Hazy Oracles in Deep Learning

by

Stephan J. Lemmer

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in The University of Michigan
2023

Doctoral Committee:

Professor Jason J. Corso
Associate Professor Kira Barton
Associate Professor Jeffrey P. Bigham
Assistant Professor Anhong Guo

Stephan J. Lemmer

lemmersj@umich.edu

ORCID iD: 0000-0002-6911-786X

TABLE OF CONTENTS

LIST OF TABLES . . . . . **v**

LIST OF FIGURES . . . . . **vii**

LIST OF APPENDICES . . . . . **xvi**

ABSTRACT . . . . . **xvii**

CHAPTER

1 Introduction **1**

1.1 Supervised Training and The Oracle Assumption 2
1.2 Deferred Inference 5
1.3 Deferred Inference for Interaction 8
1.4 Contributions 9
1.4.1 Learning an Ideal Deferral Function 9
1.4.2 Development of Aggregation Functions 9
1.4.3 A general formulation and evaluation for deferred inference 10
1.4.4 Human-centered deferred inference 10
1.5 Thesis and Impact Statement 11

2 Related Work **12**

2.1 Applications Using Hazy Oracles 12
2.1.1 Exemplar HIL Applications 15
2.2 Addressing Limitations of Hazy Oracles 19
2.2.1 Machine Learning Approaches 19
2.2.2 Human-Centered Approaches 23
2.3 Conclusion 27

3 Training and Evaluating Deferral Functions **28**

3.1 Problem Statement 30
3.2 Dual-loss Additional Error Regression (DAER) 32
3.3 Experiments 33
3.3.1 Keypoint-Conditioned Viewpoint Estimation 33
3.3.2 Hierarchical Scene Classification 39
3.3.3 Ablation: Importance of Subgoals 41
3.4 Why Does DAER work? 43

3.5 Conclusion . . . 44

4 Addressing Imperfect Deferral Responses . . . 45

4.1 Quality and Effects of Crowdsourced Initializations . . . 47
4.1.1 Data Collection . . . 47
4.1.2 Semantic Quality of Bounding-Box Annotations . . . 48
4.1.3 Effect of Initialization on Tracker Performance . . . 49
4.2 The Importance of Considering the Deferral Response . . . 51
4.2.1 Evaluation Method . . . 52
4.2.2 Assuming an Oracle Deferral Response . . . 54
4.2.3 Smart Replacement: Simple Mitigation of Hazy Deferral Responses . . . 55
4.3 Conclusion . . . 57

5 Probabilistic Aggregation of Human Inputs . . . 59

5.1 Method . . . 62
5.1.1 Dimension Line Annotation Tool and Self-Filtering . . . 62
5.1.2 Automated Outlier Removal . . . 63
5.2 Evaluation . . . 66
5.2.1 Semantic Analysis and Filtering . . . 67
5.2.2 Analysis of Aggregation Function . . . 69
5.3 Conclusion . . . 72

6 Comprehensive Evaluation and General Method . . . 74

6.1 A Comprehensive Evaluation . . . 74
6.2 Proposed Method . . . 76
6.3 Exemplar Applications . . . 78
6.3.1 Single-Target VOT . . . 78
6.3.2 Referring Expression Comprehension . . . 79
6.4 Experiments . . . 80
6.5 Conclusion . . . 85

7 Setting Human-Centered Deferral Criteria . . . 86

7.1 Theory on Thresholds . . . 88
7.2 Experimental Setup . . . 90
7.2.1 Motivating Application . . . 90
7.2.2 Procedure . . . 91
7.3 Results . . . 94
7.4 Setting Deferral Criteria . . . 100
7.5 Conclusion . . . 102

8 Conclusion . . . 104

8.1 Lessons Learned . . . 104
8.2 Benefits and Ethical Concerns . . . 105
8.3 Limitations and Future Work . . . 107
8.4 Conclusion . . . 109

APPENDICES . . . 110

BIBLIOGRAPHY **142**

LIST OF TABLES

1.1 Error for the task model with No Deferral (Err @ 0) and Perfect Deferral on two applications. A perfect deferral method can reduce error by over 76%. Err @ 0 is reported as Mean and Standard Error of 100 trials. Full setup described in Section 2.1.1. 7

3.1 The AMAE of DAER and baselines across 5 folds. We note that DAER has the lowest mean, never performs worse than 25.3% over this mean, and has the lowest worst-fold AMAE. 37
3.2 AMAE for baselines and DAER on the HSC application (lower is better). Standard error is calculated across five coarse models and, for DAER, five deferral functions. . . 40
3.3 The percentage of human inputs that must be deferred for various target MAEs on the hierarchical scene classification task, as well as the percent reduction from using DAER over the next-best baseline. 41
3.4 AMAE for DAER and its individual subgoals (lower is better). 42

4.1 Performance under the assumption of an oracle deferral response. Mean and standard error are calculated using 1000 sets of the 100 videos where the initialization is randomly drawn. 54
4.2 ARMAE of various deferral functions where the tracker confidence is used as the aggregation function. 57

6.1 The effect of DBSCAN parameters on the deferral-free mean error (1-IoU) of our method. Lower is better. 79
6.2 Number of expressions for every target object in the three major referring expression comprehension datasets. 80
6.3 The DEV and Err @ 1 metrics for baselines and our method (Err @ 0 shown in Table 6.4). Our method performs best across all applications and splits by a significant margin. 81
6.4 Err @ 0 (No deferral) and Err @ 1 using our method. We can reduce error by up to 48.73%. 81
6.5 The DEV for all potential deferral functions on the referring expression comprehension application. 83
6.6 DEV when only the text embedding is dropped out (text dropout) no dropout (softmax) and full dropout. Deferral score is entropy, aggregation function is belief update. . . . 84

7.1 Types of deferral responses and quantity of each seen in our experiment. 98
7.2 Mean absolute error when targeting deferral rates of 0.1, 0.2, and 0.3. Displayed tolerances are standard error. 101

7.3 Number of violations when upper bounding deferral rates of 0.1, 0.2, and 0.3. 101

LIST OF FIGURES

1.1 While deep learned models perform inference based on a single piece of human-provided information that is assumed perfect, humans input may be (A) semantically ambiguous or (B) unambiguous but misinterpreted in addition to being (C) correctly interpreted. Because of this, using modern deep-learned models as-is will result in unnecessary error. (A) is purely illustrative, while (B) and (C) are real-world responses of the UNITER model to RefCOCO queries. 1

1.2 By enforcing the framework of infer $\rightarrow$ evaluate $\rightarrow$ update model, supervised learning is limited to performing one inference at a time. 3

1.3 A simplified example of deferred inference with hazy oracles. The initial request is ambiguous, resulting in a two-in-three chance of an incorrect answer. By deferring, the robot obtains sufficient information to solve the task. 6

1.4 Our formulation of deferred inference, with the task shown in Figure 1.3 as illustration. x is the robot's perception and h_n is the human input at step n. 6

1.5 Whether a phrase is semantically ambiguous (left) or simply unclear to the model (right), a new human input can change an inference from incorrect (pink) to correct (blue). Best viewed in color. 7

2.1 A collection of problems using human inputs, displayed in a Venn diagram of input characteristics. 14

2.2 An illustration of the viewpoint estimation task. From 16

2.3 Hierarchical scene classification is a way to provide additional information to fine-grained classification tasks—for example, by helping differentiate between a french bulldog and a boston terrier 17

2.4 An illustrative example of Kendall & Gal's uncertainty taxonomy applied to a simple cat/dog classification model. *aleatoric uncertainty* refers to uncertainty in the input that can not be reduced by more training—the cat can't be seen well enough for high certainty. Epistemic uncertainty refers to uncertainty based in the model's understanding of the world: since it has never seen a pig, it can't confidently place the image in a class. 20

3.1 An example from the keypoint-conditioned viewpoint estimation application , with a heatmap of error caused by all potential clicks overlaid. Approaches focused on input-space accuracy would select the red keypoint over the yellow keypoint as it closer to the gold-standard (green) keypoint, even though this results in higher error. 28

3.2 On both the KCVE (top row) and HSC (bottom row) applications, the task model may or may not base its answer solely on immutable data. For KCVE, the gold-standard click is shown as a green circle, while the overlaid heatmap shows error from low (green) to high (red). For HSC, the gold-standard is bolded, correct answers are green, and incorrect answers are red. 29
3.3 A flowchart of deferred inference on a single sample $(x, h_c) \in \mathcal{D}$. The deferral function, $g(x, h_c) \in \{0, 1\}$ seeks to defer samples for which using the candidate human input results in worse performance than using the gold-standard. 30
3.4 Evaluation considers when a candidate human input (red) outperforms the gold-standard (green). 30
3.5 DAER separates the regression of additional error into two components: predicting whether the human input is correct through a correctness loss, and predicting the additional error through a regression loss that is only backpropagated if the human input is incorrect. For illustration, we include an example from the hierarchical scene classification application. 32
3.6 Prediction layers for the KCVE task model (left) and deferral function (right), which accept the keypoint and image embeddings from 34
3.7 The interface provided to crowd workers for crowdsourcing keypoint clicks. 35
3.8 Distribution of distances between candidate and gold-standard keypoints. 36
3.9 Select example cases from KCVE. Ideal accept location—the coverage where sorting by additional error would accept a human input—is given by the white star. Overlaid heatmaps are from green (low error) to red (high error) 37
3.10 Quantitatively chosen KCVE heatmaps. The gold-standard human input is shown in green, the candidate human input is shown in red, and a red-yellow-green heatmap gives the additional error for that keypoint click. Methods closer to the white "ideal" star are better for that example. (A) The four cases where the gold-standard human input provided the worst absolute performance. (B) The four cases where the candidate human input improved upon the gold-standard human input the most. (C) The four cases with the highest additional error. 38
3.11 Our deferral function architecture and output format for the HSC task. Each potential candidate human input is given two outputs that are multiplied to estimate the additional error. 39
3.12 The MAE-coverage plot for the HSC application. Dark lines represent the mean of all runs, shaded area represents one standard error. 40
3.13 Geodesic error from the task model (top) compared to the additional error prediction from a DAER deferral function (bottom). Error is overlaid from high (red) to low (green). Predictions are normalized per-image. 42
3.14 The sensitivity of Click-Here CNN to click location. The majority of images have few potential clicks capable of causing significant error, and 36.3% do not respond to the keypoint click at all. 43
3.15 The sensitivity of plugin networks to coarse classification. For 67.3% of images, an incorrect human input does not result in a correct fine-grained classification becoming incorrect. 43

4.1 In single-target VOT, the tracker must be initialized with a bounding box to designate which object to follow. *Smart replacement* minimizes the number of initializations for a target accuracy by allowing the deferral function to accept the first initialization if it performs well, and allowing the aggregation function to choose between the first and second initialization if inference is deferred. We see the importance of these functions above, where nearly identical initializations result in dramatically different performances. 45
4.2 Although it is tempting to only examine the quality of the semantic input, the complexity of deep-learned models means that not all high-quality human inputs provide equal performance. 46
4.3 The instructions (left) and interface (right) provided to crowd workers for the bounding box annotation. Workers are given vertical and horizontal guidelines to assist in bounding box construction. Instruction images were taken from CelebA and the ImageNet Large Scale Visual Recognition Challenge 47
4.4 The histogram of first-frame IoU scores between crowdsourced and gold-standard annotations after removing malicious annotators. 93.1% of annotations have an IoU greater than or equal to 0.5, corresponding to a successful detection in the literature . . 48
4.5 Example bounding boxes, showing perceptual similarity of various IoUs. 48
4.6 Examples of the three categories of annotation error. The red box represents the crowdsourced initialization, the green box represents the gold-standard initialization. . 49
4.7 Calculating additional error exclusively on valid frames produces a more meaningful evaluation metric. Despite the candidate initialization (red) tracking a different object than the gold-standard initialization (yellow), additional error is low when all 174 frames are used, but high if only valid frames (frames 0-50) are used. 50
4.8 The IoU of the crowdsourced initialization with the gold-standard initialization is not the sole determining factor in an initialization's performance, as evidenced by successful initializations with poor performance. 51
4.9 The mean additional error for all methods and coverages. Shaded region represents one standard error. 55
4.10 The RMAE under the oracle assumption (*Naive Replacement*). Note the minima near 85% coverage. 55
4.11 ARMAE of various smart replacement methods, where the deferral and aggregation functions are the same. 56
4.12 RMAE-Coverage curves for the case where the deferral and aggregation functions are the same. 56
4.13 RMAE-Coverage curves for the case where tracker confidence is used as the aggregation function. 57

5.1 Annotating pose estimation is a challenging task for crowd workers, yet drawing bounding boxes (yellow) and length, width, and height lines (red, green, and blue) is straightforward. We propose a method that enables this by intelligently aggregating across annotators and video frames. 59

5.2 A small pixel error in 2D can be amplified in 3D, resulting in a severe position error. The vehicle image on the left shows a crowdsourced *height entry* dimension line annotation (in red) and the corresponding ground truth (in green). The z-dimension estimate can be calculated from the focal length and the object's actual height, which was 721 pixels and 3.59 meters in our experiment, respectively. The three-pixel difference in dimension line leads to a 26-meter difference in 3D location. 61
5.3 Overview of Popup. From workers' dimension line annotation input and additional input of real-world dimension values of the target vehicle (looked up from an existing knowledge base), Popup estimates the position and orientation of the target vehicle in 3D. 61
5.4 Step-by-step instructions with good and bad examples are provided. 62
5.5 Interactive Web UI that crowd workers can use to create, adjust, erase, and redraw *length, width* and *height* lines. 63
5.6 Perceptual distance calculation. The distances (arrows) between endpoints (grey dots of the red line) of an annotation (red line) and corresponding projected hypothesis 3D line pairs (orange, green, blue, pink) are calculated. The distances corresponding to the best-fitting 3D line pair are used to calculate probability. 65
5.7 Example frames where more than three out of five workers self-filtered. The cases include limited side view, occlusion, and low resolution. 67
5.8 Results of filtering annotations and dimension lines. 68
5.9 Estimation error of our baseline and proposed method. 70
5.10 State estimation error of baseline vs. particle filtering without inter-frame referencing . 71
5.11 State estimation error of particle filtering without vs. with inter-frame referencing . . . 72

6.1 Our proposed aggregation function quickly combines complementary information to achieve higher certainty and accuracy than previous methods such as taking the mean of two outputs or selecting the better output (Smart Replacement). Target object boxed in blue. Original image cropped vertically for space. View in color. 77
6.2 Marginal plots showing the effect of the DDC (left) and DR (right) on the VOT (top) and Referring Expression Comprehension (bottom three) applications. Shaded area represents one standard error across 100 trials. View in color. 82
6.3 Effect of perturbing text embeddings on error. 84

7.1 Two notable shortcomings caused by the collection procedure of the RefCOCO dataset are (a) the ability to specify clicks instead of objects and (b) the common practice of backchannel communcation. The target bounding box is shown in pink. 87
7.2 Under many data collection methods , annotators are rewarded for guessing, leading to ambiguous phrases. 87

7.3 The four screens in our interface. The user begins in the *Initial Screen* and is tasked with cropping the object in the green box. After entering text on the initial screen, the AI may choose to defer or infer. If the AI chooses to defer, the user is asked to provide another input on the *Deferred Inference* screen. After inference, the user is presented with either the *Correct Inference* screen or the *Incorrect Inference* screen. In both cases, the removed region is darkened. Indicated regions shown below images. The color of region (D) depends on whether the inference was correct. Inputs for correct and incorrect inferences were *three-seater sofa* and *far right sofa*, which were provided by participants to identify the cropped objects. 92
7.4 Relationship between performance measures—error (A) and deferral rate (B)—and user satisfaction. Error is binned at intervals of 10%. Error bars represent one standard deviation. 94
7.5 The probability of the first n samples being different from the next half of the remaining samples (pink line) and the two halves of the remaining samples after n being different (blue line). When the first condition is true and the second is false, represented by the blue vertical line, the users' mental models have settled. 95
7.6 The relationship between task number and deferral score, probability of error, and expression length. Mean across the five adjacent task numbers shown in pink. Burn-in period shown with a dashed vertical line. Shaded area is one standard error. 96
7.7 Kernel Density Estimate plots of deferral scores frequency for four different users. Despite the pink and blue users having the same overall accuracy, both pairs are visibly and statistically ($p < 0.05$ by Mann-Whitney U) different. Kernel Bandwidth set by Scott's rule. 97
7.8 The mutual information conditioned on individual users (pink lines) superimposed on a distribution of unconditioned mutual information (blue histogram). The green line represents the p ¡ 0.05 significance threshold. 99

A.1 Please draw a bounding box around all parts of the person in the white shirt, but not their backpack. 110
A.2 Please draw a bounding box around the head and helmet of the bicyclist. 110
A.3 Please draw a bounding box around all parts of the skier. 111
A.4 Please draw a bounding box around the head of the tiger toy. 111
A.5 Please draw a bounding box around the head and neck (above the shoulders) of the person. 111
A.6 Please draw a bounding box around the white car following the black van. 111
A.7 Please draw a bounding box around the helmet of number 59 on the blue team. 112
A.8 Please draw a bounding box around the circuit board connected to 3 wires. 112
A.9 Please draw a bounding box around the head of the person holding the trophy. 112
A.10 Please draw a bounding box around all parts of the person in the far right of the image 112
A.11 Please draw a bounding box around the black pickup truck with its brake lights off. . . 113
A.12 Please draw a bounding box around all parts of the person, excluding their front leg and foot. 113
A.13 Please draw a bounding box around the white sports utility vehicle (SUV). 113
A.14 Please draw a bounding box around all parts of the female figure skater. 113

A.15 Please draw a bounding box around all parts of the male figure skater, except the hidden arm. . . . 114
A.16 Please draw a bounding box around the white car. . . . 114
A.17 Please draw a bounding box around all parts of the person, except the front foot. . . . 114
A.18 Please draw a bounding box around all parts of the dancer, excluding their hands and feet. 114
A.19 Please draw a bounding box around the box which holds the pencils, but not the pencils themselves. . . . 115
A.20 Please draw a bounding box around the parts of the large bird that are covered in feathers. 115
A.21 Please draw a bounding box around the body (not the wings or legs) of the rightmost bird. 115
A.22 Please draw a bounding box around the Clif bar. . . . 115
A.23 Please draw a bounding box around the head of the person standing up. . . . 115
A.24 Please draw a bounding box around the head of the person. . . . 115
A.25 Please draw a bounding box around all parts of the figure skater on the left, except their hands. . . . 116
A.26 Please draw a bounding box around the animal toy hanging from the string. . . . 116
A.27 Please draw a bounding box around all parts of the person dressed in all black, except their back leg. . . . 116
A.28 Please draw a bounding box around all parts of the person in the black shirt. . . . 116
A.29 Please draw a bounding box around the head of the guitarist closest to the pianist. . . . 116
A.30 Please draw a bounding box around all parts of the sprinter in lane 4. . . . 116
A.31 Please draw a bounding box around all parts of the doll except its hands. . . . 117
A.32 Please draw a bounding box around the white car. . . . 117
A.33 Please draw a bounding box around the body and head of the fish statue. . . . 117
A.34 Please draw a bounding box around the light-colored car on the left. . . . 117
A.35 Please draw a bounding box around the minivan. . . . 117
A.36 Please draw a bounding box around all parts of the pole vaulter. . . . 117
A.37 Please draw a bounding box around the silver car. . . . 118
A.38 Please draw a bounding box around all parts of the diver, excluding their arms. . . . 118
A.39 Please draw a bounding box around the person's head. . . . 118
A.40 Please draw a bounding box around all parts of the panda. . . . 118
A.41 Please draw a bounding box around the person's head. . . . 118
A.42 Please draw a bounding box around the head of the person in the trenchcoat. . . . 118
A.43 Please draw a bounding box around the body of the white suv. . . . 119
A.44 Please draw a bounding box around the person's face. . . . 119
A.45 Please draw a bounding box around the character on the poster. . . . 119
A.46 Please draw a bounding box around all parts of the person. . . . 119
A.47 Please draw a bounding box around all parts of the person. . . . 119
A.48 Please draw a bounding box around all parts of the person with the bag. . . . 119
A.49 Please draw a bounding box around all parts of the person closest to the crosswalk. . . . 120
A.50 Please draw a bounding box around all parts of the person. . . . 120
A.51 Please draw a bounding box around all parts of the person in the street. . . . 121
A.52 Please draw a bounding box around all parts of the person. . . . 121
A.53 Please draw a bounding box around the head of the surfer. . . . 121
A.54 Please draw a bounding box around the athlete wearing green surrounded by athletes wearing white, from the top of their head to their knees, excluding their arms. . . . 121

A.55 Please draw a bounding box around all parts of the sprinter in lane 4, except their arms. 122
A.56 Please draw a bounding box around the head of the kite surfer. 122
A.57 Please draw a bounding box around the box labeled “PENTAX” 122
A.58 Please draw a bounding box around the person’s head. 122
A.59 Please draw a bounding box around the toy cat. 122
A.60 Please draw a bounding box around the person’s face. 122
A.61 Please draw a bounding box around all parts of the person in the sleeveless shirt. . . . 123
A.62 Please draw a bounding box around all parts of the person in the white shirt. 123
A.63 Please draw a bounding box around the coupon. 123
A.64 Please draw a bounding box around the person’s face. 123
A.65 Please draw a bounding box around the toy tiger’s head. 124
A.66 Please draw a bounding box around all parts of the dancer, from the top of their head to their knees. 124
A.67 Please draw a bounding box around the person’s head. 124
A.68 Please draw a bounding box around the body and head (no ears or arms) of the toy. . . 124
A.69 Please draw a bounding box around all parts of the ice skater, excluding the straight leg. 125
A.70 Please draw a bounding box around the baby’s head. 125
A.71 Please draw a bounding box around the bicycle and its rider. 125
A.72 Please draw a bounding box around the bottle. 125
A.73 Please draw a bounding box around the can of diet coke. 126
A.74 Please draw a bounding box around the head of the person standing up. 126
A.75 Please draw a bounding box around all parts of the two people crossing the street together, except the back foot. 126
A.76 Please draw a bounding box around the person’s face. 126
A.77 Please draw a bounding box around all parts of the person in pink pants. 127
A.78 Please draw a bounding box around all parts of the person in light pants, except their arms. 127
A.79 Please draw a bounding box around the robot. 127
A.80 Please draw a bounding box around the head of the dog toy. 127
A.81 Please draw a bounding box around the head of the deer, from the base of its ear to the tip of its nose. 127
A.82 Please draw a bounding box around the singer, from the top of their head to the end of their jacket. 127
A.83 Please draw a bounding box around all parts of the singer in the white dress. 128
A.84 Please draw a bounding box around the red vehicle. 128
A.85 Please draw a bounding box around all parts of the figure skater, except their arms and feet. 128
A.86 Please draw a bounding box around all parts of the person, except the arm closer to the trash cans. 128
A.87 Please draw a bounding box around the box of tea. 128
A.88 Please draw a bounding box around the rubik’s cube. 128
A.89 Please draw a bounding box around all parts of the gymnast, except their arms. 129
A.90 Please draw a bounding box around the helmet of the player holding the ball. 129
A.91 Please draw a bounding box around the vehicle. 129
A.92 Please draw a bounding box around the motorcycle and its rider. 129

A.93 Please draw a bounding box around all parts of the pedestrian in blue pants. 129
A.94 Please draw a bounding box around the person's face. 129
A.95 Please draw a bounding box around the head of the red superhero. 130
A.96 Please draw a bounding box around the person's head. 130
A.97 Please draw a bounding box around the person's face. 130
A.98 Please draw a bounding box around the body (not head, legs, or tail) of the dog. 130
A.99 Please draw a bounding box around the car with its brake lights on. 130
A.100 Please draw a bounding box around the person's head. 130

B.1 Deferral rate against error for the VOT application and DDC=1. 131
B.2 Deferral Rate against Error for the VOT application and DDC=2. 131
B.3 Deferral rate against error for the VOT application and DDC=3. 132
B.4 Deferral rate against error for the VOT application and DDC=4. 132
B.5 Deferral rate against error for the VOT application and DDC=5. 132
B.6 Deferral rate against error for the VOT application and DDC=6. 132
B.7 Deferral rate against error for the VOT application and DDC=7. 132
B.8 Deferral rate against error for the VOT application and DDC=8. 132
B.9 Deferral rate against error for the VOT application and DDC=9. 133
B.10 Deferral rate against error for the VOT application and DDC=10. 133
B.11 Deferral rate against error for the val split of the RefExp application and DDC=1. . . . 134
B.12 Deferral rate against error for the val split of the RefExp application and DDC=2. . . . 134
B.13 Deferral rate against error for the val split of the RefExp application and DDC=3. . . . 134
B.14 Deferral rate against error for the val split of the RefExp application and DDC=4. . . . 134
B.15 Deferral rate against error for the val split of the RefExp application and DDC=5. . . . 135
B.16 Deferral rate against error for the val split of the RefExp application and DDC=6. . . . 135
B.17 Deferral rate against error for the val split of the RefExp application and DDC=7. . . . 135
B.18 Deferral rate against error for the val split of the RefExp application and DDC=8. . . . 135
B.19 Deferral rate against error for the val split of the RefExp application and DDC=9. . . . 136
B.20 Deferral rate against error for the val split of the RefExp application and DDC=10. . . 136
B.21 Deferral rate against error for the testA split of the RefExp application and DDC=1. . . 136
B.22 Deferral rate against error for the testA split of the RefExp application and DDC=2. . . 136
B.23 Deferral rate against error for the testA split of the RefExp application and DDC=3. . . 137
B.24 Deferral rate against error for the testA split of the RefExp application and DDC=4. . . 137
B.25 Deferral rate against error for the testA split of the RefExp application and DDC=5. . . 137
B.26 Deferral rate against error for the testA split of the RefExp application and DDC=6. . . 137
B.27 Deferral rate against error for the testA split of the RefExp application and DDC=7. . . 138
B.28 Deferral rate against error for the testA split of the RefExp application and DDC=8. . . 138
B.29 Deferral rate against error for the testA split of the RefExp application and DDC=9. . . 138
B.30 Deferral rate against error for the testA split of the RefExp application and DDC=10. . 138
B.31 Deferral rate against error for the testB split of the RefExp application and DDC=1. . . 139
B.32 Deferral rate against error for the testB split of the RefExp application and DDC=2. . . 139
B.33 Deferral rate against error for the testB split of the RefExp application and DDC=3. . . 139
B.34 Deferral rate against error for the testB split of the RefExp application and DDC=4. . . 139
B.35 Deferral rate against error for the testB split of the RefExp application and DDC=5. . . 140
B.36 Deferral rate against error for the testB split of the RefExp application and DDC=6. . . 140

B.37 Deferral rate against error for the testB split of the RefExp application and DDC=7. . . 140
B.38 Deferral rate against error for the testB split of the RefExp application and DDC=8. . . 140
B.39 Deferral rate against error for the testB split of the RefExp application and DDC=9. . . 141
B.40 Deferral rate against error for the testB split of the RefExp application and DDC=10. . 141

LIST OF APPENDICES

A Crowd Queries Used for VOT 110

B DR-Error Plots Conditioned on DDC 131

ABSTRACT

While deep learning problems are often motivated as enabling technologies for human-computer interaction—a robot, for example, must align natural language referents and sensor readings to operate in a human world—assumptions of these works make them poorly suited to real-world human interaction. Specifically, evaluation typically assumes that humans are oracles that provide semantically correct and unambiguous information, and that all such information is equally useful. While this is enforced in controlled experiments via carefully curated datasets, models operating in the wild will need to compensate for the fact that humans are *hazy oracles* that may provide information that is incorrect, ambiguous, or misaligned with the features learned by the model. For example: given a choice of three mugs, a robot would not be able to satisfy a request to retrieve *the mug*, but would likely be able to retrieve *the orange mug*.

A natural question follows: how can we use deep learning models trained via the oracle assumption with hazy humans? We answer this question via a method we call *deferred inference*, which allows models trained via supervised learning to solicit and integrate additional information from the human when necessary. Deferred inference begins with a method for determining if the model should defer inference and wait until more human-provided information is provided. Past work has generally simplified this into one of two questions: *is the human-provided information correct?* or *is the output correct?* However, we find that these approaches are insufficient due to the complex relationship between human inputs, sensor readings, and deep models: low-quality human-provided information may not cause error, while high-quality human-provided information may not correct it. To address the misalignment between input and output error, we introduce Dual-loss Additional Error Regression, or DAER, a method that successfully locates instances where a new human input can reduce error.

Although introduction of such an effective *deferral function* is necessary to optimize the trade-off between human effort and error, we must additionally consider that the deferral response is also subject to the effects of hazy oracles. For this reason, we must not only consider how to find error caused by human input, but also how to integrate deferral responses and measure the performance of the team. For this, we introduce *aggregation functions* that allow us to integrate information across multiple inferences and a novel evaluation framework that measures the trade-off between error and additional human effort. Through this evaluation, we show that we can reduce error by up

to 48% under a reasonable level of human effort without any changes to training or architecture.

Last, we consider how shifting from dataset-based evaluation to an individual human affects deferred inference. Specifically, while crowdsourced datasets work well for rapid implementation and evaluation of deferral and aggregation functions, they do not accurately model human-computer interaction: the mechanisms used to procure high-quality data most likely cause shifts in the distribution, and the failure to track the inputs of individual annotators makes the tacit assumptions that all humans are the same and inputs do not change over time or deferral depth. Through a human-centered experiment, we find that these assumptions are not true: an ideal deferral function must be calibrated for a specific user, users learn the model over time, and the deferral response is likely to be of lower quality than the initial query. Further, we show that despite the shortcomings of crowdsourced data, our deferral method does still reduce error.

While deep learned models have been proposed for many applications that require cooperation between humans and computers, deploying models that were trained and evaluated across carefully curated datasets remains a challenge due to the hazy nature of human inputs. In this dissertation, we propose deferred inference as a method for addressing this challenge while respecting the paradigm of supervised training. By demonstrating components of deferred inference on four disparate problems, we provide meaningful insights into its challenges, benefits, and generalizability that motivate and lay the foundation for the eventual deployment of deep-learned human-in-the-loop models in the wild.

CHAPTER 1

Introduction

Deep neural networks have demonstrated unprecedented performance on applications previously considered the realm of science fiction. Commercially available deep-learned models are capable of performing tasks such as removing objects from photos [3], drawing images based on text inputs [4], and producing high quality answers to linguistic requests [5]. The most important of these potential applications promises to improve lives by allowing humans and computers to interact in a complex, multimodal world: Visual Question Answering [6] is an exciting assistive technology for the visually impaired [7] and the ability to localize an object based on language—Referring Expression Comprehension [8]—will be required if, for example, human support robots [9] are expected to ameliorate the effects of the predicted shortfall of 151,000 elder care workers in 2030 [10].

Despite this enormous potential, deep learned models are typically unsuitable for human interaction without additional modification. Consider the scenario shown in Figure 1.1: a human support robot that is designed to perform tasks for the elderly or those with mobility impairments [9].

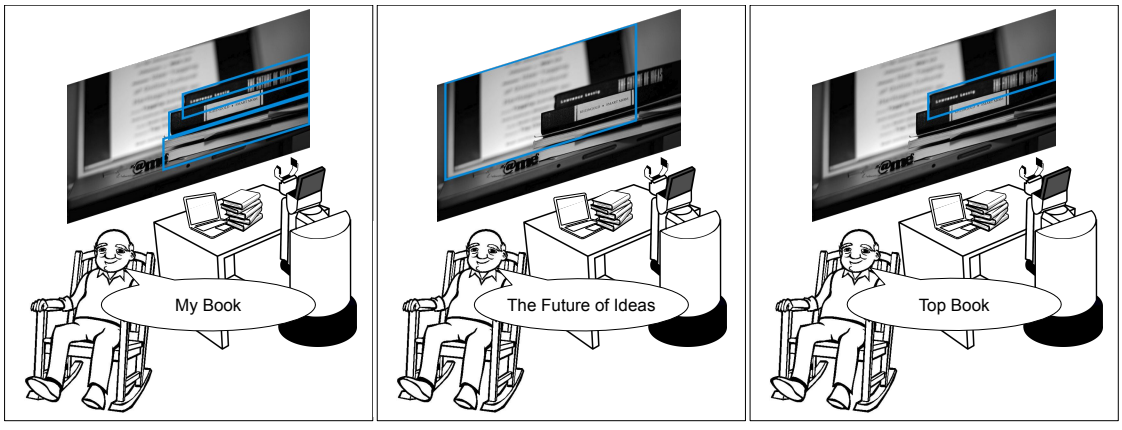


Figure 1.1: While deep learned models perform inference based on a single piece of human-provided information that is assumed perfect, humans input may be (A) semantically ambiguous or (B) unambiguous but misinterpreted in addition to being (C) correctly interpreted. Because of this, using modern deep-learned models as-is will result in unnecessary error. (A) is purely illustrative, while (B) and (C) are real-world responses of the UNITER [1] model to RefCOCO [2] queries.

The human is looking for his book, *The Future of Ideas*, which is the top book in the stack located on his desk. There is no screen-based interface and it is prohibitively difficult for the human to get the book himself, so he must ask the robot to get his book using natural language.

Since the human is unable to perform the action himself, the model must be able to detect and respond to potential failures. While it is permissible for Conversational Virtual Assistants (CVAs) such as Alexa to decline inference or occasionally return in incorrect answer, such a method is unacceptable for a robot: if the robot chooses not to perform the action the human will not get his book, while if the robot performs an incorrect action it, at best, imposes significant latency as the human waits for the wrong object and, at worst, fails catastrophically by breaking fragile objects—such as a laptop (Figure 1.1)—or returning incorrect answers to important visual questions like oven temperature [11].

Despite the potential consequences of model misbehavior, deep learned models typically do not have a way of detecting and addressing such failures during inference. Instead, they typically make the *oracle assumption*: all human-provided information is semantically correct and unambiguous, and all such information is equally useful. For this reason, the human is given the chance to provide one request that the model must use during inference.

In contrast, we observe that humans are *hazy oracles*: information they provide may be incorrect, ambiguous, or misaligned with the features learned by the model. An incorrect human input may be factually incorrect or based on a false premise [12] and is likely to cause error if not detected, while ambiguous information may permit multiple answers (Figure 1.1-A). If the human input is correct and unambiguous, the model may still produce incorrect inferences because its learned features do not match the human's expectations. For example, this particular model can't read, so while the request *The Future of Ideas* (Figure 1.1-B) is unambiguous, the robot will still perform an incorrect action.

1.1 Supervised Training and The Oracle Assumption

Under the oracle assumption, the human provides a single input and the model provides a single output. If the input provided by the human is ambiguous either in truth or because it is not aligned with the training data, the model provides its best guess as to the answer regardless of its confidence. While this work develops inference-time strategies that compensate for the shortcomings of the oracle assumption when interfacing with hazy human oracles, we discuss here the training process for deep learned models to demonstrate the origin—and necessity—of the oracle assumption. To be clear, *the oracle assumption is not simply an artifact of choices within architecture design or data collection, but fundamentally entwined with the paradigm of supervised learning.*

In supervised training, shown in Figure 1.2, we begin with a dataset that consists of a collection of inputs and a corresponding target values. At every step, a random set of inputs and outputs are combined into a batch. This batch is presented to the model, which produces a guess as to the label of every input (*Infer*). These guesses are then compared to the target labels (*Evaluate*) and the model parameters are updated based on the result of this comparison (*Update Model*). This process is repeated over all examples across many epochs, until some stopping criteria has been met (*e.g.,* error is no longer decreasing).

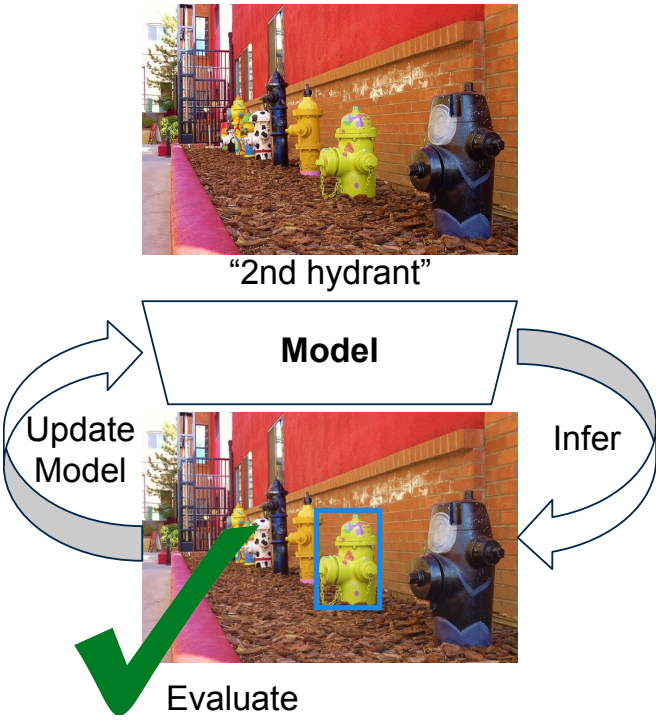


Figure 1.2: By enforcing the framework of infer → evaluate → update model, supervised learning is limited to performing one inference at a time.

We established previously three components of the oracle assumption: i) every human-provided input is correct, ii) every human input is unambiguous, and iii) all correct and unambiguous human-provided information is equally useful. If the human-provided information is incorrect during this training process, there is either no target with which to train the model (*e.g.,* if you ask *what color is the cat's tie?* there is no answer if there is no cat wearing a tie [12]), or you negate the benefit of human-provided information by training the model to ignore it. If the human-provided information is ambiguous, there is no way to be sure of the target—at best the model indicates its uncertainty via a calibrated output, but it is also likely that the model will return a confidently incorrect answer [13]. For these reasons, dataset procurement uses methods such as two-player games [2], [6], [14] or review of web data [15] to ensure high-quality human inputs. The third aspect of the oracle assumption is a consequence of the standard deep learning formulation: since every input has a one-to-one mapping to an output, measuring the correctness of every output is the natural evaluation.

Different facets of these shortcomings have been addressed in previous work, primarily through three different approaches: work in *input-space optimization* attempts to improve performance by locating low-quality human inputs explicitly during inference, some works attempt to *remove model blind spots* which would make it acceptable to only analyze input-space quality, while works in *selective prediction* attempt to locate incorrect inferences to mitigate their effects.

Input-Space Optimization Input space optimization can be used during inference to locate and compensate for human inputs that are semantically incorrect or ambiguous. Some works do this by treating the detection of semantically poor human inputs as a classification problem: Ray *et al.* [16] and Mahendru *et al.* [12] propose novel datasets and architectures to determine whether the premise of a visual question is correct, while Bhattacharya *et al.* [17] use a pre-defined taxonomy to annotate why crowd workers disagree on the answer to visual questions posed by visually-impaired users, then propose a model to make predictions within this taxonomy.

In other cases, particularly in the crowdsourcing domain, semantic quality is increased by changing the interface or more effectively aggregating multiple inputs. As an example of the former, Papadopoulos *et al.* [18] show that quality of bounding box annotations can be improved by clicking on extreme points instead of drawing bounding boxes, while for the latter Song *et al.* [19] propose aggregating multiple types of annotations for producing 3D annotations from 2D images and Bernstein *et al.* [20] propose a word processor editing pipeline with several complementary revision steps. Jain & Grauman [21] split the difference, simplifying the process of segmentation by clicking object edges across multiple steps instead of directly coloring in the segmentation.

Although the aforementioned works are meaningful, there are notable shortcomings in their approaches. First and foremost, none of these works consider potential gaps in the model's understanding of the world, which we show in this work is critical. Next, aggregating in the input space is tractable for categorical or continuous real-valued inputs, but impossible in large input spaces such as language. Last, pre-defined categories of incorrect are inherently limited, as will be any supervised training of the same.

Removing Model Blind Spots If most failures occur due to the human input being misaligned with the model's learned features, a straightforward way to reduce error is by improving the deep model's architecture, dataset, or training procedure. Architectural improvements include progressions such as the transition in referring expression comprehension from language-based models [22] to visiolinguistic transformers with object detectors [1] to integrating the detector into the transformer architecture [23], [24]. Dataset-based improvements aim to collect more balanced data, therefore removing the effectiveness of—and dependence on—undesired priors [25]. When training, many works alter the objective to address specific flaws in the model: for example, Cadene and Dancette [26] focus on the ability to remove unimodal biases (*e.g.,*, the answer to *what color is the banana?* will usually be *yellow*, even without the information in the image), and some works [27], [28] focus on the shortcoming of reading text from images mentioned in our motivating example.

While one can argue that the goal of machine learning is to produce a task model that does not have any gaps in its knowledge, we do not consider this a reasonable goal: not only do we not expect

a human to communicate flawlessly, but the space of potential inputs is large—can we reasonably expect a training set to contain all potential images, utterances, and combinations thereof? Further such a hypothetical dataset would still be subject to problems related to class imbalance—the tendency for deep-learned models to ignore classes that occur infrequently [29]. Although class imbalance has been addressed in many ways, such as importance weighting [30], [31], grouping into subcategories [32], or training specifically on hard examples [33], the stubbornness of this problem demonstrates the challenge of a perfect classifier. Similarly, adversarial examples [34]—inputs that look correct to the human but cause model failures—are addressed with approaches such as robust training [35], explicitly training rejection of adversarial examples [36]–[40], and model distillation [41], but remain indicative of how difficult it is to create a perfect decision boundary in high dimensions.

Selective Prediction In selective prediction [42]–[44] the goal is to only perform inference when the model is confident in its output. The most commonly stated assumption under these scenarios is that the correct answer will be provided by the human when confidence is low. While this works well for applications such as second opinions in the medical space [45], [46], it is impractical in cases such as our motivating example: the robot is there to assist the human, since he can not retrieve the object himself. In the case of a visually impaired user asking for help with a visual question [11], the user cannot determine the answer without assistance, and many conversational virtual assistants do not have an alternate input mode. If the user can't perform the task without assistance, and the model has no oracle to appeal to, we are left with the unsatisfying [47] approach of simply reporting an error and requiring the human to start over.

1.2 Deferred Inference

Our discussion of the oracle assumption, how it is used, and its shortcomings leads to what we call the *hazy oracle assumption*, and our formulation of the deferred inference problem [48]:

> An automated agent is asked to perform a task, such as cropping an image based on a language request or tracking an object through a video. This agent has some probability of solving the task on its own, but may also defer to a hazy oracle that can provide additional information at some cost. The information provided by the hazy oracle may be semantically ambiguous either in truth (*e.g.,* more than one output satisfies the request) or because it is mismatched with the features learned by the agent (*e.g.,* a deep learning model trained in English will not understand a Spanish query, regardless of the information it contains). With the goal of minimizing error subject to constraints on human effort or human effort subject to constraints on error, the agent must determine

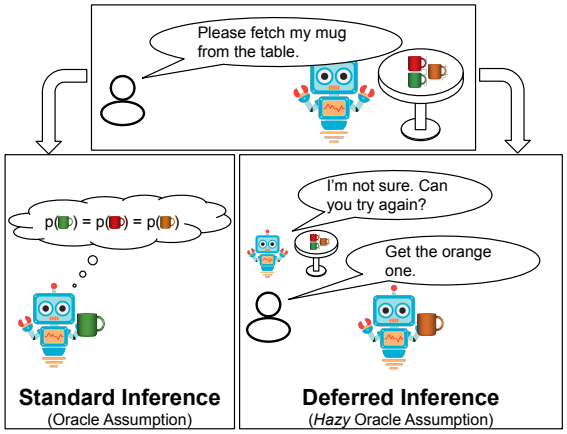


Figure 1.3: A simplified example of deferred inference with hazy oracles. The initial request is ambiguous, resulting in a two-in-three chance of an incorrect answer. By deferring, the robot obtains sufficient information to solve the task.

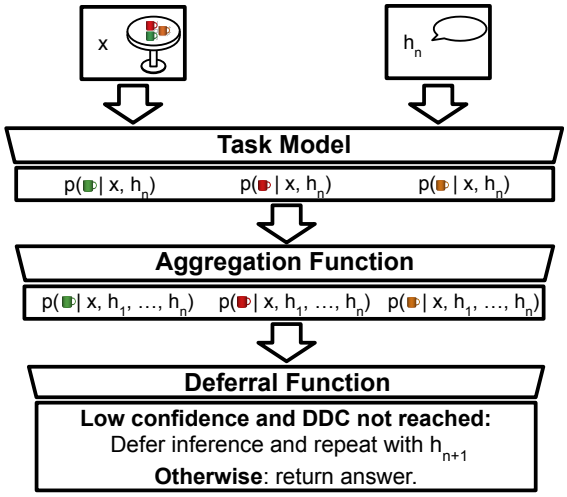


Figure 1.4: Our formulation of deferred inference, with the task shown in Figure 1.3 as illustration. x is the robot's perception and h_n is the human input at step n.

whether to defer its decision and request information from the hazy oracle. If the agent chooses to defer its decision, it must additionally determine how best to integrate the additional, potentially noisy, information provided.

We show an abstraction of such an agent in Figure 1.4, based on the motivating scenario shown in Figure 1.3. The abstraction consists of three parts: the *task model*, the *aggregation function* and the *deferral function*. We define these three components here as independent entities, but note that that is not necessarily true: a recurrent neural network with a deferral score similar to SelectiveNet [49] would execute all three functions in a single step.

The goal of our first component, the task model, is to integrate human-provided information with prior knowledge or sensor readings to produce an output. Throughout this work we use pre-existing task models from the literature to demonstrate that task models can be implemented and improved in their relevant sub-fields, then further enhanced via deferred inference. Task models are typically trained under the oracle assumption using a standard supervised learning method and architecture: visiolinguistic models such as UNITER [1] or ViLBERT [50] are trained for applications such as Referring Expression Comprehension [8] or Visual Question Answering [6], lightweight models are trained to enable hand gesture recognition [51], and models such as ToMP [52] or OSVOS [53] are trained to propagate human-provided information—bounding boxes or segmentations—across video frames. Throughout this work, we attempt to place as few assumptions as possible on the task model, since we seek to minimize the amount of additional development required to implement deferred inference on novel architectures. Specific assumptions are discussed in the relevant chapters.

		Referring Expression Comprehension		
	VOT	Val	TestA	TestB
No Deferral	0.329 ± 6.6e^{-4}	8.82 ± 0.03	8.27 ± 0.04	9.67 ± 0.04
Perfect Deferral	0.276	2.07	1.92	2.82
Abs. Improvement	0.053	6.75	6.35	6.85
% Improvement	16.11%	76.53%	76.78%	70.84%

Table 1.1: Error for the task model with No Deferral (Err @ 0) and Perfect Deferral on two applications. A perfect deferral method can reduce error by over 76%. Err @ 0 is reported as Mean and Standard Error of 100 trials. Full setup described in Section 2.1.1.

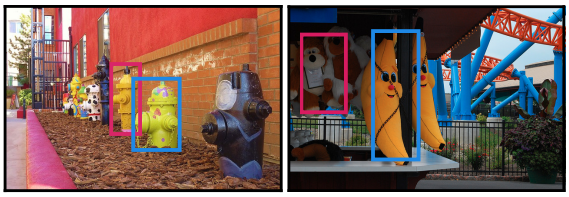


Figure 1.5: Whether a phrase is semantically ambiguous (left) or simply unclear to the model (right), a new human input can change an inference from incorrect (pink) to correct (blue). Best viewed in color.

Our second component, the aggregation function, accepts one or more predictions from the task model and produces a unified prediction. The structure of the aggregation function largely depends on the structure of the task model: if the task model produces outputs such as a discrete bounding box—common in the application of video object tracking [54]—the best aggregation function may be a method that simply picks the best output, while if the output is a softmax distribution—common in visual question answering [6] and referring expression comprehension [8]—the product of the distributions is likely to provide the best output by resolving ambiguities.

Last, the deferral function accepts the output of the aggregation function and determines whether or not the model needs to solicit additional information by assigning a *deferral score* to the output of the aggregation function and applying a threshold to this score. As we discuss in Chapter 3, the deferral function presents an interesting challenge in that it must not only defer when the output is likely to be incorrect—as in the selective prediction setting described above—but when the output is likely to be incorrect *and* a new human input is likely to correct it. We address this through a novel, trained method called Dual-loss Additional Error Regression (DAER), but also demonstrate more generally applicable deferral functions based on output distributions.

Motivation We motivate deferred inference through the applications of single-target video object tracking, where the goal is to propagate a first-frame bounding box through all subsequent frames, and referring expression comprehension, where the goal is to draw a bounding box around the object described by a text query. We show the benefit of a perfect deferred inference method—that is, one that is able to select the best human input from the dataset—quantitatively in Table 1.1: for the validation split of the referring expression comprehension task, *using the best human input can reduce error by over 76%*.

In Figure 1.5 we demonstrate the benefit qualitatively by showing four human inputs and their matching outputs on the application of referring expression comprehension. On the left we see the more intuitive case, where the first expression, *yellow fire hydrant*, can be reasonably thought

to refer to four objects, while *2nd hydrant* is mostly unambiguous.[1] On the right, we see a case where the referring expression *left banana* isn't truly ambiguous, but the model produces the wrong answer due to shortcomings in its understanding of language. Though the latter failure—a human input being outside of the model's understanding—is typically considered a shortcoming of the model, the fact that a new referring expression can successfully solve both tasks means that both applications would benefit from deferred inference.

1.3 Deferred Inference for Interaction

Having motivated deferred inference from a machine learning perspective, we situate our formulation in the interaction domain. Human-computer interaction operates across many assumption sets, and even perspectives framed as opposites often find commonality [55]. Despite this wide variety of assumption sets, our work embraces two assumptions that, when taken in tandem, are unique among interaction approaches: 1) the human cannot operate in the output space and 2) inference is performed via unmodified task models.

Our first assumption is motivated by the fact that the majority of intelligent interfaces in use today require some form of manipulation in the output space. For example, AI advising [56] and annotation [57] require the human to make the final decision, selective prediction [58] and learning to defer [44] require the human to provide the correct answer if the AI is uncertain, and simple applications such as search allow the human to review answers and revise queries. While effective, this is impractical in some important use-cases: if visual question answering models are used to assist the visually impaired [7] the human cannot provide or confirm the correct answer, an elderly individual instructing a robot to retrieve an object [59] may not be physically capable of getting it without assistance, and conversational virtual assistants such as Alexa [60], do not have a secondary input mode that can be leveraged.

Our second assumption motivates the use of a reformulation-based deferral. While this is not inherently necessary for deferred inference—some works have proposed generative text models for similar problems [61], [62]—this assumption has a number of implementation advantages. First, since we do not have to create novel architectures, we can easily extend our methods to novel task models across a variety of applications, which is important in the rapidly evolving deep learning space. Additionally, generating relevant follow-up queries is difficult even in constrained spaces. In addition to generative models generally being considered more challenging than discriminative models, such methods either utilize a particularly accomodating problem space—such as in Referring Expression Comprehension [61]—or datasets that, while usable for producing follow-up

[1] While data collection was designed to produce unambiguous referring expressions, we find a non-negligible number of semantically ambiguous examples. This is likely due to the requirement that annotators make a guess for every expression, and is discussed further in Chapter 7.

questions, do not allow for true introspection [62]. Additionally, meaningful follow-up queries may not exist for applications with non-linguistic human inputs such as Visual Object Tracking [54] and Keypoint-Conditioned Viewpoint Estimation [63].

1.4 Contributions

The contributions of this dissertation relate to introducing and motivating a formulation for deferred inference with hazy oracles, then addressing the shortcomings of current work via approaches that simultaneously consider the semantic quality of the human input, the model's response to specific inputs, and the combined behavior of the human and model across deferrals. To do this, we introduce novel evaluations and solutions that consider both the initial input and the deferral response via four contributions: we learn a deferral function that effectively isolates error caused by the human from fixed error in multimodal inference, consider how best to compensate for the fact that deferral responses are hazy via aggregation functions, develop a general formulation for both evaluation and implementation of deferred inference, and shift the analysis from crowdsourced data to individual user responses.

1.4.1 Learning an Ideal Deferral Function

Due to the cost of acquiring additional information—measured both in dollars and user satisfaction—it is important to defer inference only when it would be helpful. While previous works have deferred or declined inference by locating cases where the human input [12], [16], [17] or the output [42], [49], [58], [64], [65] is incorrect, we show that these approaches are insufficient: the counterintuitive behaviors of deep networks mean that incorrect human inputs may result in performance that is the same or better than the corresponding correct input, and imperfect outputs may not be improved by acquiring a new human input. To address this, we introduce a novel evaluation method, centered around the metric of additional error, that measures not whether the answer is correct, but whether a new human input would be helpful. We use this to evaluate our novel method of Dual-loss Additional Error Regression (DAER), and find that considering the human input in tandem with the model allows us to outperform selective-prediction and input-space optimization inspired baselines, even when given knowledge of the ideal human input.

1.4.2 Development of Aggregation Functions

In addition to deferring inference appropriately, it is important to consider the best course of action after deferral has occurred. We first show that if we naively accept the deferral response—similar to the reformulation strategy of conversational virtual assistants [66]—a local optimum is

reached: beyond a certain deferral rate, error and human effort increase simultaneously. In other words, the current paradigm is such that the more often a rephrase is requested, the worse the overall performance becomes.

To address this shortcoming, we introduce two aggregation functions. The first, *smart replacement*, performs inference using the human input that it believes provides the highest-quality output. We demonstrate that this straightforward method allows us to improve performance effectively on any task for which a confidence measure can be derived. Although it is almost universally generalizable, smart replacement does not allow us to utilize complementary, ambiguous, information. To address this, we perform experiments on the applications of 3D video pose estimation, video object tracking, and referring expression comprehension that demonstrate the benefit of probabilistic aggregations of predictions produced by human inputs. Specifically, we are able reduce error by up to 48% by integrating multiple ambiguous human inputs to produce a high-quality output.

1.4.3 A general formulation and evaluation for deferred inference

While a number of deferred inference methods have been proposed in isolation [61], [62], [67], [68] such works have thus far been unrepeatable due to their reliance on human experiments, non-standard datasets, and deferral criteria determined a-priori that may not produce optimal overall performance. To address this, we introduce a general formulation for deferred inference. This formulation formalizes the concepts of task model, deferral function, and aggregation function, as well as an evaluation method for deferred inference that simultaneously considers the error, deferral rate, and maximum deferral depth. We demonstrate the importance of our evaluation by showing that changing the constraint set changes the best methods, even though such constraints are set arbitrarily in previous works.

