{7.14}$$

where $\xi = -1/\ln c$ is the *persistence length* of the chain, measured in monomers. Note that c is an analytic function of bending rigidity as the singular parts of the two terms exactly cancel out. The large-scale statistics of the polymer chain with finite bending rigidity are those of a *persistent random walk*. A persistent walk of n steps of length a is statistically similar to a independent random walk of $n' = n/\xi$ steps of length $a' = a\xi$. Specifically, the probability distribution for the displacement between the start and the end of the random walk is approximately Gaussian:[208]

$$p(\vec{r}) = \frac{1}{(2\pi n' a'^2)^{3/2}} \exp\left(-\frac{3\vec{r}^2}{2n'a'^2} \right). \tag{7.15}$$

A loop is a random walk on n steps that ends approximately in the same place where it starts (so $\vec{r} = 0$), up to the size of a monomer (a volume of a^3). Note that a ring of n monomers has n bonds as opposed to $n-1$ in an open chain. The partition function of the ring is then:

$$\mathcal{Z}_{\mathsf{wl\ ring}} \approx e^{S_b n} a^3 p(\vec{r} = 0) = \frac{e^{S_b n}}{(2\pi \xi n)^{3/2}}. \tag{7.16}$$

Compared to the linear chains, the loop partition function has an additional power law decay factor with exponent $-3/2$. The decay is driven by the fact that it gets harder and harder for a long random walk to encounter its own beginning. For this reason, long loops are always entropically suppressed compared to long chains. However, loops have one more bond that can be energetically favorable. We will see that this entropy-energy tradeoff is one of the main design limitations of the self-assembly system.

Note that the expression (7.16) only holds for loops long enough to be able to bend on themselves, that is $n > \xi$. For very short loops, the bending energy will be prohibitive for the loop to form at all. The simplest way to deal with such short loops is to manually exclude them from the computation of the full self-assembly partition function, as we will do in the next section.

One way to reduce the entropy penalty is to use building blocks with binding sites that are not strictly at the opposite sides of the block, but at a finite angle. The lowest bending energy configuration will then correspond to a bond bend by that specific angle, thus directly building curvature into the assembling structure and favoring rings over chains. However, we will not consider this case in the following calculation in order to highlight other, more fundamental design tradeoffs.

7.3 Chain sequence combinatorics

This section details the computation of the cluster partition function (7.3) for our specific model system. We first introduce the hierarchical sum of the partition function over the topologies, lengths, and sequences. We then define the transfer matrix as the central element of the combinatorial calculation. We derive closed-form partition functions for linear chains and rings and indicate their convergence criteria. We show how the convergence and other properties of the partition function relate to the spectral properties of the transfer matrix. As we approach the divergence point, the chain and ring partition functions diverge at different rates, thus informing the entropic trade-off between them. We conclude the section with deriving the total concentrations and absolute yields of building blocks in closed form.

7.3.1 Hierarchical structures

In order to start computing the partition functions, we need to look back at the hierarchical taxonomy of structures that can be assembled (Fig. 7.1). The structure index s so far did not refer to this hierarchy and was treated as one-dimensional. Without invalidating any of the summations in s, we can use it to refer to the topology, the length, and the specific sequence of the structure. In this case we can write the cluster partition function as a triple sum:

$$\mathcal{Z}_{cl} = \sum_{\text{topology}} \sum_{\text{length}} \sum_{\text{sequence}} Q_s. \tag{7.17}$$

The outermost sum is over the possible topologies. As we stated before, the monomers with two binding sites can only assemble two different topologies: chains and rings. For these topologies, we will write down two different expressions and add them up. The sum in length is relatively straightforward since the structures are one-dimensional, so the possible lengths are enumerated by a single set of natural numbers with some starting point. The sum in possible sequences is more tricky. However, since the energies of bonds and chemical potentials of the monomers are additive, the corresponding statistical weights are multiplicative and can be expressed via matrix products by following the approach of Ref. [91]. The following sections derive the partition function expressions from the bottom up and develop the tools for their analysis.

7.3.2 The transfer matrix

The central mathematical object of our theory is the *transfer matrix* $\mathbf{T}$ that can be defined elementwise as following:

$$T^i_j = \exp\bigl(\beta(-E^i_j + \mu^j) + S_b\bigr). \tag{7.18}$$

The transfer matrix corresponds to adding one more element to the chain. The first index i accounts for the possible key binding sites exposed by the preceding part of the chain. The second index j accounts for the possible monomers and their corresponding locks that the key can bind with. Note that in constructing this matrix we explicitly broke the symmetry by the convention of adding the new monomer on the right (in the index j). We could place the chemical potential either on the left or on the right, so long as in a long chain chemical potentials alternate with binding energies, with special care of what happens at the ends of the chain. The bending entropy depends on the bending rigidity of the chain, but not on the identities of the monomers and is thus a constant multiplicative factor in the transfer matrix.

This definition of the transfer matrix closely follows the version used in solving spin lattice models,[107] with suitable modifications to account for the grand canonical ensemble. It is also very similar to the coupling tensors in Chapter VI and Ref. [80]. While the coupling tensors only accounted for two-point interactions, here we also absorb the one-point interaction (chemical potential) into the same object.

7.3.3 Partition functions

A linear chain consists of three components: the initial monomer driven only by the chemical potential, some number of subsequent monomers described by the transfer matrix, and the interaction of the last monomer with the solvent. The selection of the initial monomer is given by the vector of fugacities $\underset{\rightarrow}{z} : z_i \equiv e^{\beta\mu^i}$. The number of intervening monomers varies between zero and infinity and is accounted by the sum in length. The terminus is taken as $\vec{1}$, a vector of all 1's since we assume that monomers don't have any specific interactions with the solvent. In this case, the partition function of all possible chains is given by:

$$\mathcal{Z}_{chain} = \sum_{n=0}^{\infty} \underset{\rightarrow}{z}\mathbf{T}^n\vec{1}. \tag{7.19}$$

Note that for each term n, the chain has n bonds that contribute energy and entropy and $n+1$ monomers that contribute the chemical potential. The contraction of the transfer matrix with the fugacities on the left and the terminus on the right can be taken outside of the summation due to linearity. The internal sum is then a simple geometric series in the matrix $\mathbf{T}$ that we will call the *propagator* $\mathbf{D}$:

$$\mathbf{D} \equiv \sum_{n=0}^{\infty} \mathbf{T}^n = \left(\mathbf{I} - \mathbf{T}\right)^{-1}, \tag{7.20}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix. The geometric series does not always converge, and consequently the inverse does not always exist. This will have important physical implications below. For now, we can write down the chain partition function in a much more concise form:

$$\mathcal{Z}_{chain} = \underset{\rightarrow}{z}\mathbf{D}\vec{1}. \tag{7.21}$$

By analogy with the chain, we can derive the ring partition function. The key difference between the chain and the ring is that there is no initial monomer and no terminus. Instead, the last monomer in the chain is bound to the first one. In order to account for this, we take the trace of the appropriate power of the transfer matrix. While the bending entropy of the bonds is already accounted for in the transfer matrix, the ring entropy penalty needs to be added manually. We also manually exclude the very short loops since they need excessively high bending. The resulting ring partition function is:

(7.22)
$$\mathcal{Z}_{ring} = \sum_{n=n_{min}}^{\infty} \frac{1}{(2\pi\xi n)^{3/2}} \operatorname{Tr}(\mathbf{T}^n) = \frac{1}{(2\pi\xi)^{3/2}} \left(\operatorname{Tr}\big(\operatorname{Li}_{3/2}(\mathbf{T})\big) - \sum_{n=1}^{n_{min}-1} \frac{\operatorname{Tr}(\mathbf{T}^n)}{n^{3/2}} \right),$$

where Li_q is the polylogarithm function of order q (here we use $q = 3/2$).[209] The polylogarithm is usually defined for a scalar argument, but we will discuss a suitable generalization to matrix arguments below. Since the polylogarithm is defined via a sum starting from $n = 1$, we need to manually subtract a finite number of terms of the sum. The polylogarithm remains the leading term that will govern the divergence properties of the ring partition function, described below.

The total partition function of all possible structures in this system is the sum over the two possible topologies:

(7.23)
$$\mathcal{Z}_{cl} = \mathcal{Z}_{chain} + \mathcal{Z}_{ring}.$$

This cluster partition function is the basis for computing the total monomer density and absolute yields of target structures. However, to predict those quantities it is crucially important to understand the properties of the divergence of both the geometric series and the polylogarithm. Those divergences, in turn, are directly related to the spectral properties of the transfer matrix, as discussed below.

7.3.4 Spectral analysis

Both the propagator and the polylogarithm are defined as sums of series in powers of the transfer matrix. In order to clarify the conditions of convergence of this series, we can analyze it in terms of the spectrum of the transfer matrix. The eigendecomposition of the transfer matrix can be computed as following:

$$\mathbf{T} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}, \tag{7.24}$$

where $\mathbf{P}$ is a transformation matrix of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues.[2] The eigenvectors and eigenvalues together form the spectrum of the matrix, which for small matrices can be easily computed numerically once and looked up after that. The properties of $\mathbf{T}$ impose some constraints on its spectrum. If none of the binding energies are positive-infinite, the matrix consists of positive elements only, which by Perron-Frobenius theorem guarantees that the top eigenvalue λ_0 is real, positive, and an all-positive eigenvector can be found for it. However, since the transfer matrix $\mathbf{T}$ depends on the binding energy matrix $\mathbf{E}$, which is not symmetric, there is no guarantee that the rest of the spectrum is real, only that it lies within a complex circle of radius λ_0.