1.4.4 Human-centered deferred inference

Although aggregate evaluation of deferred inference on crowdsourced data can be informative, it does not consider specifically when deferral should occur. In order to do this, we need to consider the individual with whom the model is interacting. This individual is likely to have unique characteristics—such as particular dialects—that must be considered, and is likely to systematically change the query after deferral. For this reason, we perform a user study on deferred inference motivated by a language-powered image cropping task. Through this, we demonstrate various aspects of human interaction with such a model—most notably that the deferral response will typically be less clear to the model—and show that it is necessary to consider the individual user when setting deferral criteria.

1.5 Thesis and Impact Statement

By allowing the human to provide additional information upon request, we can meaningfully improve both quantitative performance and qualitative user experience in a human-AI team. Previous work has considered detection of both low-quality inputs and low-quality outputs, but generally ignored what to do after these failures have been detected. In this work, we consider not only how to detect low-quality predictions, but also what to do afterwards: what is the effect of subsequent information, how do we integrate it with previous inferences, and how does it affect user experience?

By demonstrating evaluations and methods across many applications and architectures, we provide a simple pathway to using state-of-the-art deep models trained via supervised learning in practice. Such models, and therefore deferral methods, will be critical for impactful applications such as assistive visual question answering technologies [11] and service robots for elder care [9].

CHAPTER 2

Related Work

This dissertation focuses on problems in which an AI agent uses human input in tandem with world knowledge and sensor readings to produce a desired output. Such problems span a number of applications and strategies that we describe here. After describing a selection of these problems—both in general and in depth for our exemplar applications—we discuss various strategies used when collaborating with hazy oracles.

2.1 Applications Using Hazy Oracles

A number of works require input from hazy oracles—humans—in order to accomplish their goals. Here, we describe a number of applications that use hazy oracles in tandem with automated procedures. Broadly speaking, we group these works into three categories with different assumptions: techniques for dataset procurement and enhancing human abilities allows the user to directly review putative outputs, while accessibility methods typically do not have this option.

Dataset Procurement Hazy oracles are often used to crowdsource datasets, particularly when the gold-standard information is hard to obtain without computational assistance. For example, a task like 3D scene reconstruction is difficult and slow to do manually, but can be made tractable through intelligent aggregation techniques [19], [69] or deep learning [63]. Similarly, segmentation—while not a cognitively difficult task—requires attention to detail and fine motor skills, leading to a variety of collaborative assistants: Uijlings *et al.* propose a collaborative assistant for panoptic segmentation that allows a human and computer to take turns correcting a putative output, Jain & Grauman [21] allow the human to reduce the hypothesis space by clicking on the target's output, while Song *et al.* [70] aggregate imperfect annotations from different, simple, tools to produce a high-quality answer. We provide more details on crowdsourcing techniques in Section 2.2.2.

Enhancing Human Ability A well-publicized capability of deep learning is the production of novel visual art: applications such as generating faces [71], [72], style transfer [73], [74] and domain transfer [75] that previously required many hours of work by a trained artist could now be done in seconds by anyone with a web browser. Many of these methods interact with hazy oracles: inpainting techniques for both video [76] and photos (*e.g.,* Google's *Magic Eraser* [3]) use the human to outline an object that should be removed from the image or video, and Dall-E shows a remarkable ability to produce high-quality drawings based on a text query [4].

Although image manipulation tasks have dominated the public discussion of machine learning, these are not the only ways that deep learned models work to enhance human abilities: Machine translation [77] accepts a natural language human input and produces a translation that can be understood without requiring employment of a highly educated translator. Although the GPT-3 architecture [78] has been the subject of controversy and discussion on the ethics of AI [79], [80], it has proven useful in applications as diverse as feedback summarization, semantic search, and automated dialog [5]. Perhaps more impactful is its ability to power the OpenAI codex [81], which produces executable source code from natural language queries, and has even been used to allow robot control via motion primitives [82].

Not only do generative models have the ability to create art or perform translations from raw human inputs, deep models can help augment the performance of *e.g.,* medical professionals who need to find previous examples to be able to come to a diagnosis. For eample, Cai *et al.* [56] propose a human centered visual search tool for viewing tissue samples and Caruana *et al.* [83] propose an explainable method for predicting hospital readmission for pneumonia. Although all of the above are important tools, we primarily focus on situations where the human does not have access to the output space, and the model must therefore make the final inference.

Accessibility Technologies HIL inference may also be used to increase accessibility. For example: the VizWiz dataset [7] was motivated by allowing visually impaired individuals to operate in a visual world (though the work that originated the VQA problem [6] used this as a motivation, users were asked to *stump a smart robot*, not perform meaningful accessibility tasks). This particular application has spawned various approaches, from early work in crowdsourcing answers to questions posed by visually impaired users [11] to transformer architectures [22] with vision-language pre-training [1], [24], [84]. Other approaches to HIL accessibility include improved image captioning techniques, such as multi-layered exploration [85], that produce varying levels of captions based on human inputs.

Work in robotics also promises to improve the lives of the elderly or physically impaired [9], [86]–[88], particularly when the significant predicted shortfall of elder care workers is considered [10]. As in VQA, these applications require not only an ability to understand the human's commands,

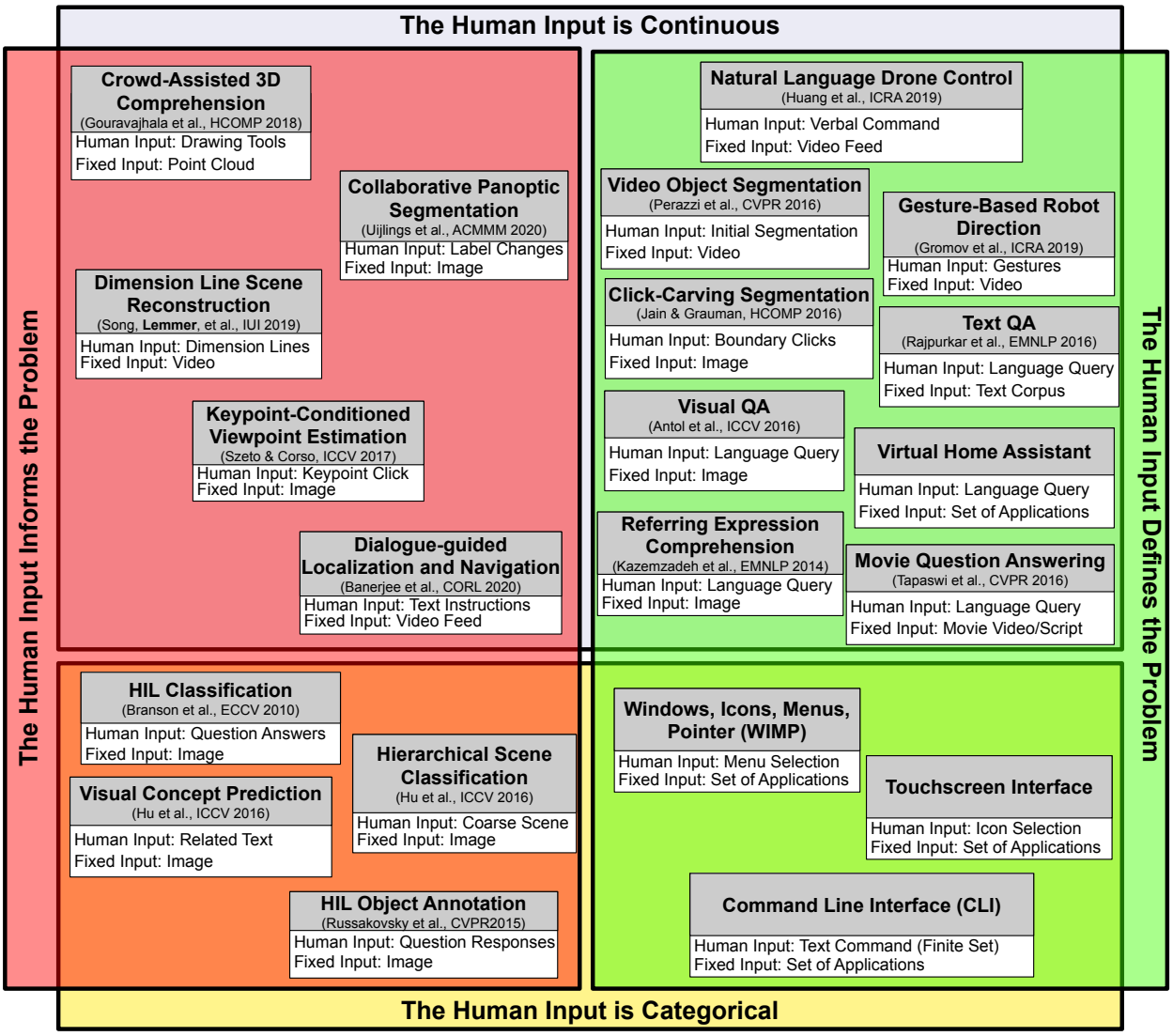


Figure 2.1: A collection of problems using human inputs, displayed in a Venn diagram of input characteristics.

but also an ability understand them in the context of the robot's knowledge and sensor readings. Gouravajhala *et al.* [89] consider these applications from a crowdsourcing perspective, focusing on point clouds, while Lasecki *et al.* [90] use crowdsourcing to allow an inexpensive robot to understand natural language commands. Due to the intuitive nature of the referring expression comprehension task, many works apply it directly to robots [61], [67], [68], [91], [92]. Similarly, vision-language navigation [14], [93]–[96] combines human input with sensor data to allow a human operator to intuitively send a robot to a desired location.

A Useful Taxonomy It is useful to consider human inputs across two axes: whether they are categorical or continuous, and whether they inform or define the problem. We categorize a number of HIL problems along these axes in Figure 2.1.

If the human input is categorical—for example, in Hierarchical Scene Classification [98], HIL classification [97], and Multi-Label Image Annotation [104]—it can be easily defined as correct or incorrect. Since it can easily be defined as correct or incorrect, there exists a trivial answer to the question of whether or not error can be reduced via a new human input. That is, if the human input is correct, a second human input will not improve the answer regardless of the output confidence. [1] In cases where this is not true, it is less straightforward to determine whether or not a deferral can help. While some explicit definitions of an incorrect continuous input [12], [17] exist, they do not consider the model's understanding of the world, and therefore a "correct" human input could still be improved by a deferral. We discuss this in Chapter 4.

Similar to problems with categorical human inputs, problems where the human input is used to inform the model—such as keypoint-conditioned viewpoint estimation [63] and hierarchical scene classification [104] allow some shortcuts to be made when determining when to defer. Specifically, we can choose only to defer when the provided human information can change the output which, as we see in Chapter 3, is often not the case.

In the case of defining vs. informing the problem, the distinction is more straightforward: if a model could perform the task with a better-than-chance success rate without the human input, the human input informs the problem, otherwise the human input defines the problem. As an example, we can consider the tasks of visual question answering [6] and keypoint-conditioned viewpoint estimation [63]. In the former case, the task is undefined until the human input (the text question) is provided—the user can't reasonably expect the task model to provide the correct answer without it. However, in the latter case, the question of "what is the viewpoint?" is defined, and therefore can be answered, prior to the human input being given.

2.1.1 Exemplar HIL Applications

While there are many HIL problems, we perform our evaluations on four specific applications: Keypoint-Conditioned Viewpoint Estimation [63], Hierarchical Scene Classification [104], Single-Target Video Object Tracking [100], and Referring Expression Comprehension [2]. We describe them in detail here.

[1] While it is possible for an incorrect human input to provide a better result than a correct one, we assume that the user will attempt to provide the correct answer and can never expect the "better" result. We discuss this in Chapter 3.

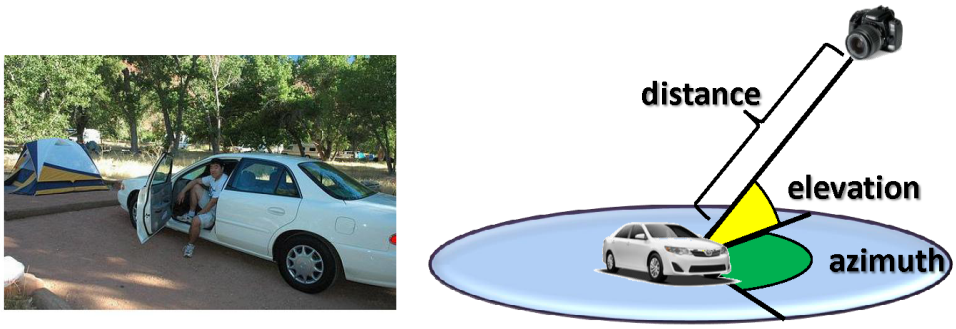


Figure 2.2: An illustration of the viewpoint estimation task. From [105].

Keypoint-Conditioned Viewpoint Estimation Keypoint-conditioned viewpoint estimation is an HIL extension of the monocular viewpoint estimation application. In monocular viewpoint estimation, the goal is to produce the rotation between the imaging axis and the (canonical) axis of a target object in terms of azimuth, elevation, and tilt. Early work used multiple images obtained via motion or a stereo pair as input to a closed-form [106] or least-squares [107] solution to the Perspective-n-Point (PnP) problem. As in many computer vision problems, the features used for PnP have shifted from handcrafted features such as SIFT [108], to deep representations [109].

In cases like ours where multiple camera views are not available, some works have attempted to resolve the ambiguity by placing priors on the object's shape. Broadly, these fall into two categories: deformable parts models (DPMs) and matching approaches. For DPMs, a prior is given in the form of parts and their relationships, and the solver penalizes potential solutions that deviate from the standard relationship. While these constraints were initially proposed for object detection in two dimensions [110], the concept extends easily to the 3D space, where models may be defined by a cuboid [111] or by semantic parts [105], [112], [113]. For matching approaches [114], [115], a detection method and similarity metric are used to locate and align the 3D model most similar to the pictured object.

Another line of work has abandoned the idea of strong priors and focused on training a deep neural network end-to-end. Initial work [116] trained solely on image datasets such as PASCAL3D+ [105], while later work [117] leveraged synthetic renders from the ShapeNet [118] dataset to improve performance. While these end-to-end approaches are attractive due to their simplicity and the general success of learned feature representations, these models generally struggle due to input issues such as occlusion and truncation [105], as well as an inability to compensate for the symmetry of many man-made objects [116]. Keypoint-conditioned viewpoint estimation was proposed by Szeto & Corso [63] to resolve these ambiguities and significantly improve performance by integrating the semantic understanding of humans—the ability to answer questions such as *where is the front-left tire?*—with the speed and numeric precision of DNNs.

Hierarchical Scene Classification Hierarchical scene classification is a human-in-the-loop extension of fine-grained visual classification (FGVC), loosely defined as any classification task where differences between classes are small. While this definition is subjective, there is rarely controversy as to whether a benchmark is fine or coarse: where CIFAR10 [119] asks the model to select between the 10 classes such as *dog* and *horse*, the Stanford Dogs dataset [120] asks the model to distinguish between a 120 breeds, including those as similar as the Boston Terrier and French Bulldog (Figure 2.3).

This similarity between classes forces the network to learn subtle features instead of relying on shortcuts such as background information [121]. Since the subtle semantic differences between classes are known (*e.g.,*a Boston Terrier has more pointed ears than a French Bulldog [122]), many works in FGVC propose methods that encourage the network to attend to relevant semantic features. Early works that used this approach assumed that part-level bounding boxes were available at training time [123]–[125], while recent work attempts to learn important regions in a semi-supervised manner [126], [127].

Figure 2.3: Hierarchical scene classification is a way to provide additional information to fine-grained classification tasks—for example, by helping differentiate between a french bulldog and a boston terrier [120].

While this approach is the most intuitive, it is not the only approach that has been attempted on this problem. One common approach is *bilinear pooling* [128], [129], which combines feature maps from different trained classifiers to capture a greater diversity of features. Other approaches include lifting the 2D image into a 3D space [130], adjusting losses to prevent overconfidence caused by irrelevant features [131], [132], comparing similar classes during inference [133], and using text descriptions of distinct parts as the human input [134].

Like a number of works in fully automated FGVC [128], [135], hierarchical scene classification (HSC) [98], [104], [136] exploits the hierarchy that is often available in these problems by allowing a user to define if the scene falls into a coarse class like indoor, outdoor natural, or outdoor man-made.

Single-Target Video Object Tracking In our chosen formulation of Single-Target Video Object Tracking (VOT)—also often referred to as Single-Target Visual Object Tracking—a bounding box is drawn around a semantic object in the first frame of a video and propagated through the remaining frames. The earliest work on this application [137] focus on the ability to measure displacements and, in doing so, produce segmentations of moving objects. Other early work focuses on tracking keypoints [138] or ellipses specifically designed to enclose faces [139]. Because of

long history of this application, evaluation methods and datasets have been inconsistent: under the assumption of per-frame bounding box annotations, evaluations included center error [140], region overlap [141], tracking length [142], failure rate [143], and others [144]. Noting the difficulty of obtaining per-frame annotations, Wu *et al.* [145] proposed a cycle-consistency metric, while two other works [54], [146] converged to similar solutions still used today: an aggregation of new sequences and previously collected datasets such as Berkeley segmentation [147], FERET [148], VIVID [149], CAVIAR [150], and others [140], [151], [152] measured via robustness (number of lost tracks) and accuracy (mean IoU across valid frames). Both works acknowledge the potential for perturbations in the human-provided initialization, such as a 10% error in bounding box size and variations in the starting frame, but do not consider the effect of individual annotations on the quality of the output.

Similar to the variety of evaluation metrics, a number of solutions for single-target VOT have been proposed. Early work focused on handcrafted local features [137]–[139], while more recent works implement deep learning-based approaches through fine-turning the network at inference time [153]–[155], asking the network to directly regress bounding box location [156], or placing template and search images in a common feature space [157]–[160]. We performed experiments using the Distractor-Aware Siamese Region Proposal Network (DaSiamRPN) [160], which won the short-term real-time category of the 2018 VOT challenge [161] by improving the negative mining strategy used to create the common feature space of the Siamese Region Proposal Network [158], and the ToMP tracker [52], which learns the weights for a tracking model.

Referring Expression Comprehension In referring expression comprehension, a model is asked to identify—via bounding box or segmentation—an object in an image described by a natural language expression. The motivation for this task is straightforward: referring expressions are natural linguistic features that will be needed to communicate with intelligent agents such as robots [61], [67], [91].

While the first work to consider this application [162] referred to it as text-to-image coreference and performed their experiments on the indoor scenes of the NYU-RGBD V2 Dataset [163], most work follows the formulation of Mao *et al.* [8], which uses referring expressions collected on the COCO dataset [15] via a two-player crowdsourcing mechanism [2]. Under this formulation, the model is correct if the predicted bounding box has an IoU of greater than 0.5 with the target bounding box and incorrect otherwise.

Referring expression comprehension has since become a staple in the visiolinguistic literature, with strategies changing over time. Shortly after the Mao *et al.* work, Hu *et al.* shifted from bounding boxes to segmentations, using fully convolutional networks to produce response maps. In addition to the precision @ IoU formulation of Mao *et al.* this work reports the mean IoU.

Like many early strategies [8], [164]–[168], this work encoded the referring expression using LSTM neurons and, like many language problems, work has since shifted towards attention and transformer architectures. Though early work using attention did allow for some manual feature engineering [22], more recent work [1], [24], [50], [169] has utilized pure transformer models with extensive visiolinguistic pretraining. When the output is a bounding box—which is common due to the challenges of providing visual information to a transformer—most works accept object proposals as their fixed input and treat referring expression comprehension as a classification across these proposals, with the notable exception of MDETR [24] which directly accepts image features from a convolutional network.

2.2 Addressing Limitations of Hazy Oracles

We discussed above various applications that utilize hazy oracles, but have not discussed strategies for compensating for hazy oracles. In this chapter, we discuss how different works treat the fact that human inputs are noisy, roughly defined in two groups: machine learning approaches, where the goal is typically to optimize the output under the oracle assumption, and human-centered approaches, where the goal is to improve team performance via methods such as aggregation or explainability.

2.2.1 Machine Learning Approaches

Due to the supervised learning paradigm of machine learning approaches, the oracle assumption is typically used without modification—it is assumed that the input is correct and unambiguous, and the model must produce an answer. There are a handful of exceptions to this paradigm, where human input may be noisy or the model may be uncertain, that we describe here.

ML Centered HIL Inference Machine learning approaches have undeniably led to improved performance on meaningful applications: accuracy in Visual Question Answering has improved from 54.06% in the originating work [6], to 84.34% at the time of this writing [170], and similar improvements have been shown on referring expression comprehension [8], text question answering [102], single-target video object tracking [54], and more [76], [96], [100], [171]. The dataset-focused supervised-learning approach of these evaluations is both a blessing and a curse: while performance has undeniably improved, the human input is considered an oracle that can always be interpreted correctly. In other words, there is no understanding of what happens when the human input is incorrect or simply outside of the model's understanding.

There are, however, a few dedicated problems that consider effects of noisy human inputs in a supervised learning framework: evaluation for single-target VOT [54], [145] often applies

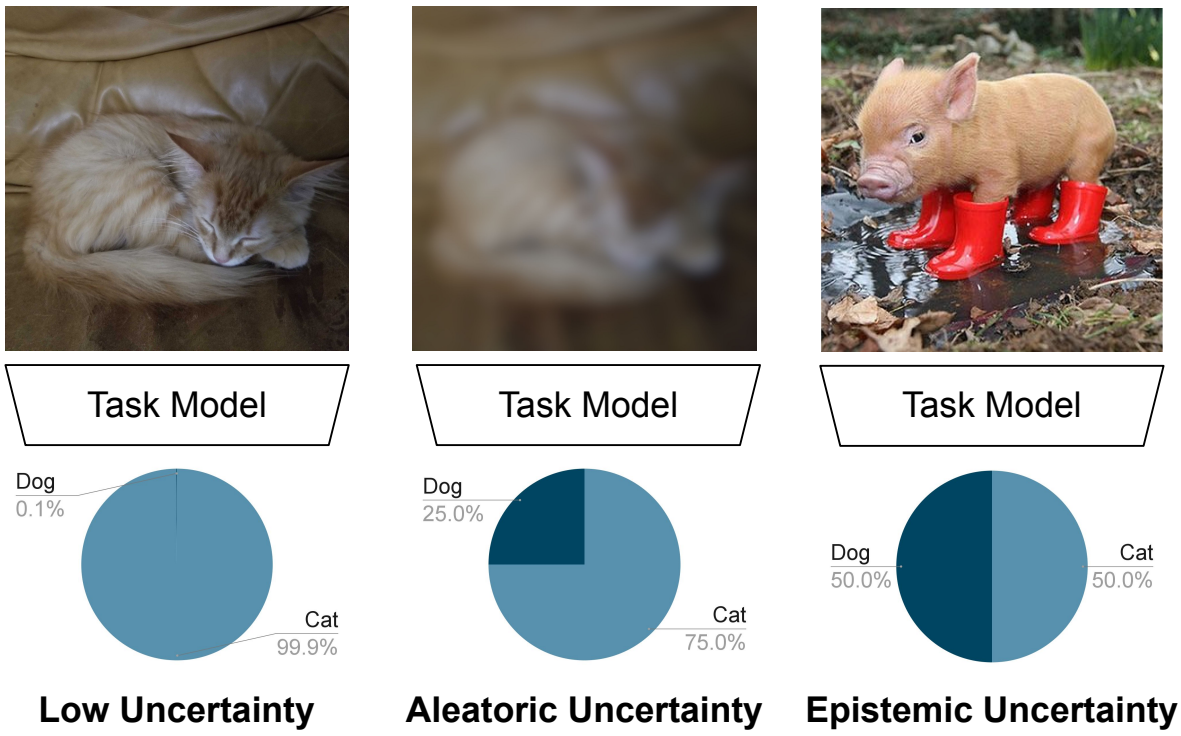


Figure 2.4: An illustrative example of Kendall & Gal's uncertainty taxonomy [175] applied to a simple cat/dog classification model. *aleatoric uncertainty* refers to uncertainty in the input that can not be reduced by more training—the cat can't be seen well enough for high certainty. Epistemic uncertainty refers to uncertainty based in the model's understanding of the world: since it has never seen a pig, it can't confidently place the image in a class.

small perturbations the initialization to evaluate the tracker's robustness, and works in keypoint-conditioned viewpoint estimation [172] and video object segmentation [173] determine which question is best to ask (*e.g.,* which frame provides the best result when annotated). Other works treat low-quality human inputs as a separate supervised learning problem. For example, SQuAD 2.0 [174] introduces questions that can not be answered based on the provided text, while Bhattacharya *et al.* [17] provide a taxonomy for visual questions that cannot be answered, and Mahendru *et al.* [12] attempt to find visual questions that are based on false premises. Critically, none of these works holistically consider the interaction between the model and the human operator: either they measure the model's sensitivity to different correct inputs, choose which class of input will produce the best output, or ignore the fact that semantically correct inputs can produce incorrect outputs.

Uncertainty and Selective Prediction Recognizing the importance knowing how confident a prediction is (an autonomous vehicle that has 60% confidence in class *road* and 40% confidence in class *person* should stop), a number of methods have been proposed to detect low confidence inferences. The majority of these detect *aleatoric uncertainty*—uncertainty inherent in the data—

as opposed to *epistemic uncertainty*—or gaps in the model's understanding of the world [175] (Figure 2.4). The reason for this is simple: aleatoric uncertainty models the amount of information available in the data, including features such as size and blur, and can therefore learned from the data. Epistemic uncertainty, on the other hand, is related to the knowledge of the model, which is very difficult to estimate in the supervised learning framework.

We begin with a discussion of methods for aleatoric uncertainty. Interestingly, not only does the data-dependent nature of aleatoric uncertainty make it easy to learn, but it is learned implicitly in the softmax loss. Although these outputs are known to be overconfident, this confidence is low dimensional, meaning it can be calibrated via a low-dimensional temperature scaling procedure [13]. A number of other works have proposed more complex or principled calibration procedures: Mozafari *et al.* [176] extend the work of Guo *et al.* [13] by learning the scaling coefficient during training instead of during post-hoc calibration, Mukhoti *et al.* [177] use temperature scaling in tandem with a focal loss, Kull *et al.* [178] use dirichlet calibration instead of temperature scaling, and Brian Lucena [179] proposes a spline-based calibration method.

While the most common formulation for regression—minimizing the mean squared error—does not implicitly capture the aleatoric uncertainty the way a softmax does, it is trivial to do so by changing to a negative log likelihood loss and adding output neurons to match. Kendall & Gal [175] introduce this for pose estimation, Choi *et al.* [180], Kraus and Dietmayer [181], and Le *et al.* [182] use this approach for object detection, with the latter showing the loss adjustment works better than a method that aggregates redundant boxes instead of using non-maximal suppression. Li & Lee [183] incorporate a more challenging formulation—a mixture density network—for the problem of human pose estimation.

Since it can't be learned, estimating epistemic uncertainty is much more challenging. The most conceptually straightforward techniques—Bayesian DNNs—place uncertainties on the weights and sample from these distributions at inference time. Blundell *et al.* [184] learn weight distributions explicitly via a method they call Bayes by Backprop, Maddox *et al.* [185] save weights at various point during training and place a distribution over those checkpoints, and Gal and Gharamani [186] assert that DNN weights are Bernoulli distributions and therefore can be sampled by enabling dropout at inference time. Liu *et al.* note that this does not properly model uncertainty outside of the support, and propose a sampling-based method using normalized weights and a Gaussian process output [187]. Some sampling free methods have also been proposed: Lakshminarayanan *et al.* [188] demonstrate that ensembles of networks can estimate both epistemic and aleatoric uncertainty and Postels *et al.* [189] inject noise into their training to capture epistemic uncertainty.

Though not explicitly treated as uncertainty estimation, a very closely related problem is Out Of Distribution detection (OOD)—in fact, some aleatoric [190] and epistemic [188] uncertainty methods were tested on this problem. In OOD detection, the goal is to determine if the given sample

matches the training distribution which is useful, for example, in determining whether or not to perform inference or locating adversarial inputs [34]. The simplest approach to this task is outlier exposure [191], where the model is trained to compare the target distribution to various out of distribution sets and it is assumed that novel OOD data will be closer to the trained OOD classes than the distribution in the decision space.

Other works follow the intuition that the only available training data is in distribution. Such methods include perturbing inputs and measuring the response [192] and building generative models. When the model maps directly to a likelihood (*e.g.,* Generative Flows [193], PixelCNN [194], Variational Autoencoders [195]), this is straightforward: all inputs are mapped to a location on the distribution, that can be used to divide high probability (in-distribution) and low probability (out-of-distribution) samples. Different strategies improve upon this basic idea: Ren *et al.* [196] mitigate the effect of the background through addition of noise, Xiao *et al.* [197] compare the difference between the actual generative model and an optimal generative model, Serrá *et al.* [198] demonstrate that lower complexity images typically have higher likelihood regardless of whether or not they are in distribution, and Choi *et al.* [199] use ensembling in tandem with the Watanabe Akaike Information Criterion [200].

Evaluating Uncertainty Measures There are many ways to evaluate uncertainty measures, each of which provides a slightly different framing of the problem. The most straightforward of these are measures such as the log likelihood or brier score [188], which can be applied directly to every prediction and aggregated across a dataset. While these are proper scoring rules (the maximum score is returned if and only if the predicted and true distributions match), they do not distinguish between a method that is overconfident and occasionally correct and a method that is always correct but not confident. For example, consider the log likelihood for a binary classification of 3 inferences:

$$\text{nll} = -\sum_{n=1}^{3} y_n log(\hat{y}_n) + (1 - y_n) log(1 - \hat{y_n}). \tag{2.1}$$

If the model makes two correct classifications with 97% confidence, and one incorrect classification with 86% confidence, the NLL is:

$$-(ln(0.97) + ln(0.97) + ln(0.14)) \approx 2.02 \tag{2.2}$$

whereas if the model makes three correct classifications with 51% confidence:

$$-(ln(0.51) + ln(0.51) + ln(0.51)) \approx 2.02 \tag{2.3}$$

Because of this, a number of works measure calibration directly—if a model predicts 65% confidence, is there actually a 65% chance of it being correct—using the metrics of Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) [13]. While intuitively satisfying, this evaluation results in two major shortcomings: first, confidence can be meaningful without being calibrated, leading to poor ECE for an otherwise useful confidence score. Second, it is fundamentally histogram based, which means that the selection of bin width can produce meaningfully different results. Selective prediction [42], [49], [58], [64], [201], [202] avoids these problems by evaluating the quality of confidence estimates ordinally. This is highlighted by Geifman *et al.* [203], who proposed using selective prediction to evaluate uncertainty estimation.

Of course, the best evaluation is dependent on the final application. Weather prediction favors the ECE and MCE [204], since the data must ultimately be human interpretable. For medical classification tasks, a selective prediction approach is more appropriate, since the goal is to determine when to request a second opinion: For diabetic retinopathy detection, Raghu *et al.* [45] use the AUROC metric to evaluate uncertainty performance, while Leibig *et al.* [46] use the AUROC metric as well as the tradeoff between accuracy and coverage used in selective prediction [58]. For segmentation, the path is less straightforward, but NLL, Brier score, and ECE [205], [206] have all been used with some success. In language tasks, some work simply notes the improvement in the base metric (*e.g.,* mean squared error) when uncertainty is explicitly accounted for [207], while others evaluate ECE [208], or an error-coverage tradeoff similar to selective prediction [209]. For camera relocalization, Kendall & Cipolla [210] illustrated the relationship between uncertainty and error visually, while probabilistic object detection [211] proposed a novel evaluation based on the pairwise Probability-based Detection Quality (PDQ) metric that combines bounding box regression and object classification uncertainties.

Notably, none of these works consider a deferral response. Of works that do consider a deferral response, method and evaluation is inconsistent: many works generate complementary text queries [61], [62], [91], [92], ask for a rephrase [68], or allow the human to identify and resolve local minima in tasks with a long time horizon (*e.g.,* adding instructions to a pick-and-place task [67]). Such methods conduct human experiments with a-priori thresholds, and report the change in accuracy from the deferral-free condition. This is not only unrepeatable, but neglects to address the tradeoff between accuracy and user effort.

2.2.2 Human-Centered Approaches

The goals of the machine learning approach are straightforward: increase performance relative to some metric given a problem formulation. In contrast, human-centered work generally chooses between three potential goals: i) measure how the human and AI act as a team, ii) produce the best annotations using many inputs, or iii) measure how satisfied the human is with the interaction.

Improving Team Performance AI agents in are used in tandem with individual humans to improve performance in a wide variety of domains. In the medical space, Cai *et al.* [56] introduce novel deep-learning enabled tools for searching tissue images for pathology, while Caruana *et al.* [83] introduce GA^{2}Ms, a pairwise addition to Generalized Additive Models (GAMs) [212] that allow medical professionals to manually adjust the influence of different factors and their interactions. For fact checking, Nguyen *et al.* [213] introduce a method that collects supporting evidence and produces sub-inferences that can be adjusted to produce a final output. For a chatbot, Jain *et al.* [214] propose parsing human text input and providing a visual representation of the model's understanding.

In most of these teaming works, the AI acts as an advisor and the human makes the final decision. This makes explainability, trust, and accurate mental models the most important components of the interaction. Though some works treat explainability as a machine learning problem [215]–[217], we focus here on cases where explainability's impact on the human operator is considered. Kulesza *et al.* [218] allow the human to ask for reasoning behind a decision, and adjust the reasoning when it's incorrect. Ghai *et al.* [219] use local explanations to improve active learning performance, while Zhang *et al.* [220] note some shortcomings with this approach in their evaluation of trust calibration. Wang *et al.* [221] propose a user-centric framework for explainable AI, while Tjoa and Guan [222] similarly note that explainability must be human-centered. Motivated by the observation that human-AI teams are often worse than just the AI [223], Bansal *et al.* provide a focus on mental models—how does the human understand the model and how does effect the team [224]? Interestingly, they find that decision boundaries must account for the human (*i.e.,* the best model is not the best teammate) [225] and that any changes to a model must consider how the human interprets the model and how they will adapt to those

Because the goal is to evaluate qualities such as trust and mental modeling, many teaming works abstract away the deep learning model by using simplified models, Wizard-of-Oz (WoZ) studies, or examining the inputs to characterize interactions. For example, the work of Chang *et al.* [226], [227] seeks to enable verbal interaction with videos, but uses a finite list of commands (10 main commands, with 17 variants in [226]), whereas Zhao *et al.* [228] seek the same goal, but use a WoZ study. Rosenblatt *et al.* [229] evaluate user satisfaction with a WoZ voice-based IDE. Little work considers the human teamed with a modern deep learning system in cases where the deep learning system must make the final inference. For this reason, we examine in-depth the interaction between human inputs and various deep learning models, as well as propose different ways to examine and evaluate this interaction.

Crowdsourcing In crowdsourcing, a number of Human Intelligence Tasks (HITs) are submitted to a central service such as Amazon Mechanical Turk, where a pool of workers can access and perform these tasks without the overhead of hiring and firing workers. This ability to easily access human intelligence as an API call has proven a boon for both real-time processing and dataset collection, but must compensate for unskilled or malicious workers—one study found that 30% of responses were generated by spammers [230]—via simple tasks, heavy training and filtering, or post-hoc review and aggregation.

As examples of simple interfaces, both OpenSurfaces [231] and MSCOCO [15] allow users to produce segmentations by simply drawing on the image, while Song *et al.* [70] combine four intuitive tools—trace, pin-placing, drag-and-drop, and floodfill—to produce a segmentation. Sorokin *et al.* [232] propose a pipeline for object grasping based on crowd workers providing object outlines, groupings, and comparisons. Bigham *et al.* [11] provide a method where workers answer visual questions, while CrowdMask [233] extends this by asking users to determine whether or not sensitive information is present in part of an image, assuming that this information can be removed prior to presenting the full image for processing.

With respect to filtering, some approaches increase the quality of the input by preventing low quality work before it is submitted. For example, determining the optimal way to train workers in a crowdsourced setting [234], making sure workers are qualified via initial [15] or random [235] screening, allowing users to choose not to annotate when they are uncertain [236]–[238], or providing the proper incentive structure [239], [240]. Combining the last two strategies, Shah and Zhou [237] showed the benefit of incentivizing workers to self-filter low-confidence annotations.

Filtering may also be done post-hoc by aggregating different workers' responses to the same question. This is straightforward when the human input is categorical: Dawid & Skene [241] use the Expectation Maximization algorithm to produce optimal answers for medical diagnoses when patient data is imperfect, while Ipeirotis *et al.* consider Amazon Mechanical Turk directly and estimate worker quality, making to separate error and bias, the latter of which can still improve performance over ignoring the "bad" annotations if it is accurately modeled [230]. Whitehill *et al.* [242], Raykar & Yu [243], and Welinder *et al.* [244] follow a similar strategy of modeling the user. Bragg *et al.* use simple majority voting to confirm a category [245], as does ImageNet when confirming the results of an image search [246]. Revolt [236] introduced method by which workers were allowed to discuss specific annotations with each other when there is disagreement. When the human input is not categorical, post-hoc filtering is a bit more challenging. Some works use a similar voting technique to confirm the quality of segmentations [15], [231], while Song *et al.* attempt to use summary statistics to filter bounding boxes and dimension line annotations, but observe that these methods are imperfect and suggest using probabilistic filters [69] or tools with different biases [19], [70] to mitigate remaining errors.

Human Satisfaction There are many ways for an individual to interact with an AI agent, but the most commonly surveyed interaction mode is linguistic. This is for two reasons: humans have a natural understanding of this input mode, and such systems are currently both commercially desirable—80% of businessees surveyed in 2016 claimed they would like to implement some form of chatbot to reduce expenses or improve customer experience [247]—and deployed in practice via mechanisms such as tech support phone trees or Conversational Virtual Assistants (CVAs) such as Siri.

Broadly, these studies have shown that people are not happy with such interfaces: a 2016 survey found that Interactive Voice Response (the tech-support phone-tree technology) systems are only satisfactory to 10% of users [248]. Though the technology is advancing, dissatisfaction remains: a 2019 study found that for both IVRs and Smart Assistants (*e.g.,* Siri), users are more likely to express disappointment, confusion, or unease than happiness with an interaction [249]. Generally speaking, this is due to poor or limited performance by the CVA. A 2017 study [250] performed across 54 users in India and the United States found that these chatbots were no better than the baseline condition—searching Google—for most tasks. Further, people were aware of this: 9 out of 17 non-users cited limited utility as their reason for not owning a smart speaker [251]. There were a few exceptions, Zamora *et al.* [250] and Luger & Sellen [47] both found CVAs to be useful for menial tasks and situations when the user can not be fully engaged with the keyboard (*e.g.,* handsfree scenarios). Put less charitably: voice control is only best when there is no other option.

The fact that such methods result in low quality inferences, in addition to human's ability to discover the decision boundaries and failure modes of systems such as decision support systems [224], means that humans don't speak to conversational agents in the same way as they speak to other humans: humans tend to shift from speaking colloquially to simple terms [47], shorter messages [251], [252], or more formal language [253]. Unfortunately, the fact that people expect to be able to learn the model means that they often blame themselves for poor performance that may not actually be their fault [47], [254].

Based on these findings, several guidelines for conversational agents are suggested. The most prominent of these guidelines is that the user should be told about the agent's capabilities upfront and often [47], [254] and, in doing so, their expectations should be appropriately set [250], [255]. While providing such instruction is outside the scope of this work, some suggestions are directly relevant to deferred inference: Jain *et al.* [254] suggest that the agent understand when it lacks requisite knowledge and either admit it or cover it up, and proactively asking questions about ambiguous queries to reduce the search space. Both Jain *et al.* and Luger & Sellen [47] suggest retaining conversational context, which is relevant when considering deferred inference.

2.3 Conclusion

Deferred inference sits at an intersection between deep learning and human-computer interaction. Although capable of remarkable tasks when cooperating with humans, the common use of the supervised learning leads to the prevalence of the oracle assumption during both training and evaluation, even when attempts are made to model uncertainty. Similarly, the simplifications made to accurately explore human behavior and attitudes mean that such work does not adequately consider the role of the deep model in the human-AI team. For this reason, the next chapter explores the often counterintuitive interaction between human inputs and deep learning models.

CHAPTER 3

Training and Evaluating Deferral Functions

The first component of deferred inference is the deferral function, which determines whether or not inference should be performed. Past works generally treat this in one of two ways: locating inferences that are likely to be incorrect [42], [58], [61] or detecting semantically incomplete or incorrect inputs [12], [17], [258]. However, a high-quality deferral function must consider these questions together: is there error *and* can changing the human-provided component of the input reduce this error. In order to implement this holistic approach, we must therefore understand both the task model's response to specific inputs and whether or not the cause of error is the human-provided input.

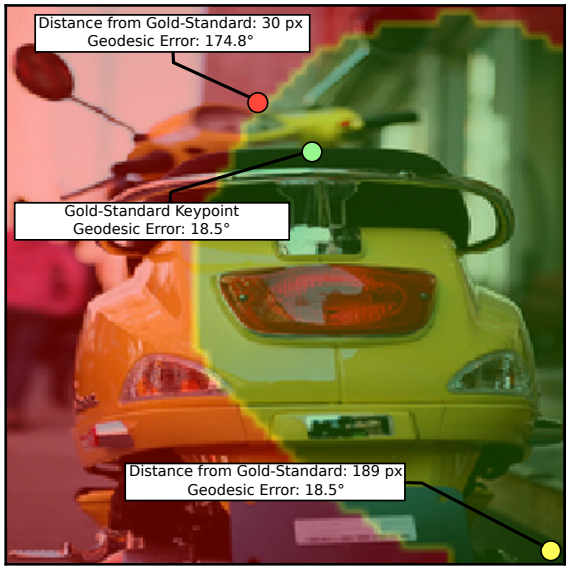


Figure 3.1: An example from the keypoint-conditioned viewpoint estimation application [63], with a heatmap of error caused by all potential clicks overlaid. Approaches focused on input-space accuracy [15], [19], [69], [99], [246], [256], [257] would select the red keypoint over the yellow keypoint as it closer to the gold-standard (green) keypoint, even though this results in higher error.

Understanding the Task Model Response: The need to understand the task model's response requires us to consider how a human's intuition of input quality differs from its effect on the quality of the output. Consider Figure 3.1: in this figure, the task model uses human-provided keypoint clicks on a semantic location—such as rear seat—to resolve perceptual ambiguities. While the gold-standard click location—shown in green—results in the lowest error (green overlay), many click locations that are incorrect in the input space (*e.g.,* the yellow circle) do not degrade performance, while many that are nearly correct in the input space (*e.g.,* the red circle) perform significantly worse than the gold-standard. It follows that methods that are designed to optimize accuracy in the input space [12], [15], [17], [69], [99], [246], [256]–[258] optimize the wrong objective: they maximize accuracy in the input space at the potential cost of output accuracy.

Understanding the Cause of Error: Since the goal of deferred inference is not to locate incorrect answers but to minimize error by acquiring additional human information, we must consider whether or not the answer can be improved by additional human information. If we inspect Figure 3.2, we can see that this is not always the case: a new human input may not be able to improve the inference because it will be correct even for an incorrect human input (Figure 3.2-left) or incorrect even for a correct human input (Figure 3.2-center). Specifically, this highlights the inadequacy of methods such as selective prediction [42], [58], which only seek to detect low-quality outputs.

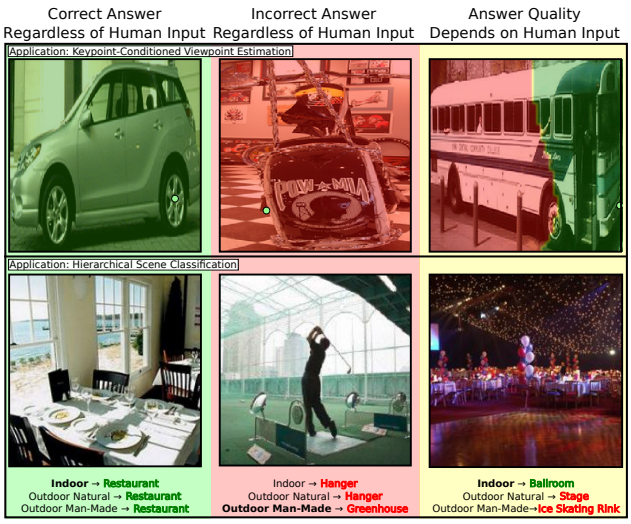


Figure 3.2: On both the KCVE (top row) and HSC (bottom row) applications, the task model may or may not base its answer solely on immutable data. For KCVE, the gold-standard click is shown as a green circle, while the overlaid heatmap shows error from low (green) to high (red). For HSC, the gold-standard is bolded, correct answers are green, and incorrect answers are red.

Implementation and Evaluation To address the challenges of deferral functions, we propose Dual-loss Additional Error Regression (DAER), a novel training method developed for this problem. DAER considers the two challenges discussed above separately during training, and combines them during inference to predict the effect of a candidate human input on the downstream task. We evaluate the performance of DAER on two applications: keypoint-conditioned viewpoint estimation [63], which is a human-in-the-loop extension of the canonical viewpoint estimation task [109], [116], [117], [259], [260], and hierarchical scene classification [104]—a method that improves performance on fine-grained classification [261]–[264] by integrating a coarse scene classification.

To evaluate DAER, we introduce a task-agnostic evaluation method for deferral functions, centered around three metrics designed specifically to assess the ability to detect inferences that would benefit from deferral: Additional Error (AE), Mean Additional Error (MAE), and Area under the Mean Additional Error curve (AMAE). Unlike existing metrics, such as selective risk [203], these metrics focus on the potential benefit of a new human input, instead of an oracle label of the target value that is prohibitively difficult to obtain at crowdsourcing scales, and may be impossible for important applications such as robotic assistance of individuals with mobility impairments.

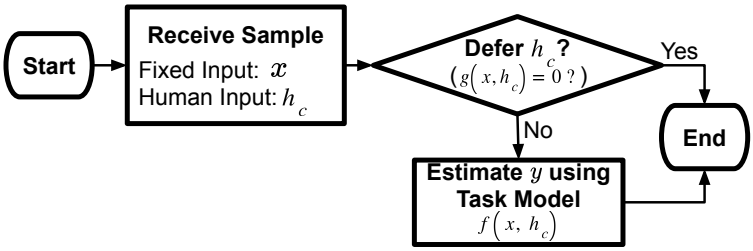


Figure 3.3: A flowchart of deferred inference on a single sample $(x, h_c) \in \mathcal{D}$. The deferral function, $g(x, h_c) \in \{0, 1\}$ seeks to defer samples for which using the candidate human input results in worse performance than using the gold-standard.

3.1 Problem Statement

We show the overall formulation of our problem in Figure 3.3. This formulation is based around a task model, $\hat{y} = f(x, h)$, that accepts a fixed input, x, and human-provided input, h. Given these inputs, the task model provides an estimate of a target value, y, with the goal of minimizing some task-specific performance measure, ℓ.

We refer to the human-provided input as the *candidate* input, h_c, and seek to determine if it degrades performance when compared to corresponding gold-standard human input, h_{gs}. We measure this degradation using the *additional error* (AE), calculated:

$$\begin{aligned} \text{AE}(\hat{y}_c, \hat{y}_{gs}, y|\ell) &= \max(\ell(\hat{y}_c, y) - \ell(\hat{y}_{gs}, y), 0) \\ \text{where } \hat{y}_c &= f(x, h_c) \text{ and } \hat{y}_{gs} = f(x, h_{gs}) \ . \end{aligned} \tag{3.1}$$

Figure 3.4: Evaluation considers when a candidate human input (red) outperforms the gold-standard (green).

The $\max$ operator enforces the constraint that a semantically incorrect candidate input cannot be considered better than the corresponding gold-standard input. This is important in cases such as the one shown in Figure 3.4, where there exist human inputs that result in less error than the gold-standard but we can not expect a human tasked with returning the gold-standard input to provide it.

We seek a deferral function, $g(x, h_c) \in \{0, 1\}$, such that human inputs with low additional error are accepted ($g(x, h_c) = 1$), and human inputs with high additional error are deferred ($g(x, h_c) = 0$). While an ideal deferral function would perfectly divide human inputs that cause error from those that do not, in practice the goal is to optimize a tradeoff between the proportion of inputs that are not deferred (referred to as *coverage*)[1] and an aggregate measure of the

[1]Since there is a limit of one deferral per human input, this chapter uses coverage instead of deferral rate to more closely match the evaluation of selective prediction. We note that coverage is simply $1 -$ deferral rate.

task model's performance over the accepted set. The use of coverage is required because deferral budgets will often be limited: the deferral function may need to accept human inputs that cause more error than the gold-standard but less than other human inputs (this is particularly important for continuous performance measures [54], [63], [100]) or may need to balance its confidence that a deferral will improve inference with the cost of deferring. Because of this, the deferral function produces a deferral score that ranks human inputs by the potential benefit of deferral, and coverage corresponds to a normalized ranking of inputs by this score.

Aggregate Metrics We now define the aggregate metrics used for both parameter tuning and comparing the performance of deferral functions on a test set, $\mathcal{D}$. We begin with the Mean Additional Error (MAE), which corresponds to the mean of all additional errors across an accepted set of samples:

$$\text{MAE} = \frac{\frac{1}{|\mathcal{D}|} \sum\limits_{(x,h_c,h_{gs},y)\in\mathcal{D}} g(x,h_c)\,\text{AE}(\hat{y}_c,\hat{y}_{gs},y|f,\ell)}{\frac{1}{|\mathcal{D}|}\sum\limits_{(x,h_c)\in\mathcal{D}} g(x,h_c)} \quad . \tag{3.2}$$

Although a target coverage or MAE will be chosen based on an application constraint (*e.g.,* budget), a general comparison of deferral functions requires a single summary statistic. For this, we introduce the Area under the Mean Additional Error-coverage curve (AMAE) metric. This metric is found by calculating the mean additional error at all coverages, then calculating the area under this curve.

To calculate this value, we sort the samples in our test set, $\mathcal{D}$, by the pre-threshold output of $g(x, s_c)$. This allows us to generalize the mean additional error formula such that we can define an arbitrary coverage as our target: by indexing the sorted set of samples using j, we can interpret our deferral problem as $g(x, h_c) = (j \leq i)$, where $\frac{i}{|\mathcal{D}|}$ is the target coverage. The mean additional error for coverage $\frac{i}{|\mathcal{D}|}$ is then:

$$\text{MAE} = \frac{\sum_{j=0}^{i} \text{AE}(\hat{y}_c^{(j)},\hat{y}_{gs}^{(j)},y^{(j)}|\ell)}{i} \quad . \tag{3.3}$$

We sum the MAE at every potential coverage:

$$\sum_{i=1}^{|\mathcal{D}|} \frac{\sum_{j=0}^{i} \text{AE}(\hat{y}_c^{(j)},\hat{y}_{gs}^{(j)},y^{(j)}|\ell)}{i} \tag{3.4}$$

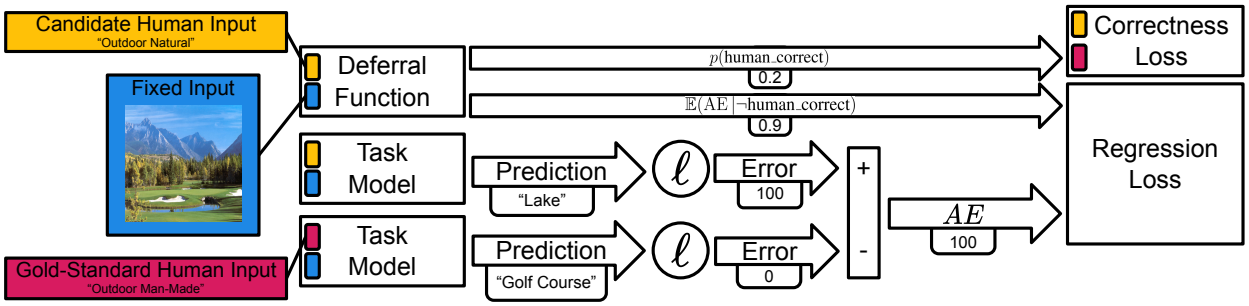


Figure 3.5: DAER separates the regression of additional error into two components: predicting whether the human input is correct through a correctness loss, and predicting the additional error through a regression loss that is only backpropagated if the human input is incorrect. For illustration, we include an example from the hierarchical scene classification application.

and scale such that the curve has a width of one:

$$\text{AMAE} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\frac{\sum_{j=0}^{i}\text{AE}(\hat{y}_c^{(j)}, \hat{y}_{gs}^{(j)}, y^{(j)}|\ell)}{i} \quad . \tag{3.5}$$

The AMAE can then be used to directly compare deferral functions across all target coverages. For all proposed metrics (AE, MAE, AMAE), a lower value corresponds to a better performance.

3.2 Dual-loss Additional Error Regression (DAER)

As our deferral function, we propose regressing the additional error directly through a novel method we call *Dual-loss Additional Error Regression* (DAER), shown in Figure 3.5. Core to DAER is the separation of the additional error regression into two components that correspond to the challenges described in the introduction. The correctness loss, which addresses the subgoal *understanding the cause of error*, is a binary classifier that estimates the probability that human input is correct. The regression loss, which addresses the subgoal *understanding task model response*, estimates the additional error given that the human input is incorrect. This conditional is enforced by only updating weights for this loss when the given human input is incorrect.

Mathematically, the correctness and regression outputs can be used to calculate the expected additional error:

$$\begin{aligned}\mathbb{E}(\text{AE}(\hat{y}_c, \hat{y}_{gs}, y|\ell)) = p(\text{human\_correct})\mathbb{E}(\text{AE}\,|\text{human\_correct}) + \\ p(\neg\text{human\_correct})\mathbb{E}(\text{AE}\,|\neg\text{human\_correct}) \quad .\end{aligned} \tag{3.6}$$

Since the additional error for a correct human input is zero by definition, this simplifies to:

$$\mathbb{E}(\mathrm{AE}(\hat{y}_c, \hat{y}_{gs}, y|\ell)) = p(\neg\text{human\_correct})\mathbb{E}(\mathrm{AE}\,|\neg\text{human\_correct}) \quad . \tag{3.7}$$

We use this formula to predict the additional error at inference time, but not during training. Instead, we train $p(\neg\text{human\_correct})$ and $\mathbb{E}(\mathrm{AE}\,|\neg\text{human\_correct})$ with separate losses, a method that is the key component of DAER. While DAER's training method is mathematically equivalent to regressing the additional error directly, we show in Section 3.3.3 that separating the two components significantly improves performance.