In terms of the spectrum, the propagator and the polylogarithm can be computed as following:

$$\mathbf{D} = \mathbf{P}\left(\mathbf{I} - \mathbf{\Lambda}\right)^{-1}\mathbf{P}^{-1} \tag{7.25}$$

$$\mathrm{Li}(\mathbf{T}) = \mathbf{P}\,\mathrm{Li}(\mathbf{\Lambda})\mathbf{P}^{-1}, \tag{7.26}$$

where the inverse and the polylogarithm of the diagonal matrix can be computed elementwise. The convergence of both of them is then dependent on the top eigenvalue. Importantly, the convergence condition $\lambda_0 < 1$ is the same for both functions. Since λ_0 depends on the components of the transfer matrix, then as these components vary, both the chain and the ring partition functions will diverge at exactly the same point, though at different rates which we analyze below.

[2]Usually the matrix $\mathbf{P}$ will be unitary, since the eigenvectors are typically orthogonal to each other for most matrices that occur in physics. However, in the present case we are dealing with a non-symmetric matrix $\mathbf{T}$. While the spectrum can be found even for a non-symmetric matrix, the orthogonality of eigenvectors is not guaranteed. Because of that, the inverse of the transform matrix does not necessarily coincide with the conjugate transpose $\mathbf{P}^{-1} \neq \mathbf{P}^{*}$.

While the top eigenvalue is sufficient to determine whether or not the sum over all statistical weights diverges, the structure of that sum also depends on all other eigenvalues. A simple way to see that is to examine the sequence of weights of the rings of valuable size. The weights are proportional to the traces of powers of the transfer matrix, which in turn are related to sums over eigenvalues:

$$Q_n \propto \mathrm{Tr}(\mathbf{T}^n) = \sum_{k=0}^{N-1} \lambda_k^n, \tag{7.27}$$

where N is the number of monomers, equal to the dimension of the matrix $\mathbf{T}$ and thus the number of eigenvalues. Even though the eigenvalues themselves can be complex, the sum over them is always real, resulting in a real Q_n. However, for certain values of ring length n all of the eigenvalues can add up in phase and give a boost to the statistical weight. Such a boost will be an evidence of the successful programming of a target structure into the energy matrix, as shown in Supplementary Results section below. A similar spectral analysis can be carried out for the assembly of linear chains.

In this way, the spectrum of the transfer matrix serves simultaneously two purposes: while the lead eigenvalue governs the overall divergence of the partition function, all other eigenvalues determine whether specific target structures are robustly formed close to that divergence. Therefore, the spectrum is a *collective metric* that quantifies the properties of a given set of building blocks. In order to design an optimal set of building blocks for self-assembly, we can state the desired spectral properties of the transfer matrix and then attempt to pick the blocks that can realize those properties.

7.3.5 Divergence analysis

The divergence properties of the partition function contain important knowledge about the design limitations. In order to elucidate these limitations, we need to look at the asymptotic behavior of the partition functions. Generically, as we approach the divergence in the space of chemical potentials, the lead eigenvalue of the transfer matrix will have the following form:

$$\lambda_0 \simeq \exp(\mu - \mu^*), \tag{7.28}$$

where the divergence point μ^* depends in a complicated way on the properties of the binding energies. However, the divergence point is the same for the linear chains and the rings, thus we can compare the asymptotic forms of the two partition functions. For the chain partition function, the asymptotic form is quite simple:

$$\mathcal{Z}_{chain} = \underset{\rightarrow}{z}(\mathbf{I} - \mathbf{T})^{-1}\vec{1} \propto A(1-\lambda_0)^{-1} \propto A\left(\mu^* - \mu\right)^{-1}, \tag{7.29}$$

where A is a non-singular positive constant.

For the ring partition function, the analysis is slightly more complicated and has to do with the properties of the polylogarithm function. As the argument $x \equiv e^w$ of the polylogarithm $\mathrm{Li}_q(x)$ approaches $x = 1$, the singular term can be separated from the analytic term:[209]

$$\mathrm{Li}_q(e^w) = \Gamma(1-q)(-w)^{q-1} + \mathcal{O}\big(w^0\big), \tag{7.30}$$

where $\mathcal{O}(w^0)$ is a series in non-negative powers of w that does not affect the singularity. In our case $q = 3/2$, so the singular part is $(-w)^{1/2}$. Note that the Gamma function $\Gamma(-1/2)$ is negative, so that the singular part of the ring partition function is negative. We can therefore write it down as following:

$$\mathcal{Z}_{ring} \propto -B(\mu^* - \mu)^{1/2} + C, \tag{7.31}$$

where B, C are non-singular positive constants. Note that as $\mu \to \mu^*$, the partition function itself does not approach infinite value, even though it is not defined for $\mu > \mu^*$.

However, the interesting part of the divergence is the behavior of total concentrations, i.e. the derivative of the cluster partition function $\mathcal{Z}_{cl} = \mathcal{Z}_{chain} + \mathcal{Z}_{ring}$. The derivative shifts the divergence exponents by 1, giving us:

$$c^{\mathsf{total}} \propto \frac{\partial \mathcal{Z}_{cl}}{\partial \mu} = \underbrace{A(\mu^* - \mu)^{-2}}_{\text{chain}} + \underbrace{B(\mu^* - \mu)^{-1/2}}_{\text{ring}}. \tag{7.32}$$

Each of the two divergent terms corresponds to the number of building blocks bound up in either chains or rings. We are interested in which of these terms if larger. At any finite density, $(\mu^* - \mu) > 0$, and the relative balance of the two terms depends on the values of A, B, which in turn depend on all of the design space variables in a complicated way. Figuring out this complicated dependence is the goal of design. A system well designed for assembly of chains will have the term with A dominate, whereas a system well designed for assembly of rings will have the term with B dominate. However, the terms still diverge at different rates. Regardless of our design efforts, for very high concentrations c^{total} we will get arbitrarily close to μ^* and the assembly of chains will always dominate.

It is important to note that these very high concentrations c^{total} are not always possible, since the building blocks will either reach close-packing density, or different clusters will be so close to each other that ignoring their interactions will be impossible. In either case, the cluster-based theory is likely to break down in that limit.

There is two more qualitative features that can be extracted from the analysis of this divergence. First, close to the divergence point Eqn. (7.32) becomes a simplified form of an equation of state that connects chemical potential to the concentration. For example, for a system dominated by chains, we can easily solve for the chemical potential in terms of the experimentally controllable concentration:

$$\mu \simeq \mu^* - \left(\frac{c^{\mathsf{total}}}{A}\right)^{-1/2}. \tag{7.33}$$

This approximate value of chemical potential can then be plugged into any other μ-dependent expressions in the grand canonical theory. For example, the lead eigenvalue of the transfer matrix is then $\lambda_0 \simeq 1 - (c^{\mathsf{total}}/A)^{-1/2}$, asymptotically approaching 1 as expected.

The second qualitative feature is the asymptotic behavior of the conversion ratio at large concentrations. The partition function diverges because it is a sum over an *infinite number* of statistical weights Q_s. Each statistical weight is an analytical function of chemical potential μ, thus it does not diverge at μ^*. Similarly, if the target structure aggregates a *finite* number of structures s, its statistical weight does not diverge either. Therefore, the asymptotic behavior of the conversion ratio is governed exclusively by the divergence of the total concentration:

$$R_{target} = \frac{1}{\beta c^{\mathsf{total}}} \frac{\partial Q_{target}}{\partial \mu} \propto \frac{1}{c^{\mathsf{total}}}. \tag{7.34}$$

In other words, optimal conversion of raw monomers into target structures will always be achieved at finite building block concentration. Arbitrarily large building block concentrations favor formation of larger and larger structures, thus they will always suppress the formation of any finite target structure.

7.3.6 Computing yields

In order to complete the theory, we need to predict the absolute yields and total concentrations of structures of interest. Above we showed the general formula for total concentration in vector derivative format. In order to perform that derivative analytically, it will be more convenient to rewrite the expression in index notation:

$$c_l^{\mathsf{total}} = c_0 \frac{1}{\beta} \frac{\partial \mathcal{Z}_{cl}}{\partial \mu^l}, \tag{7.35}$$

where we highlight that a derivative of a scalar ($\mathcal{Z}_{cl}$) with respect to a column vector (μ^l) is a row vector (c_l^{total}). Similarly, the vector derivative of any higher-rank object adds an extra index to the result. For an example, an important building block of our theory is the transfer matrix, the derivative of which is:

$$\frac{1}{\beta} \frac{\partial T^i_{\ j}}{\partial \mu^l} = T^i_{\ j} \delta_{jl}, \tag{7.36}$$

where there is no summation in repeated index j. Since each index might be repeated more than twice and not always summed over, throughout this section we do not assume the Einstein convention and instead write the summations out explicitly.

Through the definition of the propagator $D^i_{\ j}$ we can also get the derivative of the propagator:

$$\frac{1}{\beta} \frac{\partial D^m_{\ k}}{\partial \mu^l} = \sum_{i,j} D^m_{\ i} T^i_{\ j} \delta_{jl} D^j_{\ k} = \left(\sum_i D^m_{\ i} T^i_{\ l} \right) D^l_{\ k}. \tag{7.37}$$

The propagator derivative allows us to write down the derivative of the chain partition function:

$$c_l^{chain} = \frac{c_0}{\beta} \frac{\partial \mathcal{Z}_{chain}}{\partial \mu^l} = \frac{c_0}{\beta} \frac{\partial}{\partial \mu^l} \left(\sum_{i,j} z_i D^i_{\ j} 1^j \right) \tag{7.38}$$

$$= \frac{c_0}{\beta} \left[\left(\sum_j z_l D^l_{\ j} 1^j \right) + \left(\sum_{i,j} z_i D^i_{\ j} T^j_{\ l} \right) \left(\sum_j D^l_{\ j} 1^j \right) \right]. \tag{7.39}$$

While the expression above appears cumbersome, it amounts to several lookups and multiplications of matrices $\mathbf{T}, \mathbf{D}$ that are already known. In a similar way, the main component of the ring partition function is the trace of a power of the transfer matrix $\mathrm{Tr}(\mathbf{T}^n)$. Its derivative can be shown to be:

$$\frac{1}{\beta}\frac{\partial}{\partial \mu^l}\,\mathrm{Tr}(\mathbf{T}^n) = n(\mathbf{T}^n)^l{}_l, \tag{7.40}$$

so that the derivative is read off the diagonal of a matrix power (a trace will have summed up that diagonal). This expression, up to a prefactor, gives the concentration of building blocks bound up in rings of a specific size n. By summing such expressions, we can express the total concentration of building blocks bound up in rings of all sizes:

$$c_l^{ring} = \frac{c_0}{\beta}\frac{\partial \mathcal{Z}_{ring}}{\partial \mu^l} = \frac{c_0}{\beta(2\pi\xi)^{3/2}}\left[\left(\mathrm{Li}_{1/2}(\mathbf{T})\right)^l{}_l - \sum_{n=1}^{n_{min}-1} n^{-1/2}(\mathbf{T}^n)^l{}_l\right]. \tag{7.41}$$

Note that the order of the polylogarithm and the power of n in the second sum both shifted by 1. The polylogarithm of any order diverges at the same value of the argument, and thus the polylogarithm of a matrix can be evaluated via the same spectral method.

It is important to note that the equations (7.39) and (7.41) are valid for any transfer matrix regardless of the symmetries or design decisions built in. We are still at liberty of assigning the binding energies $E^i{}_j$, the chemical potentials μ^l, and the bending rigidity P. The total concentration and related yield expressions then directly connect the points in design space (encoded in $\mathbf{T}$) with the self-assembly outcomes (encoded in c_l^{chain} and c_l^{ring}). Having obtained these fully analytic expressions, we can explore them numerically and give practical recommendations for design of particular structures.

7.4 Design for topology

There are three qualitative outcomes possible in the system we are considering here: target rings, target finite chains, or unstructured chains. The goals of self-assembly design are to pick the appropriate points in the design space that will drive the system into the regime that is desired.

In order to reliably assemble rings, the binding energy matrix needs to be circular, i.e. each building block should bind on-target to the next one, and the last one should bind to the first one. In order for rings of target size N to be preferred over other sizes, especially $N \pm 1$, the contrast between the on- and off-target binding energies needs to be sufficiently strong (see an analysis in Supplementary Results). If the target ring size is large, there is an additional suppression from the loop entropy which will require even higher contrast to ensure $\mathcal{Z}_{ring} \gg \mathcal{Z}_{chain}$.

In order to reliably assemble finite chains of target size, the last building block should only bind weakly to the first one to ensure that the chain reaches the desired size and actually terminates. There are two effects that compete with this. On one side, shorter chains are entropically preferable: while there is only one chain of N elements with $N - 1$ on-target bonds, there are two chains of $N - 1$ elements with $N - 2$ on-target bonds. To favor formation of complete chains, the on-target binding has to be strong enough. On the other side, if another off-target building block binds to the end of a chain of on-target bonds, it can effectively restart the chain. For strong enough on-target bonds, it is possible to make a chain of $2N$ blocks with $2N - 2$ on-target bonds and only 1 off-target bond. The chief way to avoid this is to make the off-target interaction as low as possible, perhaps even repulsive, but that might be limited by the substrate.

The third outcome, unstructured chains, dominates whenever the first two fail. If the binding energy contrast is too weak to provide sufficient specificity of binding and overcome the entropy, all building blocks will behave as effectively homogeneous.[91] The assembly of homogeneous polymer chains is well-studied,[201] and results in an exponential distribution of chain length. The exponential distribution is featureless in that it does not have any defined peaks at target length.

Overall, in order to assemble large complex structures, we need to encode specific interactions with sufficiently high contrast, which will allow for a non-monotonic sequence of statistical weights Q_s. In the language of grand canonical ensemble, we will have to tune the system close to the divergence point in order to see the large structures. Detailed numerical investigation still needs to be undertaken to formulate the quantitative boundaries and trade-offs between different regimes.

7.5 Discussion

In this section we tally up the progress in pursuit of statistical mechanics questions stated back in Chapter III and discuss how the proposed framework can be extended to account for more complex structures.

7.5.1 Conceptual questions

This Chapter builds a version of self-assembly theory grounded entirely in fully classical modeling choices by synthesizing several previous conceptual discussions. In the following summary of results the conceptual questions of Chapter III are referred to by their number in brackets.

The first modeling choice is about distinguishing individual building blocks and clusters (1). Regardless of manufacturing details, we choose the building blocks to be distinguished only if they are of different type, and similarly for assembled clusters. In describing the particles, we do not refer to any quantum mechanical factors explicitly (2). Implicitly, the statistical weights are dependent on the thermal de Broglie wavelength λ_{th}, which in turn depends on the phase measure h. Together, these factors determine the reference monomer concentration c_0, but its exact value remains a modeling choice. This choice is directly connected to the classical bond entropy, which we implicitly absorb into the bond (free) energy (3).

Treating the bonds in the sticky limit is our implicit choice of the integral, or continuous part of the theory (4). Instead, in this Chapter we focus on accounting for the combinatorics of discrete arrangements of building blocks into clusters. We discuss in some detail the high-concentration behavior of this self-assembly theory, even though real systems have a hard upper bound on building block density as they reach close packing (5). We build up the theory initially in the grand canonical formalism (via chemical potential) but connect it to the canonical formalism (via total concentrations) to be more directly comparable to experiment and simulation (6-7). In the process we discover the chemical potential to be essentially an unmeasurable theoretical construct that nevertheless makes a lot of mathematics tractable both analytically and numerically (8). We discover divergences in the theory as chemical potential is smoothly varied, but these divergences cannot be crossed as total concentration is varied in equilibrium. The divergence exponents, however, govern the relative abundances of different structures.

Lastly, we discuss and compare several notions of self-assembly yield (9). We define the design space for heterogeneous self-assembly via the binding energy matrix, bond bending rigidities, and building block concentrations, even though the latter are only controlled indirectly via fugacities (10). The derived connection between the design space and the outcome concentrations of self-assembled clusters is the knowledge structure that will empower the self-assembly design process.

7.5.2 Implementation questions

For a specific family of building blocks we provide specific derivations and analytical predictions that address the second set of questions in Chapter III, focused on implementation.

We account for the vibrational and rotational entropies by using the worm-like chain model from polymer science (1). We derive the partition functions for two types of structures, linear chains and closed loops, via the transfer matrix method and series summations (3). The convergence criteria of the partition function are shown via the spectral properties of the transfer matrix. This makes the matrix spectrum our desired collective metric to design whole sets of heterogeneous building blocks (2). The matrix spectrum also accounts for both the energetic and entropic aspects of cross-talk and allows us to derive the bounds on the size of structures that can be reliably assembled with a given substrate (6). For linear structures, steric interactions of the building blocks are not a significant hindrance, but we manually exclude from the sums the rings that are too short.

7.5.3 Generalizations and outlook

The conceptual and mathematical framework built in this Chapter can be further generalized to account for more diverse structures. The simplest extension is to allow for building blocks with more than two binding sites on them. If such a building block bonds with an existing linear chain, it provides a way for this chain to branch off in two or more directions, thus forming a tree-like structure. So long as the linear segments of the tree are long compared to the size of the branching points, such structures can largely avoid steric interaction issues.

Certain families of tree-like structures can be directly enumerated and summed via special functions, much like we did with the polylogarithm function for the rings here. For more general forms of tree-like networks, we can construct self-consistent expressions similar to those used for study of random networks [210] and hyperbranched polymers.[211] Similar methods can help us account for formation of local loops within the structures.[212, 213] Another avenue to control of self-assembled structures is the usage of allosteric building blocks for which the different binding sites are not independent.[214, 215]

7.6 Supplementary results

7.6.1 Quantifying the in-phase boost

The boost to the statistical weight of target structure can be detected numerically by diagonalizing any given transfer matrix, but we can get a qualitative insight by looking at an especially simple transfer matrix. To illustrate this boost, consider a binding energy matrix that consists of two values: the on-target energy v and the off-target binding energy ϵ such that $v < \epsilon < 0$. Each monomer's key site binds on target with the next monomer's lock site, with the last monomer's key binding to the first. In other words, the interaction energy is the following:

$$\begin{cases} E^i_j = v, & j = i+1 \\ E^i_j = \epsilon, & j \neq i+1 \end{cases}, \tag{7.42}$$

where the equality of indices is taken modulo N. We also assume that all chemical potentials are the same $\mu^i = \mu$ and there is no bending entropy $S_b = 0$. Such a matrix attempts to optimize the assembly of a ring. The same symmetry structure propagates to the transfer matrix, which then can be analytically diagonalized with a Fourier ansatz. The resulting eigenvalues take the following form:

$$\lambda_k = \lambda e^{i2\pi \frac{k}{N}} + \Delta \delta_{k,0} \tag{7.43}$$

$$\lambda = e^{\beta\mu}\left(e^{-\beta v} - e^{-\beta\epsilon}\right) \tag{7.44}$$

$$\Delta = e^{\beta\mu} n e^{-\beta\epsilon}. \tag{7.45}$$

To ensure convergence, we assume that $\lambda + \Delta < 1$. The trace of a matrix power then takes the following form:

$$\mathrm{Tr}(\mathbf{T}^n) \approx (\lambda + \Delta)^n + \lambda^n n\delta(n, Nr), \tag{7.46}$$

where r is an integer counting how many times the full sequence has repeated. In other words, the statistical weight gets a boost every N elements. Is this boost significant? In other words, does it significantly affect the structures assembled? This boost will be significant if the second term in the sum is much larger than the first one for the weight of the target structure $n = N$:

$$N\lambda^N \gg (\lambda + \Delta)^N \tag{7.47}$$

$$\ln N > N \ln(1 + \Delta/\lambda) \tag{7.48}$$

$$\frac{\ln N}{N} > N \frac{e^{-\beta\epsilon}}{e^{-\beta v}} \tag{7.49}$$

$$\frac{N^2}{\ln N} < e^{-\beta(v-\epsilon)} \tag{7.50}$$

Turns out, requiring a substantial boost of statistical weight sets up a limit on the largest structure that can be reliably assembled given the binding energy *contrast* $(v - \epsilon)$. The contrast is typically limited by the properties of the substrate that realizes the specific interactions. The derived limit is remarkably similar to the "$\log n$ limit" shown in Ref. [91].