3.3 Experiments

Our formulation is applicable to a wide variety of problems, as it is fully specified by a task model, a deferral function architecture, a performance measure, and a definition of a correct human input. In this section, we demonstrate this by showing state-of-the-art performance on two disparate tasks: keypoint-conditioned viewpoint estimation and hierarchical scene classification.

3.3.1 Keypoint-Conditioned Viewpoint Estimation

Keypoint-conditioned viewpoint estimation [63] is a human-in-the-loop extension of the canonical computer vision application of viewpoint estimation [109], [116], [117], [259], [260]. In this application, a human annotator is given an image of a vehicle and asked to click a keypoint such as "front right tire." This human-produced information is then combined with CNN features to estimate the camera viewpoint more accurately than would be possible without the keypoint. Following convention [63], [116], [117], we measure performance using the geodesic on the unit sphere:

$$\ell(\hat{y}, y) = \frac{1}{\sqrt{2}}||log(\hat{y}, y^T)||_F, \tag{3.8}$$

where our estimate and ground-truth viewpoints are represented as rotation matrices.

Architecture In this work, Click-Here CNN (CH-CNN) [63] is our task model and, with modified output layers, our learned deferral function. CH-CNN consists of two branches that process the image and keypoint mostly independently, then concatenates the resulting features and passes them through two linear layers. Further information on the base architecture is available in the original work [63].

The output of the task model (Figure 3.6-left) is of size 3x3x360, consisting of three vehicle classes (car, bus, motorbike), three angles (azimuth, elevation, tilt) and 360 potential angle values

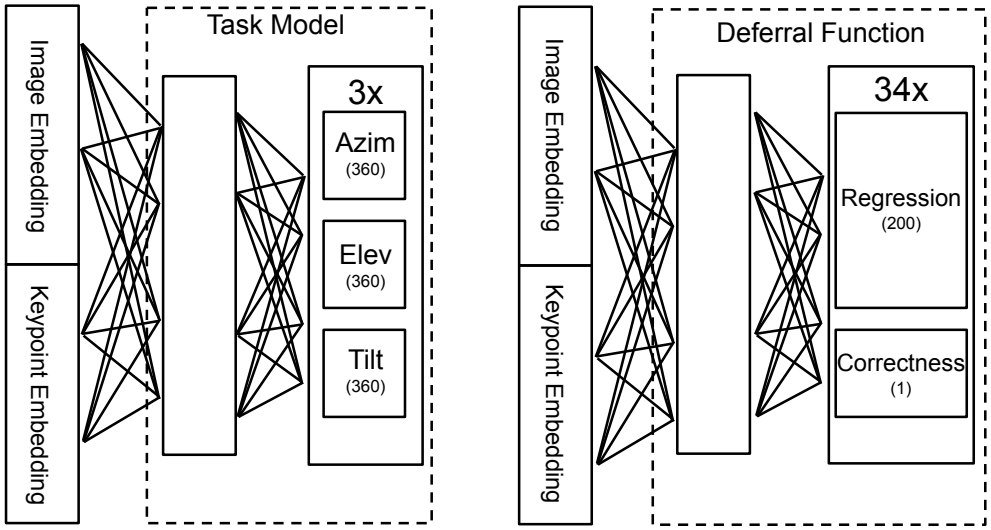


Figure 3.6: Prediction layers for the KCVE task model (left) and deferral function (right), which accept the keypoint and image embeddings from [63].

following the common convention of binned cross-entropy instead of regression [116], [117]. For our learned deferral function (Figure 3.6-right) we change the output to be of size 34x(200+1). This output consists of 34 potential keypoint classes, 200 binned outputs per keypoint class to regress the additional error, and one output per keypoint class to estimate the correctness.

Training Although we evaluate using the metric of the geodesic on the unit sphere, the computational complexity of calculating the matrix logarithm makes this measure impractical for training. For this reason, our deferral function is trained to predict rotational displacement in terms of Larochelle *et al.*'s distance [265]:

$$d = ||I - A_2 A_1^T||_F \quad . \tag{3.9}$$

While it is intuitive to define a correct human input as one that exactly matches the gold-standard, the Click-Here CNN architecture uses a 46x46 one-hot grid to process the human input, which makes it unlikely that a click randomly selected during training will match the gold-standard. Therefore, defining a correct human input in this way would result in a deferral function whose objective effectively reduces to regressing the additional error directly. Instead, we define a correct human input as one for which the additional error is zero:

$$p(\text{human\_correct}) = \begin{cases} 0 & \text{AE}(\hat{y}_c, \hat{y}_{gs}, y|\ell) = 0 \\ 1 & \text{AE}(\hat{y}_c, \hat{y}_{gs}, y|\ell) \neq 0 \end{cases} . \tag{3.10}$$

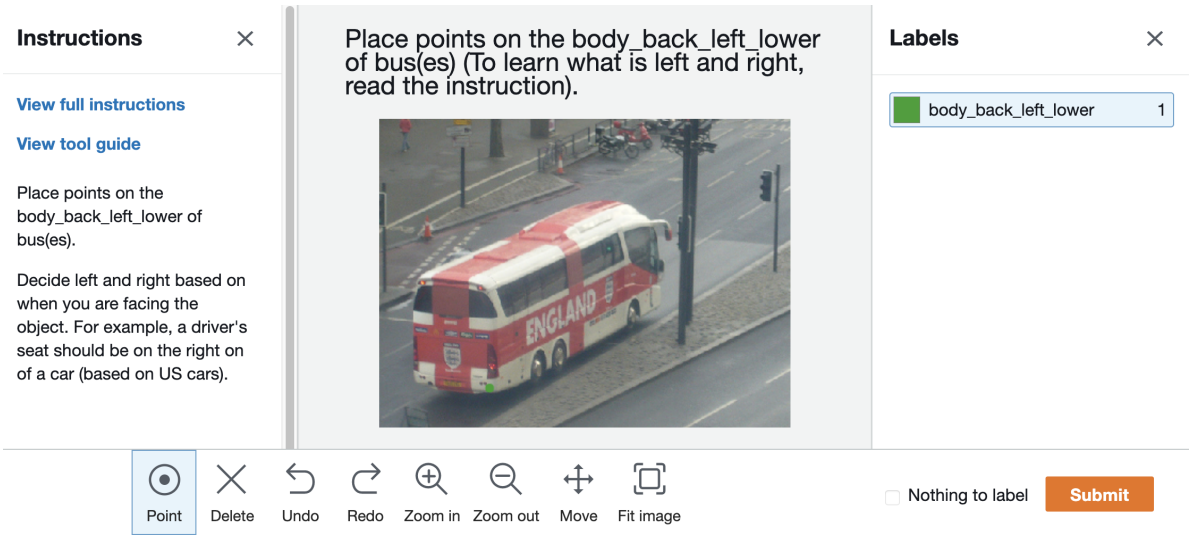


Figure 3.7: The interface provided to crowd workers for crowdsourcing keypoint clicks.

In addition to more effectively balancing correct and incorrect inputs, this encourages the deferral function to take a shortcut by learning the interaction between the task model and primary input prior to considering the human input. For example, the left and center cases in Figure 3.2 can be accepted without considering the the human input.

The deferral function was trained in two phases. In the first phase, it was trained on a combination of rendered [63] and real [105] data. Candidate human inputs were generated by randomly selecting an x-y location on the image, an Adam optimizer [266] was used with learning rate $1e^{-4}$, and early stopping was performed on the validation loss with a patience of 5 epochs. In the second phase, the deferral function was trained exclusively on the PASCAL3D+ dataset [105]. The same optimizer settings were used, but the one-hot additional error target was softened by convolving with a Gaussian kernel with standard deviation 3. Early stopping was performed on the validation loss with a patience of 100 epochs.

Regression and correctness ablations used the same training procedure, where back propagation was only performed on the appropriate loss. For the human-free ablation, a tensor of zeros was given to the deferral function in place of the keypoint map, and no further modifications were made to architecture or training.

Crowdsourcing Keypoint Clicks Performance was evaluated using a total of 6,042 keypoints on the PASCAL3D+ validation set [105] collected from US-based annotators via Amazon Mechanical Turk. Keypoint annotations were collected from US-based annotators using the interface shown in Figure 3.7: workers were shown an image containing one or more vehicles and asked to click all instances of a specific keypoint class. If an annotator responded that the keypoint class wasn't

present, we provided the query to another annotator up to two additional times. If all three annotators responded that the keypoint class wasn't present, we assumed the gold standard was incorrect or too difficult, and removed it from the evaluation.

To match the annotated keypoints with the corresponding verified gold-standard keypoint from PASCAL3D+, we used a three-step process: first, we associated all keypoints to vehicle crops that contained them. Next, we matched these keypoints to the gold-standard keypoint of the same class in that vehicle crop. Last, if a vehicle crop contained multiple candidate keypoints of the same class, we selected the one that was nearest to the gold-standard keypoint. Using this process, we receive annotations matching 6,042 of the 6,593 gold-standard keypoints.

Analyzing the distribution of matched keypoints (Figure 3.8), we found that 40% of keypoints were within 5 pixels of the matching gold-standard and 57% were within 10 pixels of the matched gold-standard keypoint. We further found that 6.3% (381) of keypoints caused additional error, while 1.3% (81) caused more than 5° additional error, and 0.5% (30) caused more than 150° additional error. We examine this more closely in Section 3.4.

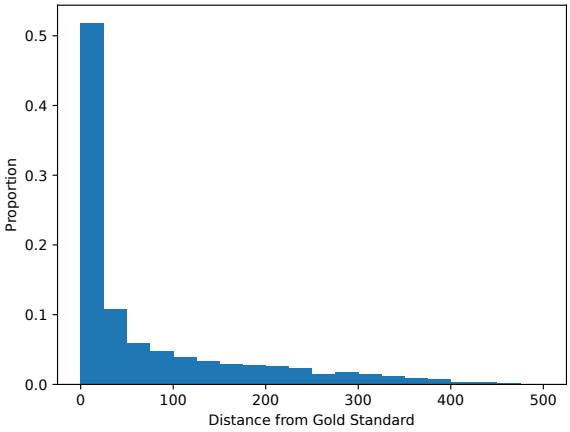


Figure 3.8: Distribution of distances between candidate and gold-standard keypoints.

Baselines For KCVE, we compared DAER to the following baselines:

- Random: A random human input from $\mathcal{D}$ is deferred at every step.
- Softmax Response (S.R.): The largest value of the softmax output. This was shown to perform best on selective prediction [58], the most similar task to ours.
- Known Distance: Oracle knowledge of the human input's Euclidian distance from the gold-standard. This simulates approaches that seek to minimize error in the input space.
- Task Network Entropy: The distributional entropy of the output of the task model.
- Task Network Percentile: 10,000 samples are taken from the task model's output distribution, and the n^{th} percentile difference between all samples and the mean is used as our deferral function. We evaluated on the 70^{th}, 80^{th}, and 90^{th} percentiles.

Results We divided the PASCAL3D+ validation set and corresponding crowdsourced clicks into five folds such that no vehicle crop appeared in more than one fold. Results are shown in Table 3.1. While no single method performed best across all folds, DAER performed best on the mean and was the most consistent overall: DAER did not perform worse than 25.3% above its mean on any fold, while the corresponding number for the best baseline (80^{th} percentile) was 80.4%. Additionally DAER has a better worst-fold AMAE (0.359) than all baselines.

	Fold					
	1	2	3	4	5	Mean
Random	1.919	1.698	0.935	1.648	1.518	1.544
Softmax Response	0.561	0.999	0.167	1.430	1.496	0.930
Known Distance	0.419	**0.254**	0.147	0.757	0.405	0.396
Task Network Entropy	0.325	0.556	0.112	0.421	0.353	0.353
70th Percentile	0.296	0.739	0.121	0.398	**0.182**	0.347
80th Percentile	**0.292**	0.558	0.125	0.370	0.201	0.309
90th Percentile	0.310	0.601	0.145	0.398	0.198	0.330
DAER	0.322	0.307	**0.109**	**0.335**	0.359	**0.286**

Table 3.1: The AMAE of DAER and baselines across 5 folds. We note that DAER has the lowest mean, never performs worse than 25.3% over this mean, and has the lowest worst-fold AMAE.

As each baseline accurately addresses one of the described subgoals while ignoring the other (e.g., distance only finds the cause of error and sampler only understands model response), this suggests that some folds contain more instances of one source of error, and again highlights the importance of considering the team holistically instead of optimizing the components separately. While focusing solely on one subgoal allowed baselines to perform well on folds where the relevant source of error was more frequent, DAER's understanding of both subgoals led to more consistent and overall better performance.

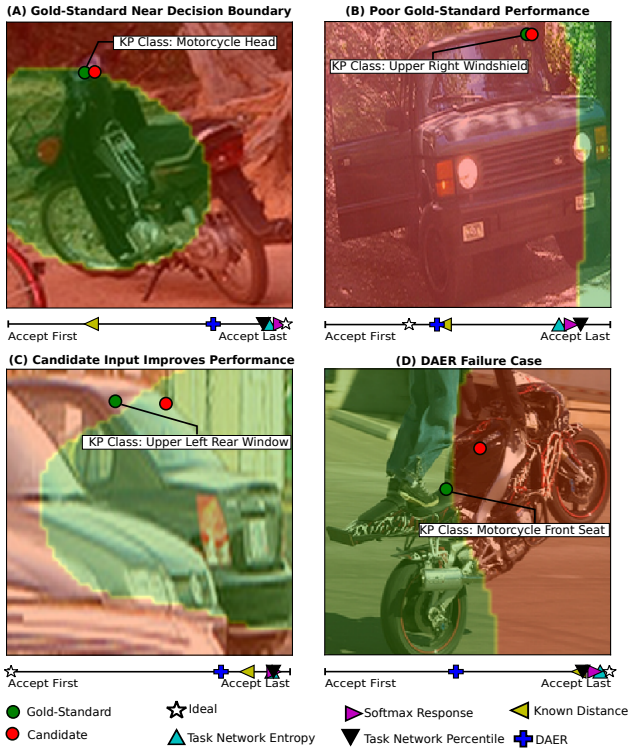


Figure 3.9: Select example cases from KCVE. Ideal accept location—the coverage where sorting by additional error would accept a human input—is given by the white star. Overlaid heatmaps are from green (low error) to red (high error)

We highlight specific examples in Figures 3.9 and 3.10, where the former were hand picked, and the latter were selected quantitatively. In 3.9-(A), the gold-standard was near the decision boundary and there was a high additional error even though the candidate is near the gold-standard. This caused the known distance baseline to fail by accepting the candidate early, while DAER and baselines based on the task model's output recognized a high probabilty of error and accepted this candidate late. In 3.9-(B), DAER successfully recognized that while the geodesic error for the candidate is high, the ground-truth will not provide an improved estimate of the camera viewpoint. In 3.9-(C) the gold-standard caused error in the output, but the candidate produced a better output, despite a mismatch between the keypoint label and location. In 3.9-(D) DAER was unable to accurately

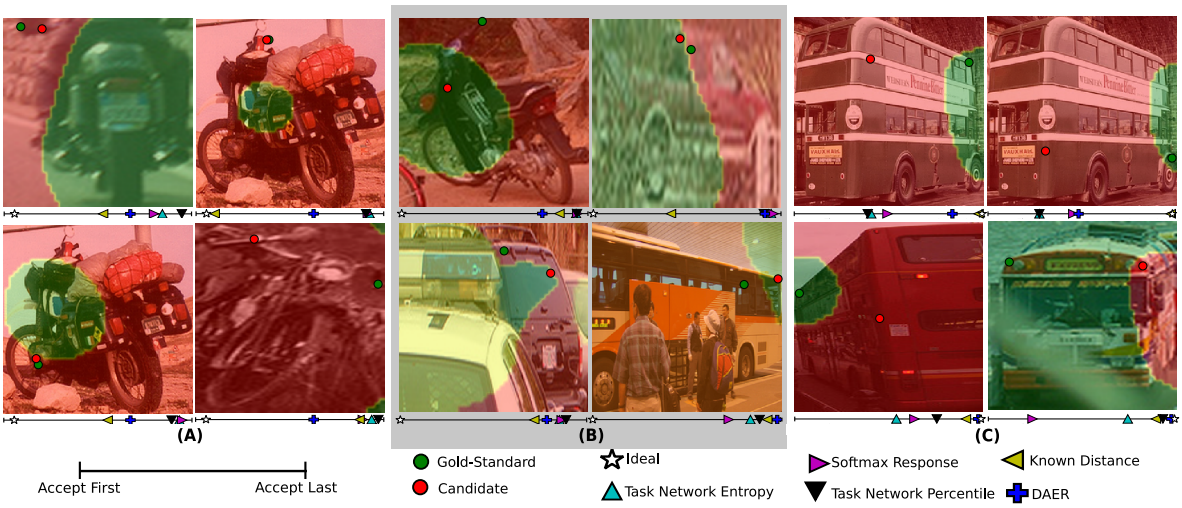


Figure 3.10: Quantitatively chosen KCVE heatmaps. The gold-standard human input is shown in green, the candidate human input is shown in red, and a red-yellow-green heatmap gives the additional error for that keypoint click. Methods closer to the white "ideal" star are better for that example. (A) The four cases where the gold-standard human input provided the worst absolute performance. (B) The four cases where the candidate human input improved upon the gold-standard human input the most. (C) The four cases with the highest additional error.

estimate the task model's decison boundary, resulting in early acceptance of a low-quality human input.

In Figure 3.10, we additionally provide four quantitatively chosen cases for each of three conditions: (A) the gold-standard human input results in the highest geodesic error, (B) the candidate human input most improves upon the gold-standard, (C) and the candidate human input causes the greatest additional error. When the gold-standard input results in high geodesic error, the input should be accepted despite poor performance, since a non-adversarial worker would continue clicking locations near the gold-standard. In these cases known distance generally performed best because the goal was to accept gold-standard inputs that produced high error, but DAER outperformed all baselines that attempted to evaluate the output quality. Similarly, DAER and known distance both perform well at detecting high additional error and, while not as catastrophic as in the case of gold-standard inputs that cause error, the various selective prediction-like approaches generally performed poorly. When the candidate human input outperforms the gold-standard (B), there is no method that clearly outperforms the others.

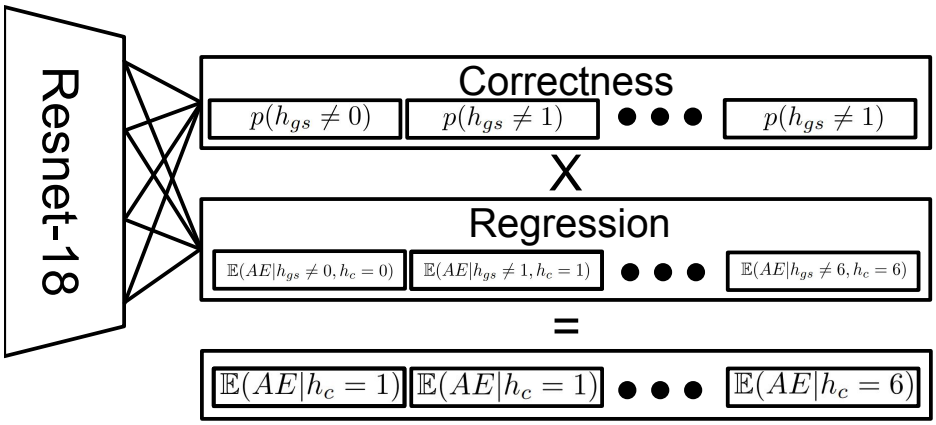


Figure 3.11: Our deferral function architecture and output format for the HSC task. Each potential candidate human input is given two outputs that are multiplied to estimate the additional error.

3.3.2 Hierarchical Scene Classification

Hierarchical scene classification [98], [104], [136] is an extension of fine-grained classification [261]–[264] where information about the coarse scene categorization—such as "indoor"—is given to a classifier alongside the image to help determine the fine-grained scene classification—such as "ballroom"—of an image. In this work, we train and evaluate on the SUN397 dataset [264], a dataset of over 130,000 images across 397 classes, and use the Plugin Network architecture developed by Koperski *et al.* [104] as our task model. Since the evaluation set is much larger for HSC than KCVE, we replace human annotators with a deep learned classifier (*Coarse Model*)—a ResNet-18 pretrained on ImageNet. For this experiment, we treat the coarse model as equivalent to a human, but we discuss the implications of the deferrable information being provided by humans instead of machines in the conclusion of this thesis.

For this problem, we define a correct human input as one that matches the gold-standard coarse classification. The performance measure is given as:

$$\ell(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 100 & \hat{y} \neq y \end{cases} . \tag{3.11}$$

With this performance measure, the MAE corresponds to the percent difference in accuracy caused by using candidate human input in place of the gold-standard at a given coverage.

Architecture and Training The architecture and output layers used for the hierarchical scene classification task are shown in Figure 3.11. As a backbone, we used an ImageNet-pretrained ResNet-18 [246], and truncated the output to 2 elements per coarse class (14 total). Seven of these

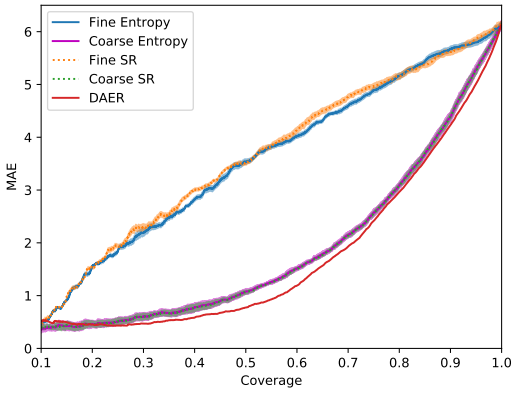


Figure 3.12: The MAE-coverage plot for the HSC application. Dark lines represent the mean of all runs, shaded area represents one standard error.

Method	AMAE
Random	6.17 ± 1.1e^{-1}
Fine Softmax Response	3.35 ± 4.1e^{-2}
Fine Entropy	3.29 ± 3.8e^{-2}
Coarse Softmax Response	1.75 ± 4.6e^{-2}
Coarse Entropy	1.74 ± 4.8e^{-2}
DAER	1.62 ± 3.4e^{-3}

Table 3.2: AMAE for baselines and DAER on the HSC application (lower is better). Standard error is calculated across five coarse models and, for DAER, five deferral functions.

outputs—the correctness outputs—were trained using a cross-entropy loss to determine

$$p(h_{gs} \neq \text{class}|x) = 1 - p(h_{gs} = \text{class}|x) \ , \tag{3.12}$$

while the other seven were trained using a binary cross entropy to find

$$\mathbb{E}(\text{AE}\,|x, s_{gs} \neq \text{class}, s_c = \text{class}) \ . \tag{3.13}$$

The model was trained for 50 epochs with learning rate 1e^{-5} and the model with the best validation AMAE was used for evaluation. The correctness-only deferral functions, regression-only deferral functions used in our ablation, and coarse models were trained identically using only the appropriate outputs, except the learning rate was increased to $1e^{-4}$ and accuracy was used in place of AMAE to select the best coarse model.

We trained five deferral functions and five coarse models, which allowed us to calculate the standard error across 5 runs for the baselines, and 25 runs for the learned deferral functions.

Baselines For hierarchical scene classification, we compared DAER to five baselines:

- Random: We defer a random human input at every coverage.
- Fine Softmax Response: We defer based on the softmax response of the task model's output.
- Fine Entropy: We defer based on the entropy of the task model's output.
- Coarse Softmax Response: We defer based on the softmax response of the coarse model that is used to simulate the human input.
- Coarse Entropy: We defer based on the entropy of the coarse model that is used to simulate the human input.

Results We see in Table 3.2 that DAER significantly outperformed baselines for hierarchical scene classification under the aggregate AMAE metric. Further, we see in Figure 3.12, that DAER outperformed all baselines on the MAE metric at every coverage greater than 0.197, which corresponds to all cases where fewer than 80.3% of human inputs are deferred. At this crossover point, the MAE was approximately 0.45: in about 1 out of every 222 samples an incorrect answer was caused by our coarse model.

To provide an additional perspective on relative performance, we consider the cases where it is acceptable that 1 out of every 100, 1 out of every 40, and 1 out of every 20 inferences are incorrect due to an incorrect human input, corresponding to acceptable MAEs of 1, 2.5, and 5 respectively. The percentage of human inputs that must be deferred ($1 -$ coverage), as well as the corresponding percent reduction in number of deferrals for these cases is shown in Table 3.3. Notably, for a target MAE of 5, DAER reduced the number of deferrals by 23.8% over the next strongest baseline.

	Target MAE		
	1	2.5	5
Fine Softmax Resonse	85.0%	67.0%	24.2%
Fine Entropy	84.4%	64.3%	22.7%
Coarse Softmax Response	51.8%	25.7%	6.7%
Coarse Entropy	51.7%	25.6%	6.7%
DAER	**43.6%**	**24.2%**	**5.1%**
Relative Reduction[1]	15.7%	5.5%	23.8%

Table 3.3: The percentage of human inputs that must be deferred for various target MAEs on the hierarchical scene classification task, as well as the percent reduction from using DAER over the next-best baseline.

3.3.3 Ablation: Importance of Subgoals

In the introduction, we proposed two subgoals: *understanding the task model response* and *understanding the cause of error*, which correspond to regression and correctness losses, respectively, in DAER. While we have shown that DAER outperforms the baselines, we have not yet examined the contributions of each subgoal. To do this, we performed three ablations:

1. Correctness: It may be adequate to simply guess whether or not the human input is correct. To test this, we use the correctness loss alone as the deferral criteria. This is analogous to methods that attempt to predict the semantic accuracy of human inputs [12].
2. Regression: The way DAER combines its outputs during evaluation is mathematically equivalent to regressing additional error directly. Therefore, we evaluate the value of splitting our loss by training a model to perform regression without the correctness loss. By doing this, we focus solely on understanding the task model's response to the given inputs.
3. No Human: While we encourage simplifying the goal of understanding the task model's response by learning which images are difficult, we would like to ensure that the model does not rely solely on this shortcut. To test if it does, we regressed the additional error without

[1] Calculated: $\frac{\text{Coarse Entropy} - \text{DAER}}{\text{Coarse Entropy}}$

Figure 3.13: Geodesic error from the task model (top) compared to the additional error prediction from a DAER deferral function (bottom). Error is overlaid from high (red) to low (green). Predictions are normalized per-image.

access to the human input. For KCVE, we provide a click map of all zeros, while for HSC this was implemented by reducing the regression output to a single element.

The results of these ablations—shown in Table 3.4—reveals two interesting phenomena that provide insight into the functionality of DAER: first, in both tasks the correctness loss outperformed the regression loss. Second, even without knowing the human input, understanding the task model's response to the image was competitive with some baselines.

The fact that the correctness loss outperformed the regression loss suggests that classifying human inputs as correct and incorrect—understanding the cause of error—is easier than estimating the additional error, and that this rough categorization combined with its implicit confidence is therefore a moderately effective deferral function. However, the fact that it is improved by a conditioned version of regressing the additional error shows us both that eliminating cases where the human input is correct results in an easier regression problem, and that a deferral function trained to solve this regression problem can learn to predict the task model's response.

Further, the fact that performance of a deferral function trained without access to the human input is comparable to baselines on both tasks suggests that it is possible, but not optimal, to defer based on the sensitivity of an immutable input and task model to the human input. We see why this might be the case in Figure 3.13: the best performance can be obtained by regressing the per-pixel additional error but deferring an unknown keypoint on the rightmost image is much more likely to reduce the mean additional error than deferring an unknown keypoint on the other example images.

	KCVE	HSC
Correctness	0.2937	$1.79 \pm 2.3e^{-2}$
Regression	1.1633	$2.05 \pm 1.1e^{-2}$
No Human	0.8002	$2.28 \pm 2.1e^{-2}$
DAER	**0.2864**	$\mathbf{1.62 \pm 3.4e^{-3}}$

Table 3.4: AMAE for DAER and its individual subgoals (lower is better).

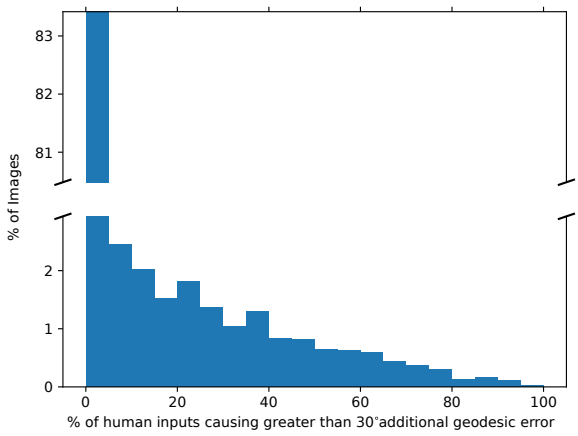


Figure 3.14: The sensitivity of Click-Here CNN [63] to click location. The majority of images have few potential clicks capable of causing significant error, and 36.3% do not respond to the keypoint click at all.

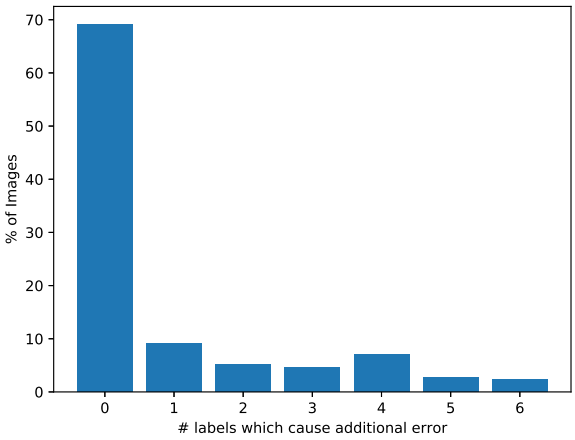


Figure 3.15: The sensitivity of plugin networks [104] to coarse classification. For 67.3% of images, an incorrect human input does not result in a correct fine-grained classification becoming incorrect.

3.4 Why Does DAER work?

Though DAER is mathematically equivalent to directly regressing the additional error, our ablation showed that the dual-loss nature of DAER is critical for state-of-the-art performance (Table 3.4). The natural question is, of course, why would this be the case? We begin by examining the distribution of model responses to all potential human inputs in Figures 3.14 and 3.15. In other words, for every potential click location (KCVE) or coarse scene category (HSC) we measure the additional error. We see that for most immutable inputs (images) in the dataset, no human input can produce additional error.

This suggests that *regressing additional error is a class-imbalance problem that DAER addresses through hierarchical learning.* This is additionally supported by our ablation results: training on correctness likely outperformed regressing the additional error directly due to it being a more balanced problem. For example, the fact that zero is the answer for 36.3% of inputs on the KCVE task, combined with the fine granularity of our deferral function (200 bins) means that a deferral function that always guesses 0 is likely to be more accurate than a deferral function that actually attempts to perform the regression.

This particular use of hierarchical learning is interesting, and likely to be applicable to a number of cases. For example, when attempting to estimate Ground Reaction Forces (GRFs) [267], there are two very common states that do not require numeric estimation: static/standing, where the GRF will be the subject's weight, and airborne (from jumping), where the GRF will be zero. Using a single classifier for all three states will result in the same problem, whereas training heirarchically

(*i.e.,* $\mathbb{E}(\text{force}|\neg\text{static}, \neg\text{airborne})p(\neg\text{static})p(\neg\text{airborne})$) will more effectively use the training data.

Interestingly, however, there has yet to be a systematic approach to or challenge for this problem. Methods that study class imbalance generally focus on imbalanced training data and balanced test data. For example, current work [31] produces imbalanced training sets for CIFAR 10 and 100 but leaves the validation set balanced, while the commonly used INaturalist Dataset [268] contains a long-tailed training set and a balanced validation set. Because of this, the goal during training is to simulate a balanced training set through strategies like under-sampling the majority class [269], adjusting how much each class contributes to the loss [31], [270], and adjusting the learning rate per class [271]. While meaningful, we believe this would fail in problems like additional error regression, since the test set maintains the imbalance of the train set, and therefore you can not simply seek to remove the effect of the class imbalance in the training data.

3.5 Conclusion

To develop an ideal deferral function, we must treat the human-AI team holistically: we cannot ask whether the human input is correct or whether the answer is correct, we must instead ask whether the answer is correct *and* whether a new human input can improve the answer. We demonstrate this via evaluations on the applications of keypoint-conditioned viewpoint estimation and hierarchical scene classification using the metric of Additional Error (AE)—the amount of error caused by the human input—as well as its aggregate complements of Mean Additional Error (MAE) and Area under the Mean Additional Error coverage curve (AMAE).

These metrics were used to evaluate the performance of various methods as deferral functions, including our novel method of Dual-loss Additional Error Regression (DAER). DAER is a training procedure that mitigates imbalance caused by overreliance on immutable inputs by separating the probability that the human input is correct from the probability of error. We demonstrate that it outperforms strong baselines under the AMAE metric and under the MAE metric at all coverages greater than 0.197, allowing us to effectively detect low-quality inferences caused by the human input.

The evaluation metrics introduced in this chapter, as well as DAER, are applicable in cases where there exists a gold-standard human input to which the candidate human input can be compared. There are two major limitations to this approach, which we address in subsequent chapters. First, while the metric of additional error makes the implicit assumption that the gold-standard human input can be retrieved, there are many cases where the replacement human input is likely to be drawn from the same distribution as an initial human input. Second, many settings—particularly linguistic tasks—can't compare to a gold-standard human input due to the dimensionality of the input space. We begin addressing these limitations by considering the effect of noisy deferral responses.

CHAPTER 4

Addressing Imperfect Deferral Responses

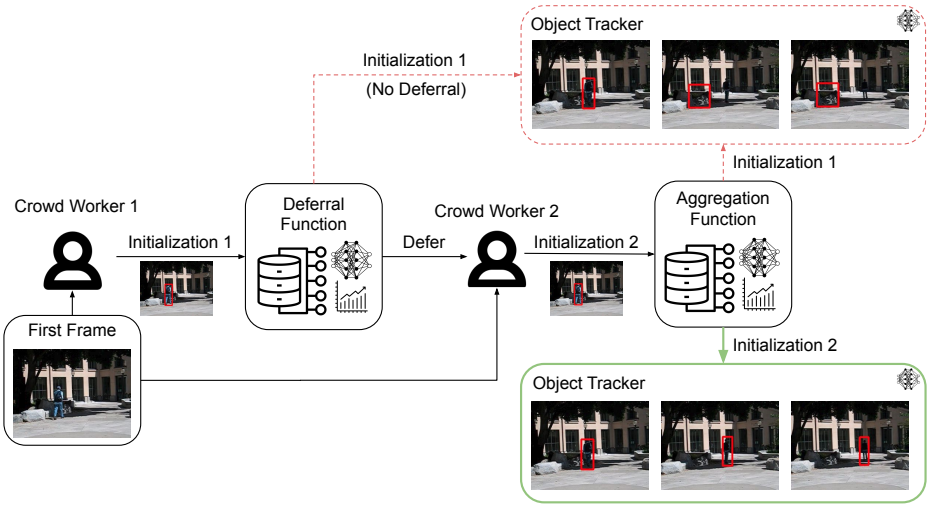


Figure 4.1: In single-target VOT, the tracker must be initialized with a bounding box to designate which object to follow. *Smart replacement* minimizes the number of initializations for a target accuracy by allowing the deferral function to accept the first initialization if it performs well, and allowing the aggregation function to choose between the first and second initialization if inference is deferred. We see the importance of these functions above, where nearly identical initializations result in dramatically different performances.

In the previous chapter we saw that a deferral function must separate error caused—and therefore correctable—by the human input from error caused by other sources. However, since this evaluation was focused on locating low-quality human inputs, it did not consider the situations where the deferral response is drawn from the same distribution as the initial query, such as crowdsourcing or interactions with an individual. This is a critical oversight, as it ignores the fact that the deferral response may also cause error. In this chapter, we address this shortcoming via the application of single-target Video Object Tracking (VOT) [54]. In this application, a human provides an initialization in the form of a bounding box drawn around a semantically meaningful object in the first frame of a video. This initialization is then propagated through the remaining video frames despite occlusions, deformations, rotations, and other visual phenomena. Although the initialization

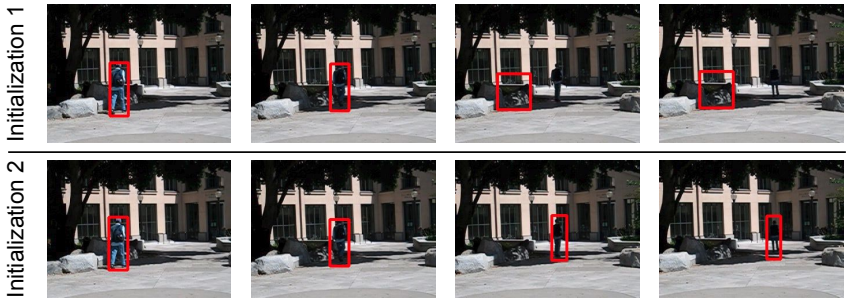


Figure 4.2: Although it is tempting to only examine the quality of the semantic input, the complexity of deep-learned models means that not all high-quality human inputs provide equal performance.

is perturbed for some evaluations, it is generally assumed that these perturbations are low in magnitude and proportional to the size of the object. The latter is incorrect, as error in bounding box annotations tends to be constant in the pixel space, not proportional to the object's size [272], the former is enforced via heavily controlled data collection (*e.g.,* [273]), and the question of how to compensate for this sensitivity is left unaddressed.

In response to these evaluation shortcomings and the sensitivity of VOT methods to small perturbations in the initialization (Figure 4.2), we examine deferred inference on a crowdsourced formulation of this application. We crowdsource 900 first-frame bounding boxes (9 for each video in the OTB-100 dataset [145]) and measure both their semantic quality and quality when used to initialize a state-of-the-art VOT model [160]. Through this analysis, we show that there is a correlation between semantic accuracy and performance but, as discussed in the previous chapter, it is not definitive: there are many (23.3%) crowdsourced initalizations that result in performance equal to the gold standard, and there are many initializations similar to the gold-standard that result in much worse performance.

From there, we find the best deferral function for this application using the AMAE-based evaluation procedure from the previous chapter, then extend it to the case where the deferral response comes from a hazy oracle via the novel evaluation metrics of Replacement Mean Additional Error (RMAE) and Area under the Replacement Mean Error-coverage Curve (ARMAE). Critically, we demonstrate that shifting this assumption changes the relative performance of the proposed deferral functions. Further, using the incorrect assumption results in local minima at relatively high coverages, meaning more deferrals may result in worse performance.

Motivated by this, we introduce the concept of aggregation functions and a straightforward aggregation function called *smart replacement* that compares the initial query and deferral response, then performs inference using the human input it believes results in the better answer. Despite its simplicity, smart replacement is both generalizable—it can be used with any application with a deferral function—and effective.

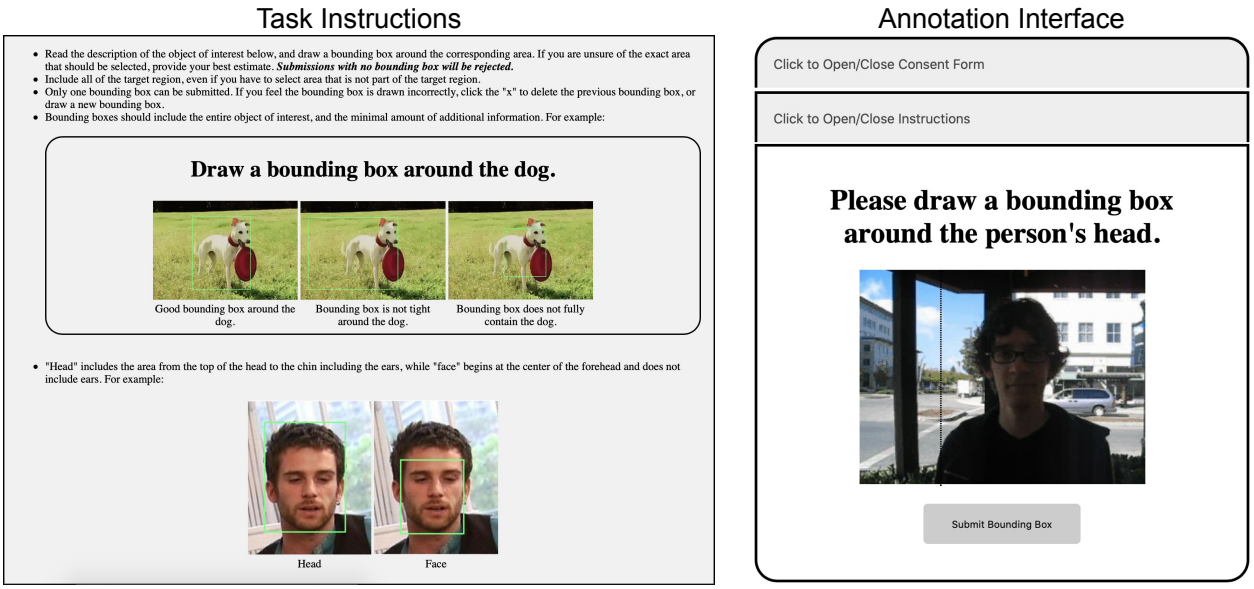


Figure 4.3: The instructions (left) and interface (right) provided to crowd workers for the bounding box annotation. Workers are given vertical and horizontal guidelines to assist in bounding box construction. Instruction images were taken from CelebA [274] and the ImageNet Large Scale Visual Recognition Challenge [272].

4.1 Quality and Effects of Crowdsourced Initializations

We begin by evaluating the quality of inputs collected from the crowd and their effect on the quality of the downstream inference. While some methods used for comparing and evaluating single-target trackers analyze the sensitivity to noise in the initialization [54], [146], no work to our knowledge examines the effect of individual crowdsourced bounding boxes on the video object tracking application. In this evaluation, we discuss the distribution of crowdsourced bounding box initializations, failure modes of crowdsourced bounding boxes, and the initialization's effect on the output of the task model (DaSiamRPN [160]). Through this, we show that most of the initializations are high quality, many of those of apparent poor quality are actually initializations around an incorrect object, and that an initialization that has a high IoU with the gold-standard on the first frame may not produce a high-quality result (and vice-versa).

4.1.1 Data Collection

Using the interface and instructions shown in Figure 4.3, we asked workers on Amazon Mechanical Turk to provide an initial bounding box based on a text description of the target object (text descriptions are provided in Appendix A). We requested nine annotations for each of

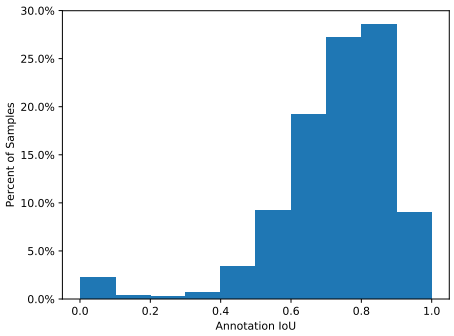


Figure 4.4: The histogram of first-frame IoU scores between crowdsourced and gold-standard annotations after removing malicious annotators. 93.1% of annotations have an IoU greater than or equal to 0.5, corresponding to a successful detection in the literature [99], [273].

the 100 videos of the OTB-100 dataset [145] and limited each annotator to one bounding box per video, but did not require that every annotator annotate all videos. Annotators were paid $0.06 per bounding box drawn, which is equivalent to approximately $12/hr based on timed data collections performed by the authors. All annotators were located in the United States.

Since bounding box annotations are a common task for crowd workers, we did not perform a qualification task. Instead, we filtered results from inattentive annotators by defining a correct annotation as one that has an IoU of greater than 0.5 with the gold-standard, consistent with designations used in object detection [99], [272], [273]. Since all annotators had an error rate of less than 15% or greater than 49%, we consider annotators for whom more than 15% of annotations were incorrect to be malicious or inattentive and remove them from our evaluation. While filtering through a comparison with the gold-standard initialization is not possible during deployment, this could be emulated via attention checks.

Overall, 26 unique annotators returned 899 of the 900 HITs with bounding boxes. Four annotators annotated more than 90 images, while 14 annotated fewer than 20 images. Our filtering process resulted in the elimination of four annotators who drew a combined 41 initial bounding boxes, resulting in evaluation being performed on 858 annotations.

4.1.2 Semantic Quality of Bounding-Box Annotations

While our filtering method eliminated inattentive or malicious workers, attentive workers still occasionally make mistakes. In Figure 4.4, we show the distribution of agreement between the accepted initializations and gold standard initializations in terms of IoU. Overall, we found that 93.1% of the filtered annotations met our definition of correct by having an IoU greater than 0.5, and most (55.4%) had an IoU with the gold-standard between 0.7 and 0.9. While a relatively small percentage (8.9%) fell within the top range of 0.9-1.0, previous work [54] has suggested that bounding boxes do not need to have an IoU near 1 to be perceptually similar, which we illustrate in Figure 4.5. Of the 858 accepted annotations, we

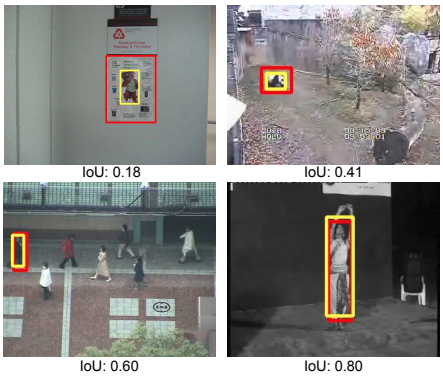


Figure 4.5: Example bounding boxes, showing perceptual similarity of various IoUs.

Figure 4.6: Examples of the three categories of annotation error. The red box represents the crowdsourced initialization, the green box represents the gold-standard initialization.

examined the 59 annotations that met our definition of incorrect to determine the failure modes of the human annotators. Broadly, we find the failure modes fit into the three categories shown in Figure 4.6: misidentified target, wrong region on target, and poorly fit. Overall, 15 (25.4%) were a misidentified target, 28 (47.5%) identified the wrong region of the target, and 16 (27.1%) were poorly fit.

4.1.3 Effect of Initialization on Tracker Performance

If we evaluate solely from the semantic perspective, the potential benefit of deferred inference looks low: since 72.9% of our semantically incorrect initializations were semantically correct for a different problem, only 27.1% of our semantically incorrect initializations (and therefore 1.9% of our total initializations) can be improved by deferral. However, the oracle assumption not only asserts that all human-provided information is high-quality, but that all high-quality information is equivalent. As we demonstrated in the previous chapter, this is not necessarily true. We discuss here the agreement—or lack thereof—between input error and output error, and this disagreement affects the deferral process.

Metric To quantify the effect of the initialization on subsequent frames, we used the additional error metric described in the previous chapter, with one modification: instead of calculating error across the full inference, we followed a similar procedure to Kristan *et al.* [54] and designated as valid frames where the track produced by the gold-standard initialization has maintained a continuous overlap with the gold-standard track. That is, all frames prior to the first frame with an IoU of zero are valid. The reason for this is shown in Figure 4.7, where processing on invalid frames results in a relatively low additional error, despite the candidate initialization having zero IoU with the

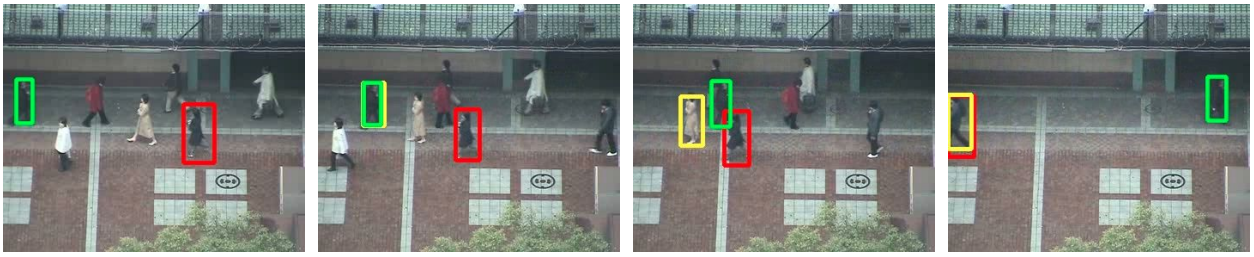


Figure 4.7: Calculating additional error exclusively on valid frames produces a more meaningful evaluation metric. Despite the candidate initialization (red) tracking a different object than the gold-standard initialization (yellow), additional error is low when all 174 frames are used, but high if only valid frames (frames 0-50) are used.

gold-standard. Put together, this means our performance measure is:

$$\ell(\hat{y}, y) = 1 - \frac{1}{N_v} \sum_{n=1}^{N_v} \frac{y_n \cap \hat{y}_n}{y_n \cup \hat{y}_n} \ , \tag{4.1}$$

where y_n represents the bounding box in track y at frame n. N_v represents the number of valid frames in the video, and we subtract this IoU from one to convert it to an error metric. From the previous chapter, our additional error is:

$$\text{AE}(\hat{y}_c, \hat{y}_{gs}, y | f, \ell) = \max(\ell(\hat{y}_c, y) - \ell(\hat{y}_{gs}, y), 0) \ , \tag{4.2}$$

Effect of Initialization on Performance We showed in Section 4.1.2 that the majority of crowdsourced annotations were of acceptable semantic quality and, of those that weren't, many of them were drawn around an incorrect object. Like the previous chapter, however, the goal is not to detect cases where the input is semantically incorrect, but to detect cases where a new human input can improve the inference. To evaluate this, we examine the relationship between initialization IoU and additional error in Figure 4.8.

This reinforces the finding that a deferral function must consider both whether there is error and whether this error can be reduced by a new human input. We see the need to consider the semantic quality if we inspect the leftmost four bins (initialization IoUs between 0.0 and 0.4): aside from two outliers, none of the initializations in these bins result in an additional error better than the third quartile of the 0.5-0.6 (the lowest IoU we regard as a correct initialization) IoU bin. Although high initialization error does relate to low output performance, this does not justify crowdsourcing techniques that rely on aggregation of multiple annotations (Section 2.2.2): 200 of our 858 (23.3%) of our initializations result in no additional error and, if we allowed one initialization per

video we could accept up to 22.93% ($\sigma = 2.75\%$) of initializations without increasing the additional error.

Although many initializations result in no additional error, it is also not a guarantee that high-quality initializations will result in high-quality outputs due to the sensitivity of the model. For example, the red box in Figure 4.8 shows instances where high-quality initializations result in high additional error. This demonstrates, like the previous chapter, that a deferral function must detect these cases instead of simply relying on the quality of the initialization. For a sufficiently high deferral rate, we would also need to consider the distribution of inliers: the additional error at the third quartile of the 0.7-0.8 IoU bin is near the median of the 0.6-0.7 IoU bin, meaning that it is better to defer the worst quarter of the 0.7-0.8 bin before the best half of the 0.6-0.7 bin.

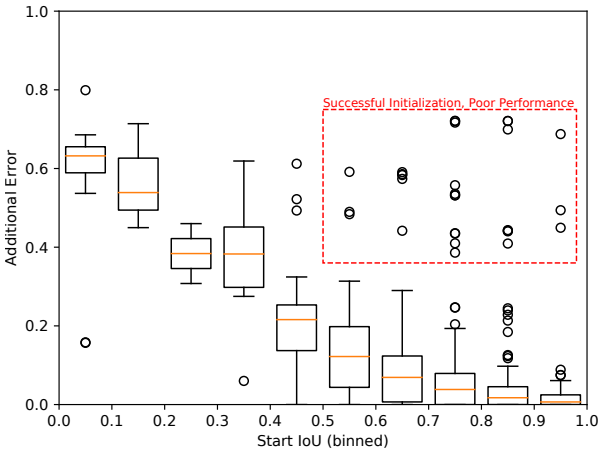


Figure 4.8: The IoU of the crowdsourced initialization with the gold-standard initialization is not the sole determining factor in an initialization's performance, as evidenced by successful initializations with poor performance.

These two findings support the idea that using crowdsourcing strategies that focus on the quality of the initialization in the input space will not perform optimally when the goal is to produce the best possible output. This is for two reasons: first, *annotations that appear good in the input space may still cause significant error on the downstream task*. Second, requiring multiple touches for every datapoint is an unnecessary expense, as *about 23% of initializations cause no additional error*.

4.2 The Importance of Considering the Deferral Response

In the previous section, we reinforced the findings of the previous chapter: we must consider both whether the human input is correct and the effect of that human input on the downstream inference. In addition, we demonstrated an insufficiency in the approach of the previous chapter: we must consider the quality of the deferral response in addition to the quality of the initial query. We do this by by comparing performance of four deferral functions under the assumptions of perfect and noisy deferral responses.

4.2.1 Evaluation Method

Metrics The evaluation metric we use depends on our assumptions about the deferral response. If we assume that the deferral response comes from an oracle, we use the MAE defined in the previous chapter:

$$\text{MAE} = \frac{\frac{1}{|\mathcal{D}|} \sum_{(x,h_c,h_{gs},y)\in\mathcal{D}} g(x,h_c)\,\text{AE}(\hat{y}_c,\hat{y}_{gs},y|\ell)}{\frac{1}{|\mathcal{D}|}\sum_{(x,h_c)\in\mathcal{D}} g(x,h_c)} \quad . \tag{4.3}$$

However, if we assume that the deferral response comes from a hazy oracle, we require a metric that measures not the error from the initial human input, but the error after the deferral response has been received. To do this, we first introduce the *aggregation function*, which accepts multiple human inputs and produces an estimate of the target value. The aggregation function is formalized as:

$$\hat{y}_c = h(x, h_{c1}, ..., h_{cn}|f) \quad , \tag{4.4}$$

and is used to calculate the *Replacement Mean Additional Error* (RMAE):

$$\text{RMAE} = \frac{1}{|\mathcal{D}|}\sum_{i=0}^{|\mathcal{D}|} AE(\hat{y}_c^{(i)}, \hat{y}_{gs}^{(i)}, y|\ell) \quad , \tag{4.5}$$

As in the previous chapter, both MAE and RMAE are dependent on coverage, requiring a summary metric. For MAE, this is the Area Under the Mean Additional Error coverage curve (AMAE)

$$\text{AMAE} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} \frac{\sum_{j=0}^{i} AE(h_c^{(j)}, h_{gs}^{(j)}, y^{(j)}|\ell)}{i} \quad . \tag{4.6}$$

In this chapter, we set the deferral depth constraint to one deferral per task. For the RMAE, we separate $\hat{y}_{c1}$, which is the output using only the initial human input, and $\hat{y}_{c12}$, which is the output using both the initial input and the deferral response. If these are sorted by the initial deferral score, the *Area under the Replacement Mean Additional Error coverage curve* (ARMAE) is calculated:

$$\text{ARMAE} = \frac{1}{|\mathcal{D}|^2}\sum_{j=0}^{|\mathcal{D}|}\sum_{i=0}^{|\mathcal{D}|}(i \geq j)AE(,\hat{y}_{c1}^{(i)}, \hat{y}_{gs}^{(i)}, y|\ell) + (i < j)AE(\hat{y}_{c12}^{(i)}, \hat{y}_{gs}^{(i)}, y|\ell) \quad , \tag{4.7}$$

Since we collect nine initializations for every video, and the AMAE metric requires one initialization for each of the 100 videos, we take 1,000 samples of the approximately 9^{100} potential combinations of initializations and select our deferral response without replacement. These samples are then used to calculate the mean and standard error of the evaluated deferral functions.