CHAPTER VIII

Conclusions and Outlook

8.1 Design as Knowledge Generation

In order to recap the conclusions of the dissertation and evaluate our progress, we need to first return to the statement of our goals. Back in Chapter I we defined design as "the act of generating knowledge for decision-making through time". With this definition, the implied goal of a study of design is to figure out how to systematically generate and store knowledge and build upon the existing knowledge. Important advances on these knowledge structures were made by the previous network-based studies,[7, 38] but they focused on documenting a single realization of the design process. In contrast to those works, this dissertation is interested in the space of available design solutions and the collective physical phenomena in that design space. How did we fare in pursuing this interest?

In Chapter II I formulated the Systems Physics approach to wicked problems as a three-stage loop Model–Compute–Learn. Each of these stages involves making qualitative choices. In order to model the original wicked problem as a tame problem, we need to define the solution space and parameterize the design objectives and pressures. On this tame problem, we can perform a variety of calculations using statistical mechanics as a toolbox. The results of those calculations become pieces of knowledge about *what* and *why* questions that we can communicate back to the wicked problem and refine our analysis on the next run of the loop. To conclude the dissertation, I am going to address all stages of the loop but out of order.

First, I highlight the last stage of the loop – Learn – since it serves as the punchline for the kinds of knowledge attainable via statistical mechanics and not otherwise. The statistical mechanics analysis of the models in Chapters IV–VII shows that the design problems can show new physical phenomena or uncommon variations of known phenomena. Second, I discuss the innovations of the Compute stage in form of analytical and numerical tools. Third, I reflect on the Model stage and discuss how exactly we make qualitative choices that turn the wicked problem into a tame one.

Lastly, there are plenty of design questions opened up by this dissertation. Some of these questions I have already formulated in statistical mechanics terms, for some I have preliminary results that are too raw to be included here in full, but many of these questions are of great interest to the design community. Thus, I end the dissertation with an outlook of where we can go further.

8.2 Learn: New Phenomena Discovered

The results from model computations in Chapters IV–VII don't only provide knowledge about those specific problems, but show off new physical phenomena that could be observed in other systems too.

The first phenomenon we discovered in Chapter IV is the trade-off between minimization of routing cost and maximization of routing flexibility that was not known before. The variation of the cost design pressure results in a phase transition smoothed out by the finite system size. Understanding this phase transition in Chapter IV allows me to explore qualitatively different behavioral regimes in Chapters V–VI via a simple parameter sweep. While the derivation of the critical cost tolerance depended on the assumptions of only the shortest paths and a square lattice in 2D space, I expect this phase transition to be inevitably present in systems where the number of possible routings grows with distance fast enough.

The trade-off between routing cost and flexibility drives the different robustness behaviors examined in Chapter V. We describe this robustness in the familiar and interpretable language of stress–strain curves, but find their shape to be highly unusual. The curves show at the same time three features characteristic of unconventional materials: residual stress, negative Young's modulus and tension softening. This observation posits that maybe the conventional robustness behavior of integrated systems is generically similar to unconventional mechanical material robustness.

A host of other unconventional behaviors results from the tight coupling of Logical and Physical Architectures studied in Chapter VI. Usually one of those Architectures dominates others, but we show that their tight interplay manifests as complex patterns of avoidance, adjacency, and association. These three patterns are highly reminiscent of physical phenomena in condensed matter systems, such as symmetry breaking, propagation of correlations, and emergent localization.

The intuitions about the interplay of different kinds of degrees of freedom allow us to pursue design of sets of building blocks for self-assembly. In Chapter VII we showed that the set of scalar binding energies can be sufficient to encode the target topology of the assembled structures – a goal previously identified as desirable but not achieved directly. The maximal size of structures that can be reliably assembled is limited by the binding energy contrast afforded by the substrate, as well as the loop entropy. We also show that the partition function of self-assembling structures generically diverges at some value of chemical potential, with the shape of the divergence informing the asymptotic trends in self-assembly yields. The divergence is driven by the assembly of structures larger than the target, thus limiting the conversion of loose monomers into the target structure at high concentrations.

8.3 Compute: New Methods Proposed

Throughout the dissertation I develop several new computational methods. Using maximal entropy approach to statistical mechanics is most definitely not a new idea and has wide applications.[216, 217, 218] However, its application to design problems, and especially it's connection to other computational tools have not been considered in such detail before.

A statistical mechanics perspective is a generalization of both mathematical optimization and constraint satisfaction.[20, 18] From optimization it inherits the discussion of the objective function landscapes, but in contrast does not strictly identify the absolute minimum of the objective with the sole desirable solution. From constraint satisfaction it inherits the accounting for sets of solutions and the search for compatibility of several design objectives, but it doesn't divide all solutions into the binary classes of "feasible" and "infeasible". Instead, statistical mechanics operates with probability landscapes and the discusses their averages corresponding to design phases and their basins corresponding to architecture classes.

Grounded in statistical physics, the Systems Physics analysis emphasizes the aggregation of microstates (detailed design solutions) into mesostates (intermediate scale design solutions). I perform this coarse-graining operation as computation of

Landau free energy in Chapters IV–VI. [107] Chapter III has a brief discussion on the origins and attribution of the "Landau free energy" that drove its usage as both a term and a tool. While textbook computations typically rely on self-similar functional form of Landau free energy in which only the coefficients "flow" under renormalization, in my computations the free energy has a different form at each scale. Moreover, in Chapter VI I show how coarse-graining can be performed in two orthogonal directions, spatial and topological detail, in order to answer different questions about the collective effects in the system. The ability to smoothly change perspective between detailed and coarse design, to zoom in and out of the solution space, was sorely missing from the existing design frameworks.

At the intermediate scale, the design spaces frequently naturally break up into several basins that I associate with qualitatively distinct architecture classes. In order to quantify the robustness of such architecture classes, I show how to compute numerically the stress–strain curves in Chapter V. While these curves have interesting features, they can be abbreviated into two metrics of ultimate design stress and strain. These two metrics constitute the two-factor robustness R^2 which allows direct comparison of architecture classes that immediately highlights the eclipsing and the trade-offs between them, similar to comparisons of solutions in multi-objective optimization.[158]

Statistical mechanics computations provide us with a powerful information structure, the partition function, that provides knowledge about the collective behaviors spanning the whole solution space. However, in order to capture more granular interactions of space and topology in the design space, I developed a new information structure framework based on tensor networks. While tensor networks have many applications in modern science (see overview in Chapter III), they so far remained disconnected from the complex networks literature. My framework allows me the use of arbitrary physical spaces and arbitrary topologies of couplings, which has wide-reaching implications for both design and physics communities. Tensor networks provide both a graphical and a computational way to work with statistical mechanics, further aided by the `TenZ` package. The usage of SVD approximations in tensor network contractions provides a qualitatively new kind of approximate calculations, which enabled me to enumerate 10^{13} design solutions in Chapter VI and up to 10^{167} states in a related work not included in the dissertation.[80]

Apart from tensor networks, I used a set of different analytical combinatorial techniques in Chapter VII to count the possible self-assembled structures at different levels of detail and quantitatively predict their yields. Expressing the results as matrix functions naturally leads to spectral analysis methods which help identify the features of the whole *set* of building blocks driving self-assembly. The proposed self-assembly theory also accounts for and interprets such important quantities as chemical potential, bond entropy, concentration, different forms of yield, and divergence and its exponents. While the work in this dissertation only considers relatively simple building blocks and assembled structures, it paves way to more detailed investigations of design for self-assembly.

8.4 Model: New Design Principles Developed

Having demonstrated the new kinds of insights that can be gained with Systems Physics, as well as the technical ways to reach those insights, we can now discuss the implications for modeling the design problems. The modeling process reduces a wicked problem to a tame one by necessarily ignoring something. So how does the statistical mechanics accounting change the pattern of what we leave out and what we leave in the problem?

Statistical mechanics forces us to explicitly consider the full space of design solutions spanned by a set of design variables. Other analysis of alternatives paradigms operated with a set of a hundred or a thousand candidate solutions, but don't scale well due to the need of looking at each solution individually. Statistical mechanics reduces each solution to its statistical weight and thus allows combing through huge spaces very fast. These design spaces also typically have a special structure driven by the sparsity of couplings: out of all possible interactions between design variables, only very few are present. I exploit the sparsity property in Chapter VI, where sparsity ensures that tensor networks work fast enough.

Along with the accounting of solution space, Systems Physics forces explicit accounting of design objectives and pressures. The essence of the maximal entropy approach is to predict the properties of the ensemble of solutions that are *exclusively* driven by the objectives put in by hand, without any external bias or prejudice. This approach attains a certain degree of mathematical purity and separates the insight given by the computational tools from the human intuition, expertise, and judgment of the designer.

The goal of Systems Physics is to provide the designer with knowledge about the relative merit of possible alternative solutions, as opposed to direct answers to the design problems. This comes in sharp contrast with the optimization school of thought, in which the minimum of the objective function is trivially identified with the solution, often without extended discussion.[20] By treating the design pressures as parameters, Systems Physics can analyze shifting objective landscapes and thus recognize the phase transitions in design (Chapter IV). In the common case of solution space fragmentation into architecture classes, Systems Physics provides knowledge about the architecture class robustness to the human designer, who can then make the qualitative, but informed decision of which class to pick.

Systems Physics can differentiate between the solution space and the design space, as well as between design objectives and design outcomes. Solution spaces enumerate all possible microstates of the system, while design spaces enumerate the space where decisions are made. For the Naval Engineering arrangement problems, the two spaces essentially match, but for the self-assembly problems the designer makes choices about energies and concentrations, but not necessarily the specific particle microstates. Similarly, the choice of a particular solution is driven by the objectives $\mathcal{O}$ weighted by design pressures λ, but the outcomes are measured as observable averages $\langle X \rangle$ that may or may not be conjugate to the pressures. In Chapter IV the design pressures are cost tolerance and bulkhead penalty, but the outcome is characterized by the correlation. In Chapter VII the considered system is explicitly thermal and thus the only allowed "objective" is the energy landscape, but the outcome is measured by several yield metrics. Recognizing both of these dichotomies reinforces the distinction of the original wicked problem described by design and outcome spaces that need to be mapped out gradually, from its tame counterpart described by solution and objective spaces that are known from the beginning.

The relaxation of the rigid objective–solution connection also relaxes the antagonistic relationship between forward and inverse design. Forward design is, loosely, a mapping from design space to outcome space, while inverse design is the opposite mapping. The analyses presented in this dissertation describe the relationship between the two spaces that can be progressively mapped out as knowledge about the whole wicked problem is generated. Since the forward–inverse design dichotomy was most informed by the self-assembly literature, it is self-assembly that most benefits from the integration of the two approaches. Instead of subordinating the building block search to the rigid target structure, or vice versa, we are free to consider them jointly and imagine the new possible structures and functional behaviors.

Lastly, Systems Physics is a design tool that directly highlights its role in the design method by pushing the designer's tacit knowledge to a higher level of abstraction. Systems Physics predicts the relationships between the design elements driven by pressures and gives user-friendly computational tools. The choice of what new knowledge to generate (which computation to run next) or what decision to make (which design variable to freeze/anchor) remains firmly in the hands of the designers, while Systems Physics computations predict the implications of such decisions. In this way, the execution of the Model–Compute–Learn loop allows us to exploit the feasibility of solving tame problems in order to ultimately solve the wicked problems posed to us in the first place.

8.5 Open Questions in Design Science

Having covered the questions already answered by the dissertation, I want to finish with the questions freshly posed by the dissertation. Some of them rely on simple extensions of scale and scope of the studies presented here, whereas others would require new conceptual developments. These questions would affect all three stages of the Model–Compute–Learn loop, and pursuing them would inevitably enrich our understanding of Systems Physics and design problems more broadly.

The simplest question is one of scale. How large a system can we study with statistical mechanics? Chapter VI enumerated 10^{13} solutions, but that number grows exponentially fast with the number of design elements, so how many elements can we consider? How much can we improve the spatial resolution of the arrangements, given that the Chapters IV–VI relied on a very coarse and cartoonish ship hull model? How do the tensor network algorithms scale with system size and target accuracy? Are there better backend software packages for `TenZ` than `PyTNR`?[137, 138] Addressing these questions of scale would allow us to generate better design insights with purely extensive methods.

A similar extensive modification is to consider not just more functional units or a higher spatial resolution, but more layers of description of the designed system and more heterogeneity in those layers. Specifically, the Logical Architecture of naval ships frequently has multiple layers, differentiating between e.g. electric cables, water pipes, and personnel corridors.[157] How do we efficiently account for the heterogeneity of couplings for different edges of the functional network? The tensor network formalism allows using a different form for each coupling tensor, and different space of states for each functional node, but how do we coherently formulate such a heterogeneous problem and extract general insights?

Apart from the extensive scaling, we can consider intensive complications of modeling, requiring new qualitative insights. The Naval Engineering design framework that informed our analysis of Logical and Physical Architecture also includes the Operational Architecture that we left out to have a more focused scope. The Operational Architecture tracks the dynamic operational variables of the functional units, such as hydrodynamic pressures, electric voltages, on/off status, or possible damage to the unit. These variables can either vary stochastically or form dynamical systems that evolve over time with a potential for cascade failures. How can we incorporate such information into our analysis and couple it with the existing spatial and topological degrees of freedom?

The time domain is not only important in analysis of ship's operations. Time also drives the design process in crucial ways. At every time moment, choices to generate knowledge and to commit to certain design elements are made in context of the previous, but not future knowledge. This generically creates path dependency of the design process.[7] In Chapter VI I briefly discussed the free energy landscape in light of the previous design decisions. However, how can we navigate the space of possible decisions histories? Is it even appropriate to try modeling the decision process that we so far left to the designer? What are the appropriate metrics of evolving knowledge and design freedom for us to track over time?

I discuss above that defining the space of solutions for consideration is one of the essential choices in taming a wicked problem. On each Systems Physics loop this choice can potentially be made differently, reflecting the new gained understanding of space. However, in the three Naval Engineering investigations in this dissertation the space was exactly the same even as we sweep across the ranges of design pressure and even as we vary the functional network topology. What could be a reasonable model of revision of the solution space as we get a better understanding of the relationship between the design space and the outcome space over time?

Lastly, the three Naval Engineering studies in this dissertation deep down rely on the same design objective function of routing cost. While this allows for an intimate understanding of this objective function and asking different kinds of questions across Chapters IV–VI, it leaves the concern whether the reported results are generalizable for other kinds of objective functions. The chosen objective function is not particularly detailed or realistic, staying at the same level of abstraction as the Ref. [41] that inspired it. To demonstrate broader applicability of Systems Physics to the modern Naval Engineering practice, it is necessary to perform studies with other objective functions. In the ideal scenario, these studies need to be lead someone who, unlike me, has Naval Engineering as opposed to physics training, and asks questions of more direct engineering relevance. This would also demonstrate that Systems Physics analysis is accessible to users and is not a tool understood only by its developers.

It is my hope that the tools and intuitions of Systems Physics can be expanded by my successors beyond the scope presented here. The study in this dissertation is a product of both physics and design studies, with a significant bias towards the former due to author's training and professional history. Applying statistical physics ideas to design studies is not only an exercise in pedagogy and translation of terms into a different professional jargon, but an earnest examination of the foundations of statistical physics philosophy and computational methods. It is ironic that the more traditional physical problem of interacting colloidal particles raises as many conceptual questions in model-building as the hereto unknown arrangement problem. As we discovered new physical phenomena and new manifestations of old phenomena in this model system, the raised practical and philosophical questions uncover a vast research program that can be pursued into the future.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Karl Popper. *The logic of scientific discovery.* Routledge, 2005.

[2] Robert Farrell and Cliff Hooker. Design, science and wicked problems. *Design studies*, 34(6):681–705, 2013.

[3] César Hidalgo. *Why information grows: The evolution of order, from atoms to economies.* Basic Books, 2015.

[4] César A. Hidalgo, Bailey Klinger, A.-L. Barabási, and Ricardo Hausmann. The product space conditions the development of nations. *Science*, 317(5837):482–487, 2007.

[5] James Manyika, Jeff Sinclair, Richard Dobbs, Gernot Strube, Louis Rassey, Jan Mischke, Jaana Remes, Charles Roxburgh, Katy George, David O'Halloran, and Sreenivas Ramaswamy. *Manufacturing the future: The next era of global growth and innovation.* McKinsey Global Institute, November 2012.

[6] Manufacturing industry: Politicians cannot bring back old-fashioned factory jobs. The Economist, January 14 2017.

[7] Colin P. F. Shields. *Investigating Emergent Design Failures Using a Knowledge-Action-Decision Framework.* PhD thesis, University of Michigan, 2017.

[8] Donald Rumsfeld. *Known and unknown: a memoir.* Penguin, 2011.

[9] C. West Churchman. Wicked problems. *Management Science*, 14(4):B141–B142, 1967.

[10] Horst W. J. Rittel and Melvin M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973.

[11] D. J. Andrews. Art and science in the design of physically large and complex systems. *Proc. Roy. Soc. London A*, 468(2139):891–912, 2012.

[12] Jeffrey A. Drezner, Mark V. Arena, Megan McKernan, Robert Murphy, and Jessie Riposo. Are ships different? policies and procedures for the acquisition of ship programs. Technical report, RAND National Defense Research Institute, Santa Monica CA, 2011.

[13] Jonathan M. Ross. A practical approach for ship construction cost estimating. *COMPIT*, 4:98–110, 2004.

[14] Sean Gallagher. German navy experiences "lcs syndrome" in spades as new frigate fails sea trials. Ars Technica, February 7 2018.

[15] M. Mitchell Waldrop. The chips are down for moore's law. *Nature News*, 530(7589):144, 2016.

[16] Japan captures TOP500 crown with arm-powered supercomputer. TOP500, June 22 2020.

[17] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[18] E. Tsang. *Foundations of Constraint Satisfaction.* Computation in Cognitive Science Series. Academic Press, 1993.

[19] Melanie Mitchell. *An introduction to genetic algorithms.* MIT press, 1998.

[20] Panos Y. Papalambros and Douglass J. Wilde. *Principles of optimal design: modeling and computation.* Cambridge university press, 2000.

[21] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[22] Joshua Schimel. *Writing science: how to write papers that get cited and proposals that get funded.* OUP USA, 2012.

[23] Thomas McKenney and David Singer. Set-based design: A concurrent engineering approach with particular application to complex marine products. *Marine Technology*, 2014.

[24] J. Harvey Evans. Basic design concepts. *Naval Engineers Journal*, 71(4):671–678, 1959.

[25] Rachel Pawling, Victoria Percival, and David Andrews. A study into the validity of the ship design spiral in early stage ship design. *Journal of Ship Production and Design*, 33(2):81–100, 2017.

[26] Herbert D. Benington. Production of large computer programs. *Annals of the History of Computing*, 5(4):350–361, 1983.

[27] Julie Chalfant. Early-stage design for electric ship. *Proceedings of the IEEE*, 103(12):2252–2266, 2015.

[28] Peter E.D. Love, Purnendu Mandal, and Hui Li. Determining the causal structure of rework influences in construction. *Construction Management & Economics*, 17(4):505–517, 1999.