Deferral Functions In this chapter, we evaluate the following four deferral functions:

1. Tracker Confidence: The Distractor-Aware Siamese Region Proposal Network architecture [160] that we use for our experiments returns a confidence score, which corresponds to the confidence that a bounding box contains the object which is being tracked. In the tracking algorithm, this score is used to select between potential bounding boxes and to determine when a track has been lost and re-acquired for long-term tracking. We used the mean of this tracking score over all valid video frames to produce a single metric.
2. Regression of per-frame IoU: While it is not tractable to train a model to predict the additional error directly due to the large number of non-parallelizable inferences that would need to be performed for a single training step, it is possible to predict the IoU of a bounding box on a given frame, similar to the work of Gurari *et al.* [275]. We formulated this regression as a classification problem that attempts to predict which of 10 evenly spaced IoU bins the bounding box has with the gold-standard, which is hidden from the classifier. We used a pretrained ResNet-18 [276] backbone, added a fourth input channel that accepts a binary mask representing the candidate initialization, and trained the model using perturbed bounding boxes on the MSCOCO [15] dataset. The predicted IoU on a single frame is the mean of the classifier's output distribution, while the deferral score is the mean across all valid frames.
3. Cycle Consistency: Proposed by Wu *et al.* [145] to evaluate trackers without a ground-truth annotation, cycle consistency appends the reversed video to the end of the forward video, and runs the tracker across this forward-backward video. The final bounding box is compared to the initial bounding box, and the more they agree, the higher the predicted quality of the track. The original work used the distance between the prior and posterior densities as the comparison, but since this is not compatible with modern methods, we used the IoU between the initial and final bounding boxes.
4. Cycle Consistency + IoU Regression (Combined C+I): The cycle consistency score discussed above has a lower bound of zero, which occurs for any initialization that results in a lost track. This means that no distinction is made between tracks that are lost early in the video and tracks that are lost late. To compensate for this, we separated cycle consistency scores into "hit" and "miss" bins based on the previously defined threshold of 0.5. The hit bin is accepted in order of the cycle consistency score, then the miss bin is accepted in the order given by the IoU regressor.

	AMAE	Rank	ARMAE (Naive)	Rank
Combined C+I	**0.06541 ± 3.88e^{-4}**	**1**	**0.08710 ± 2.61e^{-4}**	**1**
IoU Regression	0.08178 ± 4.80e^{-4}	4	0.08792 ± 2.52e^{-4}	2
Cycle Consistency	0.06735 ± 3.79e^{-4}	2	0.08839 ± 2.65e^{-4}	3
Tracker Confidence	0.07070 ± 3.57e^{-4}	3	0.08896 ± 2.63e^{-4}	4

Table 4.1: Performance under the assumption of an oracle deferral response. Mean and standard error are calculated using 1000 sets of the 100 videos where the initialization is randomly drawn.

4.2.2 Assuming an Oracle Deferral Response

We begin by evaluating under the assumption that the deferral response is perfect both when it is correct (calculating the MAE), and incorrect (calculating the RMAE). To maintain the assumption of a perfect deferral response when using the RMAE (which requires an aggregation function), we assume that an inference can only be deferred once and always use the deferral response. We call this *Naive Replacement*, and it can be written:

$$h(x, h_{c1}, h_{c2}|f) = f(x, h_{c2}) \ . \tag{4.8}$$

We see the performance under this assumption numerically in Table 4.1, and visually in Figures 4.9 and 4.10. Under the oracle assumption—both when it is correct and when it is incorrect—the Combined C+I method performed best, likely due to the complementary behavior of the IoU regressor and cycle consistency metric. Examining the AMAE plot (Figure 4.9), we see that the cycle consistency metric performed very well at low coverages—it detected when a track was high quality—but it could not distinguish between low quality inferences at high coverages. In contrast, the IoU regressor could determine the exact frame where a track was lost, differentiating between different low-quality tracks in a way that the cycle consistency metric could not, but struggled with accuracy at low coverages.

Examining the behavior when the deferral response is not provided by an oracle, we note first that the ARMAE was substantially higher than the AMAE simply because error contributed by the replacement initializations is not considered when calculating the AMAE. Although both methods had the same best performer, the IoU regressor outperformed cycle consistency on the mean—going from worst to second best—under the updated assumptions. We speculate that this is due to the fact that cycle consistency is excellent at detecting lost tracks that are related to qualities of the video such as occlusion, motion blur, or fast motion. Due to this, there is a high likelihood that replacing an initialization that causes a lost track will still result in a lost track because these methods do not adequately separate the causes of error. The IoU regressor, on the other hand, detects lost tracks less reliably than the cycle consistency metric, but is more sensitive to failure modes that are video independent, meaning the samples it defers first are highly likely to benefit from being replaced. In

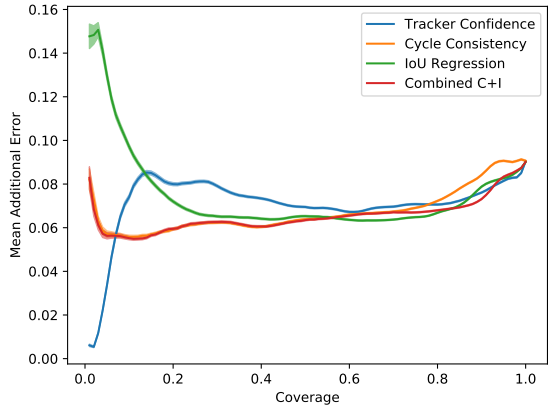


Figure 4.9: The mean additional error for all methods and coverages. Shaded region represents one standard error.

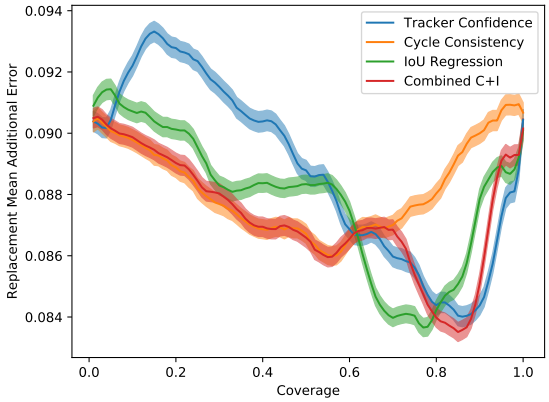


Figure 4.10: The RMAE under the oracle assumption (*Naive Replacement*). Note the minima near 85% coverage.

other words, when the deferral response is hazy the qualities we need to look for when choosing tasks to defer change.

We also highlight that three of our four methods had minima near a coverage of 0.85 when evaluating with the RMAE. At this high coverage, these methods identified poorly performing initializations well, which resulted in replacement initializations having a high likelihood of outperforming the first initialization. As the coverage decreased (we replaced more and better initializations), it was no longer a near guarantee that the replacement initialization would provide better performance, and the replacement initializations often performed worse than the initialization they were meant to replace. In fact, under the conditions of this experiment the error at zero coverage will always be the same as the error at coverage one. This is further supported by the fact that cycle consistency is the sole deferral function that did not have this minimum, since it does not have any discriminative power within the 20% of samples for which it returned a score of zero.

4.2.3 Smart Replacement: Simple Mitigation of Hazy Deferral Responses

The previous section demonstrated two things: first, the assumptions we make about the deferral response affect which deferral function is best and second, incorrectly assuming a perfect deferral response will cause performance to get worse with more deferrals. The former is important when selecting deferral functions, and the latter is undesirable because it simultaneously increases human effort and decreases performance. To address the former, we simply need to choose our deferral function under the appropriate condition, while for the latter we introduce the aggregation function of *smart replacement*:

$$h(x, h_{c1}, h_{c2}|f) = (g(x, h_{c1}) \leq g(x, h_{c2}))(f(x, h_{c1})) + (g(x, h_{c1}) > g(x, h_{c2}))(f(x, h_{c2})) \quad (4.9)$$

	ARMAE
Combined C+I	0.08600 ± 2.96e^{-4}
IoU Regression	0.08796 ± 2.82e^{-4}
Cycle Consistency	0.08843 ± 3.05e^{-4}
Tracker Confidence	**0.08205 ± 2.91e^{-4}**

Figure 4.11: ARMAE of various smart replacement methods, where the deferral and aggregation functions are the same.

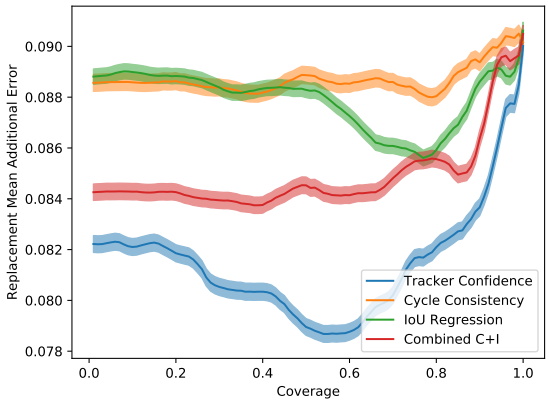


Figure 4.12: RMAE-Coverage curves for the case where the deferral and aggregation functions are the same.

where $g(x, h)$ returns the deferral score instead of a deferral decision (that is, $g(x, h) \in \mathbb{R}$ instead of $g(x, h) \in \{0, 1\}$). Put more simply: smart replacement inspects the initial query and the deferral response, and uses the one it thinks is better.

We begin by using the same scoring function for both deferral and aggregation. The results of this are shown in Figure 4.12 and Table 4.11. Notably, IoU regression and cycle consistency did not perform significantly better when using smart replacement in place of naive replacement, while the C+I and Tracker Confidence methods did. The former two cases—where smart replacement does not improve the overall performance—are of particular interest: why would it not be beneficial to use the better of the two initializations?

To answer this question, we inspect Figure 4.12. While these two methods outperformed naive replacement at zero coverage—meaning they were better than random at differentiating between good and bad initializations for a given video—their RMAE never gets as low as that of naive replacement (Figure 4.10. In other words, they were not sufficiently better than random when the quality of the track was poor. A similar inspection shows that Combined C+I converged near the minimum RMAE of naive replacement, while Tracker Confidence converged below the minimum reached by naive replacement, even though it still reaches a local minima prior to zero coverage. The improved behavior of tracker confidence due to its ability to evaluate different tracks for the same video further reinforces the need to evaluate under the correct assumptions: tracker confidence went from being the worst performer under the oracle assumption (naive replacement) to being the best performer under the hazy oracle assumption (smart replacement).

	ARMAE
Combined C+I	0.08269 ± 2.94e^{-4}
IoU Regression	0.08351 ± 2.90e^{-4}
Cycle Consistency	0.08294 ± 2.89e^{-4}
Tracker Confidence	**0.08205 ± 2.91e^{-4}**

Table 4.2: ARMAE of various deferral functions where the tracker confidence is used as the aggregation function.

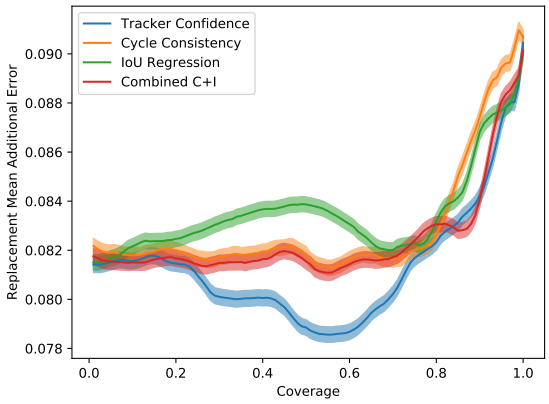


Figure 4.13: RMAE-Coverage curves for the case where tracker confidence is used as the aggregation function.

Tracker Confidence as Aggregation Function Since tracker confidence had the best ability to select between initializations, we evaluated all four deferral functions using tracker confidence as the aggregation function. Ideally, this combines the best ability to locate low quality tracks (combined C+I) with the best ability to determine the better of two tracks for a video (tracker confidence) to create the best overall performance. However, as we see in Figure 4.13 and Table 4.2, this was not the case.

All of the methods were significantly improved by using tracker confidence as the aggregation function, returning lower ARMAEs and having a lower RMAE at zero coverage, but using tracker confidence as both the deferral function and aggregation function performs significantly better than other deferral functions despite the intuition of inter- and intra- video discrimination being separate. In other words, tracker confidence as a deferral function defers inference only when it can differentiate between high and low-quality initializations.

4.3 Conclusion

While the deferral function is a critical component in deferred inference, it is only the first step: we must consider not only whether the model can benefit from a better human input, but also what happens when the deferral response is also subject to noise. We examined this question on the application of crowdsourced single-target video object tracking and found that while most initializations were semantically correct, it is inefficient to aggregate information to confirm that an initialization is semantically correct and, as in the previous chapter, a semantically correct initialization does not guarantee a high-quality output.

We then evaluated candidate deferral functions under the assumption that the deferral response

will be of high quality under both the AMAE metric—where it is true—and the RMAE metric—where we consider the deferral response to be subject to noise. Critically, we find that the relative performance of our deferral functions changes when the deferral response is taken into account and that, if care is not taken, error and human effort may rise simultaneously. We address this using the method of smart replacement, which chooses between the best of two potential inputs.

Although our evaluation and method of smart replacement demonstrated the importance of considering the deferral response, it is somewhat incomplete: it does not compensate for deferral depths greater than one, and does not consider cases where incomplete but complementary behavior is provided by the human. In the next chapter, we temporarily ignore the role of the deferral function and focus on a crowdsourced setting, where we must compensate for human inputs that are both ambiguous and noisy by intelligently aggregating them across time and annotators.

CHAPTER 5

Probabilistic Aggregation of Human Inputs

Previous chapters showed the importance of acknowledging that humans are hazy oracles, both with respect to the initial query and the deferral response, and addressed this via smart replacement, an aggregation function that chooses between the deferral response and the initial query based on which it believes will perform better. Although better than naively accepting the deferral response, the approach of smart replacement will be insufficient if both human inputs are ambiguous. In these cases, we must combine complementary pieces of human-provided information to produce an optimal answer.

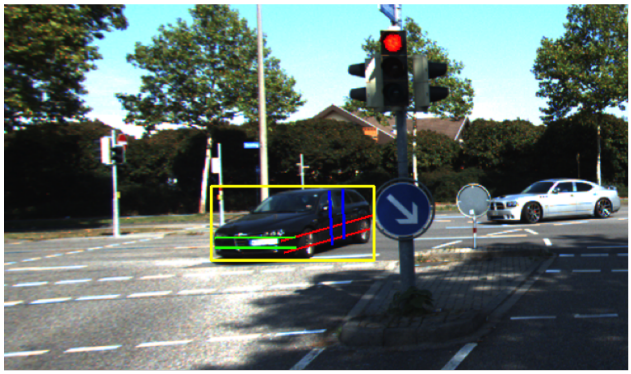


Figure 5.1: Annotating pose estimation is a challenging task for crowd workers, yet drawing bounding boxes (yellow) and length, width, and height lines (red, green, and blue) is straightforward. We propose a method that enables this by intelligently aggregating across annotators and video frames.

In this chapter, we set ourselves in the crowdsourcing domain by assuming that multiple human inputs will be available for every task, and a deferral function is not necessary. This approach is a major driver of artificial intelligence: an overwhelming number of deep learning works in computer vision have at least pre-trained on ImageNet [272], while task specific crowdsourced datasets exist for domains as diverse as birds [261], road sign text [27], dialog-based navigation [277], and others [57], [231], [278].

The domain of the data collection proposed in this chapter is rare traffic events. While autonomous vehicles collect large quantities of training data by operating in their target environment [279], uncommon events such as traffic accidents [280] are underrepresented in datasets for two reasons. The first reason is magnitude: in 2018 Waymo's autonomous research vehicles [281] traveled and recorded approximately 25,000 miles every day on public roads [282], while Americans drive a total of nearly three trillion miles every day [280], a factor of 120 million. Second, autonomous vehicles are specifically designed to avoid such critical roadway incidents, which saves money and maintains public trust, but does not allow us to acquire training and evaluation data.

Fortunately (from a data perspective), while autonomous vehicles avoid traffic accidents, humans collect them for education or entertainment: at the time of writing, a search for *traffic accident* on YouTube returns approximately 64,000 results. Creating realistic simulated 3D scenes from such an abundant source of existing data is a more reasonable—and safe—method for training and evaluating such rare events at scale, but lifting 2D monocular videos to 3D scenes is a non-trivial challenge: manual annotations are generally still necessary to resolve ambiguities that automated methods cannot resolve on their own, and processes for providing such annotations are time and knowledge intensive. For example, the annotation process of the PASCAL3D+ dataset [105] is as follows:

> The annotator first selects the 3D CAD model that best resembles the object instance. Then, he/she rotates the 3D CAD model until it is aligned with the object instance visually. [...] Based on the 3D geometry and the rough pose of the CAD model (after alignment), we compute the visibility of the landmarks. After this step, we show the visible (not self-occluded) landmarks on the 3D CAD model one by one and ask the annotator to mark their corresponding 2D location in the image. For occluded or truncated landmarks [...] we assign a visibility state to landmarks that we identify for each category: 1) visible: the landmark is visible in the image. 2) self-occluded: the landmark is not visible due to the 3D geometry and pose of the object. 3) occluded-by: the landmark is occluded by an external object. 4) truncated: the landmark appears outside the image area. 5) unknown: none of the above four states.

For this reason, we argue that crowdsourced pose estimation requires a task model that can translate intuitive annotations—height, width, and length lines—into a 4D (X, Y, Z, yaw) pose. Although this input modality has a straightforward analytical solution, this analytical solution is quite sensitive to noise: as we see in Figure 5.2, an analytic solution given a three-pixel error in the 2D annotation of vehicle height will return an estimate with a 26-meter error in 3D position estimation.

We therefore implemented several layers of quality control. In the input space, we perform statistical outlier detection and allow workers to selectively skip annotations that they have low confidence in the accuracy of (which we call *self-filtering*) [236]–[238]. Self-filtering can be particularly useful in annotation of unstructured video, as it may be impossible to generate the correct annotation due to factors such as motion blur, angle of view, or truncation [283]. While increasing the accuracy of collected annotations, this filtering comes at a cost: for example, it will create an under defined problem if all workers self-filter the same input.

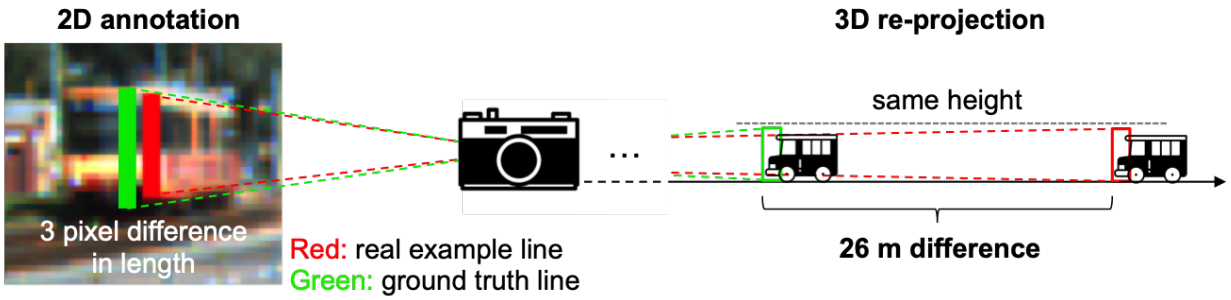


Figure 5.2: A small pixel error in 2D can be amplified in 3D, resulting in a severe position error. The vehicle image on the left shows a crowdsourced *height entry* dimension line annotation (in red) and the corresponding ground truth (in green). The z-dimension estimate can be calculated from the focal length and the object's actual height, which was 721 pixels and 3.59 meters in our experiment, respectively. The three-pixel difference in dimension line leads to a 26-meter difference in 3D location.

To address this, we introduce Popup, a novel hybrid human-AI pipeline for 3D video reconstruction shown in Figure 5.3. Popup is centered around a particle filter-based aggregation function that mitigates ambiguity and noise by aggregating across both annotators and timesteps. Popup consists of three main components: (1) dimension line annotation with self-filtering, (2) outlier filtering of submitted annotation sets, and (3) particle filtering based estimation of the 3D position and orientation.

We validate this method on videos from the KITTI dataset [278], and show that our proposed approach reduces the relative error by 33% in position estimation compared to a baseline condition that does not aggregate human-provided annotations across time. We additionally highlight that this aggregation function is robust to missing annotations, where the baseline method would fail due to the problem being underdetermined. Last, because Popup's ability to resolve ambiguity enables self-filtering, annotation time for challenging frames can be reduced by 16%.

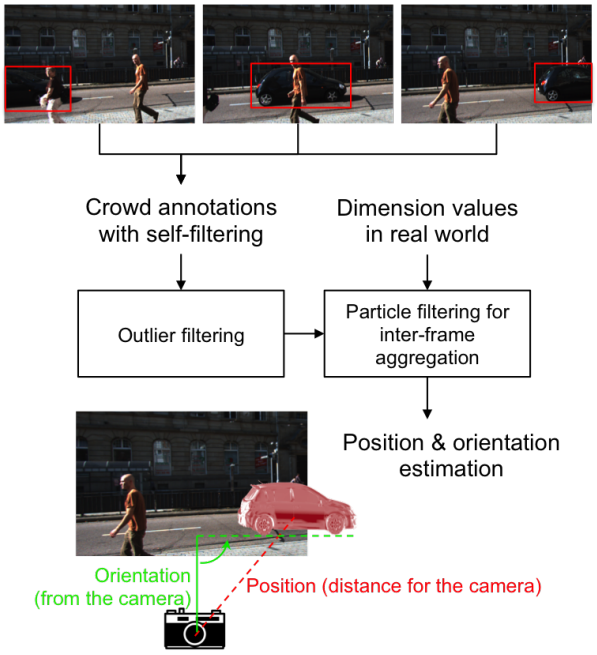


Figure 5.3: Overview of Popup. From workers' dimension line annotation input and additional input of real-world dimension values of the target vehicle (looked up from an existing knowledge base), Popup estimates the position and orientation of the target vehicle in 3D.

5.1 Method

Popup consists of three parts: first is the annotation tool, which allows annotators to crop the target vehicle, draw dimension lines, and self-filter uncertain annotations. The second part is a pair of automated statistical filters in the input space that remove low-quality annotators and low-quality annotations from otherwise high-quality annotators. Last, the data that is not removed by the semantic filter data is passed through a particle filter, which allows us to resolve ambiguity by aggregating information across annotators and timesteps.

5.1.1 Dimension Line Annotation Tool and Self-Filtering

The interface of Popup consists of a visualization and annotation web application that allows crowd workers to crop the object of interest from a video frame, then draw length, width, and height lines on the cropped object. This interface allows annotators to operate directly on video frames to capture the 3D state of an object without any complicated three dimensional interactions (*e.g.,* rotation and scaling of a cuboid) that would require familiarity with interactive 3D tools.

When a crowd worker reviews the dimension line annotation task, an explanation of the goal of the task is first given (Figure 5.4-①). Then, step-by-step instructions are provided, along with pictures exemplifying desired and undesired annotations as in Figure 5.4-②. The instructions ask workers to click the "I cannot draw" button whenever they are not confident in their ability to accurately annotate a particular dimension (Figure 5.4-③). Once the worker accepts the task, they can perform the first step: cropping the target object. The worker can click and drag on the given video frame to draw a box, and adjust the size and ratio of the box, as needed. The coordinate information of the box is used in the post-hoc outlier filtering step, as explained in the next section. Once the worker is done cropping the target object, they can click the 'Done with Step 1' button and proceed to the next step. Note that a worker annotates one frame at a time. The rate of frames to be annotated can be arbitrarily chosen by the user.

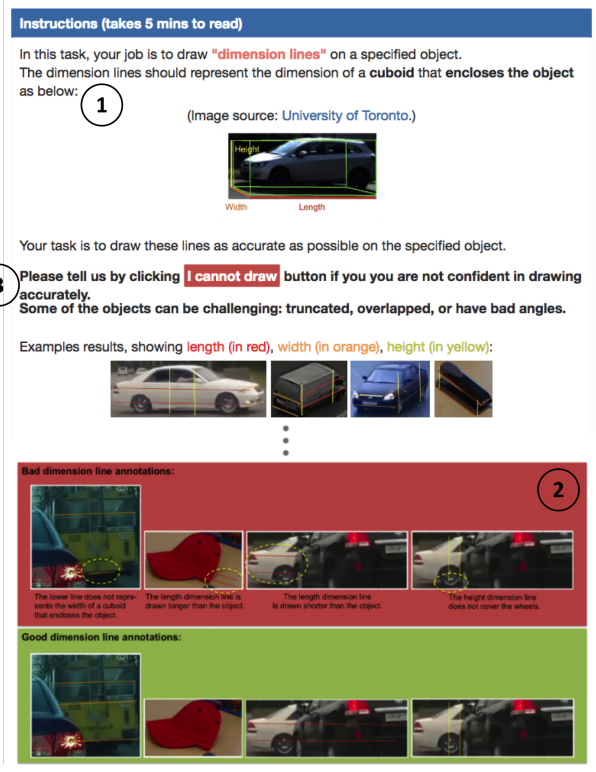


Figure 5.4: Step-by-step instructions with good and bad examples are provided.

Figure 5.5: Interactive Web UI that crowd workers can use to create, adjust, erase, and redraw *length, width* and *height* lines.

The second step is drawing the dimension line entries (length, width, and height) on the cropped vehicle. The interface has buttons that open a pop-up window to allow workers to draw dimension line annotations for each dimension entry. Workers can choose which they want to draw first. The interactive pop-up window is shown in Figure 5.5. After drawing a line, a message appears at the end of the line and asks workers "Is this end closer to the camera than the other end of the line?" The worker can answer this using a radio button. We describe this step because it affects the total time of completing the dimension line annotation task: while we we initially asked this question to avoid ambiguity due to the Necker cube illusion [284] that makes it impossible to distinguish the closer ends of the edges of a cube, we found that answers varied too much to use for pose estimation. This variance is likely due to the number of ambiguous cases that arise, such as when a car is nearly 90-degrees to the camera, making it hard to perceive which horizontal end is closer to the camera. The interface asks workers to draw more than one line per dimension in order to proceed to the next step. The interface allows adjusting already drawn dimension lines or redrawing them anytime if needed. Workers are provided with the "I cannot draw" button (Figure 5.5-(2)) which they can click on to self-filter dimension line annotations if they are not sure about their answer.

5.1.2 Automated Outlier Removal

Popup is designed to robustly handle aggressively-filtered annotation sets via two post-hoc filtering modules. The post-hoc modules assume multiple submissions per frame so that distribution statistics can be found.

Filtering Annotation Sets The first step calculates the median bounding box location of submissions to filter incorrect annotation sets (all dimension lines from one annotator). For each target object, the worker crops the object of interest from the given frame. Our assumption is that a malicious worker, careless worker, or bot will fail to crop the correct target object. For width and height independently, if a cropped box does not overlap more than 50% with the median of the cropped boxes, we assume the worker annotated the wrong object and drop the annotation set of all three entries (length, width, and height). This is designed to entirely filter poor submissions.

Filtering individual Annotation The second step compares the distance of the length and angle of submitted dimension line annotations from the medians. If a dimension line is outside $1.5\times$Interquartile Range (IQR) from the median, it is filtered. This is useful for filtering out low quality annotations and mistakes such as a height entry mistakenly drawn as a length entry. We filter based on relative distances instead of absolute values because the size of an object can differ from 30 pixels to 300 pixels.

Aggregating Annotations via Particle Filter To prevent the AI from amplifying the effect of annotation errors, it is necessary to remove annotations that are low quality in the semantic space. However, removing annotations can lead to semantic ambiguity, meaning we must use an aggregation function that can compensate for this. To do this, we use a particle filter to leverage not only multiple annotators, but also the temporal coherence of video data: while some information may be unavailable in certain cases (*e.g.,* a vehicle driving away has no length), that information may be available in future frames (*e.g.,* when the vehicle turns).

Particle filtering is a recursive Bayesian method that estimates a probability density function across a state space by maintaining a large set of particles, each of which represents a potential state ("hypothesis") [285]. Particle filters are commonly used in simultaneous localization and mapping (SLAM) systems [286], as well as face [287], head [288], and hand tracking [289] systems. This state estimation method has three main advantages in our setting: first, particle filters can utilize information from neighboring state estimates in tandem with temporal constraints (e.g., the object has a maximum speed) to refine the state estimate. Second, particle filters can support the complex measurement functions that are required to compare 2D annotations and 3D states. Last, the particle filter does not assume an underlying distribution, which allows it to maintain multimodal and non-Gaussian distributions. This is particularly useful for ambiguous data, as incomplete annotations permit multiple correct hypotheses. A particle filter can be thought of in three steps:

1. The *previous distribution* is the distribution of hypotheses at time $t-1$.
2. The *transition distribution* is the distribution of hypotheses at time t given the distribution at time $t-1$.
3. The *measurement distribution* is the transition distribution, updated to include evidence obtained at time t.

Popup embodies these three probability distributions as follows:

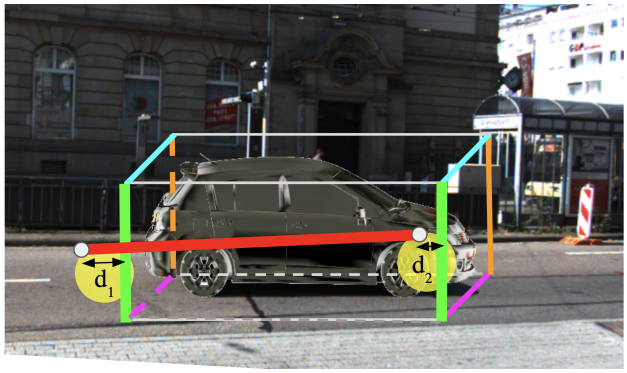


Figure 5.6: Perceptual distance calculation. The distances (arrows) between endpoints (grey dots of the red line) of an annotation (red line) and corresponding projected hypothesis 3D line pairs (orange, green, blue, pink) are calculated. The distances corresponding to the best-fitting 3D line pair are used to calculate probability.

(1) Previous Distribution The previous state distribution corresponds to the final distribution of the previous time step. As we do not have any information about the vehicle's initial pose, we set the distribution at $t = 0$ as uniform within the bounds described in Experimental Setting section.

(2) Transition Distribution The transition distribution describes the probability of a particle being in a new location given its previous location. This distribution allows the filter to maintain knowledge of potential states across time, which has two important implications: first, it means a fully determined system is not necessary at every time step, so the system is tolerant to both self- and automated filtering with aggressive thresholds. Next, it applies a spatiotemporal constraint by limiting how far the vehicle can move in successive frames, narrowing the solution space. Typically the transition distribution is based on knowledge of the vehicle's kinematics and control inputs, but we use uncorrelated zero-mean Gaussian noise that spans set of reasonable vehicle motions because our system has no knowledge of the vehicle's control signals.

(3) Measurement Distribution The measurement distribution begins by projecting the cuboid produced by the hypothesis into the 2D space of the image. It then measures how close its endpoints are to an appropriate pair of edges (Figure 5.6). This distance is placed on a normal distribution with a mean of 0 pixels and a standard deviation of 22 pixels. We also calculate the difference between the lengths of the annotation line and corresponding projected hypothesis line, and place that on a normal distribution with a mean of 0 pixels and a standard deviation of 22 pixels. The sum of these two probabilities is used as the probability of an annotation. This function is referred to as ERR in Algorithm 1.

Implementation The pseudocode for our particle filtering implementation is shown in Algorithm 1. The state space consists of five dimensions: x, y, z, θ, and f, where x, y, z denote the relative 3D position of an object from the camera and θ denotes the orientation as illustrated at the bottom of Figure 5.3. The last dimension, f, denotes the focal length of the camera. $RESAMPLE(S)$

Algorithm 1 Particle filter algorithm for Popup

Let $S = \{(s_1, w_1) \dots (s_N, w_N)\}$ be the set of N particles, where each particle $s_i = \{x_i, y_i, z_i, \theta_i, f_i\}$ is one hypothesis with probability $P(s_i) = w_i$. Let the initial set of particles S_0 be sampled uniformly from the given range for s_i:

for Every Frame t **do**
  $RESAMPLE(S)$
  for Every (s_i, w_i) in S **do**
    Next State Step: $s_{i,t} \leftarrow s_{i,t-1} + \mathcal{N}(0, \sigma)$
    $z \leftarrow 0$
    for Every Annotation Line **do**
      $z \leftarrow z + ERR(AnnotationLine, ParticleState)$
    end for
    $w_{i,t} = w_{i,t-1} \cdot z$
  end for
  $S \leftarrow NORMALIZE(S)$
  $estimate \leftarrow ARGMAX(w_i)$
end for

generates N particles (potential states) based on the existing particles and their probabilities (w). $NORMALIZE(S)$ normalizes all the updated probabilities (w) calculated in the previous for-loop such that the probabilities sum to one. When analyzing across a single frame, we perform the measurement and resampling steps after iterating through every annotation set.

5.2 Evaluation

We perform our analysis on annotations provided by 170 workers recruited from Amazon Mechanical Turk via the LegionTools [290] toolkit. All workers were located in the U.S. and had an approval rate of over 95%. All workers had to first read the instructions to proceed to the task. Evaluation was performed using data from the KITTI dataset [278], which contains traffic scenes recorded from a moving vehicle using multiple sensor modalities. The scenes include occluded, truncated, and cropped objects, as well as ground-truth measurements of distance and orientation of the objects, making it ideal for evaluating Popup on challenging, real-world scenarios. We excluded clips with no vehicle or vehicles that do not span our sampling rainge, resulting in 17 or the 21 KITTI clips being used. In each video, we targeted one object and sampled 10 frames from each video clip at a rate of two frames per second. For each video clip, we recruited 10 workers to provide annotations. Each worker annotated every other sampled frame, for a total of five frames. That is, for each frame, annotation sets from five different workers were collected. Each worker was paid \$1.10 per task, a pay rate of ∼\$9/hr. To understand the reason why some annotations were self-filtered, we presented the workers with a multiple-choice question when an annotation was

self-filtered. The choices were: 1) "The object is heavily occluded", 2) "I don't understand the instruction", and 3) "Other". We asked the workers to still draw the dimension line *after* reporting "I cannot draw" to directly compare accuracy with and without workers' self-filtering.

5.2.1 Semantic Analysis and Filtering

Filtering Annotation Sets The first outlier filtering step removed low-quality annotation sets (all dimension lines from one worker) based on bounding box coordinates of each submission. 7% of 850 submissions were filtered in total. We found that few incorrect objects (under 2%) still remained after the filtering step, which occurred when the majority of workers (at least three out of five) annotated an incorrect object.

Video #1, frame #3: **4 out of 5 workers self-filtered 'length' entry.**
Reason: No side view.

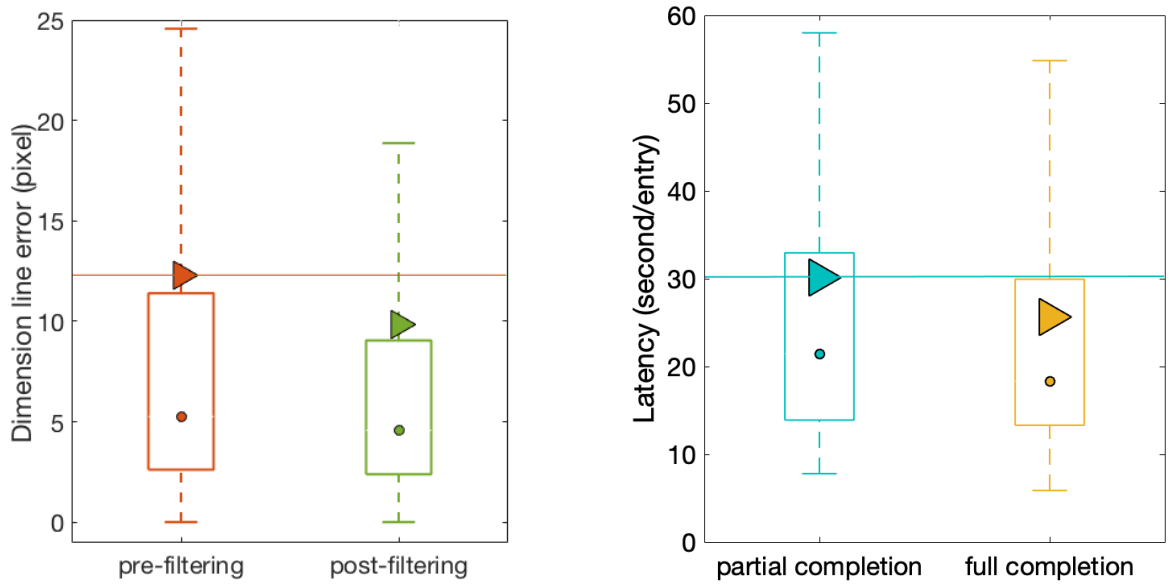
Video #5, frame #10: **4 out of 5 workers self-filtered 'width' entry.**
Reason: Occluded.

Video #9, frame #2: **all 5 workers self-filtered 'length' entry.**
Reason: Low resolution.

Figure 5.7: Example frames where more than three out of five workers self-filtered. The cases include limited side view, occlusion, and low resolution.

Result of Self-Filtering After the first step of outlier filtering, 793 annotation sets remained in the collection. Each annotation set has three dimension entries (length, width, and height), resulting in a total of 2379 entries submitted. 176 (7%) of these were self filtered. Of the self-filtered entries, 34% were filtered for the reason "The object is heavily occluded", and 66% were filtered for the reason "Other". There were no instances where the "I don't understand the instruction" option was chosen. When the "Other" option was chosen, workers could manually enter the reason behind their decision. Most explanations were related to insufficient visual information, e.g., "the object runs off the given image", "it's mostly back view", and "Bad angle, low resolution" as shown in Figure 5.7. We initially expected a higher self-filtering rate because many of our selected scenes contained objects that are hard to annotate (*e.g.,* truncated or occluded).

We believe that this discrepancy between expected and actual self-filtering rate is due to workers



(a) Height dimension-line error before and after filtering. The circle denotes the median and the triangle denotes the mean.

(b) Average latency when at least one worker (partial completion) self filtered compared to none (full completion). The circle denotes the median and the triangle denotes the mean.

Figure 5.8: Results of filtering annotations and dimension lines.

not being properly informed about or incentivized for self-filtering. In our post survey, we asked workers who completed our task if they think it is better to provide an answer or not when they are not confident about the answer being correct. One worker answered, "I think an attempt at an answer is better than none at all. Even if you aren't sure an attempt at least shows your [*sic*] trying to help the study and not just wasting everyone's time". Another answered, "Try my level best to satisfy the requester". The survey response tells that crowd workers are willing to help the requester but they might not know what is most helpful, resulting in them submitting low-confidence annotations even when they should be self-filtered. Therefore, providing clear instructions on how to benefit the task would lead to better usage of self-filtering. An appropriate incentive mechanism may also help: Shah et al. [237] gave a clear incentive to the workers, which encouraged them to use the self-filtering option ("I'm not sure") wisely. This resulted in the highest data quality in their experiments. Thus, requesters should clearly design an incentive mechanism and mention in the task how they would like workers to use the self-filtering option.

Result of Filtering Individual Annotations In the final outlier filtering step, we filtered individual annotations based on the dimension line's length and angular distance from the median. Of the individual annotations, 13% were considered outliers and filtered from the collection. We found

that a few (under 3%) outlier annotations did not get filtered with our method. These were cases where the object was relatively small in the scene, and the variance within good annotations was very close to the difference between good and poor annotations.

Accuracy of Dimension Line Annotations We examined the effect of our input-space filters on the average accuracy of dimension line annotations. Since the dimension line ground truth is not provided by the KITTI dataset, we projected the actual vehicle height of the target vehicle onto the image plane, and compared the difference from the projected height line with the annotated dimension line in pixel units. This analysis was not performed on width and length dimension lines as they are not parallel to the image plane. In our experiment, the distributions were all approximately normal, but with positive skew. Because the distributions were skewed, we computed p-values using Wilcoxon Rank-Sum test. As shown in Figure 5.8a, the pre-processing filtering reduced the average error of dimension lines by 20% ($p < .05$) on average. Note that the mean error after filtering is under 10 pixels (9.8 pixels). Given the frame heights are 375-pixel, the average error is under 3% of the full height of a video frame.

Time Savings from Self-Filtering We investigated the average latency of partially- and fully-completed sets of annotations. Because the distributions were skewed normal, we computed p-values using Wilcoxon Rank-Sum test. As shown in Figure 5.8b, we found annotations took approximately 16% longer for annotations where at least one worker self-filtered ($p < .005$). The result suggests two things: first, self-filtering can reflect a worker's confidence level as we intended in the design stage. This can be inferred by the fact that it took significantly more time for those who did not self-filter the entry, implying that it was also more challenging for them. Second, we can reduce total latency in annotation collection if we encourage workers to self-filter the challenging entries, because they can save time on the drawing activity by skipping them. In this experiment, we could save 16% of the annotation time for the hard annotations when workers self-filtered annotations.

5.2.2 Analysis of Aggregation Function

In this section, we evaluate our proposed aggregation function under different conditions by comparing it to the ground truth from the KITTI dataset [278]. For all evaluations, we dropped outliers: any data point outside $1.5\times$Interquartile Range (IQR) was removed for fair comparison between conditions.

The true 3D dimensions of the annotated vehicles were drawn from the ground-truth information included in the KITTI dataset. In a real-world deployment of Popup, the dimensions would be found online or in appropriate documentation. For our particle filter, we set the following bounds:

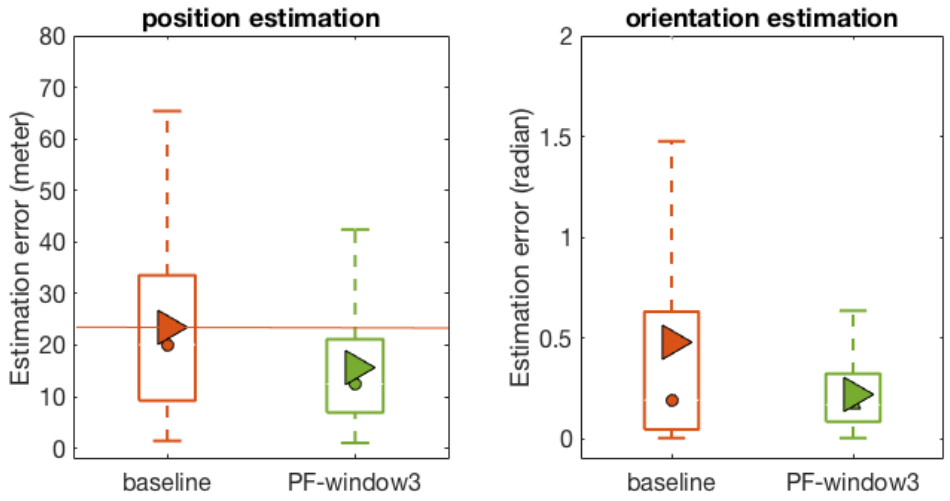


Figure 5.9: Estimation error of our baseline and proposed method.

$-30 \leq x \leq 30$, $-4 \leq y \leq 4$, $1 \leq z \leq 140$, $0 \leq \theta < \pi$, and $500 \leq f \leq 1000$. Position is given in meters, orientation in radians, and focal length in pixels. We used 50,000 particles for all the particle filtering based conditions.

Evaluation Metrics Our evaluation considers separately two errors: distance, measured by the Euclidean distance between the ground truth and the estimate, and angular error, measured as the smallest angular difference between estimated orientation and the ground truth orientation:

$$\Delta D = \sqrt{(x_g - x_e)^2 + (y_g - y_e)^2 + (z_g - z_e)^2} \tag{5.1}$$

$$\Delta\theta = |(\theta_g - \theta_e) \; mod \; \pi/2| \tag{5.2}$$

where x_g, y_g, and z_g are the 3D ground truth position, x_e, y_e, and z_e are the estimate, θ_g is the ground truth orientation, and θ_e is the orientation estimate.

Baseline The baseline method reprojects a 3D cuboid onto a given video frame and compares the corner location of the reprojection with the endpoints of the average dimension lines drawn for the target vehicle. This comparison is used as the cost function, and the L-BFGS-B optimization method [291] is used for minimization. L-BFGS-B is a well-studied optimization algorithm that is used in the state-of-the-art techniques for estimating distributions in various applications, e.g., medical image processing [292] and computer graphics [293]. Critically, the baseline method cannot handle cases where a whole entry (e.g., all height, length, and width annotations) is missing, as the problem is underdetermined. Since the baseline method cannot refer to other frames' annotations by utilizing spatiotemporal constraints, the baseline was only run for single frame based estimation.

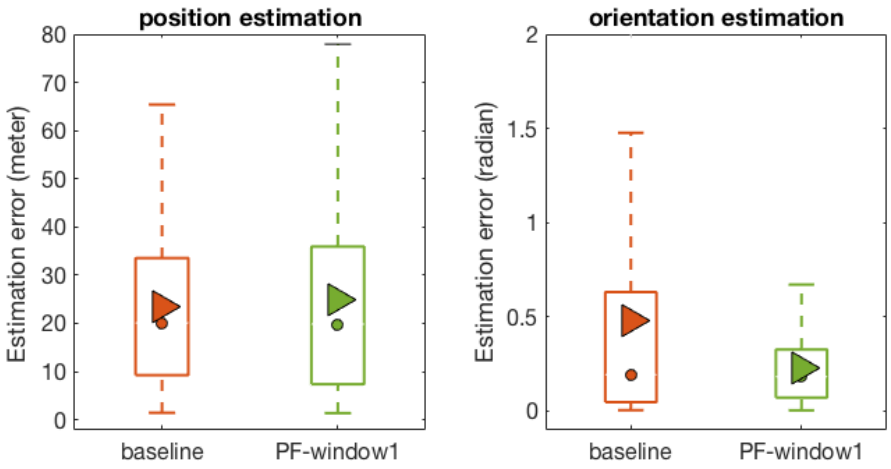


Figure 5.10: State estimation error of baseline vs. particle filtering without inter-frame referencing

Popup vs. Baseline We begin by comparing performance of Popup with our baseline using a three-frame reference window. Figure 5.9 shows the position and orientation error compared to the baseline. The average position error was reduced by 33% ($p < .001$), while orientation error was reduced by an average of 54%, but with low confidence, as the results were only approaching statistical significance ($p = .105$). This result shows that the proposed aggregation and estimation strategy for crowdsourcing image annotations is better than the baseline when annotations are noisy.

Effect of Temporal Aggregation Having shown that our particle filter based aggregation function is better than the baseline, we now seek to show that temporal aggregation is the cause of this performance increase. We do this by comparing the particle filter with a window size of one to the baseline, as we see in Figure 5.10.

In terms of position estimation, the baseline and the proposed particle filtering method perform similarly (no significant difference was observed, $p = 0.89$, Wilcoxon Rank-Sum). In terms of orientation estimation, we observed a 53% lower mean for our proposed particle filtering method compared to the baseline. However, while the effect size was medium-large ($d = 0.65$, Wilcoxon Rank-Sum), the results were only approaching statistical significance ($p = 0.11$, Wilcoxon Rank-Sum). In other words, temporal aggregation is an important component of this method.

Effect of Window Size Our comparison to the baseline was performed with window size three—that is, we use the adjacent frames to calculate the current frames. However, window size can range from one to ten. As shown in Figure 5.11, we evaluated four different window sizes—one, three, five, and ten. The window size of three frames had the lowest average state estimate error. Referencing three frames results in a 37% improvement in accuracy compared to not referencing neighboring

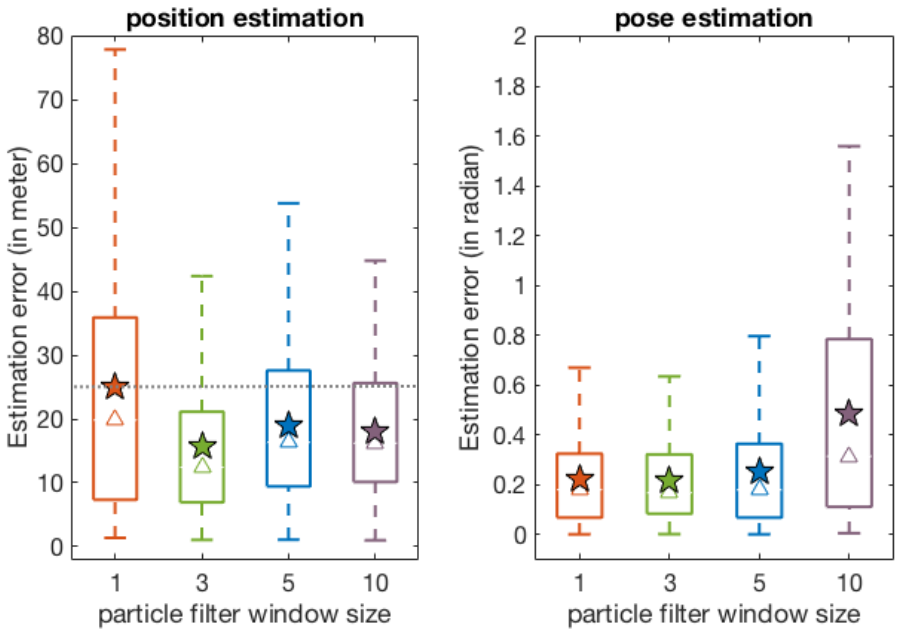


Figure 5.11: State estimation error of particle filtering without vs. with inter-frame referencing

frames in terms of position estimation ($p < .001$). However, orientation estimation accuracy did not improve by referring to neighboring frames. We speculate that the reason we did not observe progressive improvement in accuracy with increased window size is because of propagation of bad annotations. If one frame is poorly annotated, it will affect all other frames within the window. It follows that a larger window size allows local errors to affect more frames, which results in a larger aggregated overall error. For example, a critical error in frame k will affect only frame $k-1$ and $k+1$ in a three-frame window, but will affect all 10 frames in a 10-frame window.

5.3 Conclusion

Previous chapters showed that we must consider both when to defer inference—deferring our response until more data can be collected—and how to compensate for noisy deferral responses. The aggregation function of smart replacement compensates for many noisy responses by choosing the better of the two human-provided inputs, but ignores the important fact that different human inputs may contain complementary pieces of information.

For this reason we introduced Popup, which probabilistically combines hypotheses across time and annotator using a particle filter. We demonstrated Popup on the application of crowdsourced 3D scene reconstruction from monocular video, where the ability to handle complementary information is particularly important due to the fact that not only are annotators noisy and task models sensitive, but a single frame may truly have insufficient semantic information to solve the problem. Through a

variety of experiments, we show that this ability to aggregate information across multiple frames has consequences with respect to annotation time—allowing the human to ignore difficult annotations speeds up annotation—semantic accuracy—this increased flexibility allows us to more aggressively filter low-quality aggregations—and, of course, output accuracy.

This chapter clearly demonstrated the benefit of aggregating information in a probabilistic manner. However, the amount of human effort—five annotations per frame, ten frames per video—will be intractable if a single user is required to provide all annotations. For this reason, we re-enter the domain of deep learning, and seek to implement probabilistic aggregation in the output space in tandem with a deferral function.

CHAPTER 6

Comprehensive Evaluation and General Method

Thus far, we have demonstrated the importance of using the correct goal when choosing which inferences to defer and the importance of appropriately handling the deferral response. However, these findings are incomplete: our evaluations have focused on individual questions instead of an overall evaluation of deferred inference, and our deferral and aggregation functions have not been sufficiently generalizable across applications.

Motivated by this, we first describe an evaluation that comprehensively considers the tradeoff between error and human effort, where human effort is described in terms of the deferral rate and the deferral depth constraint. This evaluation includes not only a summary statistic that can be used to compare methods, but evaluates the effect of increasing human effort along multiple axes. We then introduce an aggregation function that generalizes the particle-filter based method of the previous chapter, which allows us to use the same fundamental deferral framework on the disparate tasks of referring expression comprehension and single-target video object tracking. This approach outperforms a number of baselines, including the smart replacement approach of Chapter 4.

6.1 A Comprehensive Evaluation

Deferred Error Volume We evaluate the performance of a deferred inference method with respect to three different factors: i) the error, which is a property of the application; ii) the Deferral Rate (DR), which is the expected number of deferrals that will occur for each task; and, iii) the Deferral Depth Constraint (DDC), which is the maximum number of times that a task can be deferred. Since evaluating at only a single DR-DDC pair—as is standard in previous work—does not provide an adequate analysis of a deferral method, we introduce the Deferred Error Volume (DEV), which calculates the volume under the surface produced by plotting the error at every DR/DDC pair.

While both DR and DDC are theoretically unconstrained, calculating the volume under a surface requires bounds to be placed. To produce these bounds, we make the least restrictive assumptions possible: the deferred inference method is capable of deferring every task at least once and the deferral function consists of a deferral score followed by a threshold. The former places an upper

Algorithm 2 Calculating DEV

```
DEV ← 0
DDC ← 1
while DDC ≤ 10 do
    tasks ← draw_tasks()
    DEV ← DEV + calc_error(tasks) / 10(len(tasks)+1)
    N ← 0
    while N < len(tasks) do
        cur_task ← find_task_to_defer(tasks, DDC)
        response ← get_new_input(cur_task)
        updated_task ← aggregation_fn(cur_task, response)
        update_tasks(tasks, updated_task)
        DEV ← DEV + calc_error(tasks) / 10(len(tasks)+1)
        N ← N + 1
    end while
    DDC ← DDC + 1
end while
```

bound on DR at one and a lower bound on DDC at one, and the latter allows a thorough evaluation of the relationship between DR and error. We set an upper bound of ten on the DDC, which captures all practical deferral depths, and divide by ten to scale the width of this dimension to one. We discuss the implications of this upper bound in our results.