[29] Colin P. F. Shields and David J. Singer. Naval design, knowledge-based complexity, and emergent design failures. *Naval Engineers Journal*, 129(4):75–86, 2017.

[30] Andrew Kusiak. *Concurrent engineering: automation, tools, and techniques.* John Wiley & Sons, 1992.

[31] David J. Singer, Norbert Doerry, and Michael E. Buckley. What is set-based design? *Naval Engineers Journal*, 121(4):31–43, 2009.

[32] Jeffrey K. Liker. *The Toyota Way.* McGraw-Hill, New York, 2004.

[33] James M. Morgan and Jeffrey K. Liker. *The Toyota product development system*, volume 13533. Productivity Press, New York, 2006.

[34] Joshua Ian Bernstein. *Design methods in the aerospace industry: looking for evidence of set-based practices.* PhD thesis, Massachusetts Institute of Technology, 1998.

[35] Eric A. Specking, Cliff Whitcomb, Gregory S. Parnell, Simon R. Goerger, Edward Pohl, and Naga Sai Achyuth Kundeti. Literature review: Exploring the role of set-based design in trade-off analytics. *Naval Engineers Journal*, 130(2):51–62, 2018.

[36] Robert J. Dufresne, William J. Leonard, and William J. Gerace. Knowledge structure. Scientific Reasoning Research Institute.

[37] Shatiel Edwards, Matthew Cilli, Troy Peterson, Michael Zabat, Craig Lawton, and Liliana Shelton. Whole systems trade analysis. In *INCOSE International Symposium*, volume 25, pages 1133–1146. Wiley Online Library, 2015.

[38] Conner Goodrum. *Conceptually Robust Knowledge Generation in Early Stage Complex Design*. PhD thesis, University of Michigan, 2020.

[39] Dorian Brefort, Colin Shields, Agnieta Habben Jansen, Etienne Duchateau, Rachel Pawling, Koen Droste, Ted Jasper, Michael Sypniewski, Conner Goodrum, Mark A. Parsons, et al. An architectural framework for distributed naval ship systems. *Ocean Engineering*, 147:375–385, 2018.

[40] James H. Stock and Mark W. Watson. New indexes of coincident and leading economic indicators. *NBER macroeconomics annual*, 4:351–394, 1989.

[41] Colin P. F. Shields, Michael J. Sypniewski, and David J. Singer. Understanding the relationship between naval product complexity and on-board system survivability using network routing and design ensemble analysis. In Ulrik Dam Nielsen and Jorgen Juncher Jensen, editors, *Proceedings of the 13th International Symposium on PRActical Design of Ships and Other Floating Structures (PRADS' 2016)*. Technical University of Denmark, 2016.

[42] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

[43] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Rev.*, 53(3):464–501, 2011.

[44] Patrick Grim, Daniel J. Singer, Steven Fisher, Aaron Bramson, William J. Berger, Christopher Reade, Carissa Flocken, and Adam Sales. Scientific networks on data landscapes: Question difficulty, epistemic success, and convergence. *Episteme (Edinburgh)*, 10(4):441, 2013.

[45] David J. Wales. Energy landscapes: some new horizons. *Current opinion in structural biology*, 20(1):3–10, 2010.

[46] Kurt Binder and A. Peter Young. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern physics*, 58(4):801, 1986.

[47] Ulrike Feudel. Complex dynamics in multistable systems. *International Journal of Bifurcation and Chaos*, 18(06):1607–1626, 2008.

[48] Bryan C. Daniels, David C. Krakauer, and Jessica C. Flack. Control of finite critical behaviour in a small-scale social system. *Nature communications*, 8(1):1–8, 2017.

[49] Benoit B. Mandelbrot. *The fractal geometry of nature*. W.H. Freeman and Co., 1983.

[50] Mark Newman, Albert-László Barabási, and Duncan J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[51] George M. Whitesides and Bartosz Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[52] Linda S. Hirst. *Fundamentals of soft matter science*. CRC Press, 2019.

[53] C. Grant Willson, Ralph R. Dammel, and Arnost Reiser. Photoresist materials: a historical perspective. In *Metrology, Inspection, and Process Control for Microlithography XI*, volume 3050, pages 38–51. International Society for Optics and Photonics, 1997.

[54] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[55] Shuguang Li, Richa Batra, David Brown, Hyun-Dong Chang, Nikhil Ranganathan, Chuck Hoberman, Daniela Rus, and Hod Lipson. Particle robotics based on statistical mechanics of loosely coupled components. *Nature*, 567(7748):361–365, 2019.

[56] P. Traub and M. Nomura. Structure and function of escherichia coli ribosomes: Vi. mechanism of assembly of 30 s ribosomes studied in vitro. *Journal of molecular biology*, 40(3):391–413, 1969.

[57] Tong Wang, Ruojie Sha, Remi Dreyfus, Mirjam E. Leunissen, Corinna Maass, David J. Pine, Paul M. Chaikin, and Nadrian C. Seeman. Self-replication of information-bearing nanoscale patterns. *Nature*, 478(7368):225–228, 2011.

[58] Zorana Zeravcic and Michael P. Brenner. Self-replicating colloidal clusters. *Proceedings of the National Academy of Sciences*, 111(5):1748–1753, 2014.

[59] Ofer Kimchi, Carl P. Goodrich, Alexis Courbet, Agnese I. Curatolo, Nicholas B. Woodall, David Baker, and Michael P. Brenner. Self-assembly based post-translational protein oscillators. *bioRxiv*, 2020.

[60] Edward Barry and Zvonimir Dogic. Entropy driven self-assembly of nonamphiphilic colloidal membranes. *Proc. Natl. Acad. Sci. U.S.A.*, 107(23):10348–10353, 2010.

[61] Matthew R. Jones, Nadrian C. Seeman, and Chad A. Mirkin. Programmable materials and the nature of the dna bond. *Science*, 347(6224), 2015.

[62] James Sethna. *Statistical mechanics: entropy, order parameters, and complexity*, volume 14. Oxford University Press, 2006.

[63] Michael E. Cates and Vinothan N. Manoharan. Celebrating soft matter's 10th anniversary: Testing the foundations of classical entropy: colloid experiments. *Soft Matter*, 11(33):6538–6546, 2015.

[64] Guangnan Meng, Natalie Arkus, Michael P. Brenner, and Vinothan N. Manoharan. The free-energy landscape of clusters of attractive hard spheres. *Science*, 327(5965):560–563, 2010.

[65] Chrisy Xiyu Du, Greg van Anders, Richmond S. Newman, and Sharon C. Glotzer. Shape-driven colloidal crystal–crystal transitions. *Proc. Natl. Acad. Sci. U.S.A.*, 114:E3892–E3899, 2017.

[66] William Zygmunt, Erin G. Teich, Greg van Anders, and Sharon C. Glotzer. Topological order in densely packed anisotropic colloids. *Physical Review E*, 100(3):032608, 2019.

[67] James Antonaglia, Greg van Anders, and Sharon C. Glotzer. Mapping entropy distributions in entropically ordered crystals. *In preparation*, 2018.

[68] Ellen D. Klein, Rebecca W. Perry, and Vinothan N. Manoharan. Physical interpretation of the partition function for colloidal clusters. *Physical Review E*, 98(3):032608, 2018.

[69] Joshua A. Anderson, Jens Glaser, and Sharon C. Glotzer. Hoomd-blue: A python package for high-performance molecular dynamics and hard particle monte carlo simulations. *Computational Materials Science*, 173:109363, 2020.

[70] Mena Youssef, Theodore Hueckel, Gi-Ra Yi, and Stefano Sacanna. Shape-shifting colloids via stimulated dewetting. *Nature communications*, 7(1):1–7, 2016.

[71] Zihao Ou, Binbin Luo, Chang Liu, and Qian Chen. Liquid-phase tem imaging of self-assembly pathways of anisotropic nanoparticles. *Microscopy and Microanalysis*, 25(S2):1414–1415, 2019.

[72] Greg van Anders, Daphne Klotsa, Andrew S. Karas, Paul M. Dodd, and Sharon C. Glotzer. Digital Alchemy for Materials Design: Colloids and Beyond. *ACS Nano*, 9:9542–9553, 2015.

[73] Greg van Anders, Daphne Klotsa, N. Khalid Ahmed, Michael Engel, and Sharon C. Glotzer. Understanding shape entropy through local dense packing. *Proc. Natl. Acad. Sci. U.S.A.*, 111:E4812–E4821, 2014.

[74] Sharon C. Glotzer and Michael J. Solomon. Anisotropy of building blocks and their assembly into complex structures. *Nat. Mater.*, 6:557–562, 2007.

[75] Robert J. Macfarlane, Byeongdu Lee, Matthew R. Jones, Nadine Harris, George C. Schatz, and Chad A. Mirkin. Nanoparticle superlattice engineering with dna. *science*, 334(6053):204–208, 2011.

[76] Paul W. K. Rothemund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[77] Po-Ssu Huang, Scott E. Boyken, and David Baker. The coming of age of *de novo* protein design. *Nature*, 537(7620):320–327, 2016.

[78] Ran Niu, Chrisy Xiyu Du, Edward Esposito, Jakin Ng, Michael P. Brenner, Paul L. McEuen, and Itai Cohen. Magnetic handshake materials as a scale-invariant platform for programmed self-assembly. *Proceedings of the National Academy of Sciences*, 116(49):24402–24407, 2019.

[79] Greg van Anders, N. Khalid Ahmed, Ross Smith, Michael Engel, and Sharon C. Glotzer. Entropically patchy particles: Engineering valence through shape entropy. *ACS Nano*, 8:931–940, 2014.

[80] Andrei A. Klishin and Greg van Anders. When does entropy promote local organization? *Soft Matter*, 16:6523–6531, 2020.

[81] Pablo F. Damasceno, Michael Engel, and Sharon C. Glotzer. Predictive Self-Assembly of Polyhedra into Complex Structures. *Science*, 337(6093):453–457, 2012.