Since evaluation will be performed on finite datasets, the volume under the curve when using rectangular integration is the mean of error under all constraint sets:

$$\text{DEV} = \frac{1}{10(N+1)} \sum_{\text{DDC}=1}^{10} \sum_{n=0}^{N} \ell(DR = \frac{n}{N}, DDC) \ , \tag{6.1}$$

where $\ell(DR, DDC)$ is the error at a specific DR and DDC, and N is the number of tasks in the dataset. We show the calculation of DEV in Algorithm 2: after an initial error calculation with one randomly drawn human input for every task (draw_tasks), find_task_to_defer finds the highest deferral score where the DDC constraint is not exceeded, draws another human input from the dataset (get_new_input), uses the aggregation function to update the prediction, updates the DEV, and repeats the process.

Such a thorough dataset-based evaluation has only one major requirement: *there must be a method by which deferral responses can be provided*. This requirement can be satisfied in a number of ways: the deferral response may be of the same form as the initial piece of human information with a dataset containing multiple pieces of human input per task (the approach of this work), the initial query and deferral response may be from a set of pre-defined attributes [97], or an external

agent capable of answering additional queries—potentially with access to oracle knowledge—may be developed [62].

Marginals To illustrate the effect of individual constraints on a method's performance, we marginalize out the DR and DDC and plot the result, referring to the measurement as *mean error*. Notably, calculating these marginals requires no inferences beyond those already used to calculate the DEV.

Error To provide an intuitive measure of the performance improvement, we report error at two specific locations: deferral rate of zero (deferral-free inference, or err @ 0), which is the base error of the task model, and deferral rate of one (err @ 1), which corresponds to the mean error across DDCs when the number of deferrals is equal to the number of tasks. We note that this does not mean every task will have one deferral, as some tasks will be performed with high confidence without deferring, while others may receive multiple ambiguous inputs. Since these errors correspond to the first and last points on the relevant marginal plot, no additional calculation is required to obtain these values.

6.2 Proposed Method

Formulation Here, we generalize the probabilistic particle-filter based method developed in the previous chapter to make it applicable to common problems and state-of-the-art task models. Underlying our method are two assumptions: first, we assume the task model output is a distribution. This is trivial for a softmax output, but may require additional consideration for other output formats [294]. Second, we assume the deferral function is based on this distribution. If we treat human inputs $h_1, ..., h_n$ as independent and represent the non-deferrable portion of the input (*e.g.,* an image) as x, the probability of a specific output, y, is:

$$p(y|x, h_1, ..., h_n) \propto \prod_{n=1}^{N} p(y|x, h_n) \ , \tag{6.2}$$

where $p(y|x, h_n)$ is the distribution produced by the task model, $f(x, h_n)$.

Throughout this work, we formulate the deferral as a request for the human operator to rephrase the initial input. This has two major benefits for general implementation: first, it allows us to use unmodified models and weights to perform inference. Second, it does not require novel datasets or the design of generative architectures.

Implementation of this formulation can be illustrated intuitively for a classification task: an initial x and h_1 are given to the task model, resulting in a softmax output. If the deferral function decides that inference should be deferred based on this output, h_2 is solicited and passed through

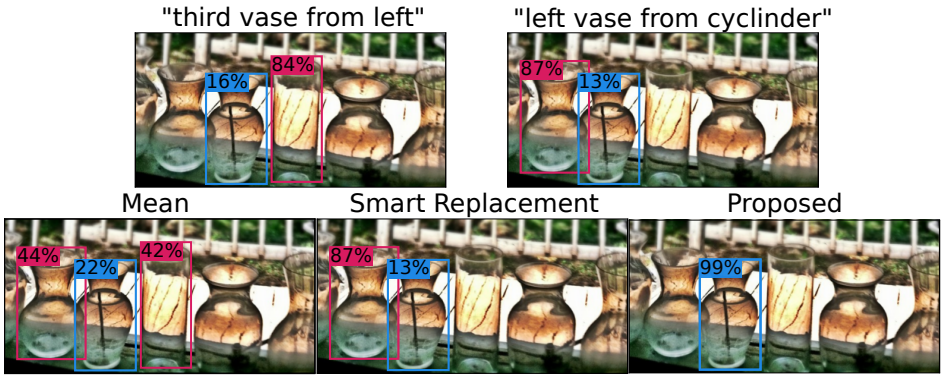


Figure 6.1: Our proposed aggregation function quickly combines complementary information to achieve higher certainty and accuracy than previous methods such as taking the mean of two outputs [68] or selecting the better output (Smart Replacement [295]). Target object boxed in blue. Original image cropped vertically for space. View in color.

the task model alongside x, resulting in a second softmax output. The two output vectors are then multiplied elementwise, and the resulting vector is normalized. The deferral function is then executed on this normalized vector to determine if another deferral is necessary. In addition to being immediately compatible with supervised training and much simpler than training text generation models [61], [62], we see an example of why this method works well in Figure 6.1: it quickly identifies the target object with high certainty, while aggregation functions such as using the better of the two outputs [295] or taking the mean of the two outputs [68] would perform additional deferrals or return an incorrect answer depending on the given constraints.

Convergence Although the ability to aggregate complementary pieces of human-provided information is likely sufficient motivation for our method, combining multiple inferences has an additional, critical, benefit: it reduces the importance of a manually imposed DDC by converging rapidly under simple assumptions. We show this here theoretically, first by defining the set of received human inputs as $\mathcal{H} = \{h_1, h_2, ...h_n\}$ and modifying equation 6.2:

$$p(y|x, \mathcal{H}) \propto \prod_{h \in \mathcal{H}} p(y|x, h) \quad . \tag{6.3}$$

We then extend $\mathcal{H}$ to include all possible human inputs, where each human input has some probability, $p(h|x, y)$, of occurring over n human inputs:

$$p(y|x, \mathcal{H}) \propto \prod_{h \in \mathcal{H}} p(y|x, h)^{np(h|x,y)} \quad . \tag{6.4}$$

which trivially reduces to:

$$p(y|x, \mathcal{H}) \propto \left(\prod_{h \in \mathcal{H}} p(y|x, h)^{p(h|x,y)} \right)^n \quad . \tag{6.5}$$

If we define the set of potential outputs as $\mathcal{Y}$, we can calculate the normalized probability:

$$p(y|x, \mathcal{H}) = \frac{p(y|x, \mathcal{H})}{\sum_{y_o \in \mathcal{Y}} p(y_o|x, \mathcal{H})} \quad . \tag{6.6}$$

We can see that as $n \to \infty$, the normalized value will converge to an impulse—a one-hot for a classification task—at the output with the largest product of weighted probabilities.

6.3 Exemplar Applications

6.3.1 Single-Target VOT

Goal In single-target Video Object Tracking (VOT), a human defines the task by drawing a bounding box around an object in the first frame of a video. Using only previous frames (the online tracking setting) the model then propagates this bounding box through the video. Despite the relatively low dimension of the input space, deferred inference is properly motivated for this application: since tracking algorithms are often sensitive to small differences in initialization, perturbation tests are a standard component of evaluation [54], [146].

Model and Dataset We perform our analysis using the crowdsourced data from Chapter 4, which consists of nine first-frame annotations for every video in the OTB-100 dataset [146]. Unlike Chapter 4, we assume that all initializations indicate the correct object and remove instances in this dataset where the initialization has an IoU of less than 0.5 with the gold-standard. As our task model, we use the ToMP tracker [52] with a ResNet-50 backbone [276] and weights provided by the original authors.

Base Error Metric We measure performance using the mean intersection-over-union (IoU), which is commonly used in evaluation of the VOT task [54], [146]. To maintain the notion of error, we subtract the IoU from its maximum value of one. Unlike previous evaluations, we allow the model's inference to continue when the object track is lost: resetting the tracker requires a ground-truth box on every frame, which is intractable both in a real-world application and in the context of our implementation of deferred inference.

Deferral Implementation The output of ToMP is a single bounding box at every frame. To convert this to a distribution for our deferral and aggregation functions, we first produce stochastic bounding boxes by performing each inference 100 times with Monte Carlo dropout enabled [186] similar to previous work on object detection [294]. Every bounding box is represented as a tuple of (TLX, TLY, width, height) and expectation maximization [296] is performed on every frame to transform these representations into a Mixture of Gaussians. To determine the number of Gaussians for expectation maximization, we use DBSCAN [297], which relies on two hyperparameters: ϵ and *minimum samples*. We performed a gridsearch across these two parameters—shown in Table 6.1—to determine that the best parameter set for our experiments was epsilon 10 and minimum samples 20.

	Minimum Samples				
	3	5	10	15	20
ε = 1	0.3314	0.3304	0.3344	0.3371	0.3391
ε = 3	0.3382	0.3269	0.3240	0.3249	0.3260
ε = 5	0.3375	0.3268	0.4779	0.3234	0.3229
ε = 10	0.3377	0.3278	0.3243	0.3235	**0.3228**
ε = 15	0.3369	0.3280	0.3250	0.3240	0.3231
ε = 20	0.3364	0.3284	0.3255	0.3246	0.3237
ε = 50	0.3350	0.3295	0.3283	0.3284	0.3277

Table 6.1: The effect of DBSCAN parameters on the deferral-free mean error (1-IoU) of our method. Lower is better.

Using these distributions, we produced the deferral score by randomly sampling 500 bounding box pairs and measuring the mean IoU between them. For both our method and baselines, we created an output bounding box by taking 10,000 samples from the mixture, and using the one with the highest likelihood. For our method, these samples were scattered by adding a normally-distributed random value with standard deviation 7 to every dimension, which allowed us to combine distributions that were close in Euclidean space but several standard deviations apart.

6.3.2 Referring Expression Comprehension

Goal In referring expression comprehension, a task is defined by an image and text query, and the task model draws a bounding box around the object described by the text. The high dimensionality of the input space means that there is much room for both semantic ambiguity and gaps in the task model's knowledge that can be corrected or clarified after a deferral.

Model and Dataset For the task model we used the UNITER architecture [1], which formulates referring expression comprehension as classification over a set of externally-provided bounding boxes. We provided ground-truth detections for these bounding boxes to minimize the influence of an external object detector. We trained and evaluated on the RefCOCO [2] dataset because it contains multiple references to all but one target object, which is substantially better than both the RefCOCO+ [2], and RefCOCOg [8] datasets (Table 6.2). Our model was trained on a single GeForce GTX Titan XP GPU using the training settings given by the original authors with a few small modifications: we used full precision floating point operations, adjusted the batch size from 128 to 64, and accumulated gradients over two steps.

	RefCOCO			RefCOCO+			RefCOCOg
Expression Count	Val	TestA	TestB	Val	TestA	TestB	Val
1	1	0	0	110	29	90	257
2	605	288	354	482	162	336	2309
3	3197	1667	1498	3169	1763	1361	7
4	8	20	17	43	21	11	0
5	0	0	1	1	0	0	0
Total Objects	3811	1975	1870	3805	1975	1798	2753
Total Expressions	10834	5657	5275	10758	5726	4889	4896

Table 6.2: Number of expressions for every target object in the three major referring expression comprehension datasets.

Base Error Metric Performance on this application was measured through the standard method of the inference being classified as correct (error = 0) if the predicted bounding box has an IoU of greater than 0.5 with the ground-truth bounding box, and incorrect (error = 100) otherwise [8]. Aggregate error can then be interpreted as the percent of tasks that are completed incorrectly. We maintain the val, testA, and testB splits from previous works [165], but note our evaluation measures per-task performance instead of per-phrase performance, making it incorrect to directly compare our results to other evaluations.

Deferral Implementation Because the UNITER referring expression comprehension model outputs a softmax distribution, we used entropy as our deferral score and implement our proposed aggregation function by multiplying and normalizing the outputs. We improved the model's ability to detect ambiguity by performing Monte Carlo dropout with 100 passes, matching the number of passes in the original work [186].

6.4 Experiments

Baselines We compare our proposed method to four aggregation functions adapted from previous work:

- Naive Replacement: If a deferral is performed, the second input is always selected over the initial input. If no DDC is specified this is analogous to a selective prediction approach [42], [58], where the user must restart the task if the inference is declined.
- Mean: If the inference is deferred, DDC new inputs are taken, and the mean of all responses is used. This is equivalent to the aggregation function of Hatori *et al.* [68], who implicitly defined a DDC of one.
- Consensus: If a deferral is performed, DDC new inputs are taken and the consensus of all outputs is returned as the answer. If there is no consensus, an answer is chosen randomly from the potential outputs with equal occurrences. This is a basic approach often used in crowdsourcing [69], [246]. We did not implement this baseline on the VOT application due to the high number of potential outputs.

	VOT		RefExp (Val)		RefExp (TestA)		RefExp (TestB)	
Method	DEV	Err @ 1	DEV	Err @ 1	DEV	Err @ 1	DEV	Err @ 1
Naive Replacement	0.3279 ± 1.6e^{-4}	0.3269 ± 5.7e^{-4}	6.92 ± 7.9e^{-3}	6.56 ± 7.6e^{-3}	5.92 ± 8.2e^{-3}	5.65 ± 1.1e^{-2}	7.60 ± 9.1e^{-3}	6.90 ± 9.3e^{-3}
Mean	0.3286 ± 1.4e^{-4}	0.3277 ± 4.4e^{-4}	7.26 ± 8.2e^{-3}	6.46 ± 7.5e^{-3}	6.48 ± 1.1e^{-2}	5.68 ± 1.0e^{-2}	8.09 ± 1.1e^{-2}	7.20 ± 1.1e^{-2}
Consensus			7.94 ± 7.9e^{-3}	7.47 ± 8.1e^{-3}	7.33 ± 1.2e^{-2}	6.92 ± 1.2e^{-2}	8.80 ± 1.2e^{-2}	8.32 ± 1.1e^{-2}
Smart Replacement	0.3280 ± 1.6e^{-4}	0.3264 ± 5.0e^{-4}	6.51 ± 5.6e^{-3}	5.68 ± 5.7e^{-3}	5.41 ± 8.1e^{-3}	4.55 ± 7.5e^{-3}	7.29 ± 1.0e^{-2}	6.17 ± 9.3e^{-3}
Ours	**0.3263 ± 1.3e^{-4}**	**0.3245 ± 4.4e^{-4}**	**6.13 ± 6.4e^{-3}**	**5.23 ± 5.9e^{-3}**	**5.16 ± 8.6e^{-3}**	**4.24 ± 7.8e^{-3}**	**6.92 ± 8.6e^{-3}**	**5.74 ± 8.7e^{-3}**

Table 6.3: The DEV and Err @ 1 metrics for baselines and our method (Err @ 0 shown in Table 6.4). Our method performs best across all applications and splits by a significant margin.

- Smart Replacement: If inference is deferred, the deferral score between all responses is compared, and the output corresponding to the best deferral score is used. As with the Mean baseline, we extend the approach in Chapter 4 by allowing the DDC to be greater than one.

Results We see in Table 6.3 that deferred inference improves performance over the deferral-free condition under both the DEV, since the DEV of the deferral-free condition is simply the error, and err @ 1 metrics for all methods and problem settings. In other words, any aggregation function is better than the deferral free condition for the evaluated deferral functions. Further, as shown in Table 6.4, our proposed aggregation function outperforms all baselines in all settings on the evaluated metrics, and provides a large reduction in error between the deferral-free condition and deferral rate 1: error decreases by 1.37% for VOT, 40.7% for RefExp-Val, 48.7% for RefExp-TestA, and 40.6% for RefExp-TestB. In other words, *our method is effective on two very different applications, and can reduce error by over 48% (from 8.27% to 4.24%) without any change to the model.*

	VOT	RefExp (Val)	RefExp (TestA)	RefExp (TestB)
Err @ 0	0.329 ± 6.6e^{-4}	8.82 ± 0.03	8.27 ± 0.04	9.67 ± 0.04
Err @ 1 (Ours)	0.325 ± 4.4e^{-4}	5.23 ± 5.9e^{-3}	4.24 ± 7.8e^{-3}	5.74 ± 8.7e^{-3}
Improvement	1.22%	40.70%	48.73%	40.64%

Table 6.4: Err @ 0 (No deferral) and Err @ 1 using our method. We can reduce error by up to 48.73%.

Marginals We now consider the effect of individual constraints by marginalizing out the DR (Figure 6.2-Left) and DDC (Figure 6.2-Right). By examining the former, we aim to answer two specific questions: what is the effect of our DDC range on the ordinal results of the DEV metric, and what is the effect of the DDC on performance? For the former question, we see that our method was unambiguously better—that is, best or within one standard error of best at all DDCs—on both applications. However, the improved performance of our method on the VOT task is primarily due to its ability to effectively handle greater DDCs: if our evaluation were limited to DDC ≤ 2, the DEV would be within one standard error of the Smart Replacement and Mean baselines.

Further consideration of the interaction between DDC and mean error provides meaningful insight into both our method and the findings of previous work. First, while other methods began

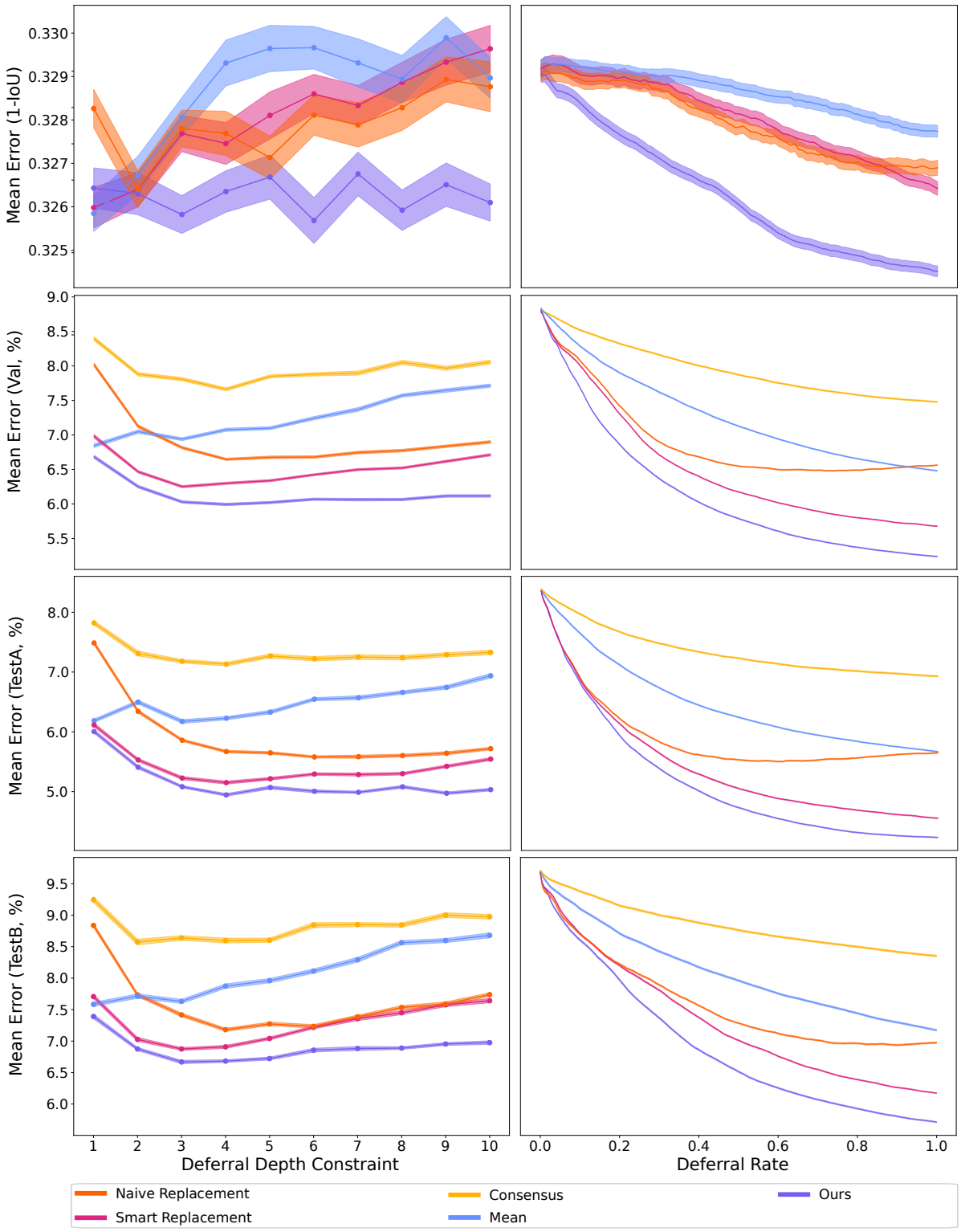


Figure 6.2: Marginal plots showing the effect of the DDC (left) and DR (right) on the VOT (top) and Referring Expression Comprehension (bottom three) applications. Shaded area represents one standard error across 100 trials. View in color.

Naive Replacement

Distribution	Scoring Fn	Val	TestA	TestB
Softmax	SR	7.22 ± 6.7e^{-3}	6.22 ± 9.3e^{-3}	7.95 ± 9.9e^{-3}
Softmax	Entropy	7.21 ± 6.5e^{-3}	6.16 ± 8.5e^{-3}	7.99 ± 9.7e^{-3}
Dropout	SR	**6.93 ± 6.8e^{-3}**	**5.94 ± 7.5e^{-3}**	**7.61 ± 1.0e^{-2}**
Dropout	Entropy	**6.92 ± 7.9e^{-3}**	**5.92 ± 8.2e^{-3}**	**7.60 ± 9.1e^{-3}**

Mean

Distribution	Scoring Fn	Val	TestA	TestB
Softmax	SR	7.91 ± 8.2e^{-3}	7.09 ± 1.3e^{-2}	8.74 ± 1.2e^{-2}
Softmax	Entropy	7.96 ± 9.1e^{-3}	7.18 ± 1.1e^{-2}	8.87 ± 1.3e^{-2}
Dropout	SR	**7.19 ± 8.4e^{-3}**	**6.42 ± 1.1e^{-2}**	**8.05 ± 1.1e^{-2}**
Dropout	Entropy	7.26 ± 8.2e^{-3}	6.48 ± 1.1e^{-2}	8.09 ± 1.1e^{-2}

Consensus

Distribution	Scoring Fn	Val	TestA	TestB
Softmax	SR	8.29 ± 9.6e^{-3}	7.52 ± 1.3e^{-2}	9.12 ± 1.5e^{-2}
Softmax	Entropy	8.29 ± 8.8e^{-3}	7.54 ± 1.3e^{-2}	9.19 ± 1.4e^{-2}
Dropout	SR	**7.95 ± 8.8e^{-3}**	**7.31 ± 1.2e^{-2}**	**8.74 ± 1.3e^{-2}**
Dropout	Entropy	**7.94 ± 7.9e^{-3}**	**7.33 ± 1.2e^{-2}**	8.80 ± 1.2e^{-2}

Smart Replacement

Distribution	Scoring Fn	Val	TestA	TestB
Softmax	SR	6.94 ± 6.3e^{-3}	5.76 ± 9.1e^{-3}	7.63 ± 6.3e^{-3}
Softmax	Entropy	6.89 ± 5.8e^{-3}	5.72 ± 9.1e^{-3}	7.65 ± 8.5e^{-3}
Dropout	SR	**6.51 ± 5.8e^{-3}**	**5.37 ± 8.0e^{-3}**	**7.31 ± 1.1e^{-2}**
Dropout	Entropy	**6.51 ± 5.6e^{-3}**	5.41 ± 8.1e^{-3}	**7.29 ± 1.0e^{-2}**

Ours

Distribution	Scoring Fn	Val	TestA	TestB
Softmax	SR	6.93 ± 7.6e^{-3}	6.31 ± 1.1e^{-2}	7.70 ± 1.0e^{-2}
Softmax	Entropy	6.84 ± 8.2e^{-3}	6.10 ± 9.7e^{-3}	7.63 ± 1.1e^{-2}
Dropout	SR	**6.12 ± 6.8e^{-3}**	**5.16 ± 8.0e^{-3}**	**6.90 ± 1.0e^{-2}**
Dropout	Entropy	**6.13 ± 6.4e^{-3}**	**5.16 ± 8.6e^{-3}**	**6.92 ± 8.6e^{-3}**

Table 6.5: The DEV for all potential deferral functions on the referring expression comprehension application.

to meaningfully degrade as DDC increased, our method did not show such severe trends. Next, the DDC of one used in previous work [62], [295] has meaningful shortcomings: all aggregation functions, with the exception of Mean, were improved by increasing the DDC beyond one on the referring expression comprehension task and, on the video object tracking task, the finding of Chapter 4 that Smart Replacement is significantly better than Naive Replacement was only supported at the DDC of one used in their evaluation.

When the DDC is marginalized, our method was best or within one standard error of best at all DRs. Broadly speaking, behavior of this marginal is as expected: error decreased as DR increased for all aggregation functions with the exception of naive replacement, which increased at higher DRs. As noted in Chapter 4, this is due to the tendency to defer correct inferences at higher DRs and replace them with potentially low-quality human inputs.

Alternate Scoring Functions For simplicity, we focused on one particular method for calculating deferral score for the referring expression comprehension task: entropy with Monte Carlo dropout [186]. For the Referring Expression Comprehension application, we can compare performance when MC dropout is and isn't enabled, as well as when softmax response or entropy is used. We show this in Table 6.5. We note three things: first, it is still true that all conditions improve over deferral-free inference. Second, conditions with MC dropout enabled performed better, and last, the performance when softmax response and entropy were used as the deferral function was roughly equivalent. Interestingly, smart replacement often matches or outperforms our proposed method when dropout isn't used, likely because the quality of distributions is more important when belief compounds over sequential human inputs.

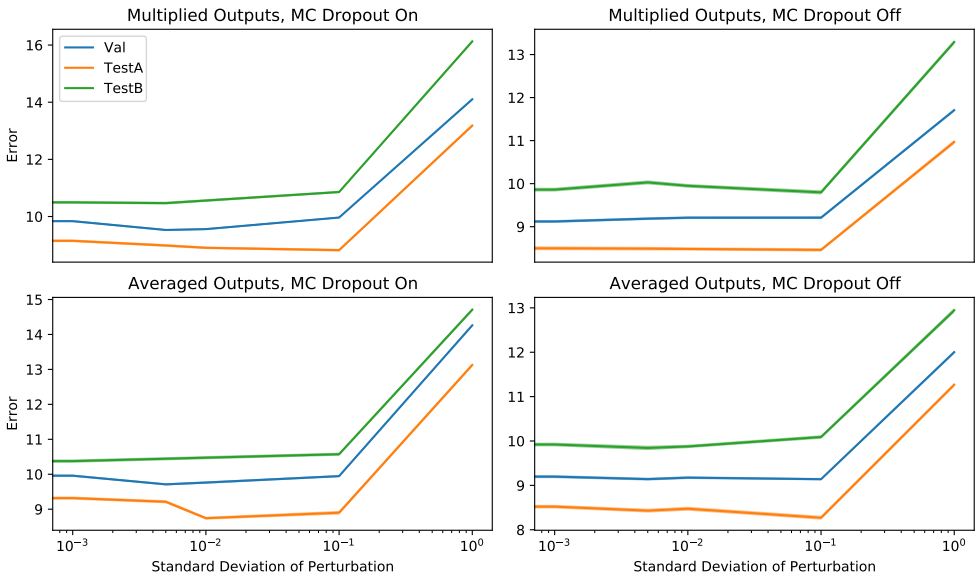


Figure 6.3: Effect of perturbing text embeddings on error.

Importance of Deferral Response To show that the semantic changes enabled by deferral occupy meaningfully different locations in the embedding space, we added random Gaussian noise of various standard deviations to the text embeddings for the referring expression comprehension application. For every task we drew 5,000 samples with the same referring expression and combined the output distributions either as the average (averaged outputs) or the product (multiplied outputs). The error under all four conditions is shown in Figure 6.3. Although there is some reduction in error due to these perturbations, it is much less than we showed using deferred inference, meaning the additional user burden of deferral is justified.

Isolating Uncertainty in Text Embeddings Chapter 3 demonstrated that significant performance benefits could be provided by only deferring when the human input is the cause of error. In an attempt to extend this finding to the problem of referring expression comprehension, we additionally calculated the performance of our aggregation function when dropout was applied to only the text embedding layers. As we see in Table 6.6, dropout on only the text embedding layers does perform better than not performing dropout at all, but is still significantly outperformed by using dropout across all layers.

Distribution	Val	TestA	TestB
Softmax	6.93 ± 7.6e^{-3}	6.31 ± 1.1e^{-2}	7.70 ± 1.0e^{-2}
Text Dropout	6.59 ± 6.5e^{-3}	5.75 ± 8.6e^{-3}	7.28 ± 9.8e^{-3}
Dropout	**6.13 ± 6.4e^{-3}**	**5.16 ± 8.6e^{-3}**	**6.92 ± 8.6e^{-3}**

Table 6.6: DEV when only the text embedding is dropped out (text dropout) no dropout (softmax) and full dropout. Deferral score is entropy, aggregation function is belief update.

6.5 Conclusion

Evaluating performance with respect to hazy oracles requires us to simultaneously consider the error, the deferral rate, and deferral depth constraint. In this chapter, we introduced the first evaluation that considers all three of these factors and showed the importance of each of them. Notably, we demonstrated the importance of the previously ignored deferral depth constraint: the conclusions of Chapter 4 on the application of single-target video object tracking were only confirmed for DDC=1.

Using this evaluation, we demonstrated a mathematically simple, yet effective and generalizable method built around updating beliefs across multiple human inputs. This method significantly outperformed baselines under the DEV metric, on the marginals, and using the error @ 1 metric. The last metric has meaningful intuitive impact: by allowing a deferral rate of one, we can reduce error by up to 48.73% on the referring expression comprehension application.

While the error reduction is meaningful and it is important to provide aggregate comparison of deferral methods, such aggregate analyses have a few important limitations. Because the datasets are crowdsourced, our deferral and aggregation functions made several implicit assumptions, such as assuming humans are identical and the deferral response is independent of previous queries. There is additionally the challenge of how to set the threshold on the deferral function such that it targets specific errors or deferral rates. We examine both of these problems through a user study in the next chapter.

CHAPTER 7

Setting Human-Centered Deferral Criteria

In deferred inference, the ultimate goal is to optimize a tradeoff between error and user effort. The first part of this, demonstrated in previous chapters, is the ability to optimally make this tradeoff as measured by aggregate analysis on crowdsourced datasets. The second part of this is setting deferral criteria that targets a specific point on this tradeoff, which cannot be done with such aggregate analysis for two reasons: first, crowdsourced data does not accurately represent the behavior of human-AI team and second, aggregate analysis ignores the fact that the deferral function must make a binary decision—should I defer or not?—instead of simply using the deferral score to rank inputs.

We begin by discussing specific ways in which crowdsourced data is misaligned with the setting of a human-AI team. First, since crowdsourced data is collected as a series of microtasks, datasets that contain multiple human inputs for the same target do not consider how the human input changes after deferral. We will prove later in this chapter that this assumption is incorrect. Second, data collected from the crowd—particularly high-dimensional linguistic data—is typically validated by formulating the data collection as a human-human interaction. For example, the data collection process for RefCOCO [2] (used in our previous experiments) is as follows:

> *Player 1:* is shown an image with an object outlined in red and provided with a text box in which to write a referring expression. *Player 2:* is shown the same image and the referring expression written by Player 1 and must click on the location of the described object (note, Player 2 does not see the object segmentation). If Player 2 clicks on the correct object, then both players receive game points and the Player 1 and Player 2 roles swap for the next image. If player 2 does not click on the correct object, then no points are received and the players remain in their current roles.

where the authors "...posted the game online for anyone on the web to play and encouraged participation through social media and the survey section of reddit. [...] also posted the game on Mechanical Turk."

Figure 7.1: Two notable shortcomings caused by the collection procedure of the RefCOCO [2] dataset are (a) the ability to specify clicks instead of objects and (b) the common practice of backchannel communcation. The target bounding box is shown in pink.

Since the data collection uses the proxy goal of telling a human how to localize a click instead of describing a target object to an AI, the distribution of utterances will be meaningfully different in several ways. First, people just talk differently to computers, typically using fewer words (but more messages), shorter words, and more profanity [252]. Next, although the stated goal of ReferItGame is to generate unambiguous referring expressions for segmented objects, success in the game (and therefore payment for crowd workers) requires neither unambiguous information nor a description of the target object. For the former, we see an example of in Figure 7.2 where, although *woman* and *her* are semantically ambiguous, the data was included in the dataset. For the latter, we see in Figure 7.1-(a) that annotators were able to specify parts of the object instead of the target object, as would be done in the motivating application. Last, humans have the desire to communicate with their game partners, leading to backchannel communications (Figure 7.1-(b)).

Annotator 1: woman
Annotator 2: her

Figure 7.2: Under many data collection methods [2], annotators are rewarded for guessing, leading to ambiguous phrases.

In this chapter, we address these shortcomings by performing a study with 25 participants interacting with a deep learning model to perform a language-based image cropping task. Using the findings of this study, along with newly developed theory on interaction, we consider two objectives for setting deferral criteria: the first seeks the deferral criteria that gets as close as possible to the desired deferral rate, while the other seeks the deferral criteria that produces the highest deferral rate without exceeding the upper-bound deferral rate. For both of these objectives, we compared deferral criteria set based on a brief interaction with the target user to deferral criteria set using interactions from multiple users, and found that using data from only the target user performs as well or better than using large datasets, despite having two orders of magnitude fewer examples for calibration.

7.1 Theory on Thresholds

When we set deferral criteria, we seek to balance error and user burden. Previous work [61], [68] has downplayed that tradeoff by making the assumption that it is sufficient to set deferral criteria based on characteristics of the task model. In this section, we show that it is impossible to set a deferral criteria that targets an error or Deferral Rate (DR) without explicitly considering the user for whom that criteria is being set. Throughout this work, we set the Deferral Depth Constraint (DDC) to one across all evaluations.

Expected Deferral Rate We begin by showing how to calculate the expected DR, $\mathbb{E}(\text{DR}|t, u)$. A deferral occurs if we have a user, u, that user produces a score, s_1, and that score is greater than the threshold, t. This gives us the formula:

$$\mathbb{E}(\text{DR}|t, u) = \int_{s_1} p(s_1 > t, s_1, u) ds_1. \tag{7.1}$$

If we expand by chain rule, assume the user is given ($p(u) = 1$) and represent $p(s_1 > t)$ with an indicator function, we get:

$$\mathbb{E}(\text{DR}|t, u) = \int_{s_1} \mathbb{1}(s_1 > t) p(s_1|u) ds_1. \tag{7.2}$$

This demonstrates clearly that while previous work often sets deferral criteria in a user-agnostic way [61], [68], we cannot target a deferral rate in a user-agnostic manner if $p(s_1)$ is dependent on the user. This motivates the research question: *do deferral scores differ meaningfully between users?*

Probability of Error To find the probability of error $p(e|t, u)$, we evaluate separately the contribution to error when a deferral does and doesn't occur. When no deferral occurs, we are looking for the condition where the user, u, produces a score, s_1, that is less than or equal to the threshold, t, and there is an error, e. Written mathematically:

$$p(e|t, u, s_1 \leq t) = \int_{s_1} p(e, s_1, s_1 \leq t, u) ds_1. \tag{7.3}$$

The formulation is similar if deferral has occurred, with the addition of the deferral score after the

second human input:

$$p(e|t,u,s_1>t)=\int_{s_2}\int_{s_1}p(e,s_2,s_1,s_1>t,u)ds_1ds_2. \tag{7.4}$$

Since these two conditions are mutually exclusive (s_1 is never simultaneously greater than and less than t), we can simply sum these two components. If we invoke the same assumptions as in Equation 7.2, we get:

$$p(e|t,u)=\int_{s_2}\int_{s_1}(p(e|s_2,u)p(s_2|s_1,u)p(s_1|u)\mathbb{1}(s_1>t)+p(e|s_1,u)p(s_1|u)\mathbb{1}(s_1\leq t))ds_1ds_2. \tag{7.5}$$

As when targeting a deferral rate, it is critical to consider the relationship between the deferral score, s_1, and the user. Additionally, we note two other questions that should be evaluated: first, if the task model's responses to the first and second human inputs are identical, we can find $p(s_2|s_1,u)$ using only initial responses ($p(s_1|u)$), significantly reducing calibration time. In other words, we ask *how do users respond when an inference is deferred?* Second, although works in calibration [13], [176], [179], [298] show a relationship between probability of error and some deferral scores, such works have never considered the role of individual users. If the relationship between probability of error and deferral score is dependent on the individual, we must consider this when finding $p(e|s_1,u)$ and $p(e|s_2,u)$, instead of simply using large datasets. In other words, we ask *does knowing the user provide additional information about the mapping between probability of error and deferral score?*

Explicitly, this leads us to five research questions:

RQ1 *How is user satisfaction related to error and deferral rate?* The goal of this thesis is to provide not only an improved accuracy—as in previous chapters—but also an improved user satisfaction. We measure this here, as well as note that it is only necessary to pursue a specific error or deferral rate—which requires user-specific deferral criteria—if these factors have an effect on overall satisfaction.

RQ2 *What are the time dependencies of error, e, and deferral score, s?* The lack of a time variable in the above formulations implicitly assumes static distributions. However, previous work [299], [300], as well as common sense assert that the users require some time to develop their mental model. Thresholds should only be set after this mental model has converged.

RQ3 *Do deferral scores differ meaningfully between users?* By not providing user identities, dataset-focused work in deferred inference [48], [68] implicitly assumes that users are interchangeable, while works that evaluate via human experiments [61], [91] set deferral

criteria a-priori and do not consider qualities of the individual. If the deferral score is different between users, the deferral criteria will need to be calibrated for individuals.

RQ4 *How do users respond when an inference has been deferred?* Previous work using our chosen deferral formulation has either accepted deferral responses as is—not comparing qualities of the deferral response to the initial query—or broken time dependency through the use of crowdsourced datasets [48], [68]. If the deferral response is significantly different from the initial response, this dependency should be considered in future work. If not, dataset-like approaches could be used to set deferral criteria for higher deferral depth constraints without collecting many deferral responses for each user.

RQ5 *Does knowing the user provide additional information about the mapping between probability of error and deferral score?* Works in model calibration [13], [176], [179], [298] demonstrate a mapping between deferral scores and probability of error, but do not explore if such mapping is consistent across users. If this mapping is not user-dependent, we can construct both $p(e|s_1, u)$ and $p(e|s_2, u)$ prior to interaction with an individual based on large non-user-specific datasets. This would greatly reduce the amount of time required to set deferral criteria.

7.2 Experimental Setup

7.2.1 Motivating Application

We used referring expression comprehension [8] as our motivating application. We presented this application to our participants as a language-based image cropping task, which was chosen for two reasons: first, cropping is a commonly performed and easily explainable task, meaning little additional instruction was necessary. Second, unlike other embodiments of referring expression comprehension—such as pick-and-place [61]—cropping can be credibly applied to already existing datasets (*i.e.,* MSCOCO [15]) and therefore does not require additional model training or dataset procurement.

As our dataset, we used a subset of target objects from the RefCOCO dataset [2]. This subset was chosen to mitigate two issues observed in our initial tests: first, there were many cases where the target object was visually ambiguous due to a high degree of overlap with other objects in the image—for example, a person standing in front of another. Second, similar to findings in Chapter 3 and on the VQA application in other works [26], there were numerous instances where the model largely ignored the text. Since our focus is on the effect of human input given a clear intent, we selected a subset of RefCOCO that meets the following criteria:

- The object does not have an Intersection-over-Union (IoU) of greater than 0.5 with any other object in the image.
- Of the referring expressions in the RefCOCO dataset [2] for this object, greater than 32% but less than 68% result in a correct answer.

We additionally iterated through the remaining examples to manually remove images that do not clearly indicate a single object or may be offensive, resulting in a total of 1,107 potential crop targets across 842 images. During evaluation, crop targets were randomly picked and an individual participant never saw the same image more than once.

7.2.2 Procedure

Participants We conducted this experiment with 28 adults (older than 18). All participants were required to have normal or corrected-to-normal full-color vision and described themselves as proficient in English. Participants were solicited via local mailing lists and located in the United States at the time of the study. Participants were asked to use a computer with a mouse and keyboard, and were supervised virtually during the experiment. Three participants were identified as malicious or inattentive actors (error greater than three standard deviations above the mean) and their data was excluded from further analysis.

Of the remaining 25 participants, 12 identified as male, 12 identified as female, and 1 preferred not to state. Mean age was 25.2 ± 2.88, technical competence was reported as 5.76 ± 1.24 out of 7, and experience with conversational virtual assistants was reported as 4.16 ± 1.76 out of 7. Our study was approved by our institution's IRB, and participants consented to participate in the study before the study started. Participants were compensated $20 for their participation.

Instructions After agreeing to the consent form but prior to any interaction with the system, participants were given a set of instructions for the study. These instructions described the overall goal of image cropping, the interface they would use, the actions the system may take (deferral, correct answer, incorrect answer), and the surveys they would be given. Instructions did not contain any example phrases in order to avoid biasing the user.

Background Survey Participants were asked to provide demographic data (age and gender) as well as their perceived technical competence (1-7 agreement with *I consider myself to be technically adept*), experience with voice assistants in general (1-7 agreement with *I am experienced with voice assistants (Alexa, Siri, etc.)*), and experience with the commercially available voice assistants Amazon Alexa, Apple Siri, Google Assistant, Microsoft Cortana, and Samsung Bixby (*How often*

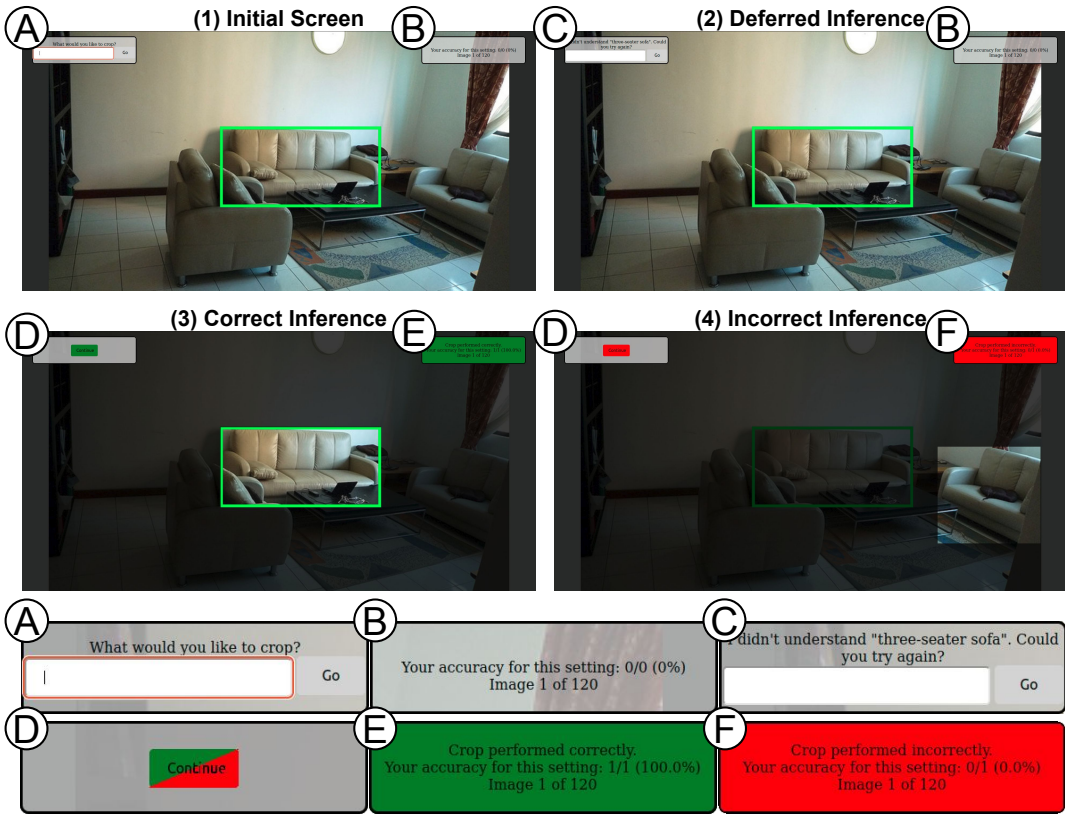


Figure 7.3: The four screens in our interface. The user begins in the *Initial Screen* and is tasked with cropping the object in the green box. After entering text on the initial screen, the AI may choose to defer or infer. If the AI chooses to defer, the user is asked to provide another input on the *Deferred Inference* screen. After inference, the user is presented with either the *Correct Inference* screen or the *Incorrect Inference* screen. In both cases, the removed region is darkened. Indicated regions shown below images. The color of region (D) depends on whether the inference was correct. Inputs for correct and incorrect inferences were *three-seater sofa* and *far right sofa*, which were provided by participants to identify the cropped objects.

do you use the following voice assistants: several times a day/several times a week/1-2 times a week/less).

Treatments Once participants completed the background survey, they were given four treatments corresponding to deferral rates of 0.0, 0.1, 0.2, and 0.3. The 0.0 deferral rate setting was given first to allow the user to gain familiarity with the system without the noise of random deferrals, while the other three deferral rate settings were presented in a randomized order. Prior to each setting, participants were informed of the beginning a new setting, but no information was provided about which variable was changed.

Participants then interacted with the cropping model via the interface shown in Figure 7.3. For every task—30 in total for each treatment—they were given a random, previously unseen image

with a green box drawn around the target object and the question "what would you like to crop?" (*Initial Screen*). After giving a referring expression corresponding to the boxed object, the model could defer or perform the inference. If the model chose to defer, participants were presented with the last entered phrase, a prompt stating *I didn't understand "[entered text]". Could you try again?* (*Deferred Inference*). If the model chose to perform the inference, the identified object was indicated by shading the removed region. If the crop was correct, the screen showed green (*Correct Inference*), while an incorrect crop showed red (*Incorrect Inference*).[1] The accuracy (number correct, number attempted, and those values expressed as a percent) for the current setting, as well as the overall progress (number of crops performed and total number of crops), were shown on the upper-right corner.

After each treatment, participants were asked to report their satisfaction by rating their agreement with the following statements on a 1-7 scale, where 1 is strongly disagree, and 7 is strongly agree:

- I was satisfied with the accuracy I was able to achieve.
- The computer asked me to repeat myself on too many pictures.

Technical Details We maintain the referring expression comprehension setting of the previous chapter: the UNITER architecture [1] with ground-truth detections and Monte Carlo Dropout [186] with 100 forward passes. We use a belief update—from the previous chapter—as our aggregation function.

We change our deferral function based on the setting. When interacting with our test participants, we seek to precisely target a deferral rate. Since we cannot do this without establishing deferral criteria—a primary goal for this work—we instead defer randomly such that the exact target deferral rate is reached for every treatment:

$$p(\text{deferral}) = \max(DR, \frac{d_r - d_e}{t_t - t_p})\mathbb{1}(d_e < d_r), \tag{7.6}$$

where d_r is the number of deferrals required for the target deferral rate (deferral rate times number of tasks in the treatment length), d_e is the number of deferrals that have been executed in this treatment, t_t is the number of tasks in the treatment (30, in our experiments) and t_p is the number of tasks that have been performed. For our analysis, we use the entropy of the output distribution as in previous chapters. This is calculated as:

$$s = -\sum_{j=1}^{o} p(y_j|x, h_1, ..., h_n)\log(p(y_j|x, h_1, ..., h_n)) \tag{7.7}$$

[1]Due to the common use of color as an attribute in the training dataset [2], we chose to restrict our study to individuals with full-color vision. Under this constraint, we used the red and green color scheme.

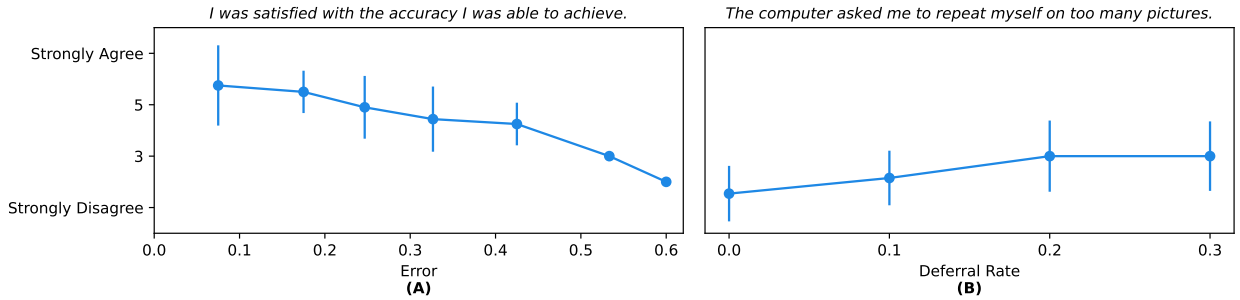


Figure 7.4: Relationship between performance measures—error (A) and deferral rate (B)—and user satisfaction. Error is binned at intervals of 10%. Error bars represent one standard deviation.

7.3 Results

RQ1: Is user satisfaction related to error and deferral rate? One motivation for our investigation is the assumption that error and/or deferral rate strongly influence user satisfaction. We plot the Likert responses of our treatment surveys against the error (A) and deferral rate (B) in Figure 7.4. We found that *satisfaction is related to both error and deferral rate*: the lower the error and fewer deferrals, the higher the reported satisfaction, suggesting we can increase user satisfaction by optimally controlling these two variables. For both performance measures, there appears to be a plateau: for error, the satisfaction for error rates between 0 and 10% (7.5% mean) was not significantly different than the satisfaction for error rates between 10% and 20% (17.5% mean) (Mann-Whitney U, $p > 0.10$), but both had a weak significance ($p < 0.10$) compared to an the 20% to 30% range (24.7% mean), and were perceived as better to a significant degree ($p < 0.05$) than the next two bins (32.7% and 42.5% on the mean). Significance could not be established for higher error rates due to small sample sizes. For deferral, the results were similar: satisfaction with deferral rate differed significantly (Mann-Whitney U test, $p < 0.05$) between deferral rates of 0.0, 0.1, and 0.2, but deferral rates of 0.2 and 0.3 both reported a mean response of 3 to the question *the computer asked me to repeat myself on too many pictures*. Because satisfaction is related to error and deferral rate, *the ideal approach for deferral is not to set deferral criteria based on model-centric qualities such as margin [61], [68], but to target a deferral rate or error directly using the formulation described in Section 7.1*.

RQ2: What are the time dependencies of error, e, and deferral score, s? In order to accurately set deferral criteria, we must be confident that the distributions on which we are working are not changing during the calibration period. If they are—as is likely [224], [299], [301]—any deferral criteria we produce will quickly become inaccurate.

To determine if and when our distributions have settled, we divided the initial queries from all

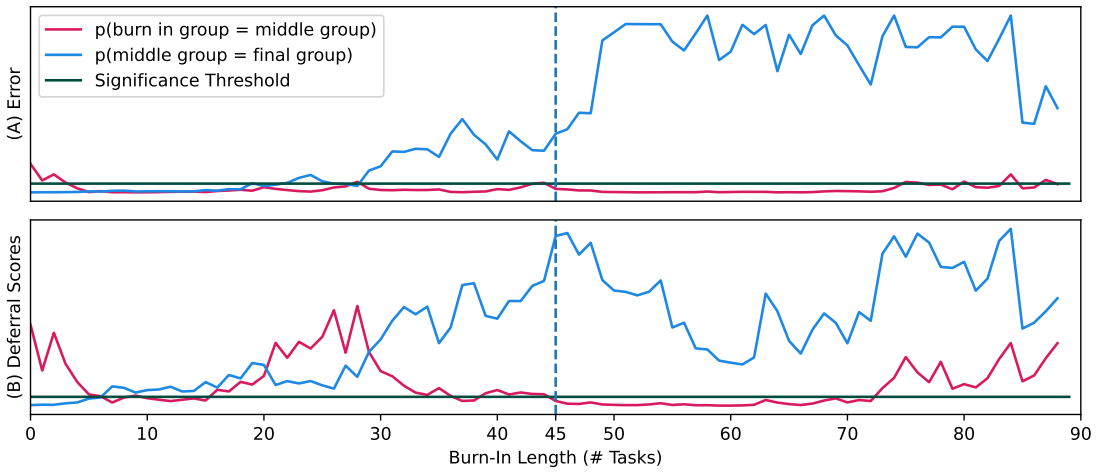


Figure 7.5: The probability of the first n samples being different from the next half of the remaining samples (pink line) and the two halves of the remaining samples after n being different (blue line). When the first condition is true and the second is false, represented by the blue vertical line, the users' mental models have settled.

participants into three groups:

1. The *burn-in* group consists of the first n tasks, where we assume the user is still learning how to interact with the model.
2. The middle group is the first half of the remaining data.
3. The final group is the second half of the remaining data.

We consider the burn-in period to be over when 1) the burn-in group is significantly different from the middle group, and 2) the middle group is not significantly different from the final group. If either of these conditions are not true, it indicates either that a substantial number of burn-in scores are within the middle group—n is too small—or a substantial number of settled scores are within the burn-in group—n is too large. Using a Fisher Exact test to measure similarity of error distributions and a Mann-Whitney U test to measure the similarity of deferral score distributions, we found that if we regard $p < 0.05$ as significant, the earliest that both of these conditions are met for several consecutive timesteps was at 45 tasks. For this reason, we use 45 as a burn-in period throughout this work. This is represented visually in Figure 7.5. We additionally show the mean values for deferral score, errors, and expression length against time in Figure 7.6, with this burn-in period indicated.