[82] Erin G. Teich, Greg van Anders, and Sharon C. Glotzer. Identity crisis in alchemical space drives the entropic colloidal glass transition. *Nature communications*, 10(1):1–10, 2019.

[83] Ali Mohraz and Michael J. Solomon. Direct visualization of colloidal rod assembly by confocal microscopy. *Langmuir*, 21(12):5298–5306, 2005.

[84] Amir Haji-Akbari, Michael Engel, Aaron S. Keys, Xiaoyu Zheng, Rolfe G. Petschek, Peter Palffy-Muhoray, and Sharon C. Glotzer. Disordered, Quasicrystalline and Crystalline Phases of Densely Packed Tetrahedra. *Nature*, 462:773–777, 2009.

[85] Yina Geng, Greg van Anders, Paul M Dodd, Julia Dshemuchadse, and Sharon C Glotzer. Engineering entropy for the inverse design of colloidal crystals from hard shapes. *Science advances*, 5(7):eaaw0514, 2019.

[86] Rose K. Cersonsky, Julia Dshemuchadse, James A. Antonaglia, Greg van Anders, and Sharon C. Glotzer. Pressure-tunable photonic band gaps in an entropic colloidal crystal. *Physical Review Materials*, 2:125201, 2018.

[87] W. Benjamin Rogers and Vinothan N. Manoharan. Programming colloidal phase transitions with dna strand displacement. *Science*, 347(6222):639–642, 2015.

[88] Zorana Zeravcic, Vinothan N. Manoharan, and Michael P. Brenner. Size limits of self-assembled colloidal structures made using specific interactions. *Proceedings of the National Academy of Sciences*, 111(45):15918–15923, 2014.

[89] Margaret E. Johnson and Gerhard Hummer. Nonspecific binding limits the number of proteins in a cell and shapes their interaction networks. *Proceedings of the National Academy of Sciences*, 108(2):603–608, 2011.

[90] Miriam H. Huntley, Arvind Murugan, and Michael P. Brenner. Information capacity of specific interactions. *Proceedings of the National Academy of Sciences*, 113(21):5841–5846, 2016.

[91] Arvind Murugan, James Zou, and Michael P. Brenner. Undesired usage and the robust self-assembly of heterogeneous structures. *Nature communications*, 6:6203, 2015.

[92] Arvind Murugan, Zorana Zeravcic, Michael P. Brenner, and Stanislas Leibler. Multifarious assembly mixtures: Systems allowing retrieval of diverse stored structures. *Proceedings of the National Academy of Sciences*, 112(1):54–59, 2015.

[93] Don Hall, Chris Williams, and Roy Conli. *Big Hero 6*. Walt Disney Studios Motion Pictures, 2015.

[94] Lawrence Sklar. Philosophy of statistical mechanics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2015 edition, 2015.

[95] Joshua A. Anderson, M. Eric Irrgang, and Sharon C. Glotzer. Scalable metropolis monte carlo for simulation of hard shapes. *Computer Physics Communications*, 204:21–30, 2016.

[96] Elizabeth R. Chen, Daphne Klotsa, Michael Engel, Pablo F. Damasceno, and Sharon C. Glotzer. Complexity in surfaces of densest packings for families of polyhedra. *Physical Review X*, 4(1):011024, 2014.

[97] Mehran Kardar. *Statistical physics of fields*. Cambridge University Press, 2007.

[98] Steven H. Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. Addison-Wesley, 1994.

[99] Edwin T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957.

[100] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 623–656, 1948.

[101] Juyong Park and Mark E. J. Newman. Statistical mechanics of networks. *Physical Review E*, 70(6):066117, 2004.

[102] Mehran Kardar. *Statistical physics of particles*. Cambridge University Press, 2007.

[103] Murray Gell-Mann and Constantino Tsallis, editors. *Nonextensive entropy: interdisciplinary applications*. Oxford University Press, 2004.

[104] Pavel L. Krapivsky, Sidney Redner, and Eli Ben-Naim. *A kinetic view of statistical physics*. Cambridge University Press, 2010.

[105] Rose K. Cersonsky, Greg van Anders, Paul M. Dodd, and Sharon C. Glotzer. Relevance of packing to colloidal self-assembly. *Proceedings of the National Academy of Sciences*, 115(7):1439–1444, 2018.

[106] Lev Davidovich Landau. On the theory of phase transitions, i. *Zh. Eksp. Teor. Fiz.*, 7:19, 1937.

[107] Nigel Goldenfeld. *Lectures on phase transitions and the renormalization group*. Addison-Wesley, Reading MA, 1992.

[108] Л. Д. Ландау and Е. М. Лифшиц. *Статистическая физика. Часть 1*, volume 5 of *Теоретическая физика*. Наука, 3 edition, 1976.

[109] L. D. Landau and E. M. Lifshitz. *Statistical Physics*, volume 5 of *Course of Theoretical Physics*. Elsevier Science, 1980.

[110] Л. Д. Ландау and Е. М. Лифшиц. *Статистическая физика. Часть 1*, volume 5 of *Теоретическая физика*. Наука, 2 edition, 1964.

[111] Л. Д. Ландау. *Собрание трудов*, volume 1. Наука, 1969.

[112] Л. Д. Ландау. *Собрание трудов*, volume 2. Наука, 1969.

[113] J. S. Langer. Metastable states. *Physica*, 73(1):61–72, 1974.

[114] Lev Petrovich Gor'kov. Microscopic derivation of the ginzburg-landau equations in the theory of superconductivity. *Sov. Phys. JETP*, 9(6):1364–1367, 1959.

[115] Leo P. Kadanoff. Scaling laws for ising models near t c. *Physics Physique Fizika*, 2(6):263, 1966.

[116] Josiah Willard Gibbs. *Elementary Principles in Statistical Mechanics.* Charles Scribner's Sons, New York, 1902.

[117] Rodney J. Baxter. *Exactly solved models in statistical mechanics.* Elsevier, 2016.

[118] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial Mathematics and its Applications*, 1:221–244, 1971.

[119] Predrag Cvitanović. *Group theory: birdtracks, Lie's, and exceptional groups.* Princeton University Press, 2008.

[120] Frank Verstraete, Valentin Murg, and J. Ignacio Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.

[121] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

[122] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[123] Garnet Kin-Lic Chan and Sandeep Sharma. The density matrix renormalization group in quantum chemistry. *Annual Review of Physical Chemistry*, 62:465–481, 2011.

[124] Igor L. Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008.

[125] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang. Second renormalization of tensor-network states. *Phys. Rev. Lett.*, 103(16):160601, 2009.

[126] Michael Levin and Cody P. Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Phys. Rev. Lett.*, 99(12):120601, 2007.

[127] Edwin Stoudenmire and David J. Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[128] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[129] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017.

[130] Jacob D. Biamonte, Jason Morton, and Jacob Turner. Tensor network contractions for# sat. *J. Stat. Phys.*, 160(5):1389–1404, 2015.

[131] Stefanos Kourtis, Claudio Chamon, Eduardo R. Mucciolo, and Andrei E. Ruckenstein. Fast counting with tensor networks. *SciPost Phys.*, 7:060, 2019.

[132] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P. Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[133] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, Danilo P. Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9(6):431–673, 2017.

[134] Ivan V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[135] Mark Newman. *Networks: an introduction*. Oxford University Press, 2010.

[136] Robert N. C. Pfeifer, Glen Evenbly, Sukhwinder Singh, and Guifre Vidal. Ncon: A tensor network contractor for matlab. *arXiv preprint arXiv:1402.0939*, 2014.

[137] Adam S. Jermyn. Automatic Contraction of Unstructured Tensor Networks. *SciPost Phys.*, 8:5, 2020.

[138] Adam S. Jermyn. Efficient tree decomposition of high-rank tensors. *Journal of Computational Physics*, 377:142–154, 2019.

[139] Kerson Huang. *Statistical mechanics*. Wiley, 1987.

[140] Kerson Huang. *Introduction to statistical physics*. CRC press, 2009.

[141] Sadi Carnot. *Réflexions sur la puissance motrice du feu et sur les machines propres a développer cette puissance*. Bachelier, Paris, 1824.

[142] Rudolph Clausius. *Abhandlungen Über Die Mechanische Wärmetheorie*. Friedrich Vieweg und Sohn, Braunschweig, 1864.

[143] Ludwig Boltzmann. *Vorlesungen über Gastheorie*. Johann Ambrosius Barth, Leipzig, 1896.

[144] James Clerk Maxwell. *Theory of heat*. Longmans, Green, 1891.

[145] J. Willard Gibbs. *The collected works of J. Willard Gibbs*. Longmans, Green, 1928.

[146] Niels Bohr. Über die serienspektra der elemente. *Zeitschrift für Physik*, 2(5):423–469, 1920.

[147] P. W. Anderson. *Basic Notions of Condensed Matter Physics*. Benjamin/Cummings, Menlo Park, 1984.

[148] Mark Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2010.

[149] P. W. Anderson. More is different. *Science*, 177:393–396, 1972.

[150] Edwin T. Jaynes. The gibbs paradox. In *Maximum entropy and bayesian methods*, pages 1–21. Springer, 1992.

[151] Daan Frenkel. Why colloidal systems can be described by statistical mechanics: some not very original comments on the gibbs paradox. *Molecular Physics*, 112(17):2325–2329, 2014.

[152] Andrei A. Klishin, Colin P. F. Shields, David J. Singer, and Greg van Anders. Statistical physics of design. *New J. Phys.*, 20(10):103038, 2018.

[153] Benjamin S. Blanchard and Wolter J. Fabrycky. *Systems engineering and analysis*, volume 4. Prentice Hall, Englewood Cliffs, NJ, 1990.

[154] Michael C. Dorneich and Nikolaos V. Sahinidis. Global optimization algorithms for chip layout and compaction. *Engineering Optimization+ A35*, 25(2):131–154, 1995.

[155] Charles H. Aikens. Facility location models for distribution planning. *European journal of operational research*, 22(3):263–279, 1985.

[156] Amine Drira, Henri Pierreval, and Sonia Hajri-Gabouj. Facility layout problems: A survey. *Annual Reviews in Control*, 31(2):255–267, 2007.

[157] Colin P. F. Shields, Douglas T. Rigterink, and David J. Singer. Investigating physical solutions in the architectural design of distributed ship service systems. *Ocean Eng.*, 135:236 – 245, 2017.

[158] Alan Brown and Juan Salcedo. Multiple-objective optimization in naval ship design. *Naval Engineers Journal*, 115(4):49–62, 2003.

[159] Christos N. Likos. Effective interactions in soft condensed matter physics. *Physics Reports*, 348(4–5):267 – 439, 2001.

[160] Marc Z. Miskin, Gurdaman Khaira, Juan J. de Pablo, and Heinrich M. Jaeger. Turning statistical physics models into materials design engines. *Proc. Natl. Acad. Sci. U.S.A.*, 113(1):34–39, 2016.

[161] Carver Mead and Lynn Conway. *Introduction to VLSI systems*, volume 1080. Addison-Wesley Reading, MA, 1980.

[162] Mark S. Daskin. *Network and discrete location: models, algorithms, and applications.* John Wiley & Sons, 2011.

[163] Rosario N. Mantegna and H. Eugene Stanley. Scaling behaviour in the dynamics of an economic index. *Nature*, 376(6535):46–49, 1995.

[164] Rosario N. Mantegna and H. Eugene Stanley. *Introduction to econophysics: correlations and complexity in finance.* Cambridge University Press, 1999.

[165] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, Jan 2002.

[166] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86:3200–3203, Apr 2001.

[167] Kwang-Il Goh, Michael E. Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proc. Natl. Acad. Sci. U.S.A.*, 104(21):8685–8690, 2007.

[168] Andrei A. Klishin, Alec Kirkley, David J. Singer, and Greg van Anders. Robust design in systems physics. 2018.

[169] Genichi Taguchi. *Introduction to quality engineering: designing quality into products and processes.* Asian Productivity Association, Tokyo, 1986.

[170] Madhan Shridhar Phadke. *Quality engineering using robust design.* Prentice Hall PTR, 1995.

[171] Barry W. Boehm. Understanding and controlling software costs. *Journal of Parametrics*, 8(1):32–68, 1988.

[172] George Ellwood Dieter and David J. Bacon. *Mechanical Metallurgy.* McGraw-Hill New York, 3rd edition, 1986.

[173] Wei Li, Amir Bashan, Sergey V. Buldyrev, H. Eugene Stanley, and Shlomo Havlin. Cascading failures in interdependent lattice networks: The critical role of the length of dependency links. *Phys. Rev. Lett.*, 108:228702, May 2012.

[174] Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025, 2010.

[175] Charles D. Brummitt, Raissa M. D'Souza, and E. A. Leicht. Suppressing cascades of load in interdependent networks. *Proc. Natl. Acad. Sci. U.S.A.*, 109(12):E680–E689, 2012.

[176] Daan Frenkel. Order through entropy. *Nat. Mater.*, 14:9–12, 2015.

[177] Eric S. Harper, Greg van Anders, and Sharon C. Glotzer. The entropic bond in colloidal crystals. *Proc. Natl. Acad. Sci. U.S.A.*, 116:16703–16710, 2019.

[178] Tamás Vicsek and Anna Zafeiris. Collective motion. *Physics reports*, 517(3-4):71–140, 2012.

[179] Jesse L. Silverberg, Matthew Bierbaum, James P. Sethna, and Itai Cohen. Collective motion of humans in mosh and circle pits at heavy metal concerts. *Physical review letters*, 110(22):228701, 2013.

[180] Debashish Chowdhury, Ludger Santen, and Andreas Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329(4-6):199–329, 2000.

[181] Andrew E. Noble, Todd S. Rosenstock, Patrick H. Brown, Jonathan Machta, and Alan Hastings. Spatial patterns of tree yield explained by endogenous forces through a correspondence between the ising model and ecology. *Proceedings of the National Academy of Sciences*, 115(8):1825–1830, 2018.

[182] Hana Koorehdavoudi and Paul Bogdan. A statistical physics characterization of the complex systems dynamics: Quantifying complexity from spatio-temporal interactions. *Scientific reports*, 6(1):1–13, 2016.

[183] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[184] H. Horii and Siavouche Nemat-Nasser. Brittle failure in compression: splitting, faulting and brittle-ductile transition. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 319(1549):337–374, 1986.

[185] James D Byerlee. Brittle-ductile transition in rocks. *Journal of Geophysical Research*, 73(14):4741–4750, 1968.

[186] P. J. Withers. Residual stress and its role in failure. *Reports on progress in physics*, 70(12):2211, 2007.

[187] Victor C. Li, Youjiang Wang, and Stanley Backer. A micromechanical model of tension-softening and bridging toughening of short random fiber reinforced brittle matrix composites. *Journal of the Mechanics and Physics of Solids*, 39(5):607–625, 1991.

[188] B. L. Karihaloo. *Fracture Mechanics and Structural Concrete*. Concrete design and construction series. Longman Scientific & Technical, 1995.

[189] Susan Hesse Owen and Mark S. Daskin. Strategic facility location: A review. *European journal of operational research*, 111(3):423–447, 1998.

[190] Joseph W. Dorsey. Brownfields and greenfields: the intersection of sustainable development and environmental stewardship. *Environmental Practice*, 5(1):69–76, 2003.

[191] David Adams and Craig Watkins. *Greenfields, brownfields and housing development*. John Wiley & Sons, 2008.

[192] Richard Hopkins and Kevin Jenkins. *Eating the IT elephant: moving from greenfield development to brownfield*. Addison-Wesley Professional, 2008.

[193] Mark E. J. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[194] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti. Bond-orientational order in liquids and glasses. *Physics Review B*, 28(2):784, 1983.

[195] J. Roth and A. R. Denton. Solid-phase structures of the dzugutov pair potential. *Phys. Rev. E*, 61:6845–6857, Jun 2000.

[196] David J Thouless. Electrons in disordered systems and the theory of localization. *Physics Reports*, 13(3):93–142, 1974.

[197] Marcel Filoche and Svitlana Mayboroda. Strong localization induced by one clamped point in thin plate vibrations. *Physical review letters*, 103(25):254301, 2009.

[198] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[199] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[200] Robert N. C. Pfeifer, Jutho Haegeman, and Frank Verstraete. Faster identification of optimal contraction sequences for tensor networks. *Physical Review E*, 90(3):033315, 2014.

[201] Filip Ilievski, Madhav Mani, George M. Whitesides, and Michael P. Brenner. Self-assembly of magnetically interacting cubes by a turbulent fluid flow. *Physical Review E*, 83(1):017301, 2011.

[202] Д. И. Менделеев. Соотношение свойств с атомным весом элементов. *Журнал Русского химического общества*, 1(9):60, 1869.

[203] Mendeleeff. The periodic law of the chemical elements. *J. Chem. Soc., Trans.*, 55:634–656, 1889.

[204] Jingshan Zhang, Sergei Maslov, and Eugene I Shakhnovich. Constraints imposed by non-functional protein–protein interactions on gene expression and proteome size. *Molecular systems biology*, 4(1):210, 2008.

[205] S. Sacanna, W. T. M. Irvine, P. M. Chaikin, and D.J. Pine. Lock and key colloids. *Nature*, 464:575–578, 2010.

[206] Yu Wang, Yufeng Wang, Xiaolong Zheng, Gi-Ra Yi, Stefano Sacanna, David J. Pine, and Marcus Weck. Three-dimensional lock and key colloids. *J. Am. Chem. Soc.*, 136(19):6866–6869, 2014.

[207] Laura Colón-Meléndez, Daniel J. Beltran-Villegas, Greg van Anders, Jun Liu, Matthew Spellings, Stefano Sacanna, David J. Pine, Sharon C. Glotzer, Ronald G. Larson, and Michael J. Solomon. Binding kinetics of lock and key colloids. *J. Chem. Phys.*, 142(17):–, 2015.

[208] Masao Doi and Samuel Frederick Edwards. *The theory of polymer dynamics*, volume 73 of *International Series of Monographs on Physics*. Oxford University Press, 1988.

[209] David Wood. The computation of polylogarithms. Technical Report 15-92*, University of Kent, Computing Laboratory, University of Kent, Canterbury, UK, June 1992.

[210] Mark E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118, 2001.

[211] Ivan Kryven. Finite connected components in infinite directed and multiplex networks with arbitrary degree distributions. *Physical Review E*, 96(5):052304, 2017.

[212] Brian Karrer and Mark E. J. Newman. Random graphs containing arbitrary distributions of subgraphs. *Physical Review E*, 82(6):066118, 2010.

[213] M. E. J. Newman. Spectra of networks containing short loops. *Physical Review E*, 100(1):012314, 2019.

[214] Jason W. Rocks, Nidhi Pashine, Irmgard Bischofberger, Carl P. Goodrich, Andrea J. Liu, and Sidney R. Nagel. Designing allostery-inspired response in mechanical networks. *Proceedings of the National Academy of Sciences*, 114(10):2520–2525, 2017.

[215] Le Yan, Riccardo Ravasio, Carolina Brito, and Matthieu Wyart. Architecture and coevolution of allosteric materials. *Proceedings of the National Academy of Sciences*, 114(10):2526–2531, 2017.

[216] Edwin T. Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[217] Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A Dill. Principles of maximum entropy and maximum caliber in statistical physics. *Reviews of Modern Physics*, 85(3):1115, 2013.

[218] Christopher W. Lynn, Ari E. Kahn, and Danielle S. Bassett. Structure from noise: Mental errors yield abstract representations of events. *arXiv preprint arXiv:1805.12491*, 2018.