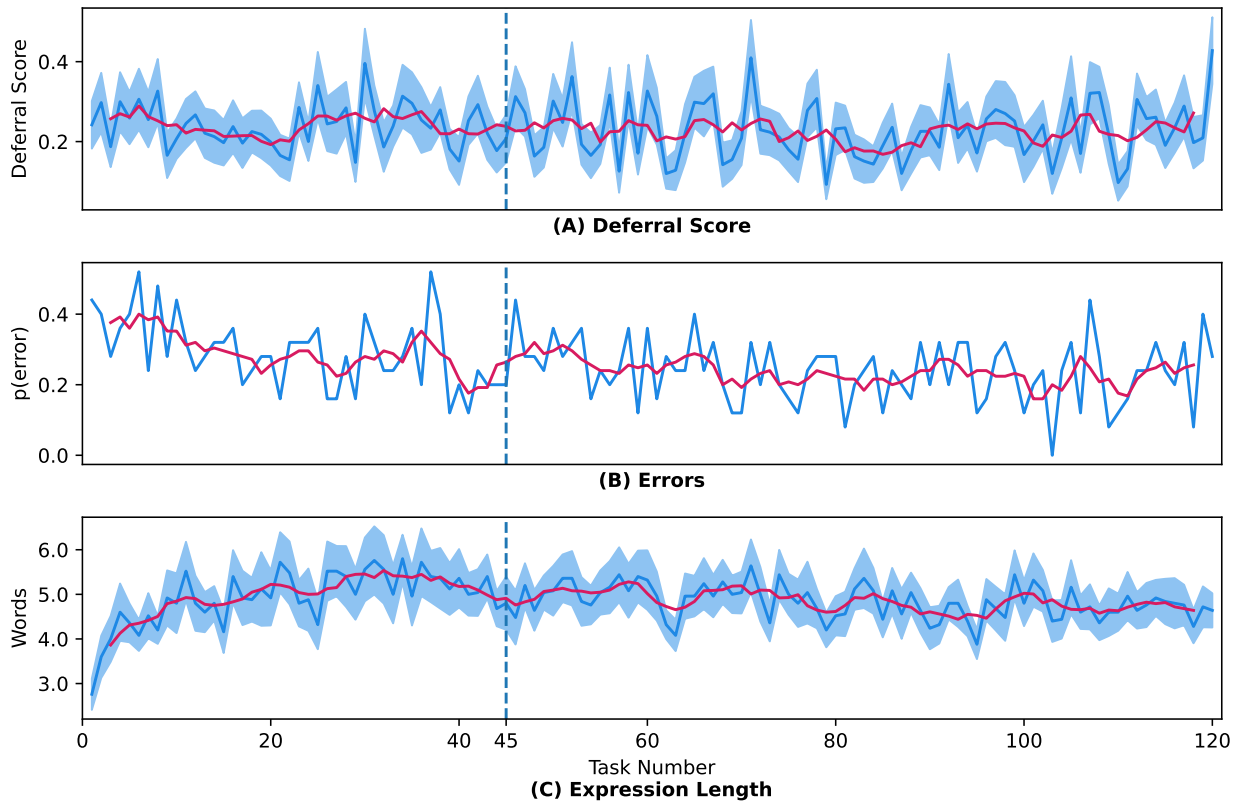


Figure 7.6: The relationship between task number and deferral score, probability of error, and expression length. Mean across the five adjacent task numbers shown in pink. Burn-in period shown with a dashed vertical line. Shaded area is one standard error.

RQ3: Do deferral scores differ meaningfully between users? We compared the deferral scores of all users together, and of pairs of individual users to determine whether the distribution of deferral scores caused by the initial query is dependent on the user. Based on our previous results, we used a 45 task burn-in. We found using a Kruskal-Wallis test that there was a significant effect of user on deferral scores ($p < 0.05$) and a Mann-Whitney U test showed that the distributions were significantly ($p < 0.05$) different for 79 of the 300 user pairings.

As we see in Figure 7.7, the distributions of scores may be different even if the final achieved accuracy is the same: the solid pink line has many more high-certainty examples balanced by more low-certainty examples, while the dashed pink line is more evenly distributed. A Mann-Whitney U test revealed that both pairs are significantly different ($p < 0.05$). This finding shows that, whether or not we control for error, users can produce significantly different deferral scores. When this is considered together with the formulation described in Section 7.1 and the relationship between error, deferral rate, and satisfaction shown in RQ1, it shows that *deferral criteria must be set based on the individual who is using the AI agent.*

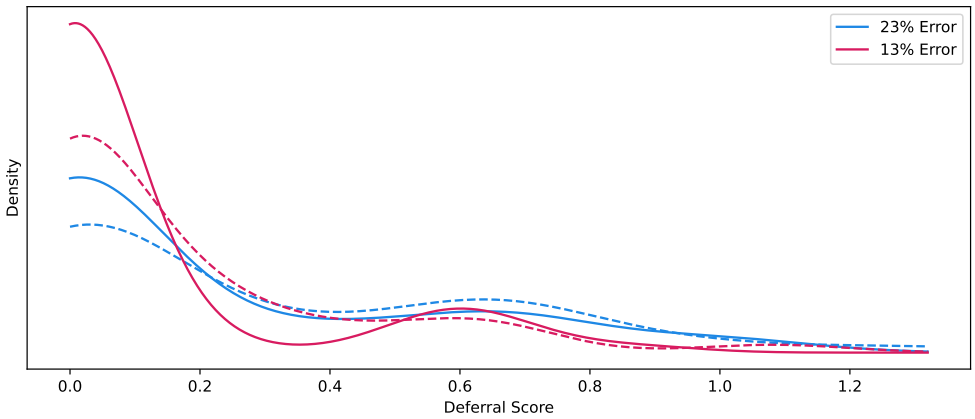


Figure 7.7: Kernel Density Estimate plots of deferral scores frequency for four different users. Despite the pink and blue users having the same overall accuracy, both pairs are visibly and statistically ($p < 0.05$ by Mann-Whitney U) different. Kernel Bandwidth set by Scott's rule.

RQ4: How do users respond when an inference has been deferred?

Analysis of Model Response We begin by analyzing the interaction of the user and deep-learned model after deferral in terms of deferral score and error. Since the data are paired and deferral only begins after the 30^{th} task, we evaluated all deferral responses without regard to the burn-in. Although the standard approach of reformulation used in conversational virtual assistants [60], [66] makes the implicit assumption that the human would provide a better utterance after deferral, we found that the deferral response was of lower quality (from the model's perspective) than the initial query: the mean output entropy in aggregate increased from 0.204 to 0.239 with significance (Wilcoxon Signed-Rank test, $p < 0.05$) and two users showed a statistically significant difference in output entropy between the initial query and deferral response, both of whom had a higher entropy after deferral (Wilcoxon Signed-Rank test, $p < 0.05$). Although we could not show significance, the error in aggregate for the first input (19.82%) was lower than the error for the second input (23.39%).

Although the deferral response was of lower quality than the initial input, we found that by using an aggregation function we could still reduce error over the deferral free condition: error decreased from 19.82% to 17.37% after deferral, 30 out of 89 (33.71%) incorrect answers were corrected, and 19 out of 360 (5.28%) correct answers were made incorrect. The results were similar when data from the burn-in was included: error decreased from 19.45% to 18.01%, 25 out of 72 (34.72%) incorrect inferences were made correct, while 18 out of 289 (6.23%) correct inferences were made incorrect. Although this reduction in overall error suggests the importance of a well-chosen aggregation function, McNemar's test did not reveal significance ($p > 0.05$).

This finding provides two important insights into these kinds of problems. First, since the

	Example	Instances
Identical	bottom left bed → bottom left bed	7
Rephrase	giraffe on the left → left giraffe	116
Same Detail	donut with chocolate sprinkles → donut at the bottom	169
More Detail	the car on the right → the car covered by snow on the right hand side	66
Less Detail	plants in green basket behind roses → plants in green basket	91

Table 7.1: Types of deferral responses and quantity of each seen in our experiment.

deferral response is generally of lower quality than the initial query, the naive reformulation approach [60] is insufficient: not only will error increase with an increased deferral rate after reaching a minimum [295], but deferral rates greater than this minimum may actually have a higher error than the deferral free condition. Therefore, it is critical to maintain state and use a meaningful aggregation function. Second, $p(s_2|s_1, u)$ can not be approximated using $p(s_1|u)$, meaning that prior to being able to target an error via deferral using Equation 7.5, we must perform multiple deferrals to characterize $p(s_2|s_1, u)$.

Input-Space Analysis In addition to examining how the model responds to initial queries and deferral responses, it is informative to characterize how users respond in the input space. On an individual basis, there were three users with a statistically significant difference in sentence length (Wilcoxon Signed-Rank test, $p < 0.05$), none of whom also had a statistically significant difference in deferral score. All of these users had a greater length for their deferral response. To provide further understanding of deferral responses, we grouped all 449 examples[2] into five broad categories: *Identical*, where the first phrase was re-used without change; *Rephrase*, where the semantic meaning and detail remained unchanged despite a change in wording; *Same Detail*, where there were meaningful semantic differences but roughly the same amount of overall information; *More Detail*, where the second input either added data to the previous phrase or used a clearly more detailed independent phrase; and *Less Detail*, where the deferral response contained less information than the initial query.

We show the number of times each category occurred in Table 7.1: most of the deferral responses were of equivalent detail, with users slightly preferring to modify semantics (same detail) over syntax (rephrase). This large proportion of rephrasing events (25.8% of deferral responses) suggests that methods used for extracting deferral responses from datasets, such as random sampling [48] or minimum word overlap [68], are likely insufficient for many settings. Although no participant systematically produced shorter responses to a statistically significant degree, aggregate analysis suggests that users believe it to be more likely that the model will understand less information better than more, consistent with previous findings that humans use shorter messages with chatbots [252].

[2]Due to a connectivity error, one user had one fewer deferral, leading to 449 responses instead of 450.

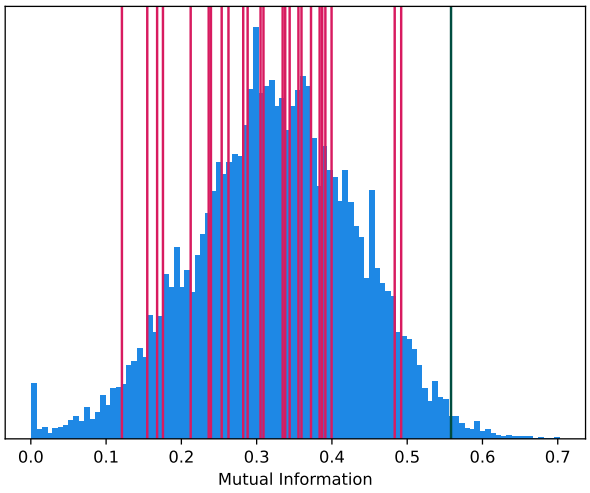


Figure 7.8: The mutual information conditioned on individual users (pink lines) superimposed on a distribution of unconditioned mutual information (blue histogram). The green line represents the p ¡ 0.05 significance threshold.

Interestingly, we did not find any cases where the deferral response was ambiguous without the initial referring expression (*e.g.,* the leftmost flowers $\rightarrow$ the yellow ones), meaning the increased entropy after deferral was likely to be due to the aforementioned shortening of messages or the fact that training data [2] consisted entirely of initial requests. Additionally, since we did not explicitly state that the AI remembered the previous interaction, this suggests that users assume the AI agent does not remember previous queries, and any deferral method with memory should therefore make this feature explicit to the user.

RQ5: Does knowing the user provide additional information about the mapping between probability of error and deferral score? From the formulation described in Equation 7.5, we see that it is important to consider the relationship between deferral score and the probability of error—$p(e|s_1, u)$ and $p(e|s_2, u)$—if we want to target an error. Due to the large number of samples required to build this distribution, it would meaningfully reduce the calibration time required to set deferral criteria if these distributions were independent of the user (*i.e.,* $p(e|s_1) = p(e|s_1, u) \ \forall \ u$). To determine if this is the case, we measured whether the deferral score gives us more information about the probability of error if it is conditioned on a user. We measured this using Mutual Information (MI), which describes the dependency of two random variables and has been used for tasks such as measuring the quality of fused images [302] and choosing network weights to prune [303], and compared the strength of this relationship when it is and isn't conditioned on an individual user using a permutation test:

1. Draw 10,000 random sets of 75 score-error pairs (120 tasks less the 45 task burn-in) and calculate the mutual information of each one using the EDGE estimator [304].
2. For every individual user, calculate the mutual information between the deferral score and the error.
3. Compare every individual user's MI to the distribution of randomly generated MIs.

We see the results of this in Figure 7.8. We see that none of our 25 evaluated users allowed us to reject the null hypothesis that knowing the user does not increase mutual information ($p > 0.05$). Though this phenomenon would benefit from further study, this finding suggests that $p(e|s_n)$ is independent of the user and *we can use dataset-based model calibration to estimate the probability of error given a deferral score*. Doing this would dramatically decrease the time required to set deferral criteria over characterizing the model based on individual user interactions.

7.4 Setting Deferral Criteria

Our work has thus far shown that user satisfaction is dependent on both deferral rate and error and that users are unique, but has not been explicitly shown that it is better to set deferral criteria based on the data from individual participants. To investigate this, we set deferral rate and threshold based on two objectives:

1. Bring the deferral rate closest to the target value. (*Minimizing Absolute Error*)
2. Produce an upper bound on the deferral rate. (*Upper Bounding*)

We considered three calibration datasets, all of which are evaluated on the final 38 human inputs (half of the tasks remaining for an individual after burn-in):

- RefCOCO: set deferral criteria using phrases from the RefCOCO dataset [2]. This dataset was constructed using a crowdsourced two-player human-to-human game, where the benefit of increased size may be outweighed by the difference in human-to-human communication described in the introduction. We used only images that met the criteria defined in our experimental setup, and removed all images that were seen by the user for whom we are setting deferral criteria.
- Multi-User: set deferral criteria using phrases collected from other users in the experiment. This dataset is slightly smaller than RefCOCO, but collected in the same setting as the test data. We remove from the calibration set all phrases from prior to the burn-in (the first 45), as well as all images that were seen by the target user.
- Individual: set the deferral criteria based on the first half (37) of phrases after the burn-in. Although the calibration set is much smaller, it will also capture the user's behavior much more accurately.

	0.1	0.2	0.3
RefCOCO	3.82 ± 0.63	5.66 ± 0.88	8.41 ± 0.90
Multi-User	4.22 ± 0.67	5.94 ± 0.88	6.97 ± 1.14
Individual	3.97 ± 0.70	6.95 ± 0.91	6.79 ± 1.09

Table 7.2: Mean absolute error when targeting deferral rates of 0.1, 0.2, and 0.3. Displayed tolerances are standard error.

	0.1	0.2	0.3
RefCOCO	7	9	7
Multi-User	10	10	9
Individual	0	1	2

Table 7.3: Number of violations when upper bounding deferral rates of 0.1, 0.2, and 0.3.

Minimizing Absolute Error The method for minimizing the absolute error with respect to a deferral rate is to find the value of the appropriate percentile in the calibration set. We see the result of this in Table 7.2: no method has a mean greater than one standard error from another. Although the performance of RefCOCO and Multi-User deferral criteria is likely stable—having approximately 2,900 and 1,650 samples, respectively—deferral criteria based on an individual may improve with a longer calibration period. In other words, while setting deferral criteria using an individual does not improve over aggregate datasets in this analysis, a longer interaction period may change this finding.

Upper Bounding Given our finding that user satisfaction is linked to both deferral and error rates, it makes sense to set an upper bound on the the respective value (*i.e.,* I want the user to be at least this happy) instead of simply attempting to match the desired value as closely as possible at the risk of great dissatisfaction for some users. This is particularly critical for faster calibrations, as minimizing the absolute error does not consider the number of examples in the calibration set.

To upper bound the error rate, we use the finding of Gascuel & Caraux [305] that when $\bar{p}$ verifies:

$$\delta = \sum_{i=0}^{k} \binom{n}{i} \bar{p}^i (1-\bar{p})^{n-i}, \tag{7.8}$$

then

$$p(p - \bar{p} \geq 0) \leq \delta, \tag{7.9}$$

where p is the true probability of deferral for the proposed criteria, n is the number of examples, k is the number of deferrals for a given deferral criteria, and δ is our desired confidence. Like Geifman & El-Yaniv [58] do for the selective classification task, we solve this using a binary search across deferral criteria (the threshold, t) with $\delta = 0.05$.

For this goal, deferral criteria produced by only examining the individual unambiguously performs better (Table 7.3). For RefCOCO and Multi-User, the deferral criteria is set with high confidence due to the size of the calibration set, but is incorrect due to the differences in score distributions between individuals. In other words, thresholding based on an individual's score distribution is necessary for producing accurate upper bounds, regardless of the calibration set size.

7.5 Conclusion

This chapter addressed the shortcomings of dataset-based works in deferred inference and provided a human-centered method for setting deferral criteria. Through a study of 25 users, we examined not only whether accuracy improves with the addition of a deferral mechanism—as in previous chapters—but also the nuances of the interaction between individual users and deep learning models. Most broadly, we report two major findings: 1) satisfaction is dependent on both error and deferral rates (RQ1), and 2) we must consider the individual user when we set deferral criteria (RQ3). The second finding is reinforced in practice by an evaluation of different methods for setting deferral criteria: despite having two orders of magnitude less data, deferral criteria set with user-specific data perform the same or better than those set on large datasets containing many individuals. We additionally find that it is critical to characterize the deferral response separately from the initial query (RQ4) but that we can characterize the model's calibration—the relationship between score and error—independently from the individual (RQ5).

Though deep neural networks are inherently unpredictable, we believe these findings of are sufficiently general to extend to other relevant applications. Many are likely to be model-agnostic: people have subtly different linguistic preferences, and the ways in which they change their language after deferral is a function of the human's perception of the model, not the model itself. The broad concepts for setting deferral criteria as a threshold on a deferral score is also likely to generalize, though the deferral function itself will have to change if the output format is different: Visual Question Answering [6] often uses a softmax output [1], [26], [306], but there is no trivial equivalent to entropy in, for example, the bounding box output of a visual object tracker [54].

This chapter concludes the work presented in this dissertation. Throughout this work, we have proposed and examined the benefit of deferring inference in human-AI teams. We have demonstrated that the high dimensionality of deep neural networks makes it more difficult than simply determining if either the input or the model is correct—we must simultaneously consider both. Having shown this, and demonstrated a method that does so, we propose a general framework and evaluation, then use it to evaluate several proposed deferral functions, reducing eror by up to 48.7% at a reasonable level of effort. We then shifted from the crowdsourced setting and evaluated how humans truly interact with deep learned models. Not only does this show that improving

accuracy and deferral rate can improve human satisfaction, but it also gave us information as to the best method for determining when to defer. Though this presents a complete view of one deferred inference implementation, there is still much opportunity to increase performance and overall experience through the mechanism of deferred inference.

CHAPTER 8

Conclusion

Deep neural networks are powerful tools, capable of combining multiple sources of high-dimensional data to produce accurate results. However, the use of the oracle assumption during training and evaluation makes them sub-optimal for interaction between humans and AI agents. For this reason, this work formalized and explored the concept of deferred inference with hazy oracles. Deferred inference allows the human to provide additional information upon request, leading to improvements in both quantitative performance and qualitative user experience in a human-AI team. While pursuing this goal, we learned many lessons, which we summarize here, alongside practical uses, ethical concerns, and avenues for future work.

8.1 Lessons Learned

We must consider the human-AI team holistically. Approaches to similar problems attempt to detect either low-quality human inputs—defined a-priori based on semantic quality—or outputs that are likely to be erroneous. However, the relationship between input quality and error in such problems is complex when deep models are asked to combine human inputs with other input modalities. Because of this, there is no guarantee that a low-quality human input will result in a low-quality output nor that a new human input will improve a low-quality output. Instead, we must set deferral criteria and evaluate holistically—is the model output incorrect *and* will a new human input improve performance?

The deferral response must be treated as imperfect. Under current paradigms, a deep model that has performed a low-quality inference will do one of two things: it will return the result with the assumption that the human will recognize the failure and try the inference again, or it will return a failure message and—again—expect the human to try again. In addition to the time cost of receiving an incorrect answer, these methods typically ignore information contained in previous queries. Not only is this undesirable from a human-AI interaction standpoint [307], but it can result in degraded quantitative performance: the deferral response is typically worse than the initial query (Chapter 7),

and naively accepting the deferral response may result in simultaneous increases of human effort and error (Chapter 4).

We additionally show that if we treat the deferral response as imperfect, we can mitigate both of these phenomena and choose the best method overall through the aggregation functions of *smart replacement*, which chooses the better of the two inferences, and a method based on multiplying subsequent probabilities. The former—smart replacement–has the benefit of being applicable in any situation where a confidence measure exists, while the latter requires the task model output to be a distribution, but is capable of combining complementary information across human inputs.

Deferral constraints play an important role in overall performance. Generally speaking, previous works set deferral constraints a-priori based on technical factors. For example, Uehara *et al.* [62] assume that every inference receives a deferral, while other works [61], [68] arbitrarily choose a margin above which deferral occurs. We find, however, that factors such as the deferral depth constraint and deferral rate meaningfully affect the performance. In other words, not only can we improve overall performance by addressing these components—most methods perform best at a DDC greater than one—but the best deferral methods change as the DDC or DR changes.

The behavior of individuals must be considered during evaluation. The choice of previous work to set deferral criteria a-priori is also consequential when the behavior of individual users is considered. Specifically, because different users have different distributions of deferral scores, any attempt to target a deferral rate or error must be customized instead of set arbritrarily or based on large datasets. Further, dataset-based evaluation breaks temporal relationships between deferrals and, in doing so, ignores the fact that deferral responses are of significantly lower quality when evaluated with the model in the loop.

8.2 Benefits and Ethical Concerns

Deployment of Deep-Learned Architectures While research continues to improve the performance of task models under the supervised paradigm, a model for human-in-the-loop inference that can be operated under the oracle assumption will most likely never be achieved. Not only is it unrealistic to expect every human input to be high quality [17], but high-quality inputs may still be misunderstood by the model due to high-dimensional decision boundaries of such models [308]. Deferred inference, therefore, can serve as a way to make models trained with the oracle assumption operate in a human-friendly way, bridging the gap between academic research and meaningful applications such as answering visual questions for the visually impaired [7], enabling service robots for the elderly [9], or simply improving reformulation handling in conversational virtual assistants [60].

Deployment of Smaller Models While current work in deep learning often focuses on improving performance relative to benchmarks by increasing the number of parameters and training datapoints [309], such large models can be undesirable for many reasons. One such reason is the environmental impact of large deep-learned models: one work [310] estimates that the powerful GPT-3 model [78] required the equivalent of 703,808.01 vehicle-kilometers of energy to train. Another reason is privacy: large models generally cannot be run on edge devices, meaning the data is sent to remote servers. Not only does this make systems fail under poor connectivity, it potentially exposes personally identifying information [311]. Both of these concerns can be addressed by smaller models, at the cost of lower accuracy. Deferred inference could be used to mitigate the effect of this trade-off by increasing the effective accuracy of the smaller models.

Fairness The fairness concerns of deferred inference are closely tied to those of deep learning in general, and can be broadly grouped into two categories: biased features and underrepresented users. Biased features are a well documented phenomena: deep models often capture culturally insensitive or otherwise undesirable correlations [312]–[314] or amplify bias present in the dataset [315]. Because works in this area typically seek to detect and address bias in the classifier [316], [317], it is difficult to predict the effect of deferred inference on models that have already been trained in a biased fashion. Given some exploration in this direction, however, there is potential for novel deferral functions that can detect and defer inferences that are likely to produce undesirable results.

A similar problem is the experience of users who are underrepresented in the training and evaluation data. For example: while we use elder-care robots as a motivating application, the 50-69 age bracket is underrepresented in the population of Amazon Mechanical Turk workers, and males are additionally underrepresented within that group [318]. If there are relevant differences between the language used in the overrepresented (ages 18-39) and underrepresented (ages 50-69, male), the experience of these groups will be substantially different.

Unlike issues related to bias, however, failures due to this kind of distribution shift are likely to be detectable using the out of distribution or uncertainty methods discussed in our related work. Because of this, deferred inference would most likely improve the experience of such underrepresented users. While deferred inference is likely preferable to inference under the oracle assumption—confidently returning the wrong answer—or selective prediction—returning no answer for these users, it is not a substitute for model improvements and more representative datasets, as the need to provide deferral responses represents an additional burden.

8.3 Limitations and Future Work

Application-Aligned User Studies Our user study focused on the difference between how the model responded to individual users and how those differences affected the deferral criteria. Because of this, no great care was taken to procure participants that represented a target demographic. Though we recruited an even number of male and female participants, age was lower and technical competence likely higher than the general adult population. The effect of this varied based on the particular research questions: the relationship between satisfaction and error, the fact that score distributions differ between users, and the mapping between probability of error and deferral score are unlikely to be affected by demographic shifts, while the particulars of how deferral responses differ from initial queries may be related to the demographics of the study group.

Additionally, the short-term nature of the study leaves a few technical questions unanswered: 1) can deferral with custom criteria reduce error to a statistically significant degree? Our choice to use random deferral instead of basing our deferral criteria on entropy led to many high-certainty (and correct) answers being deferred. Although we found that it was much more likely that the post-deferral answer was correct given an incorrect pre-deferral answer (33.71%) than the opposite (5.28%), the low quality of our deferral function resulted in a small—about 2.5%—decrease in error that could not be established as significant. 2) Can we target an accuracy? Our evidence suggests that we can use datasets to estimate the probability of error given a deferral score—mitigating some concerns about interaction length—but the nature of Bernoulli variables makes it difficult to produce meaningful evaluations at small sample sizes. 3) Is there a longer-term shift in user behavior that was not captured in our study? In other words, we may need to re-calibrate the deferral criteria over time to maintain our target value. For these reasons, we believe it will be beneficial to run longer-term analyses on the target subgroups of specific applications.

Separating Sources of Input Error While Chapter 3 showed that it is important to consider not only whether error has occurred, but whether the error can be corrected by a deferral, subsequent chapters used methods such as output entropy—which does not account for whether or not error can be corrected—as a deferral score. This was for technical reasons: to our knowledge, no work other than DAER focuses on determining the source of error in multimodal problems, and DAER is limited to cases where human inputs can easily be designated as correct or incorrect. Because of this limitation, DAER has no straightforward analogue for high-dimensional human inputs such as language.

Future work should address this shortcoming, as it has a meaningful effect on the AI agent's behavior. For example, performance can be improved in the visual question answering application

by requesting new questions, requesting new photos, or both [17]. A question-photo pair where the photo is blurry should result in a request for a new photo, while a pair where the question is ambiguous should result in a request for a rephrase. This becomes more challenging when the issue is not an input-space shortcoming that can be trained via a labeled dataset, but instead related to the training distribution and decision boundary.

Alternate Deferral Modalities Throughout this work, we used the deferral modality of *please try again* because it is familiar to the standard user of technology [60] and is compatible with the current supervised learning paradigm. However, naively requesting a reformulation is not necessarily the best way to prompt the user after deferral. This is perhaps most evident in Table 7.1, where users often shortened their utterances without providing complementary information that could be used by the model, but it also has an intuitive motivation: a human that is uncertain will typically ask specific questions related to their understanding of the request, not just ask their conversational partner to repeat the utterance.

Some works in both deferred inference [61], [62] and crowdsourcing [97], [99], [319] propose alternate query modalities such as generating questions that are easily answered by the human, but such approaches hold their own set of challenges: generating text descriptions of target objects [61] is a meaningful challenge that does not extend to other applications, while methods that generate follow-up questions for visual question answering [62] represent the same architectural challenges, and are based on explainability datasets—leading to trivial follow-up questions that do not reflect the model's true internal state. Future work should explore how to modify the form of the follow-up questions to increase both the intuitiveness of the interaction and the ability to solicit the most effective information.

Datasets for Deferred Inference In this work, we evaluated across a variety of datasets, but the evaluation in our penultimate chapter demonstrated that any interaction that takes place with individuals must consider both characteristics of the individual and the way deferral responses change across the number of deferrals. Since our evaluation datasets—as well as large-scale datasets in general—are crowdsourced, these factors are currently ignored. To make the evaluation realistic for interactions with individuals, future work must procure a dataset that contains multiple deferral responses—up to ten—for every potential task, sorted by user.

Integrating Human Confidence to Deferral One interesting finding for the hierarchical scene classification task in Chapter 3 was that deferral based on the confidence of the coarse model is more effective that deferral based on the confidence of the task model. It is intuitive that this would apply to human-provided input as well: if the human communicates indicates uncertainty when

providing their input, we may benefit more from using that signal than only the model confidence.

In our evaluated applications this is not that straightforward, as crowdsourced datasets for our applications do not measure any signal that meaningfully represents emotion. However, our ultimate goal is not simply a text-based cropping application: consumer-level tools such as CVAs and, in the future, home service robots, will be operated by voice. Although we are unaware of any work that specifically targets certainty, auditory emotion recognition is an active area of research [320] that can be leveraged to incorporate human confidence into the inference.

8.4 Conclusion

Deep learning has the potential to improve lives in meaningful ways via novel interaction modalities. However, the formulation of supervised learning is particularly unsuitable for working with humans: by expecting the human to provide one input for every output, such formulations force the user to perform incorrect inferences even when the correct answer can be easily obtained via a follow-up query. To address this, we formalized the concept of deferred inference and implemented it across a variety of applications and assumptions. Not only does this formulation allow for an interesting perspective on inference and human interaction with deep models, but it is effective: we can reduce error by over 48% in some settings. Although opportunity for improvement exists in isolating causes of error, considering multiple input modes, and examining how humans interact with deep-learned models over longer timescales and different interaction modalities, this work presents the first formalization of this problem and provides the opportunity for significant, meaningful, future work in impactful tasks such as human support robots.

APPENDIX A

Crowd Queries Used for VOT

When collecting the crowdsourced bounding boxes for our evaluation, we provide the crowd worker with a text query and ask them to produce a bounding box around the object. This appendix contains the first-frame images, gold-standard bounding boxes, and text queries used.

Figure A.1: Please draw a bounding box around all parts of the person in the white shirt, but not their backpack.

Figure A.2: Please draw a bounding box around the head and helmet of the bicyclist.

Figure A.3: Please draw a bounding box around all parts of the skier.

Figure A.4: Please draw a bounding box around the head of the tiger toy.

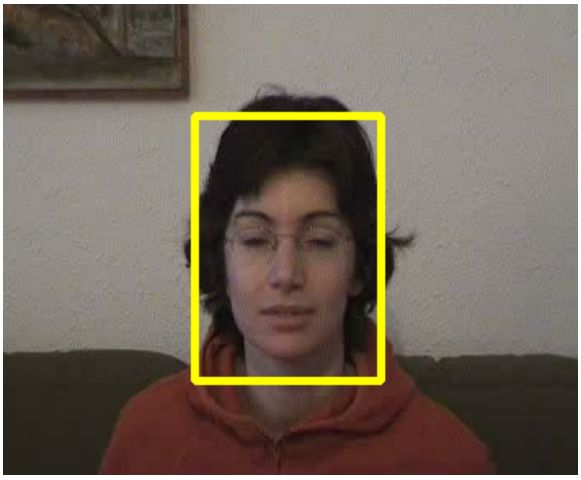


Figure A.5: Please draw a bounding box around the head and neck (above the shoulders) of the person.

Figure A.6: Please draw a bounding box around the white car following the black van.

Figure A.7: Please draw a bounding box around the helmet of number 59 on the blue team.

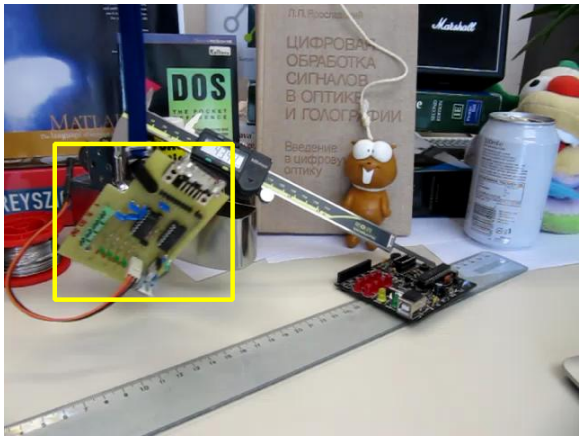


Figure A.8: Please draw a bounding box around the circuit board connected to 3 wires.

Figure A.9: Please draw a bounding box around the head of the person holding the trophy.

Figure A.10: Please draw a bounding box around all parts of the person in the far right of the image

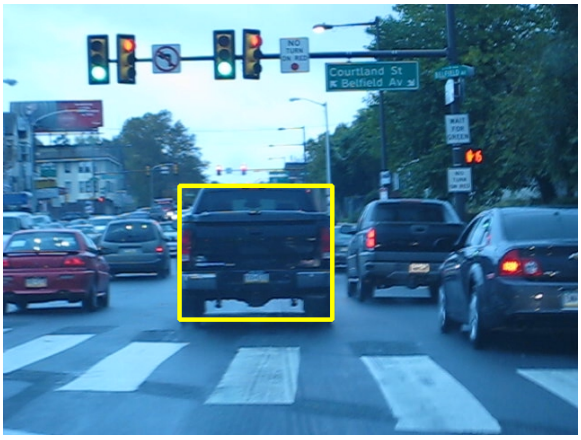


Figure A.11: Please draw a bounding box around the black pickup truck with its brake lights off.

Figure A.12: Please draw a bounding box around all parts of the person, excluding their front leg and foot.

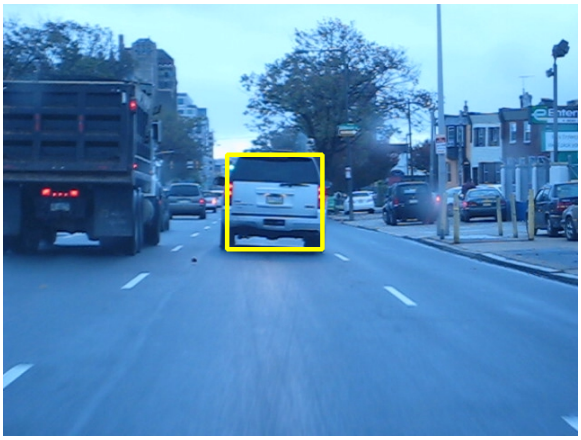


Figure A.13: Please draw a bounding box around the white sports utility vehicle (SUV).

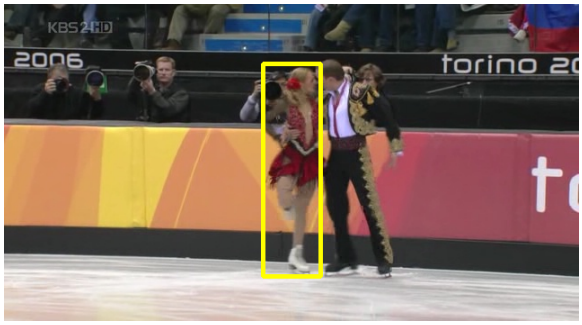


Figure A.14: Please draw a bounding box around all parts of the female figure skater.

Figure A.15: Please draw a bounding box around all parts of the male figure skater, except the hidden arm.

Figure A.16: Please draw a bounding box around the white car.

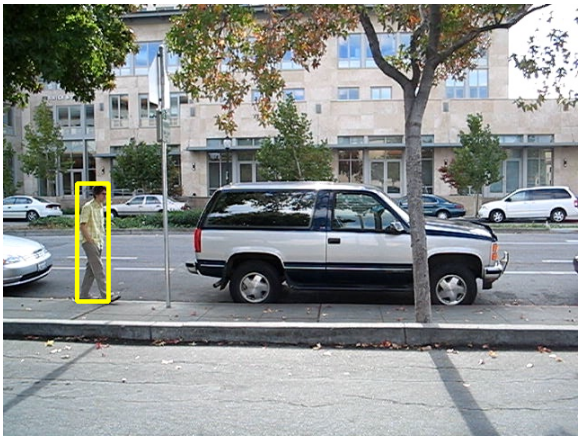


Figure A.17: Please draw a bounding box around all parts of the person, except the front foot.

Figure A.18: Please draw a bounding box around all parts of the dancer, excluding their hands and feet.

Figure A.19: Please draw a bounding box around the box which holds the pencils, but not the pencils themselves.

Figure A.20: Please draw a bounding box around the parts of the large bird that are covered in feathers.

Figure A.21: Please draw a bounding box around the body (not the wings or legs) of the rightmost bird.

Figure A.22: Please draw a bounding box around the Clif bar.

Figure A.23: Please draw a bounding box around the head of the person standing up.

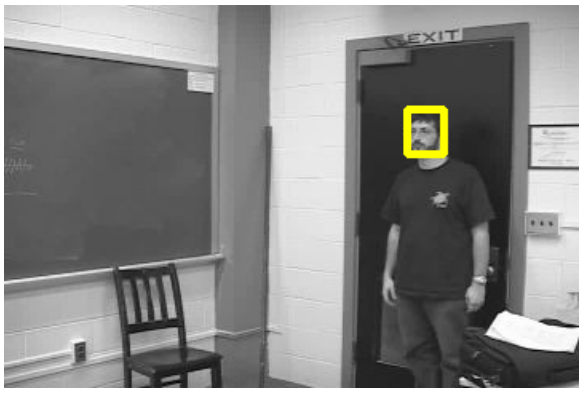


Figure A.24: Please draw a bounding box around the head of the person.

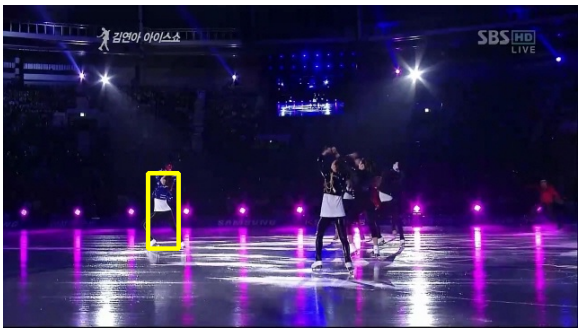


Figure A.25: Please draw a bounding box around all parts of the figure skater on the left, except their hands.

Figure A.26: Please draw a bounding box around the animal toy hanging from the string.

Figure A.27: Please draw a bounding box around all parts of the person dressed in all black, except their back leg.

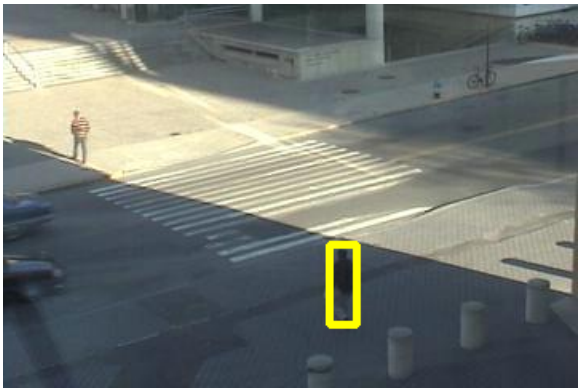


Figure A.28: Please draw a bounding box around all parts of the person in the black shirt.

Figure A.29: Please draw a bounding box around the head of the guitarist closest to the pianist.

Figure A.30: Please draw a bounding box around all parts of the sprinter in lane 4.

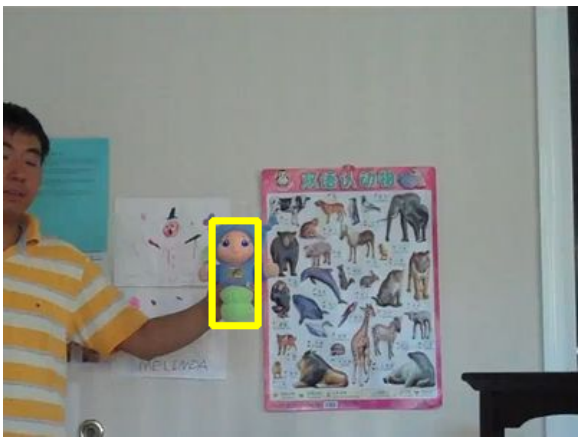


Figure A.31: Please draw a bounding box around all parts of the doll except its hands.

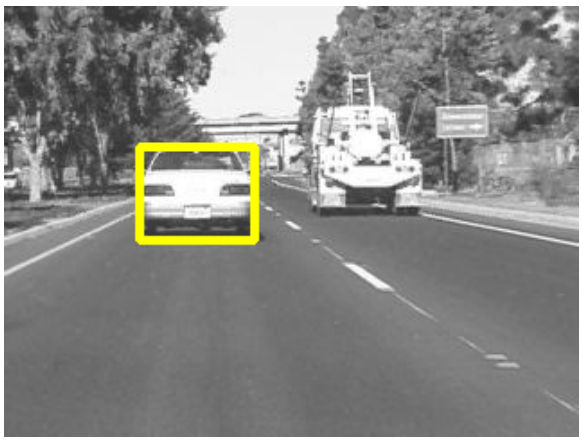


Figure A.32: Please draw a bounding box around the white car.

Figure A.33: Please draw a bounding box around the body and head of the fish statue.

Figure A.34: Please draw a bounding box around the light-colored car on the left.

Figure A.35: Please draw a bounding box around the minivan.

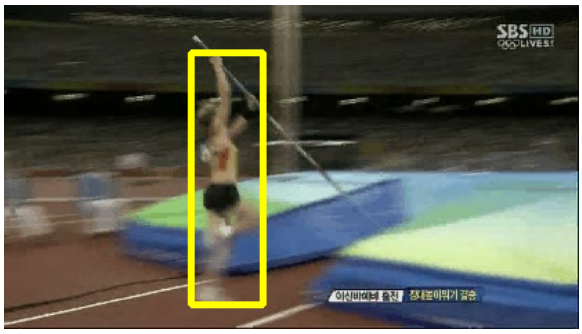


Figure A.36: Please draw a bounding box around all parts of the pole vaulter.

Figure A.37: Please draw a bounding box around the silver car.

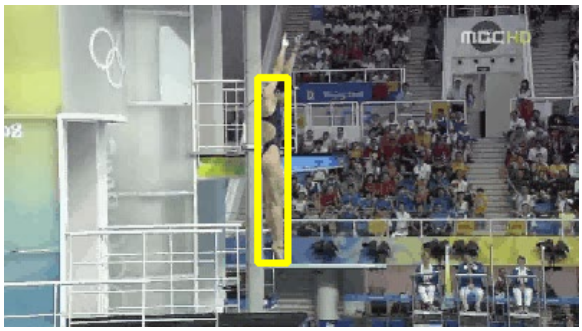


Figure A.38: Please draw a bounding box around all parts of the diver, excluding their arms.

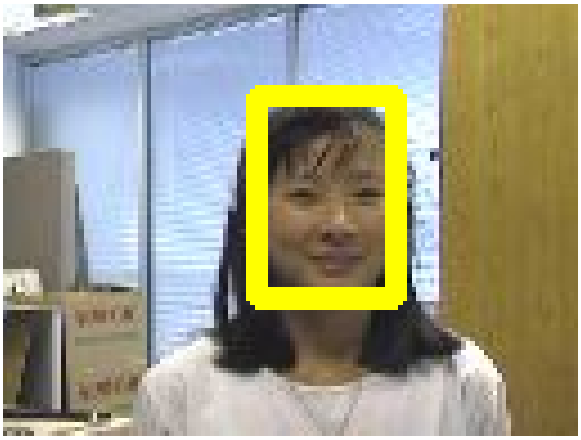


Figure A.39: Please draw a bounding box around the person's head.

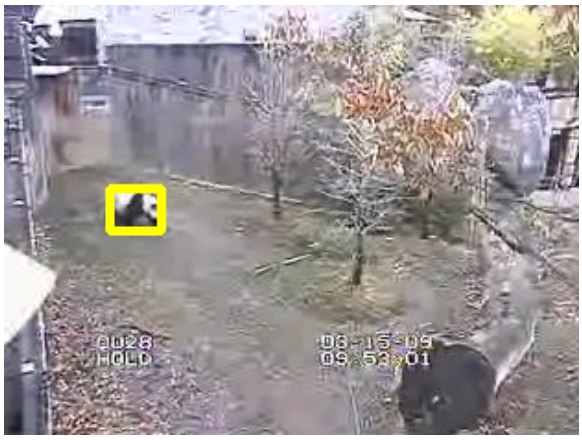


Figure A.40: Please draw a bounding box around all parts of the panda.

Figure A.41: Please draw a bounding box around the person's head.

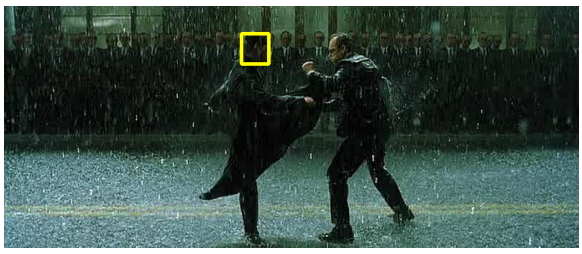


Figure A.42: Please draw a bounding box around the head of the person in the trench-coat.

Figure A.43: Please draw a bounding box around the body of the white suv.

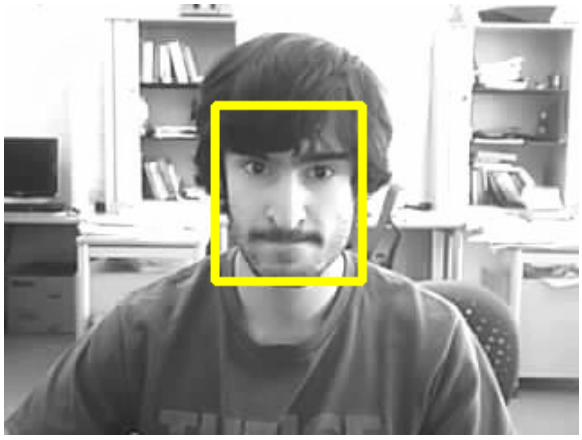


Figure A.44: Please draw a bounding box around the person's face.

Figure A.45: Please draw a bounding box around the character on the poster.

Figure A.46: Please draw a bounding box around all parts of the person.

Figure A.47: Please draw a bounding box around all parts of the person.

Figure A.48: Please draw a bounding box around all parts of the person with the bag.

Figure A.49: Please draw a bounding box around all parts of the person closest to the crosswalk.

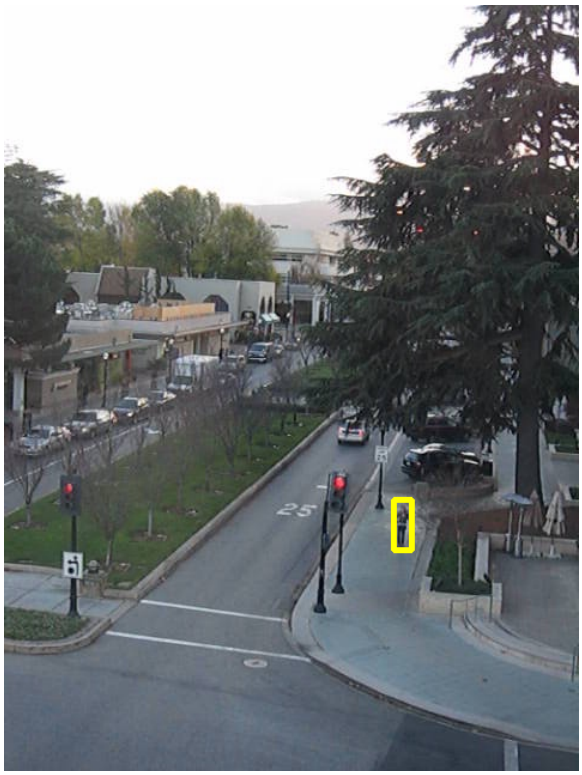


Figure A.50: Please draw a bounding box around all parts of the person.

Figure A.51: Please draw a bounding box around all parts of the person in the street.

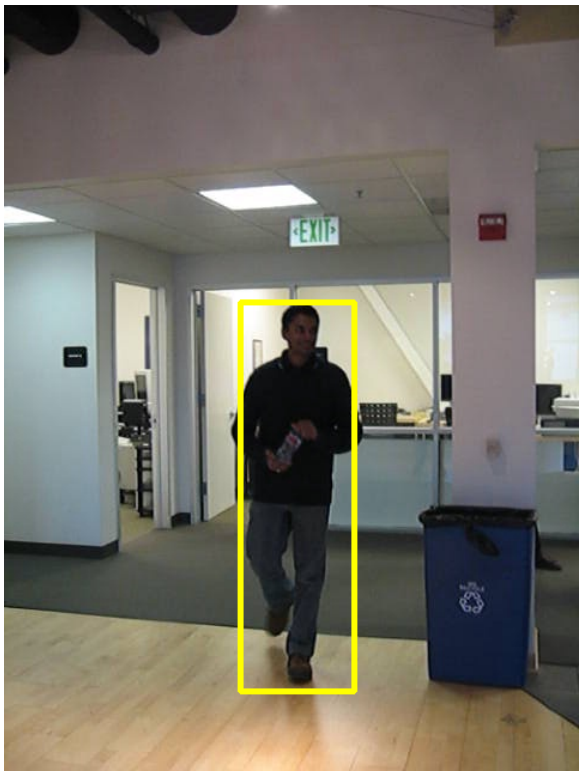


Figure A.52: Please draw a bounding box around all parts of the person.

Figure A.53: Please draw a bounding box around the head of the surfer.

Figure A.54: Please draw a bounding box around the athlete wearing green surrounded by athletes wearing white, from the top of their head to their knees, excluding their arms.

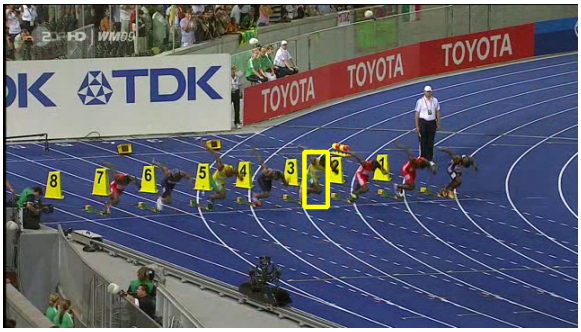


Figure A.55: Please draw a bounding box around all parts of the sprinter in lane 4, except their arms.

Figure A.56: Please draw a bounding box around the head of the kite surfer.

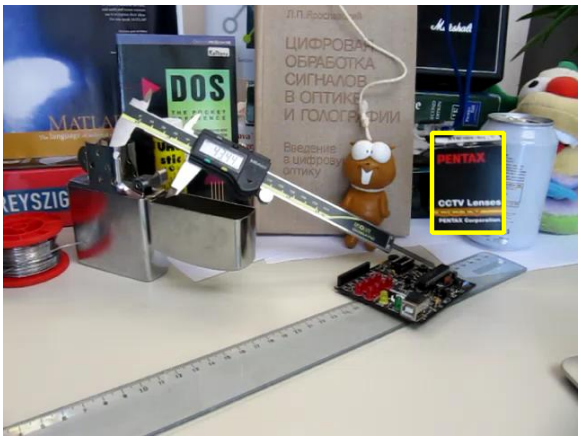


Figure A.57: Please draw a bounding box around the box labeled “PENTAX”

Figure A.58: Please draw a bounding box around the person’s head.

Figure A.59: Please draw a bounding box around the toy cat.

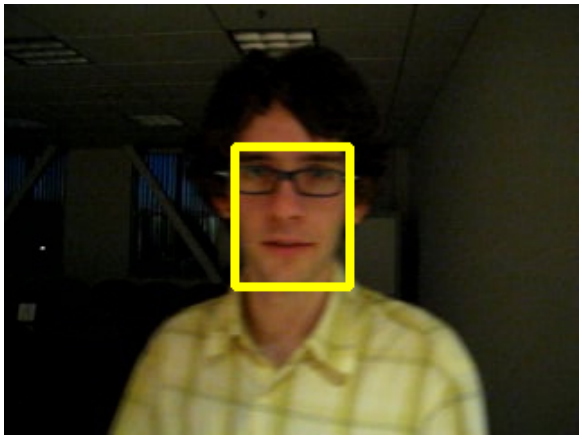


Figure A.60: Please draw a bounding box around the person’s face.

Figure A.61: Please draw a bounding box around all parts of the person in the sleeveless shirt.

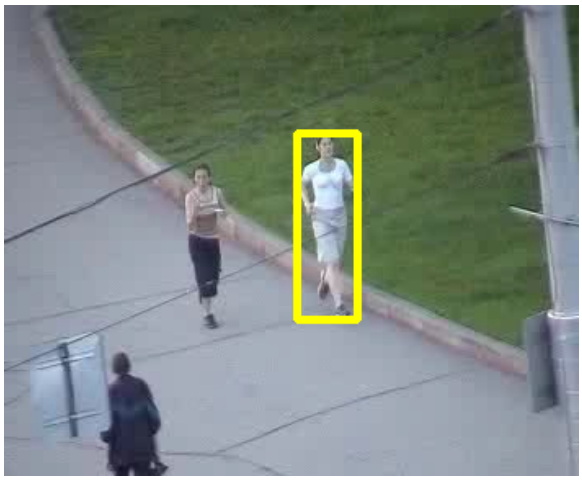


Figure A.62: Please draw a bounding box around all parts of the person in the white shirt.

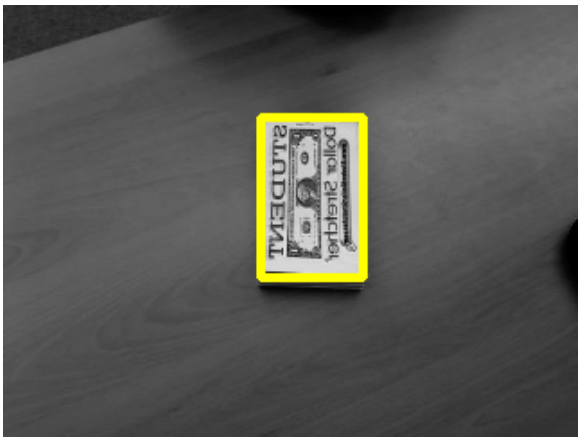


Figure A.63: Please draw a bounding box around the coupon.

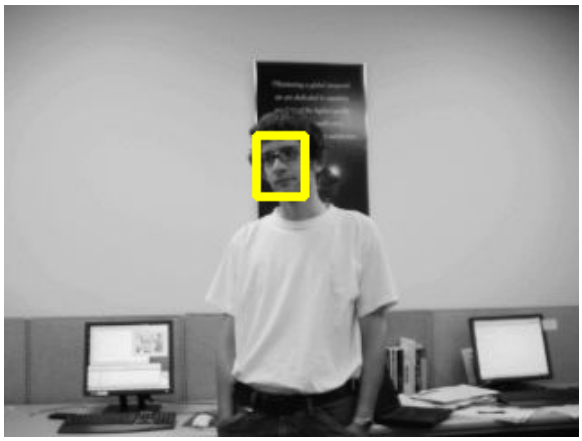


Figure A.64: Please draw a bounding box around the person's face.

Figure A.65: Please draw a bounding box around the toy tiger's head.

Figure A.66: Please draw a bounding box around all parts of the dancer, from the top of their head to their knees.

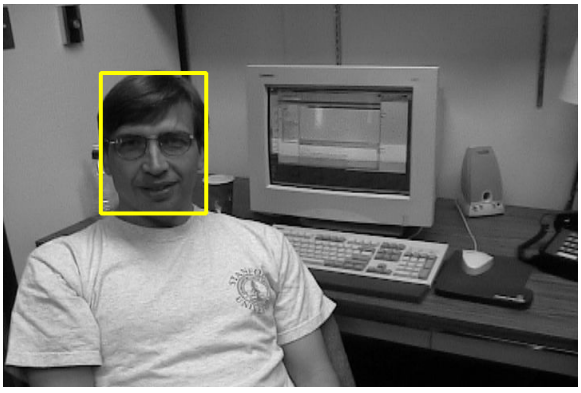


Figure A.67: Please draw a bounding box around the person's head.

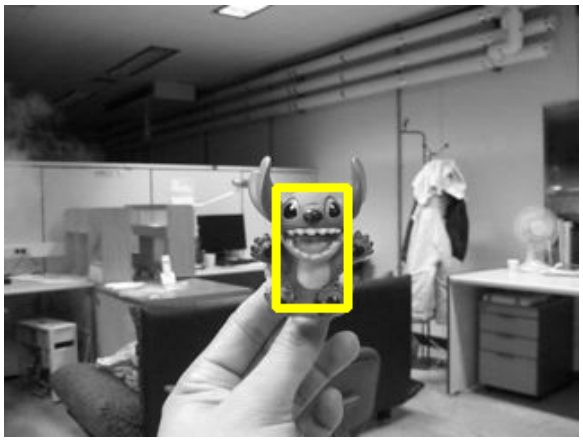


Figure A.68: Please draw a bounding box around the body and head (no ears or arms) of the toy.

Figure A.69: Please draw a bounding box around all parts of the ice skater, excluding the straight leg.

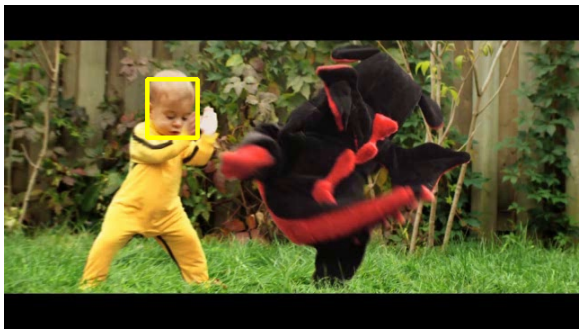


Figure A.70: Please draw a bounding box around the baby's head.

Figure A.71: Please draw a bounding box around the bicycle and its rider.

Figure A.72: Please draw a bounding box around the bottle.

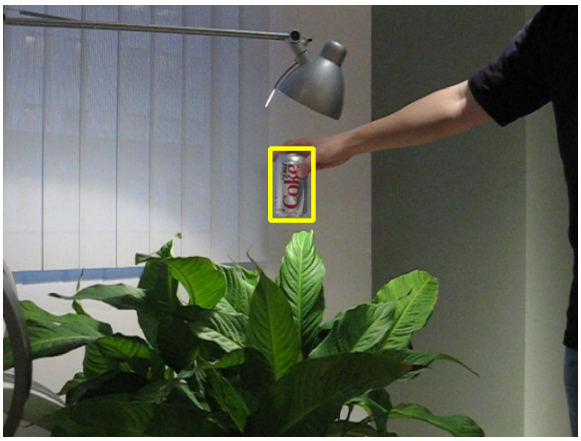


Figure A.73: Please draw a bounding box around the can of diet coke.

Figure A.74: Please draw a bounding box around the head of the person standing up.

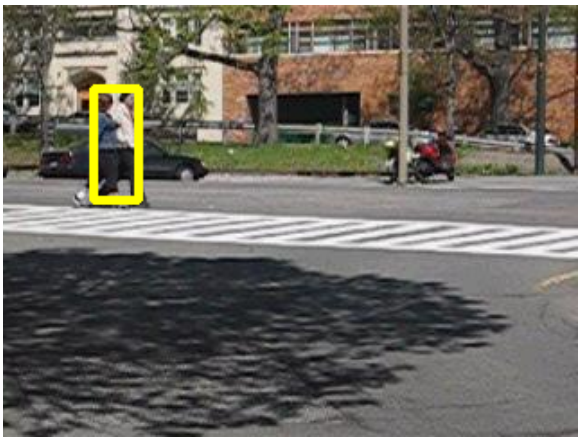


Figure A.75: Please draw a bounding box around all parts of the two people crossing the street together, except the back foot.

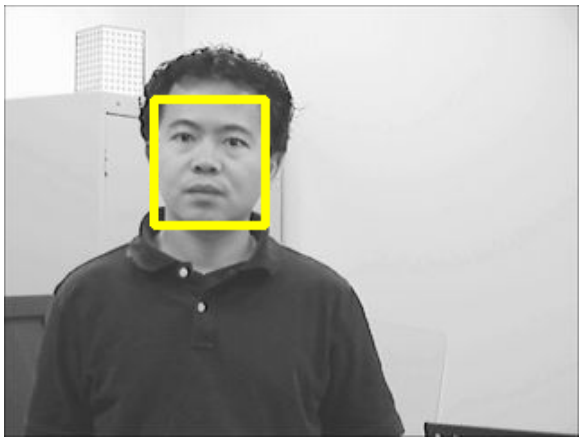


Figure A.76: Please draw a bounding box around the person's face.

Figure A.77: Please draw a bounding box around all parts of the person in pink pants.

Figure A.78: Please draw a bounding box around all parts of the person in light pants, except their arms.

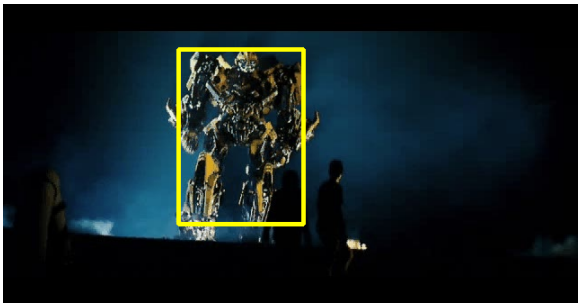


Figure A.79: Please draw a bounding box around the robot.

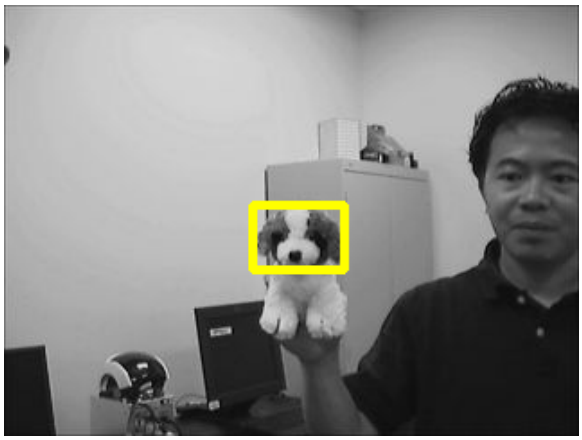


Figure A.80: Please draw a bounding box around the head of the dog toy.

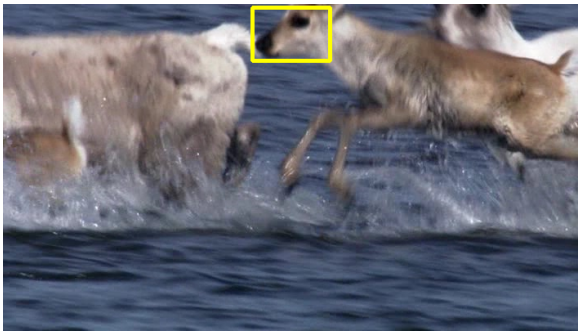


Figure A.81: Please draw a bounding box around the head of the deer, from the base of its ear to the tip of its nose.

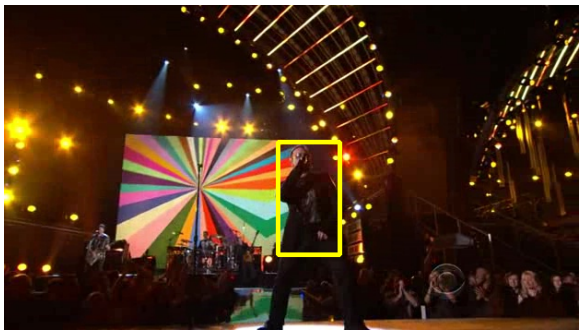


Figure A.82: Please draw a bounding box around the singer, from the top of their head to the end of their jacket.

Figure A.83: Please draw a bounding box around all parts of the singer in the white dress.

Figure A.84: Please draw a bounding box around the red vehicle.

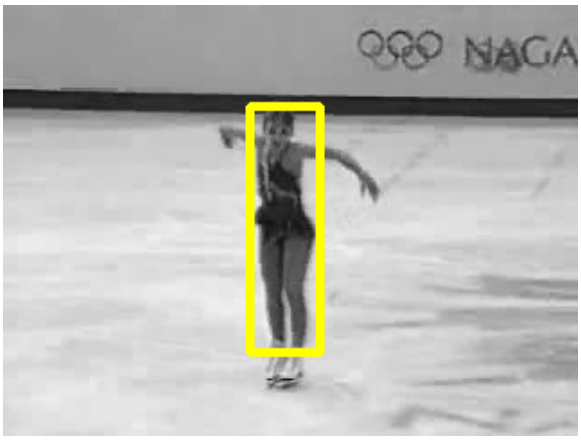


Figure A.85: Please draw a bounding box around all parts of the figure skater, except their arms and feet.

Figure A.86: Please draw a bounding box around all parts of the person, except the arm closer to the trash cans.

Figure A.87: Please draw a bounding box around the box of tea.

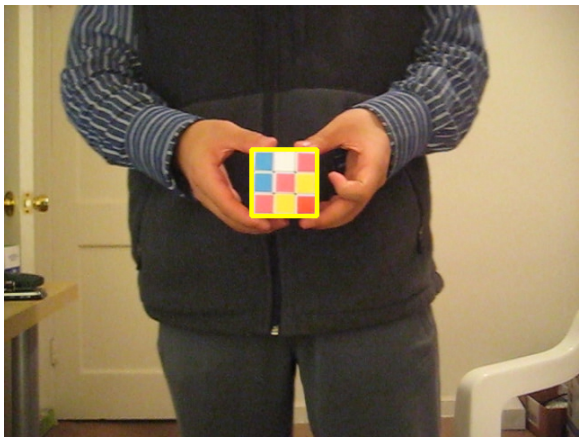


Figure A.88: Please draw a bounding box around the rubik's cube.

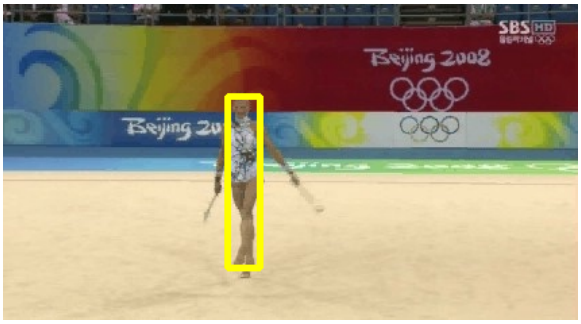


Figure A.89: Please draw a bounding box around all parts of the gymnast, except their arms.

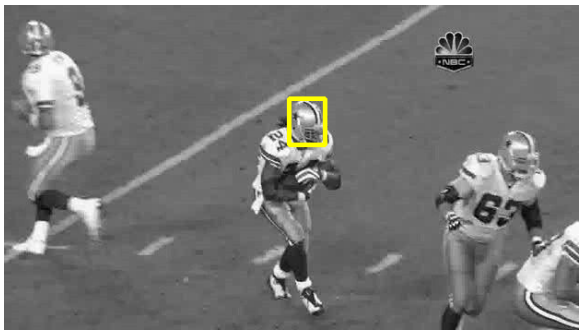


Figure A.90: Please draw a bounding box around the helmet of the player holding the ball.

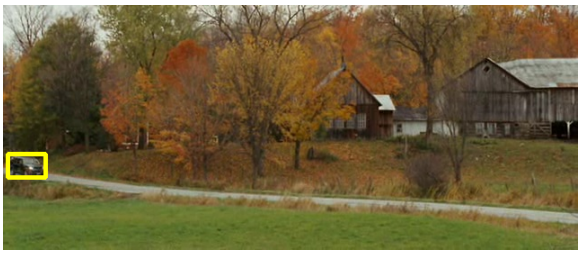


Figure A.91: Please draw a bounding box around the vehicle.

Figure A.92: Please draw a bounding box around the motorcycle and its rider.

Figure A.93: Please draw a bounding box around all parts of the pedestrian in blue pants.

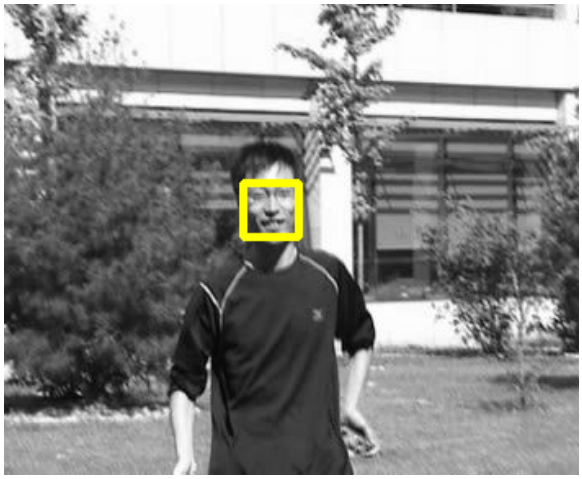


Figure A.94: Please draw a bounding box around the person's face.

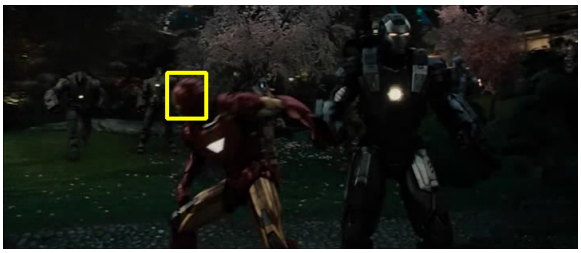


Figure A.95: Please draw a bounding box around the head of the red superhero.

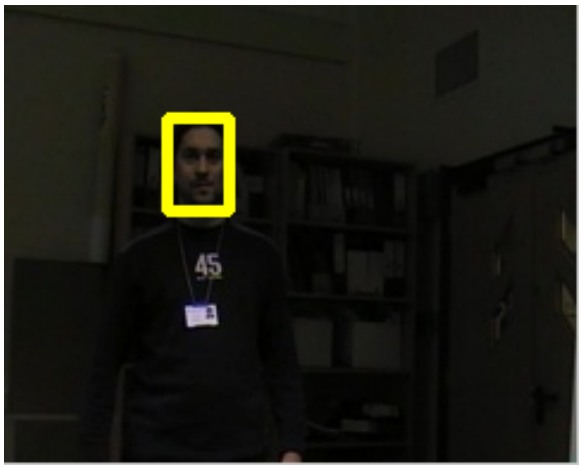


Figure A.96: Please draw a bounding box around the person's head.

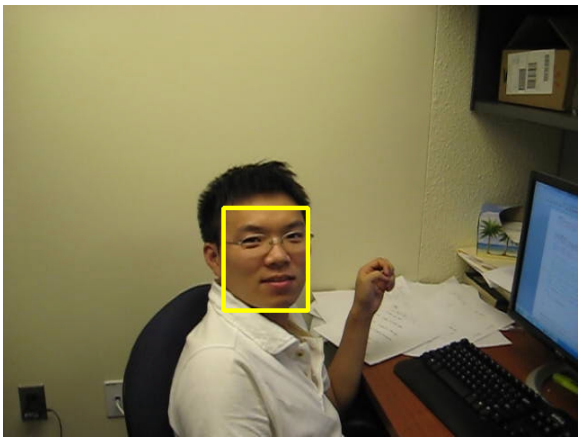


Figure A.97: Please draw a bounding box around the person's face.

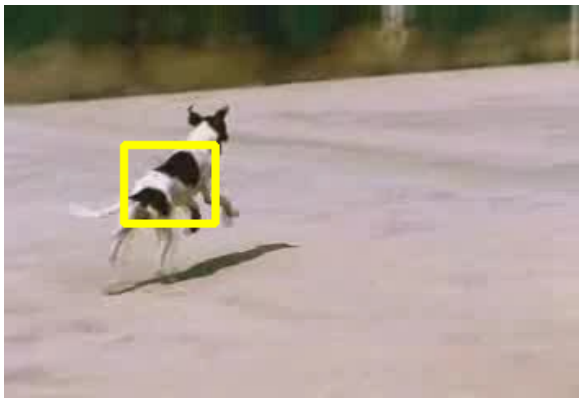


Figure A.98: Please draw a bounding box around the body (not head, legs, or tail) of the dog.

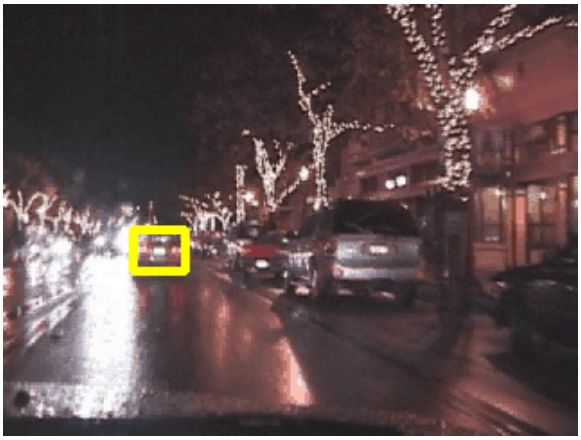


Figure A.99: Please draw a bounding box around the car with its brake lights on.

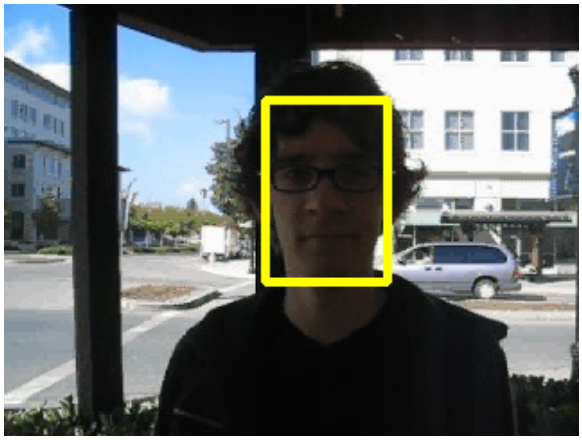


Figure A.100: Please draw a bounding box around the person's head.

APPENDIX B

DR-Error Plots Conditioned on DDC

While the analysis of Chapter 6 contained the performance for every aggregation function when the deferral rate or depth constraint were marginalized out, further insight can gained by inspecting the relationship between deferral rate and error at every deferral depth constraint independently. We provide that information here.

B.1 Video Object Tracking

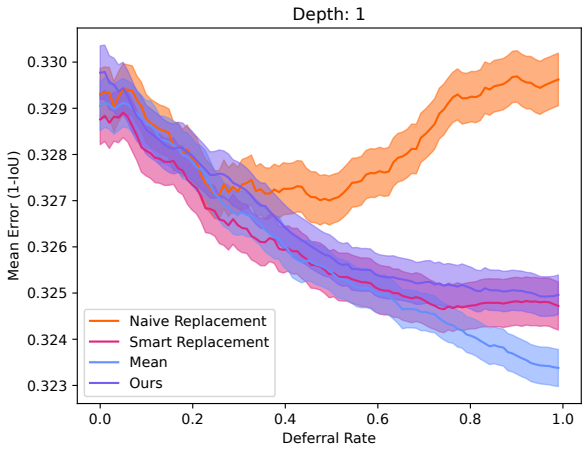


Figure B.1: Deferral rate against error for the VOT application and DDC=1.

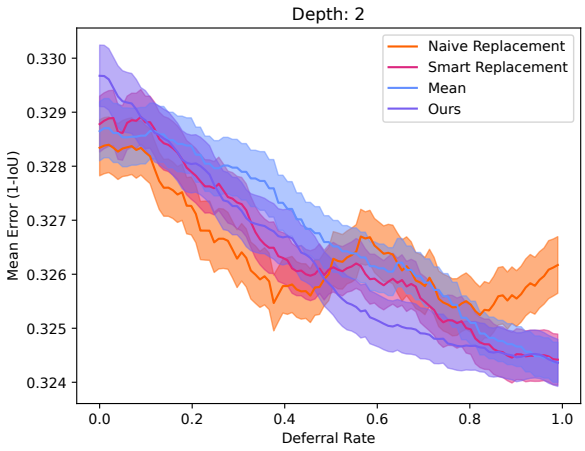


Figure B.2: Deferral Rate against Error for the VOT application and DDC=2.

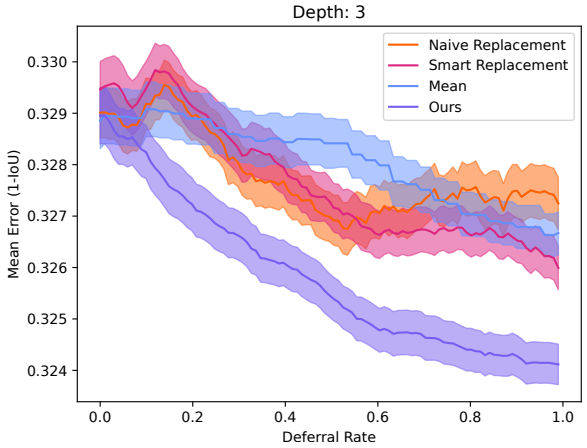


Figure B.3: Deferral rate against error for the VOT application and DDC=3.

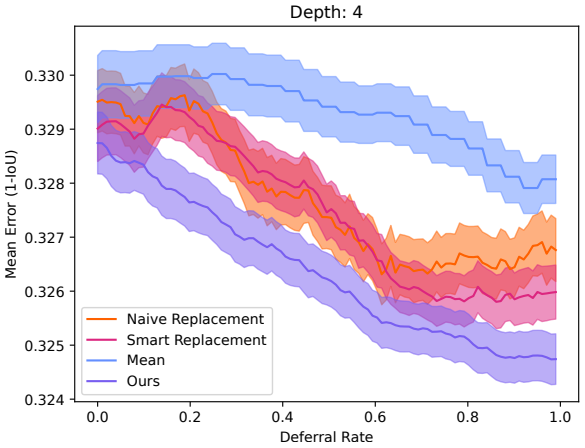


Figure B.4: Deferral rate against error for the VOT application and DDC=4.

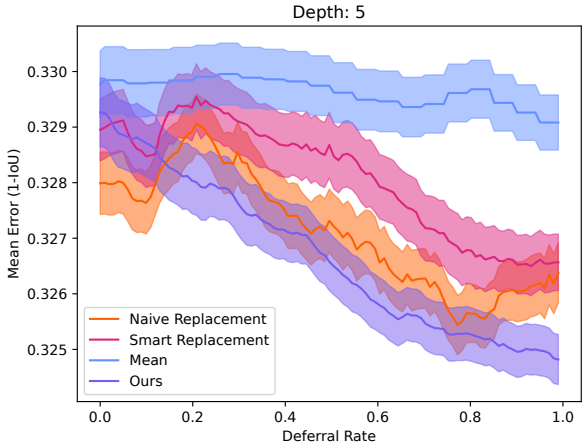


Figure B.5: Deferral rate against error for the VOT application and DDC=5.

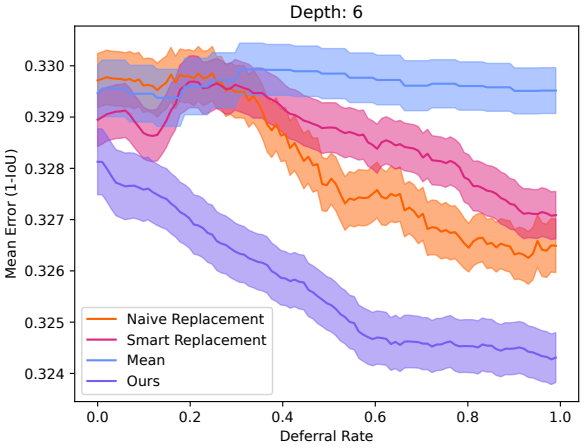


Figure B.6: Deferral rate against error for the VOT application and DDC=6.

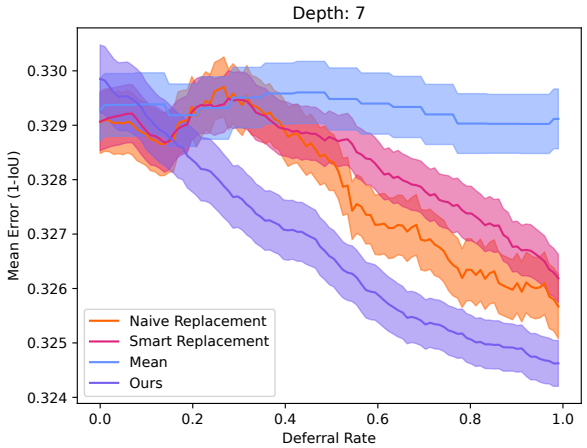


Figure B.7: Deferral rate against error for the VOT application and DDC=7.

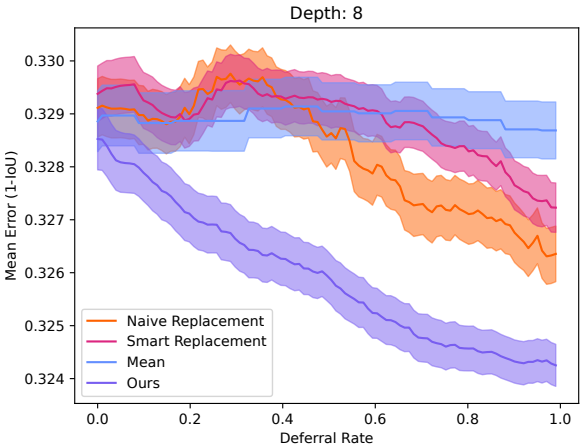


Figure B.8: Deferral rate against error for the VOT application and DDC=8.

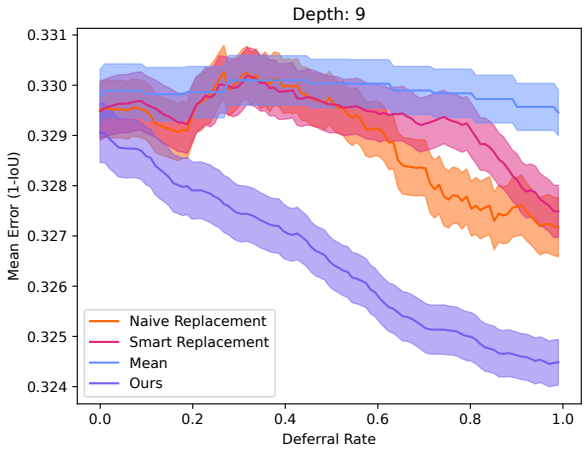


Figure B.9: Deferral rate against error for the VOT application and DDC=9.

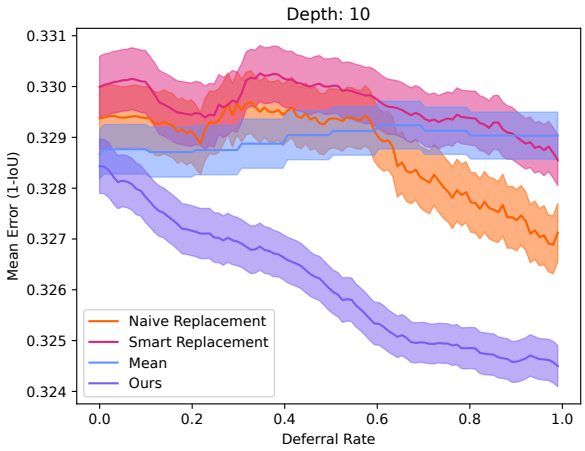


Figure B.10: Deferral rate against error for the VOT application and DDC=10.

B.2 Referring Expression Comprehension

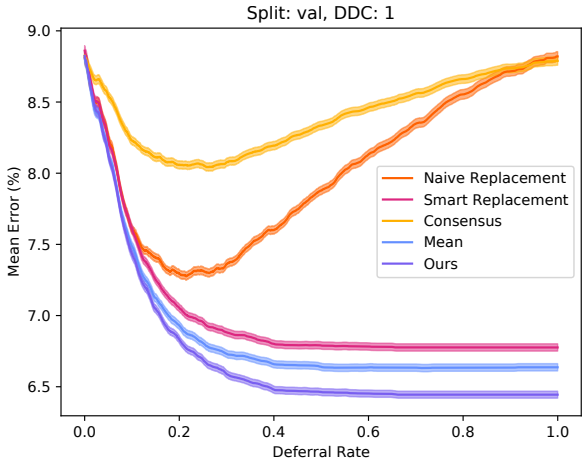


Figure B.11: Deferral rate against error for the val split of the RefExp application and DDC=1.

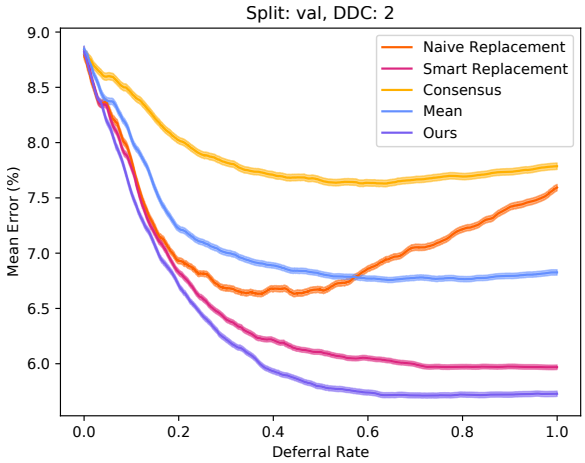


Figure B.12: Deferral rate against error for the val split of the RefExp application and DDC=2.

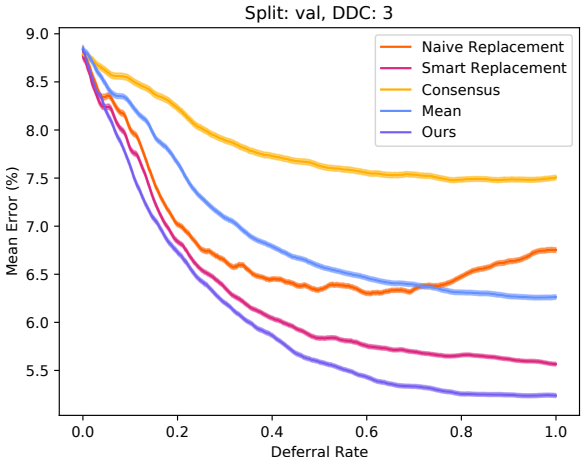


Figure B.13: Deferral rate against error for the val split of the RefExp application and DDC=3.

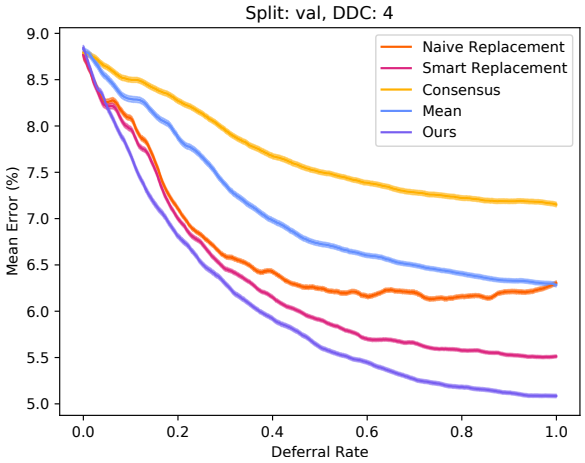


Figure B.14: Deferral rate against error for the val split of the RefExp application and DDC=4.

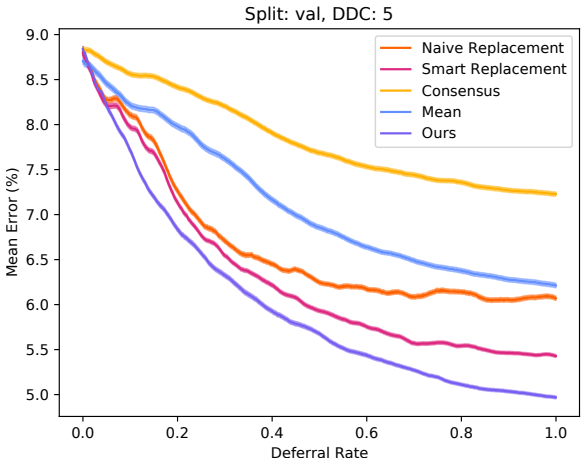


Figure B.15: Deferral rate against error for the val split of the RefExp application and DDC=5.

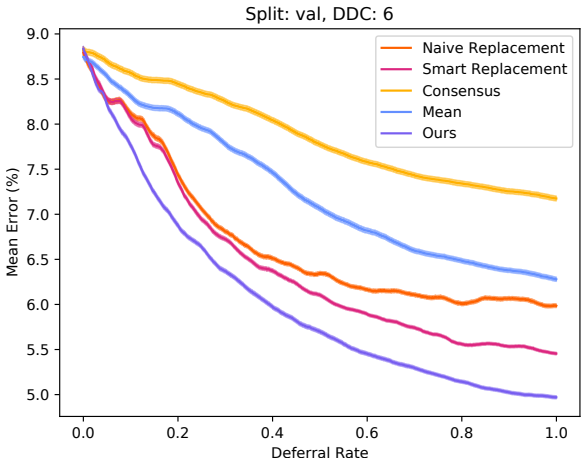


Figure B.16: Deferral rate against error for the val split of the RefExp application and DDC=6.

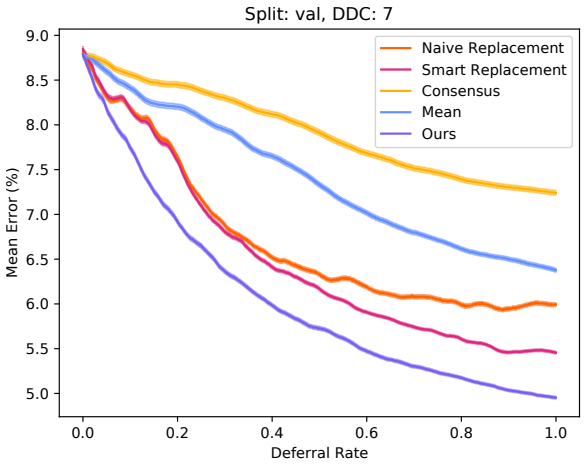


Figure B.17: Deferral rate against error for the val split of the RefExp application and DDC=7.

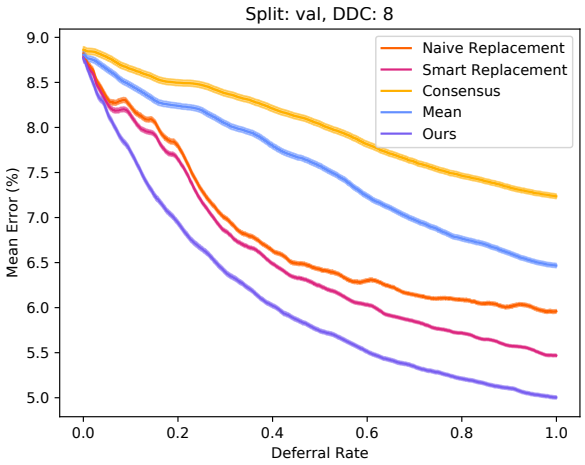


Figure B.18: Deferral rate against error for the val split of the RefExp application and DDC=8.

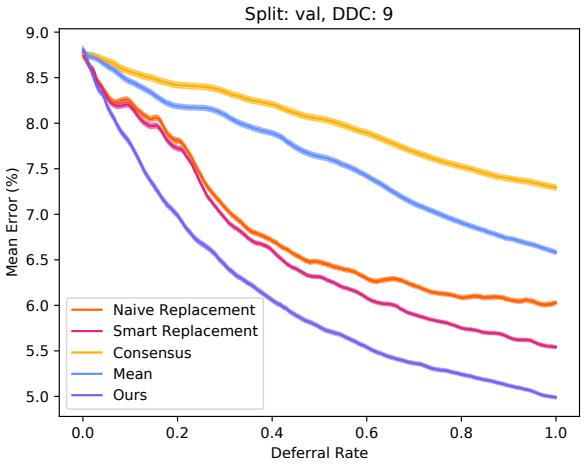


Figure B.19: Deferral rate against error for the val split of the RefExp application and DDC=9.

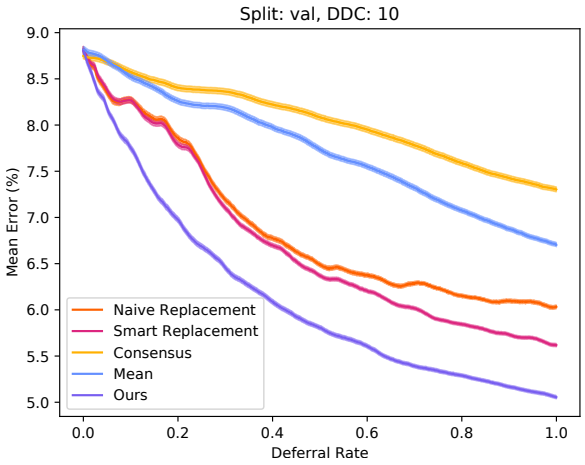


Figure B.20: Deferral rate against error for the val split of the RefExp application and DDC=10.

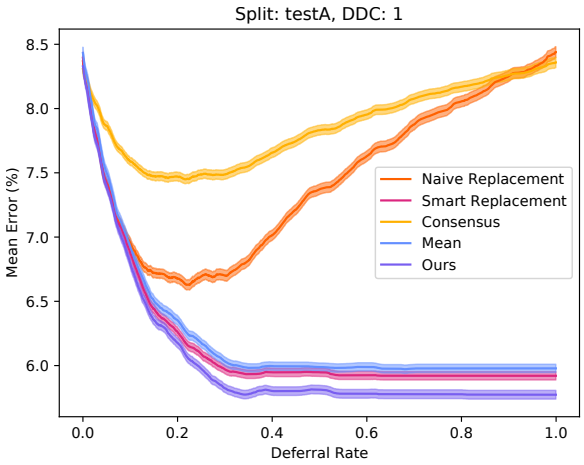


Figure B.21: Deferral rate against error for the testA split of the RefExp application and DDC=1.

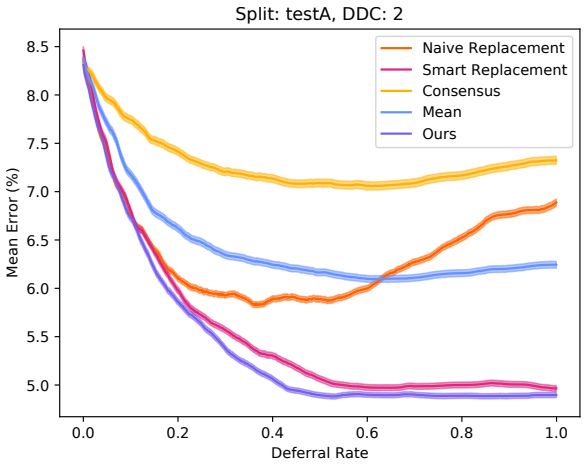


Figure B.22: Deferral rate against error for the testA split of the RefExp application and DDC=2.

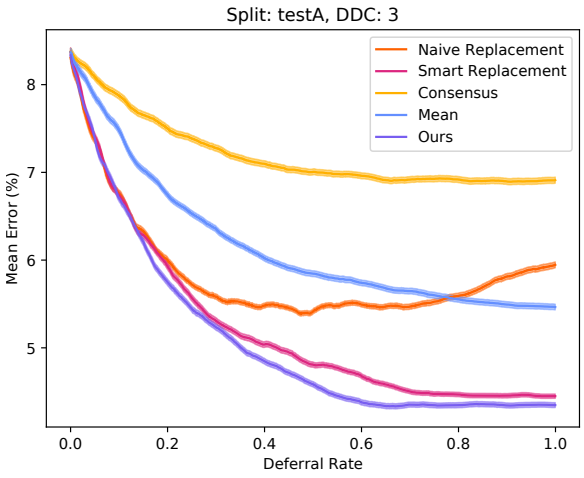


Figure B.23: Deferral rate against error for the testA split of the RefExp application and DDC=3.

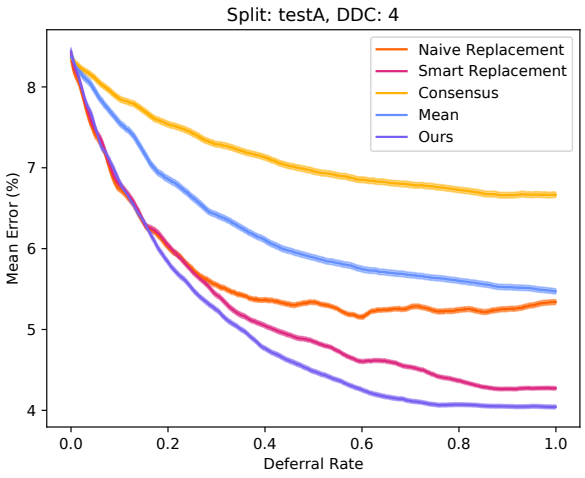


Figure B.24: Deferral rate against error for the testA split of the RefExp application and DDC=4.

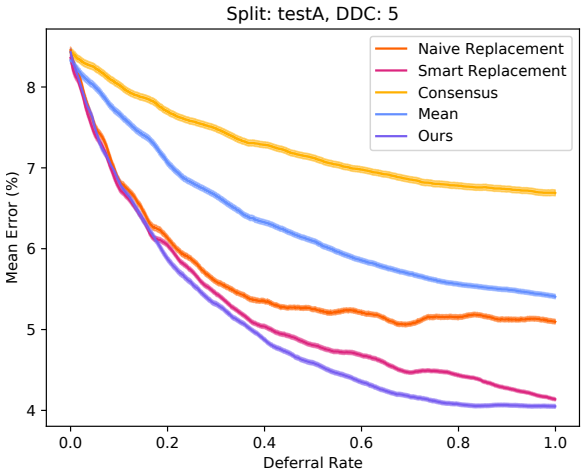


Figure B.25: Deferral rate against error for the testA split of the RefExp application and DDC=5.

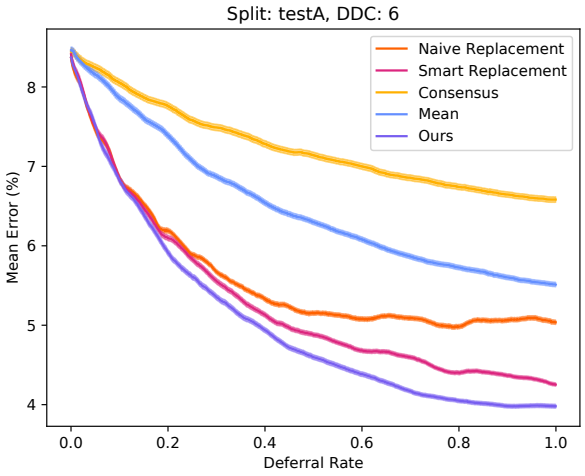


Figure B.26: Deferral rate against error for the testA split of the RefExp application and DDC=6.

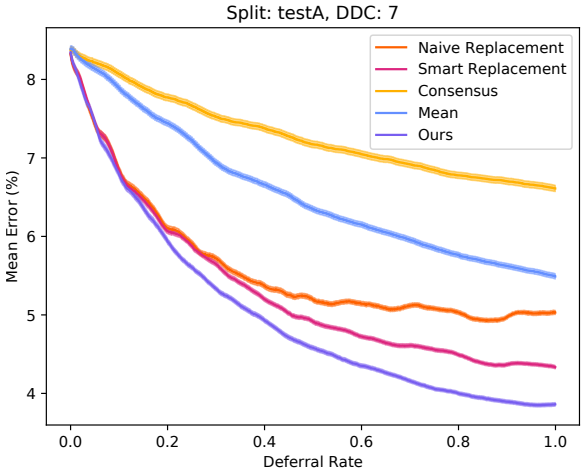


Figure B.27: Deferral rate against error for the testA split of the RefExp application and DDC=7.

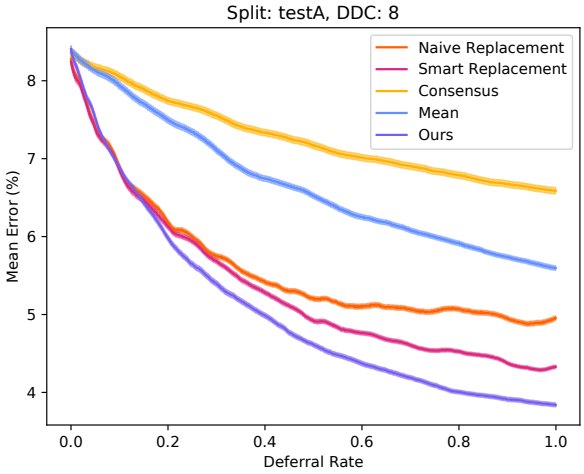


Figure B.28: Deferral rate against error for the testA split of the RefExp application and DDC=8.

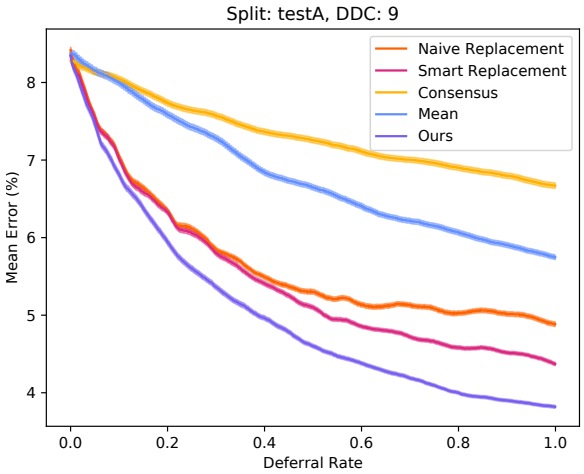


Figure B.29: Deferral rate against error for the testA split of the RefExp application and DDC=9.

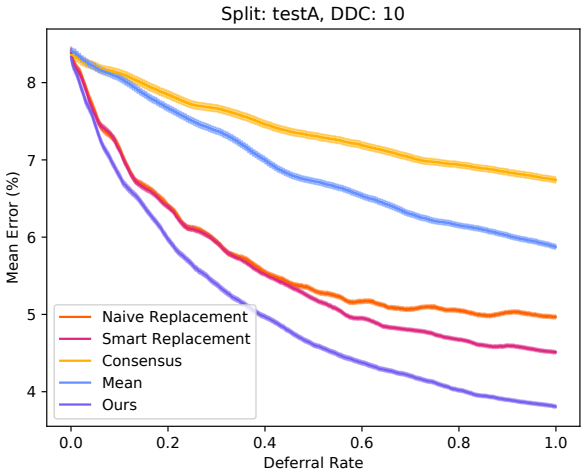


Figure B.30: Deferral rate against error for the testA split of the RefExp application and DDC=10.

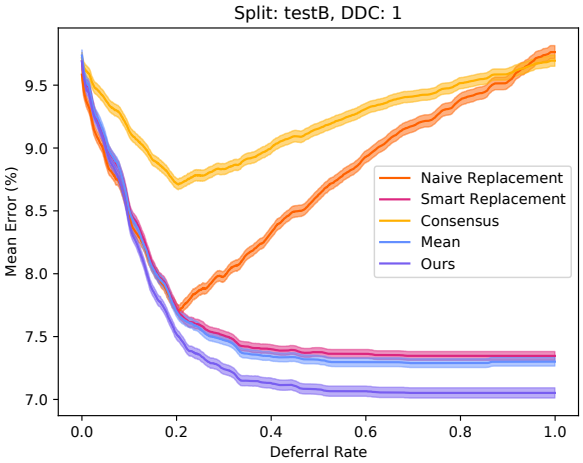


Figure B.31: Deferral rate against error for the testB split of the RefExp application and DDC=1.

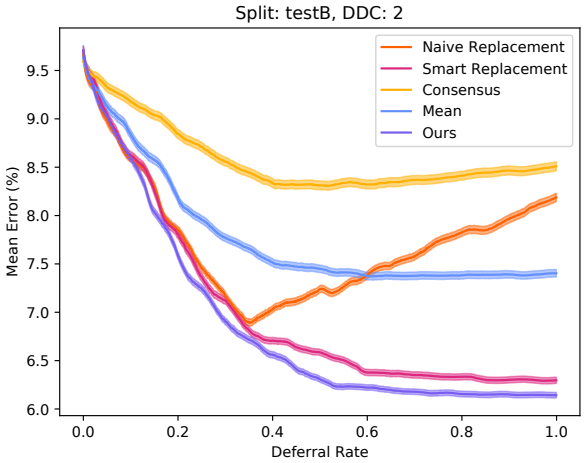


Figure B.32: Deferral rate against error for the testB split of the RefExp application and DDC=2.

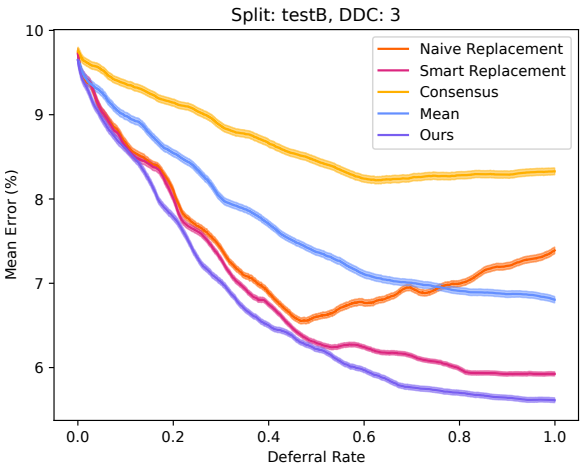


Figure B.33: Deferral rate against error for the testB split of the RefExp application and DDC=3.

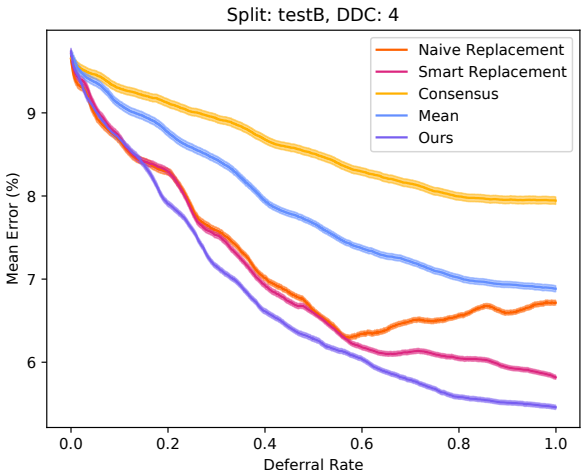


Figure B.34: Deferral rate against error for the testB split of the RefExp application and DDC=4.

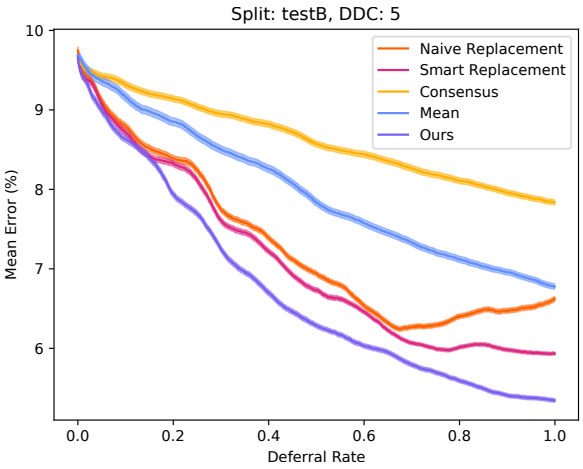


Figure B.35: Deferral rate against error for the testB split of the RefExp application and DDC=5.

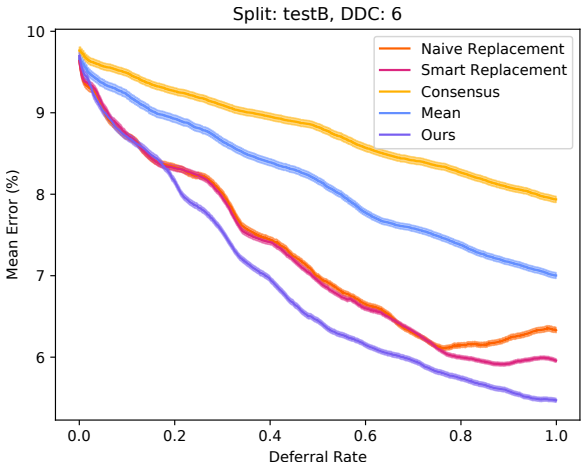


Figure B.36: Deferral rate against error for the testB split of the RefExp application and DDC=6.

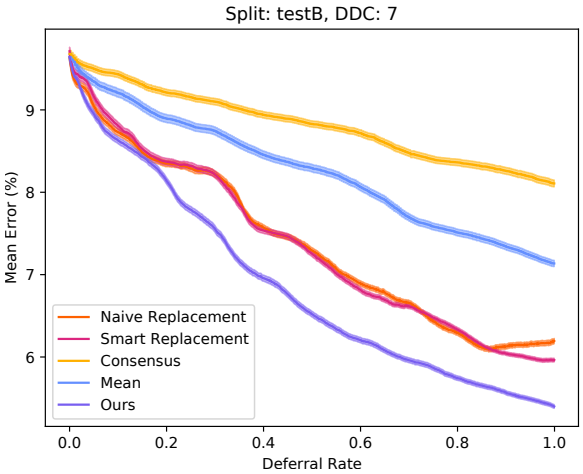


Figure B.37: Deferral rate against error for the testB split of the RefExp application and DDC=7.

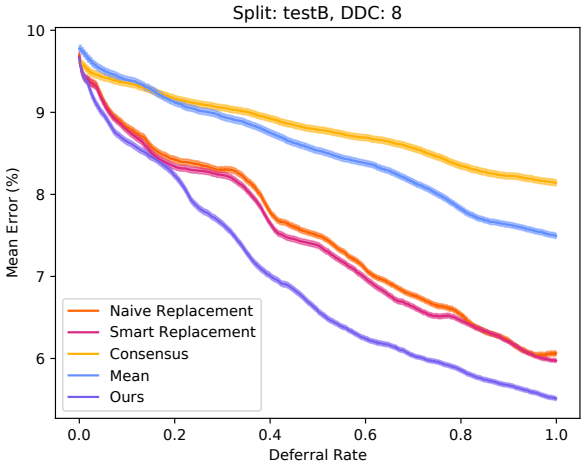


Figure B.38: Deferral rate against error for the testB split of the RefExp application and DDC=8.

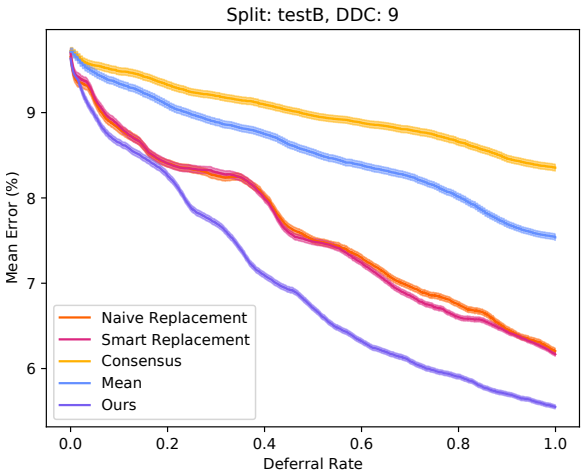


Figure B.39: Deferral rate against error for the testB split of the RefExp application and DDC=9.

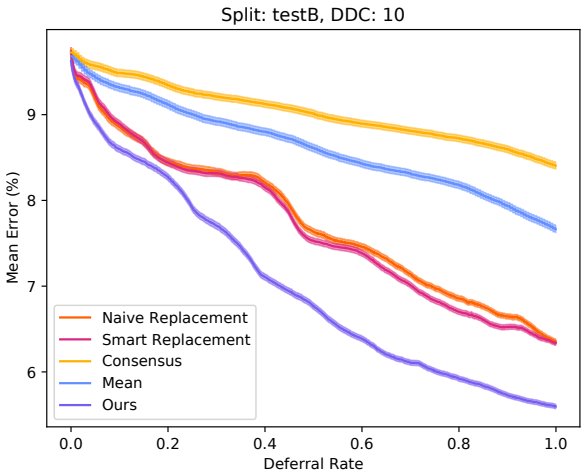


Figure B.40: Deferral rate against error for the testB split of the RefExp application and DDC=10.

BIBLIOGRAPHY

[1] Y.-C. Chen, L. Li, L. Yu, *et al.*, "UNITER: UNiversal Image-TExt Representation Learning," in *Proceedings of the 2020 European Conference on Computer Vision*, Virtual: Springer, 2020, pp. 104–120.

[2] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "ReferItGame: Referring to Objects in Photographs of Natural Scenes," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 787–798.

[3] Z. Senzer and N. Sarma, *Photobombs begone with Magic Eraser in Google Photos*, Blog, Oct. 2021. [Online]. Available: `https://blog.google/products/photos/magic-eraser/` (visited on 11/04/2022).

[4] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical Text-Conditional Image Generation with CLIP Latents*, arXiv:2204.06125 [cs], Apr. 2022. [Online]. Available: `http://arxiv.org/abs/2204.06125`.

[5] A. Pilipiszyn, *GPT-3 Powers the Next Generation of Apps*, Mar. 2021. [Online]. Available: `https://openai.com/blog/gpt-3-apps/` (visited on 11/04/2022).

[6] S. Antol, A. Agrawal, J. Lu, *et al.*, "VQA: Visual Question Answering," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, Santiago, Chile: IEEE Press, Dec. 2015, pp. 2425–2433.

[7] D. Gurari, Q. Li, A. J. Stangl, *et al.*, "VizWiz Grand Challenge: Answering Visual Questions from Blind People," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA: IEEE Press, Jun. 2018, pp. 3608–3617.

[8] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, "Generation and Comprehension of Unambiguous Object Descriptions," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 11–20.

[9] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of Human Support Robot as the research platform of a domestic mobile manipulator," *ROBOMECH Journal*, vol. 6, no. 1, p. 4, Dec. 2019.

[10] J. Famakinwa, *Report Sheds New Light on Looming Caregiving Crisis*, Jul. 2021. [Online]. Available: `https://homehealthcarenews.com/2021/07/report-sheds-new-light-on-looming-caregiving-crisis/` (visited on 12/23/2022).

[11] J. P. Bigham, C. Jayant, H. Ji, *et al.*, "VizWiz: Nearly real-time answers to visual questions," in *Proceedings of the 2010 Annual ACM Symposium on User Interface Software and Technology*, New York, New York, USA: ACM Press, 2010, pp. 333–342.

[12] A. Mahendru, V. Prabhu, A. Mohapatra, D. Batra, and S. Lee, "The Promise of Premise: Harnessing Question Premises in Visual Question Answering," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 926–935.

[13] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the 2017 International Conference on Machine Learning*, Sydney, New South Wales, Australia: PMLR, Jun. 2017, pp. 1321–1330.

[14] S. Banerjee, J. Thomason, and J. J. Corso, "The RobotSlang Benchmark: Dialog-guided Robot Localization and Navigation," in *Proceedings of the 2020 Conference on Robot Learning*, Virtual: PMLR, 2020, pp. 1384–1393.

[15] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: Common Objects in Context," in *Proceedings of the 2014 European Conference on Computer Vision*, Zurich, Switzerland: Springer, 2014, pp. 740–755.

[16] A. Ray, G. Christie, M. Bansal, D. Batra, and D. Parikh, "Question Relevance in VQA: Identifying Non-Visual And False-Premise Questions," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, USA: The Association for Computational Linguistics, 2016, pp. 919–924.

[17] N. Bhattacharya, Q. Li, and D. Gurari, "Why Does a Visual Question Have Different Answers?" In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea: IEEE Press, Oct. 2019, pp. 4270–4279.

[18] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme Clicking for Efficient Object Annotation," in *Proceedings of the 2017 IEEE/CVF International Conference on Computer Vision*, Venice, Italy: IEEE Press, Oct. 2017, pp. 4940–4949.

[19] J. Y. Song, J. J. Y. Chung, D. F. Fouhey, and W. S. Lasecki, "C-Reference: Improving 2D to 3D Object Pose Estimation Accuracy via Crowdsourced Joint Object Estimation," in *Procedings of the 2020 ACM Conference on Human-Computer Interaction*, Virtual: ACM Press, 2020, 051:1–051:28.

[20] M. S. Bernstein, G. Little, R. C. Miller, *et al.*, "Soylent: A word processor with a crowd inside," in *Proceedings of the 2010 ACM Symposium on User Interface Software and Technology*, New York, New York, USA: ACM Press, 2010, pp. 313–322.

[21] S. D. Jain and K. Grauman, "Click Carving: Segmenting Objects in Video with Point Clicks," in *Proceedings of the 2016 AAAI Conference on Human Computation and Crowdsourcing*, Austin, Texas, USA: AAAI Press, 2016, pp. 89–98.

[22] L. Yu, Z. Lin, X. Shen, *et al.*, "MAttNet: Modular Attention Network for Referring Expression Comprehension," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA: IEEE Press, Jun. 2018, pp. 1307–1315.

[23] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Proceedings of the 2020 European Conference on Computer Vision*, Virtual: Springer, 2020, pp. 213–229.

[24] A. Kamath, M. Singh, Y. LeCun, I. Misra, G. Synnaeve, and N. Carion, "MDETR – Modulated Detection for End-to-End Multi-Modal Understanding," in *Proceedings of the 2021 International Conference on Computer Vision*, Virtual: IEEE Press, 2021, pp. 1780–1790.

[25] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering," in *Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA: IEEE Press, 2017, p. 10.

[26] R. Cadene and C. Dancette, "RUBi: Reducing Unimodal Biases for Visual Question Answering," in *Proceedings of the 2019 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2019, pp. 839–850.

[27] S. Reddy, M. Mathew, L. Gomez, M. Rusinol, D. Karatzas., and C. V. Jawahar, "RoadText-1K: Text Detection & Recognition Dataset for Driving Videos," in *Proceedings of the 2020 IEEE Conference on Robotics and Automation*, Virtual: IEEE Press, 2020.

[28] Z. Raisi, M. A. Naiel, G. Younes, S. Wardell, and J. S. Zelek, "Transformer-based Text Detection in the Wild," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, Nashville, Tennessee, USA: IEEE Press, Jun. 2021, pp. 3156–3165.

[29] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018.

[30] J. Byrd and Z. C. Lipton, "What is the Effect of Importance Weighting in Deep Learning?" In *Proceedings of the 2019 International Conference on Machine Learning*, Long Beach, California, USA: PMLR, 2019, pp. 872–881.

[31] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss," in *Proceedings of the 2019 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, Oct. 2019, pp. 1565–1576.

[32] X. Zhu, D. Anguelov, and D. Ramanan, "Capturing Long-Tail Distributions of Object Subcategories," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, USA: IEEE Press, Jun. 2014, pp. 915–922.

[33] Q. Dong, S. Gong, and X. Zhu, "Class Rectification Hard Mining for Imbalanced Deep Learning," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, Venice, Italy: IEEE Press, 2017, pp. 1869–1878.

[34] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, "Intriguing properties of neural networks," in *Proceedings of the 2014 International Conference on Learning Representations*, Banff, Alberta, Canada: OpenReview, Feb. 2014, p. 10.

[35] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proceedings of the 2015 International Conference on Learning Representations*, San Diego, California, USA: OpenReview, 2015, p. 11.

[36] H. Hosseini, Y. Chen, S. Kannan, B. Zhang, and R. Poovendran, *Blocking Transferability of Adversarial Examples in Black-Box Learning Systems*, CS, arXiv: 1703.04318, Mar. 2017.

[37] J. Lu, T. Issaranon, and D. Forsyth, "SafetyNet: Detecting and Rejecting Adversarial Examples Robustly," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, Venice, Italy: IEEE Press, 2017, pp. 446–454.

[38] X. Li and F. Li, "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, Venice, Italy: IEEE Press, Oct. 2017, pp. 5775–5783.

[39] W. Xu, D. Evans, and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," in *Proceedings of the 2018 Network and Distributed System Security Symposium*, San Diego, California, USA: The Internet Society, 2018, p. 15.

[40] B. Wang, Y. Yao, S. Shan, *et al.*, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy*, San Francisco, California, USA: IEEE Press, 2019, pp. 707–723.

[41] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy*, San Jose, California, USA: IEEE Press, May 2016, pp. 582–597.

[42] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, Jan. 1970.

[43] H. Mozannar and D. Sontag, "Consistent Estimators for Learning to Defer to an Expert," in *Proceedings of the 2020 International Conference on Machine Learning*, Virtual: PMLR, 2020, pp. 7076–7087.

[44] E. Bondi, R. Koster, H. Sheahan, *et al.*, "Role of Human-AI Interaction in Selective Prediction," in *Proceedings of the 2022 AAAI Conference on Artificial Intelligence*, Virtual: AAAI Press, May 2022, pp. 5286–5294.

[45] M. Raghu, K. Blumer, R. Sayres, *et al.*, "Direct Uncertainty Prediction for Medical Second Opinions," in *Proceedings of the 2019 International Conference on Machine Learning*, Long Beach, California, USA: ACM Press, 2019, pp. 5281–5290.

[46] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific Reports*, vol. 7, no. 1, pp. 1–14, Dec. 2017.

[47] E. Luger and A. Sellen, ""Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose, California, USA: ACM Press, May 2016, pp. 5286–5297.

[48] S. J. Lemmer and J. J. Corso, "Evaluating and Improving Interactions with Hazy Oracles," in *Proceedings of the 2023 AAAI Conference on Artificial Intelligence*, Washington, District of Columbia, USA: AAAI Press, 2023, p. 9.

[49] Y. Geifman and R. El-Yaniv, "SelectiveNet: A Deep Neural Network with an Integrated Reject Option," in *Proceedings of the 2019 International Conference on Machine Learning*, Long Beach, California, USA: ACM Press, 2019, pp. 2151–2159.

[50] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks," in *Proceedings of the 2019 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2019, pp. 13–23.

[51] X. Xu, J. Gong, C. Brum, *et al.*, "Enabling hand gesture customization on wrist-worn devices," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, Lousiana, USA: ACM Press, Mar. 2022, 496:1–496:19.

[52] C. Mayer, M. Danelljan, G. Bhat, *et al.*, "Transforming Model Prediction for Tracking," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, USA: IEEE Press, 2022, pp. 8731–8740.

[53] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-Shot Video Object Segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA: IEEE Press, 2017, pp. 5320–5329.

[54] M. Kristan, J. Matas, A. Leonardis, *et al.*, "A Novel Performance Evaluation Methodology for Single-Target Trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

[55] B. Shneiderman and P. Maes, "Direct manipulation vs. interface agents," *Interactions*, vol. 4, no. 6, pp. 42–61, Nov. 1997.

[56] C. J. Cai, E. Reif, N. Hegde, *et al.*, "Human-Centered Tools for Coping with Imperfect Algorithms During Medical Decision-Making," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, UK: ACM Press, 2019, p. 14.

[57] J. R. R. Uijlings, M. Andriluka, and V. Ferrari, "Panoptic Image Annotation with a Collaborative Assistant," in *Proceedings of the 2020 ACM International Conference on Multimedia*, Virtual: ACM Press, 2020, pp. 3302–3310.

[58] Y. Geifman and R. El-Yaniv, "Selective Classification for Deep Neural Networks," in *Proceedings of the 2017 Conference on Advances in Neural Information Processing Systems*, Long Beach, California, USA: Curran Associates, 2017, pp. 4878–4887.

[59] J. M. Beer, C.-A. Smarr, T. L. Chen, *et al.*, "The domesticated robot: Design guidelines for assisting older adults to age in place," in *Proceedings of the 2012 Annual ACM/IEEE International Conference on Human-Robot Interaction*, Boston, Massachusetts, USA: ACM Press, 2012, pp. 335–342.

[60] S. Sano, N. Kaji, and M. Sassano, "Predicting Causes of Reformulation in Intelligent Assistants," in *Proceedings of the 2017 Annual SIGdial Meeting on Discourse and Dialogue*, Saarbrücken, Germany: Association for Computational Linguistics, 2017, pp. 299–309.

[61] O. Mees and W. Burgard, "Composing Pick-and-Place Tasks By Grounding Language," in *Proceedings of the 2020 International Symposium on Experimental Robotics*, La Valletta, Malta: Springer, 2020, pp. 491–501.

[62] K. Uehara, N. Duan, and T. Harada, "Learning To Ask Informative Sub-Questions for Visual Question Answering," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, New Orleans, Louisiana, USA: IEEE Press, 2022, pp. 4681–4690.

[63] R. Szeto and J. J. Corso, "Click Here: Human-Localized Keypoints as Guidance for Viewpoint Estimation," in *Proceedings of the 2017 IEEE/CVF International Conference on Computer Vision*, Venice, Italy: IEEE Press, Oct. 2017, pp. 1604–1613.

[64] M. Y. Yildirim, M. Ozer, and H. Davulcu, *Leveraging Uncertainty in Deep Learning for Selective Classification*, CS, MATH, STAT, May 2019. (visited on 05/30/2019).

[65] C. Cortes, G. DeSalvo, and M. Mohri, "Boosting with Abstention," in *Proceedings of the 2016 Conference on Advances in Neural Information Processing Systems*, Barcelona, Spain: Curran Associates, 2016, pp. 1660–1668.

[66] A. Hassan Awadallah, R. Gurunath Kulkarni, U. Ozertem, and R. Jones, "Characterizing and Predicting Voice Query Reformulation," in *Proceedings of the 2015 ACM International on Conference on Information and Knowledge Management*, Melbourne, Victoria, Australia: ACM Press, Oct. 2015, pp. 543–552.

[67] P. Sharma, B. Sundaralingam, V. Blukis, *et al.*, "Correcting Robot Plans with Natural Language Feedback," in *Proceedings of the 2022 Conference on Robotics: Science and Systems*, New York, New York, USA: MIT Press, 2022, pp. 1–12.

[68] J. Hatori, Y. Kikuchi, S. Kobayashi, *et al.*, "Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, Brisbane, Queensland, Australia: IEEE Press, 2018, pp. 3774–3781.

[69] J. Y. Song, S. J. Lemmer, M. X. Liu, *et al.*, "Popup: Reconstructing 3D video using particle filtering to aggregate crowd responses," in *Proceedings of the 2019 International Conference on Intelligent User Interfaces*, Marina del Ray, California, USA: ACM Press, 2019, pp. 558–569.

[70] J. Y. Song, R. Fok, A. Lundgard, F. Yang, J. Kim, and W. S. Lasecki, "Two Tools are Better Than One: Tool Diversity as a Means of Improving Aggregate Crowd Performance," in *Proceedings of the 2018 Conference on Intelligent User Interfaces*, Tokyo, Japan: ACM Press, 2018, pp. 559–570.

[71] D. P. Kingma and P. Dhariwal, "Glow: Generative Flow with Invertible 1×1 Convolutions," in *Proceedings of the 2018 Conference on Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, 2018, pp. 10 236–10 245.

[72] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative Adversarial Nets," in *Proceedings of the 2014 Conference on Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada: Curran Associates, Jun. 2014, pp. 2672–2680.

[73] L. Gatys, A. S. Ecker, and M. Bethge, "Texture Synthesis Using Convolutional Neural Networks," in *Proceedings of the 2015 Conference on Neural Information Processing Systems*, Montreal, Quebec, Canada: Curran Associates, 2015, pp. 262–270.

[74] R. Szeto, M. El-Khamy, J. Lee, and J. J. Corso, "HyperCon: Image-To-Video Model Transfer for Video-To-Video Translation Tasks," in *Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, Hawaii, USA: IEEE Press, Jan. 2021, pp. 3079–3088.

[75] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, Venice, Italy: IEEE Press, Oct. 2017, pp. 2242–2251.

[76] R. Szeto and J. J. Corso, "The DEVIL is in the Details: A Diagnostic Evaluation Benchmark for Video Inpainting," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, USA: IEEE Press, 2022, pp. 21 022–21 031.

[77] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention Is All You Need," in *Proceedings of the 2017 Conference on Advances in Neural Information Processing Systems*, Long Beach, California, USA: Curran Associates, 2017, pp. 5998–6008.

[78] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language Models are Few-Shot Learners," in *Proceedings of the 2020 Conference on Advances in Neural Information Processing Systems*, Virtual: Curran Associates, 2020, pp. 1877–1901.

[79] R. Daws, *Medical chatbot using OpenAI's GPT-3 told a fake patient to kill themselves*, Oct. 2020. [Online]. Available: `https://www.artificialintelligence-news.com/2020/10/28/medical-chatbot-openai-gpt3-patient-kill-themselves/` (visited on 12/29/2022).

[80] K. Hao, *OpenAI has released the largest version yet of its fake-news-spewing AI*, Aug. 2019. [Online]. Available: https://www.technologyreview.com/2019/08/29/133218/openai-released-its-fake-news-ai-gpt-2/ (visited on 12/29/2022).

[81] OpenAI, *Powering Next Generation Applications with OpenAI Codex*, May 2022. [Online]. Available: https://openai.com/blog/codex-apps/ (visited on 11/04/2022).

[82] J. Liang, W. Huang, F. Xia, *et al.*, *Code as Policies: Language Model Programs for Embodied Control*, Sep. 2022. (visited on 11/08/2022).

[83] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission," in *Proceedings of the 2015 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, New South Wales, Australia: ACM Press, 2015, pp. 1721–1730.

[84] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," in *Proceedings of the 2022 International Conference on Machine Learning*, Baltimore, Maryland, USA: PMLR, 2022, pp. 12 888–12 900.

[85] J. Lee, J. Herskovitz, Y.-H. Peng, and A. Guo, "ImageExplorer: Multi-Layered Touch Exploration to Encourage Skepticism Towards Imperfect AI-Generated Image Captions," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, Louisiana, USA: ACM Press, Apr. 2022, 462:1–462:15.

[86] C. Mucchiani, P. Cacchione, M. Johnson, R. Mead, and M. Yim, "Deployment of a Socially Assistive Robot for Assessment of COVID-19 Symptoms and Exposure at an Elder Care Setting," in *Proceedings of the 2021 IEEE International Conference on Robot & Human Interactive Communication*, Virtual: IEEE Press, Aug. 2021, pp. 1189–1195.

[87] J.-W. Kim, Y.-L. Choi, S.-H. Jeong, and J. Han, "A Care Robot with Ethical Sensing System for Older Adults at Home," *Sensors*, vol. 22, no. 19, p. 7515, Oct. 2022.

[88] E. Broadbent, C. Jayawardena, N. Kerse, R. Q. Stafford, and B. A. MacDonald, "Human-Robot Interaction Research to Improve Quality of Life in Elder Care – An Approach and Issues," in *Proceedings of Workshops at the 2011 AAAI Conference on Artificial Intelligence*, San Francisco, California, USA: AAAI Press, 2011, p. 7.

[89] S. R. Gouravajhala, J. Yim, K. Desingh, Y. Huang, O. C. Jenkins, and W. S. Lasecki, "EURECA: Enhanced Understanding of Real Environments via Crowd Assistance," in *Proceedings of the 2018 AAAI Conference on Human Computation and Crowdsourcing*, Zurich, Switzerland: AAAI Press, 2018, pp. 31–40.

[90] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham, "Real-time crowd control of existing interfaces," in *Proceedings of the 2011 annual ACM symposium on User Interface Software and Technology*, Santa Barbara, California, USA: ACM Press, 2011, pp. 23–32.

[91] M. Shridhar and D. Hsu, "Interactive Visual Grounding of Referring Expressions for Human-Robot Interaction," in *Proceedings of Robotics: Science and Systems 2018*, Pittsburgh, Pennsylvania, United States: MIT Press, 2018, pp. 1–9.

[92] D. Nyga, S. Roy, R. Paul, *et al.*, "Grounding Robot Plans from Natural Language Instructions with Incomplete World Knowledge," in *Proceedings of the 2018 Conference on Robot Learning*, Zurich, Switzerland: PMLR, 2018, pp. 714–723.

[93] B. Huang, D. Bayazit, D. Ullman, N. Gopalan, and S. Tellex, "Flight, Camera, Action! Using Natural Language and Mixed Reality to Control a Drone," in *Proceedings of the 2019 International Conference on Robotics and Automation*, Montreal, Quebec, Canada: IEEE Press, May 2019, pp. 6949–6956.

[94] K. Nguyen and H. Daumé III, "Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning," in *Proceedings of the Visually Grounded Interaction and Language (ViGIL) NeurIPS 2019 Workshop*, Vancouver, British Columbia, Canada: Curran Associates, 2019, p. 18.

[95] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "TOUCHDOWN: Natural Language Navigation and Spatial Reasoning in Visual Street Environments," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE Press, Jun. 2019, pp. 12 530–12 539.

[96] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-Dialog Navigation," in *Proceedings of the 2020 Conference on Robot Learning*, Virtual: PMLR, 2020, pp. 394–406.

[97] S. Branson, C. Wah, F. Schroff, *et al.*, "Visual Recognition with Humans in the Loop," in *Proceedings of the 2010 European Conference on Computer Vision*, Heraklion, Crete, Greece: Springer, 2010, pp. 438–451.

[98] H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori, "Learning Structured Inference Neural Networks with Label Relations," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 2960–2968.

[99] O. Russakovsky, L.-J. Li, and F.-F. Li, "Best of both worlds: Human-machine collaboration for object annotation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, Massachusetts, USA: IEEE Press, Jun. 2015, pp. 2121–2131.

[100] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 724–732.

[101] B. Gromov, G. Abbate, L. M. Gambardella, and A. Giusti, "Proximity Human-Robot Interaction Using Pointing Gestures and a Wrist-mounted IMU," in *Proceedings of the 2019 International Conference on Robotics and Automation*, Montreal, Quebec, Canada: IEEE Press, May 2019, pp. 8084–8091.

[102] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, USA: Association for Computational Linguistics, 2016, pp. 2383–2392.

[103] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding Stories in Movies Through Question-Answering," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, 2016, pp. 4631–4640.

[104] M. Koperski, T. Konopczynski, R. Nowak, P. Semberecki, and T. Trzcinski, "Plugin Networks for Inference under Partial Evidence," in *Proceedings of The 2020 IEEE Winter Conference on Applications of Computer Vision*, Snowmass Village, Colorado, USA: IEEE Press, 2020, pp. 2883–2891.

[105] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision*, Steamboat Springs, Colorado, USA: IEEE Press, Mar. 2014, pp. 75–82.

[106] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," *Computer Vision, Graphics, and Image Processing*, vol. 47, no. 1, pp. 33–44, Jul. 1989.

[107] C.-P. Lu, G. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, Jun. 2000.

[108] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 1999 IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, pp. 1150–1157.

[109] X. Zhou, A. Karpur, L. Luo, and Q. Huang, "StarMap for Category-Agnostic Keypoint and Viewpoint Estimation," in *Proceedings of the 2018 European Conference on Computer Vision*, Munich, Germany: Springer, 2018, pp. 328–345.

[110] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[111] S. Fidler, S. Dickinson, and R. Urtasun, "3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model," in *Proceedings of the 2012 Conference on Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA: Curran Associates, 2012, pp. 620–628.

[112] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3D geometry to deformable part models," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA: IEEE Press, Jun. 2012, pp. 3362–3369.

[113] N. D. Reddy, M. Vo, and S. G. Narasimhan, "CarFusion: Combining Point Tracking and Part Detection for Dynamic 3D Reconstruction of Vehicles," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA: IEEE Press, Jun. 2018, pp. 1906–1915.

[114] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of CAD Models," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, USA: IEEE Press, Jun. 2014, pp. 3762–3769.

[115] J. J. Lim, H. Pirsiavash, and A. Torralba, "Parsing IKEA Objects: Fine Pose Estimation," in *Proceedingngs of the 2013 IEEE International Conference on Computer Vision*, Sydney, New South Wales, Australia: IEEE Press, Dec. 2013, pp. 2992–2999.

[116] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, Massachusetts, USA: IEEE Press, Jun. 2015, pp. 1510–1519.

[117] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, Santiago, Chile: IEEE Press, Dec. 2015, pp. 2686–2694.

[118] A. X. Chang, T. Funkhouser, L. Guibas, *et al.*, *ShapeNet: An Information-Rich 3D Model Repository*, CS, Dec. 2015. (visited on 09/14/2020).

[119] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, Tech. Rep. 0, 2009, p. 60.

[120] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs," in *Proceedings of the 2011 Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, Colorado, USA, Jun. 2011, p. 2.

[121] K. Xiao, L. Engstrom, A. Ilyas, and A. Madry, "Noise or Signal: The Role of Image Backgrounds in Object Recognition," in *Proceedings of the 2021 International Conference on Learning Representations*, Virtual: OpenReview, Jun. 2020, p. 24.

[122] *Boston Terrier Vs French Bulldog – What's The Difference?* Apr. 2018. [Online]. Available: `https://www.rover.com/uk/blog/boston-terrier-vs-french-bulldog-whats-the-difference/` (visited on 10/28/2021).

[123] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-Based R-CNNs for Fine-Grained Category Detection," in *Proceedings of the 2014 European Conference on Computer Vision*, vol. 8689, Zurich, Switzerland: Springer, 2014, pp. 834–849.

[124] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-Stacked CNN for Fine-Grained Visual Categorization," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 1173–1182.

[125] H. Zhang, T. Xu, M. Elhoseiny, *et al.*, "SPDA-CNN: Unifying Semantic Part Detection and Abstraction for Fine-Grained Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 1143–1152.

[126] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition," in *Proceedings of the 2018 European Conference on Computer Vision*, Munich, Germany: Springer, 2018, pp. 834–850.

[127] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, Venice, Italy: IEEE Press, Oct. 2017, pp. 5219–5227.

[128] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition," in *Proceedings of the 2018 European Conference on Computer Vision*, Honolulu, Hawaii, USA: IEEE Press, 2018, pp. 595–610.

[129] S. Kong and C. Fowlkes, "Low-Rank Bilinear Pooling for Fine-Grained Classification," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu Hawaii USA: IEEE Press, Jul. 2017, pp. 7025–7034.

[130] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops*, Sydney, New South Wales, Australia: IEEE Press, Dec. 2013, pp. 554–561.

[131] A. Dubey, O. Gupta, R. Raskar, and N. Naik, "Maximum-Entropy Fine-Grained Classification," in *Proceedings of the 2018 Annual Conference on Neural Information Processing Systems*, Montreal, Quebec, Canada: Curran Associates, Sep. 2018, pp. 635–645.

[132] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell, and N. Naik, "Pairwise Confusion for Fine-Grained Visual Classification," in *Proceedings of the 2018 European Conference on Computer Vision*, Munich, Germany: Springer, 2018, pp. 71–88.

[133] P. Zhuang, Y. Wang, and Y. Qiao, "Learning Attentive Pairwise Interaction for Fine-Grained Classification," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*, Virtual: AAAI Press, Feb. 2020, pp. 13 130–13 137.

[134] J. Li, L. Zhu, Z. Huang, K. Lu, and J. Zhao, "I read, I saw, I tell: Texts Assisted Fine-Grained Visual Classification," in *Proceedings of the 2018 ACM international conference on Multimedia*, Seoul, South Korea: ACM Press, Oct. 2018, pp. 663–671.

[135] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, Massachusetts, USA: IEEE Press, Jun. 2015, pp. 842–850.

[136] T. Wang, K. Yamaguchi, and V. Ordonez, "Feedback-Prop: Convolutional Neural Network Inference Under Partial Evidence," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA: IEEE Press, Jun. 2018, pp. 898–907.

[137] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images," *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 573–579, Sep. 1976.

[138] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA: IEEE Press, Jun. 1994, pp. 593–600.

[139] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, USA: IEEE Press, Jun. 1998, pp. 232–237.

[140] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online Multiple Instance Learning," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, USA: IEEE Press, Jun. 2009, pp. 983–990.

[141] M. Godec, P. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1245–1256, Oct. 2013.

[142] J. Kwon and K. M. Lee, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive Basin Hopping Monte Carlo sampling," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, USA: IEEE Press, Jun. 2009, pp. 1208–1215.

[143] M. Kristan, S. Kovacic, A. Leonardis, and J. Pers, "A Two-Stage Dynamic Model for Visual Tracking," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 6, pp. 1505–1520, Dec. 2010.

[144] L. Č. Zajc, M. Kristan, and A. Leonardis, "Is my new tracker really better than yours?" In *Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision*, Steamboat Springs, Colorado, USA: IEEE Press, Mar. 2014, pp. 540–547.

[145] H. Wu, A. C. Sankaranarayanan, and R. Chellappa, "Online Empirical Evaluation of Tracking Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1443–1458, Aug. 2010.

[146] Y. Wu, J. Lim, and M.-H. Yang, "Online Object Tracking: A Benchmark," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, Oregon, USA: IEEE Press, Jun. 2013, pp. 2411–2418.

[147] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, May 2004.

[148] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[149] R. Collins, X. Zhou, and S. K. Teh, "An Open Source Tracking Testbed and Evaluation Web Site," in *Proceedings of the 2005 Joint IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Lausanne, Switzerland: IEEE Press, 2005, p. 8.

[150] R. B. Fisher, "The PETS04 Surveillance Ground-Truth Data Sets," in *Proceedings of the 2004 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, IEEE Press, 2004, p. 5.

[151] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, May 2008.

[152] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual Tracking: An Experimental Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[153] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning Discriminative Feature Representations Online for Robust Visual Tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, Apr. 2016.

[154] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 4293–4302.

[155] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual Tracking with Fully Convolutional Networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, Santiago, Chile: IEEE Press, Dec. 2015, pp. 3119–3127.

[156] D. Held, S. Thrun, and S. Savarese, "Learning to Track at 100 FPS with Deep Regression Networks," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, The Netherlands: Springer, 2016, pp. 749–765.

[157] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully Convolutional Siamese Networks for Object Tracking," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, The Netherlands: Springer, Jun. 2016, pp. 850–865.

[158] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA: IEEE Press, Jun. 2018, pp. 8971–8980.

[159] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the Power of Deep Tracking," in *Proceedings of the 2018 European Conference on Computer Vision*, Munich, Germany: Springer, 2018, pp. 493–509.

[160] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese Networks for Visual Object Tracking," in *Proceedings of the 2018 European Conference on Computer Vision*, Munich, Germany: IEEE Press, Aug. 2018, pp. 101–117.

[161] M. Kristan, A. Leonardis, J. Matas, *et al.*, "The Sixth Visual Object Tracking VOT2018 Challenge Results," in *Proceedings of the 2018 European Conference on Computer Vision Workshops*, Munich, Germany: Springer, 2019, pp. 3–53.

[162] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What Are You Talking About? Text-to-Image Coreference," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, USA: IEEE Press, Jun. 2014, pp. 3558–3565.

[163] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in *Proceedings of the 2012 European Conference on Computer Vision*, Florence, Italy: Springer, 2012, pp. 746–760.

[164] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Modeling Context Between Objects for Referring Expression Understanding," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, The Netherlands: Springer, Aug. 2016, pp. 792–807.

[165] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling Context in Referring Expressions," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, The Netherlands: Springer, Aug. 2016.

[166] R. Luo and G. Shakhnarovich, "Comprehension-Guided Referring Expressions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA: IEEE Press, Jul. 2017, pp. 3125–3134.

[167] L. Yu, H. Tan, M. Bansal, and T. L. Berg, "A Joint Speaker-Listener-Reinforcer Model for Referring Expressions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA: IEEE Press, Jul. 2017, pp. 3521–3529.

[168] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of Textual Phrases in Images by Reconstruction," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, The Netherlands: Springer, 2016, pp. 817–834.

[169] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang, "Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*, New York, New York, USA, Apr. 2020, pp. 11 336–11 344.

[170] X. Chen, X. Wang, S. Changpinyo, *et al.*, *PaLI: A Jointly-Scaled Multilingual Language-Image Model*, Sep. 2022.

[171] A. Das, S. Kottur, K. Gupta, *et al.*, "Visual Dialog," in *Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA: IEEE Press, 2017, pp. 326–335.

[172] M. E. Banani and J. J. Corso, "Adviser Networks: Learning What Question to Ask for Human-In-The-Loop Viewpoint Estimation," *arXiv:1802.01666 [cs]*, Oct. 2018.

[173] B. A. Griffin and J. J. Corso, "BubbleNets: Learning to Select the Guidance Frame in Video Object Segmentation by Deep Sorting Frames," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE Press, 2019, pp. 8914–8923.

[174] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," in *Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics*, Melbourne, Victoria, Australia: Association for Computational Linguistics, Jun. 2018, pp. 784–789.

[175] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In *Proceedings of the 2017 IEEE Conference on Advances in Neural Information Processing Systems*, Long Beach, California, USA: Curran Associates, 2017, pp. 5574–5584.

[176] A. S. Mozafari, H. S. Gomes, W. Leão, S. Janny, and C. Gagné, *Attended Temperature Scaling: A Practical Approach for Calibrating Deep Neural Networks*, Oct. 2018.

[177] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. S. Torr, and P. K. Dokania, "Calibrating Deep Neural Networks using Focal Loss," in *Proceedings of the 2020 Conference on Neural Information Processing Systems*, Virtual: Curran Associates, Feb. 2020, p. 23.

[178] M. Kull, M. Perello-Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration," in *Proceedings of the 2019 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2019, pp. 12 295–12 305.

[179] B. Lucena, *Spline-Based Probability Calibration*, Sep. 2018. (visited on 09/19/2019).

[180] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea: IEEE Press, Apr. 2019, pp. 502–511.

[181] F. Kraus and K. Dietmayer, "Uncertainty Estimation in One-Stage Object Detection," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference*, Auckland, New Zealand: IEEE Press, 2019, pp. 53–60.

[182] M. T. Le, F. Diehl, T. Brunner, and A. Knol, "Uncertainty Estimation for Deep Neural Object Detectors in Safety-Critical Applications," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems*, Maui, Hawaii, USA, Nov. 2018, pp. 3873–3878.

[183] C. Li and G. H. Lee, "Generating Multiple Hypotheses for 3D Human Pose Estimation With Mixture Density Network," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE Press, Jun. 2019, pp. 9879–9887.

[184] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," in *Proceedings of the 2015 International Conference on Machine Learning*, Lille, France: PMLR, 2015, pp. 1613–1622.

[185] W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson, "Fast Uncertainty Estimates and Bayesian Model Averaging of DNNs," in *Proceedings of the Uncertainty in Deep Learning Workshop at UAI 2018*, Monterey, California, USA, 2018, p. 8.

[186] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proceedings of the 2016 International Conference on Machine Learning*, New York, New York, USA: PMLR, 2016, pp. 1050–1059.

[187] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan, "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness," in *Proceedings of the 2020 Conference on Advances in Neural Information Processing Systems*, Virtual: Curran Associates, 2020, p. 25.

[188] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Proceedings of the 2017 Conference on Advances in Neural Information Processing Systems*, Long Beach, California, USA: Curran Associates, 2017, pp. 6402–6413.

[189] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, "Sampling-Free Epistemic Uncertainty Estimation Using Approximated Variance Propagation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea: IEEE Press, Oct. 2019, pp. 2931–2940.

[190] D. Hendrycks and K. Gimpel, "A Baseline for Detecting Misclassified and Out of Distribution Examples in Neural Networks," in *Proceedings of the 2017 International Conference on Learning Representations*, Toulon, France: OpenReview, 2017, p. 12.

[191] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep Anomaly Detection with Outlier Exposure," in *Proceedings of the 2019 International Conference on Learning Representations*, New Orleans, Louisiana, USA: OpenReview, May 2019, p. 18.

[192] S. Liang, Y. Li, and R. Srikant, "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks," in *Proceedings of the 2018 International Conference on Learning Representations*, Vancouver, British Columbia, Canada: OpenReview, 2018, p. 27.

[193] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proceedings of the 2017 International Conference on Learning Representations*, Toulon, France: OpenReview, 2017, p. 32.

[194] A. Van den Oord, N. Kalchbrenner, E. Lasse, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Conditional Image Generation with PixelCNN Decoders," in *Proceedings of the 2016 Conference on Advances in Neural Information Processing Systems 2016*, Barcelona, Spain: Curran Associates, 2016, pp. 4790–4798.

[195] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proceedings of the 2014 International Conference on Learning Representations*, Banff, Alberta, Canada: OpenReview, 2014, p. 14.

[196] J. Ren, P. J. Liu, E. Fertig, *et al.*, "Likelihood Ratios for Out-of-Distribution Detection," in *Proceedings of the 2019 Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2019, pp. 14 680–14 691.

[197] Z. Xiao, Q. Yan, and Y. Amit, "Likelihood Regret: An Out-of-Distribution Detection Score For Variational Auto-encoder," in *Proceedings of the 2020 Conference on Advances in Neural Information Processing Systems*, Virtual: Curran Associates, 2020, p. 12.

[198] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, "Input complexity and out-of-distribution detection with likelihood-based generative models," in *Proceedings of the 2020 International Conference on Learning Representations*, Addis Ababa, Ethiopia: OpenReview, Jan. 2020, p. 15.

[199] H. Choi, E. Jang, and A. A. Alemi, "WAIC, but Why? Generative Ensembles for Robust Anomaly Detection," in *arXiv:1810.01392 [cs, stat]*, Oct. 2018.

[200] S. Watanabe and M. Opper, "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory.," *Journal of Machine Learning Research*, vol. 11, no. 12, 2010.

[201] M. E. Hellman, "The Nearest Neighbor Classification Rule with a Reject Option," *IEEE Transactions on Systems Science and Cybernetics*, vol. 6, no. 3, pp. 179–185, Jul. 1970.

[202] G. Fumera and F. Roli, "Support Vector Machines with Embedded Reject Option," in *Proceedings of the 2002 Pattern Recognition with Support Vector Machines Workshop*, Niagara Falls, Ontario, Canada: Springer Berlin Heidelberg, 2002, pp. 68–82.

[203] Y. Geifman, G. Uziel, and R. El-Yaniv, "Bias-Reduced Uncertainty Estimation for Deep Neural Classifiers," in *Proceedings of the 2019 International Conference on Learning Representations*, New Orleans, Louisiana, USA: OpenReview, 2019, pp. 1–14.

[204] M. H. DeGroot and S. E. Fienberg, "The Comparison and Evaluation of Forecasters," *The Statistician*, vol. 32, no. 1/2, p. 12, Mar. 1983.

[205] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur, "Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 3868–3878, 2020.

[206] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation," *Computational Statistics & Data Analysis*, vol. 142, p. 13, 2020.

[207] Y. Xiao and W. Y. Wang, "Quantifying Uncertainties in Natural Language Processing Tasks," in *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA: AAAI Press, 2019, pp. 7322–7329.

[208] E. Kochkina and M. Liakata, "Estimating predictive uncertainty for rumour verification models," in *Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics*, Virtual: Association for Computational Linguistics, May 2020, pp. 6964–6981.

[209] X. Zhang, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Mitigating Uncertainty in Document Classification," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 3126–3136.

[210] A. Kendall and R. Cipolla, "Modelling Uncertainty in Deep Learning for Camera Relocalization," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, Stockholm, Sweden: IEEE Press, Sep. 2015, pp. 4762–4769.

[211] D. Hall, F. Dayoub, J. Skinner, *et al.*, "Probabilistic Object Detection: Definition and Evaluation," in *Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision*, Snowmass Village, Colorado, USA: IEEE Press, Apr. 2019, pp. 1020–1029.

[212] T. J. Hastie and R. Tibshirani, "Generalized Additive Models," *Journal of Statistical Science*, vol. 1, no. 3, pp. 297–318, 1986.

[213] A. T. Nguyen, A. Kharosekar, S. Krishnan, *et al.*, "Believe it or not: Designing a Human-AI Partnership for Mixed-Initiative Fact-Checking," in *Proceedings of the 2018 Annual ACM Symposium on User Interface Software and Technology*, Berlin Germany: ACM Press, Oct. 2018, pp. 189–199.

[214] M. Jain, R. Kota, P. Kumar, and S. N. Patel, "Convey: Exploring the Use of a Context View for Chatbots," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal, Quebec, Canada: ACM Press, Apr. 2018, pp. 1–6.

[215] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 2016 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA: ACM Press, Aug. 2016, pp. 1135–1144.

[216] Q. Li, Q. Tao, S. Joty, J. Cai, and J. Luo, "VQA-E: Explaining, Elaborating, and Enhancing Your Answers for Visual Questions," in *Proceedings of the 2018 European Conference on Computer Vision*, Munich, Germany: Springer, 2018, pp. 570–586.

[217] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Feb. 2020.

[218] T. Kulesza, S. Stumpf, W.-K. Wong, *et al.*, "Why-oriented end-user debugging of naive Bayes text classification," *ACM Transactions on Interactive Intelligent Systems*, vol. 1, no. 1, pp. 1–31, Oct. 2011.

[219] B. Ghai, Q. V. Liao, Y. Zhang, R. Bellamy, and K. Mueller, *Explainable Active Learning (XAL): An Empirical Study of How Local Explanations Impact Annotator Experience*, Sep. 2020. [Online]. Available: `http://arxiv.org/abs/2001.09219` (visited on 03/22/2022).

[220] Y. Zhang, Q. V. Liao, and R. K. E. Bellamy, "Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Barcelona, Spain: ACM Press, Jan. 2020, pp. 295–305.

[221] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, "Designing Theory-Driven User-Centric Explainable AI," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, UK: ACM, May 2019, pp. 1–15.

[222] E. Tjoa and C. Guan, "A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[223] G. Bansal, T. Wu, J. Zhou, *et al.*, "Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama, Japan: ACM Press, Jan. 2021, pp. 1–16.

[224] G. Bansal, B. Nushi, E. Kamar, W. S. Lasecki, D. S. Weld, and E. Horvitz, "Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance," in *Proceedings of the 2019 AAAI Conference on Human Computation and Crowdsourcing*, Orlando, Florida, USA: AAAI Press, 2019, pp. 2–11.

[225] G. Bansal, B. Nushi, E. Kamar, E. Horvitz, and D. S. Weld, "Is the Most Accurate AI the Best Teammate? Optimizing AI for Teamwork," in *Proceedings of the 2021 AAAI Conference on Artificial Intelligence*, Virtual: AAAI Press, May 2021, pp. 11 405–11 414.

[226] M. Chang, A. Truong, O. Wang, M. Agrawala, and J. Kim, "How to Design Voice Based Navigation for How-To Videos," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, UK: ACM Press, May 2019, pp. 701–712.

[227] M. Chang, M. Huh, and J. Kim, "RubySlippers: Supporting Content-based Voice Navigation for How-to Videos," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama, Japan: ACM Press, May 2021, 97:1–97:14.

[228] Y. Zhao, R. Jaber, D. McMillan, and C. Munteanu, ""Rewind to the Jiggling Meat Part": Understanding Voice Control of Instructional Videos in Everyday Tasks," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, Louisiana, USA: ACM Press, Apr. 2022, 58:1–58:11.

[229] L. Rosenblatt, P. Carrington, K. Hara, and J. P. Bigham, "Vocal Programming for People with Upper-Body Motor Impairments," in *Proceedings of the 2018 International Web for All Conference*, Lyon, France: ACM Press, Apr. 2018, 30:1–30:10.

[230] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," in *Proceedings of the 2010 ACM SIGKDD Workshop on Human Computation*, Washington, District of Columbia, USA: ACM Press, 2010, pp. 64–67.

[231] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "OpenSurfaces: A richly annotated catalog of surface appearance," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 1, Jul. 2013.

[232] A. Sorokin, D. Berenson, S. S. Srinivasa, and M. Hebert, "People helping robots helping people: Crowdsourcing for grasping novel objects," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, ISSN: 2153-0858, Taipei, Taiwan: IEEE Press, Oct. 2010, pp. 2117–2122.

[233] H. Kaur, M. Gordon, Y. Yang, *et al.*, "CrowdMask: Using Crowds to Preserve Privacy in Crowd-Powered Systems via Progressive Filtering," in *Proceedings of the 2017 AAAI Conference on Human Computation and Crowdsourcing*, Quebec City, Quebec, Canada: AAAI Press, Sep. 2017, pp. 89–98.

[234] U. Gadiraju, B. Fetahu, and R. Kawase, "Training Workers for Improving Performance in Crowdsourcing Microtasks," in *Proceedings of the 2015 European Conference on Technology Enhanced Learning*, Toledo, Span: Springer International Publishing, 2015, pp. 100–114.

[235] K. Hara, J. Sun, R. Moore, D. Jacobs, and J. Froehlich, "Tohme: Detecting curb ramps in google street view using crowdsourcing, computer vision, and machine learning," in *Proceedings of the 2014 Annual ACM Symposium on User Interface Software and Technology*, Honolulu, Hawaii, USA: ACM Press, 2014, pp. 189–204.

[236] J. C. Chang, S. Amershi, and E. Kamar, "Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, Colorado, USA: ACM Press, 2017, pp. 2334–2346.

[237] N. B. Shah and D. Zhou, "Double or Nothing: Multiplicative Incentive Mechanisms for Crowdsourcing," in *Proceedings of the 2015 Conference on Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada: Curran Associates, 2015, pp. 1–9.

[238] W. Lasecki, C. Homan, and J. Bigham, "Tuning the Diversity of Open-Ended Responses From the Crowd," in *Proceedings of the 2014 AAAI Conference on Human Computation and Crowdsourcing*, Pittsburgh, Pennsylvania, USA: AAAI Press, Sep. 2014, pp. 36–37.

[239] A. Mao, E. Kamar, Y. Chen, *et al.*, "Volunteering Versus Work for Pay: Incentives and Tradeoffs in Crowdsourcing," in *Proceedings of the 2013 AAAI Conference on Human Computation and Crowdsourcing*, Palm Springs, California, USA: AAAI Press, Nov. 2013, pp. 94–102.

[240] W. S. Lasecki, R. Wesley, J. Nichols, A. Kulkarni, J. F. Allen, and J. P. Bigham, "Chorus: A crowd-powered conversational assistant," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, St. Andrews Scotland, United Kingdom: ACM, Oct. 2013, pp. 151–162.

[241] A. P. Dawid and A. M. Skene, "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm," *Applied Statistics*, vol. 28, no. 1, p. 20, 1979.

[242] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise," in *Proceedings of the 2009 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2009, pp. 2035–2043.

[243] V. C. Raykar and S. Yu, "Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks," *Journal of Machine Learning Research*, vol. 13, pp. 491–518, Feb. 2012.

[244] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The Multidimensional Wisdom of Crowds," in *Proceedings of the 2010 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2010, pp. 2424–2432.

[245] J. Bragg, "Crowdsourcing Multi-Label Classification for Taxonomy Creation," in *Proceedings of the 2013 AAAI Conference on Human Computation and Crowdsourcing*, Palm Springs, California, USA: AAAI Press, Nov. 2013, p. 9.

[246] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, USA: IEEE Press, 2009, pp. 248–255.

[247] *80% of businesses want chatbots by 2020*, Dec. 2016. [Online]. Available: `https://www.businessinsider.com/80-of-businesses-want-chatbots-by-2020-2016-12` (visited on 11/29/2021).

[248] J. E. Katz, J. Groshek, and J. Walsh, "Human Touch and the Customer Service Experience," Center for Research on the Information Society, Tech. Rep., 2016, p. 26.

[249] D. B. Shank, C. Graves, A. Gott, P. Gamez, and S. Rodriguez, "Feeling our way to machine minds: People's emotions when perceiving mind in artificial intelligence," *Computers in Human Behavior*, vol. 98, pp. 256–266, Sep. 2019.

[250] J. Zamora, "I'm Sorry, Dave, I'm Afraid I Can't Do That: Chatbot Perception and Expectations," in *Proceedings of the 2017 International Conference on Human Agent Interaction*, Bielefeld, Germany: ACM Press, Oct. 2017, pp. 253–260.

[251] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, Are You Listening?: Privacy Perceptions, Concerns and Privacy-seeking Behaviors with Smart Speakers," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, p. 31, Nov. 2018.

[252] J. Hill, W. Randolph Ford, and I. G. Farreras, "Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations," *Computers in Human Behavior*, vol. 49, pp. 245–250, Aug. 2015.

[253] A. Ho, J. Hancock, and A. S. Miner, "Psychological, Relational, and Emotional Effects of Self-Disclosure After Conversations With a Chatbot," *Journal of Communication*, vol. 68, no. 4, pp. 712–733, Aug. 2018.

[254] M. Jain, P. Kumar, R. Kota, and S. N. Patel, "Evaluating and Informing the Design of Chatbots," in *Proceedings of the 2018 Designing Interactive Systems Conference*, Hong Kong, China: ACM Press, Jun. 2018, pp. 895–906.

[255] R. Kocielnik, S. Amershi, and P. N. Bennett, "Will You Accept an Imperfect AI?: Exploring Designs for Adjusting End-user Expectations of AI Systems," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, UK: ACM Press, May 2019, pp. 411–425.

[256] P. Dai, C. H. Lin, Mausam, and D. S. Weld, "POMDP-based control of workflows for crowdsourcing," *Artificial Intelligence*, vol. 202, pp. 52–85, Sep. 2013.

[257] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA: IEEE PRess, Jul. 2017, pp. 7464–7473.

[258] P. Prabhakar, N. Kulkarni, and L. Zhang, *Question Relevance in Visual Question Answering*, Jul. 2018.

[259] S. K. Mustikovela, V. Jampani, S. De Mello, *et al.*, "Self-Supervised Viewpoint Learning From Image Collections," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA: IEEE Press, Jun. 2020, pp. 3970–3980.

[260] S. Liao, E. Gavves, and C. G. M. Snoek, "Spherical Regression: Learning Viewpoints, Surface Normals and 3D Rotations on N-Spheres," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE Press, Jun. 2019, pp. 9751–9759.

[261] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010, p. 8.

[262] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[263] Z. Li and D. Hoiem, "Learning without Forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[264] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, California, USA, Jun. 2010, pp. 3485–3492.

[265] P. M. Larochelle, A. P. Murray, and J. Angeles, "A Distance Metric for Finite Sets of Rigid-Body Displacements via the Polar Decomposition," *Journal of Mechanical Design*, vol. 129, no. 8, pp. 883–886, Aug. 2007.

[266] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 2015 International Conference on Learning Representations*, San Diego, California, USA, 2015, pp. 1–15.

[267] N. Louis, J. J. Corso, T. N. Templin, T. D. Eliason, and D. P. Nicolella, "Learning to Estimate External Forces of Human Motion in Video," in *Proceedings of the 2022 ACM International Conference on Multimedia*, Lisboa, Portugal: ACM Press, Oct. 2022, pp. 3540–3548.

[268] G. V. Horn, O. M. Aodha, Y. Song, *et al.*, "The INaturalist Species Classification and Detection Dataset," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA: IEEE Press, 2018, pp. 8769–8778.

[269] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, USA: IEEE Press, Jun. 2014, pp. 1717–1724.

[270] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Z. Zhang, "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, Massachusetts, USA: IEEE Press, Jun. 2015, pp. 3982–3991.

[271] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[272] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[273] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[274] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, Santiago, Chile: IEEE Press, 2015, pp. 3730–3738.

[275] D. Gurari, S. D. Jain, M. Betke, and K. Grauman, "Pull the Plug? Predicting If Computers or Humans Should Segment Images," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, Jun. 2016, pp. 382–391.

[276] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA: IEEE Press, 2016, pp. 770–778.

[277] H. de Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela, "Talk the Walk: Navigating New York City through Grounded Dialogue," *arXiv:1807.03367 [cs]*, Dec. 2018.

[278] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Jun. 2012, pp. 3354–3361.

[279] M. Bojarski, D. Testa, D. Dworakowski, *et al.*, *End to End Learning for Self-Driving Cars*, Apr. 2016.

[280] N. Lalra and S. Paddock, "How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability," *Research Report, Rand Corporation*, p. 3, 2016.

[281] A. C. Madrigal, *Inside Waymo's Secret World for Training Self-Driving Cars*, Aug. 2017. [Online]. Available: `www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/` (visited on 10/08/2019).

[282] D. Silver, "Waymo Has the Most Autonomous Miles, By a Lot," *Forbes*, Jul. 2018.

[283] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 184–204, Jan. 2013.

[284] L. A. Necker, "Observations on some remarkable optical phenomena seen in Switzerland; and on an optical phenomenon which occurs on viewing a figure of a crystal or geometrical solid," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 1, no. 5, pp. 329–337, 1832.

[285] S. Thrun, "Monte Carlo POMDPs," in *Proceedings of the 1999 Conference on Advances in Neural Information Processing Systems*, Denver, Colorado, USA, 1999, pp. 1064–1070.

[286] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," in *Proceedings of the Eighth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, Edmonton, Alberta, Canada: AAAI Press/The MIT Press, 2002, pp. 593–598.

[287] B. Kwolek, "Model Based Facial Pose Tracking Using a Particle Filter," in *Proceedings of the 2006 International Conference on Geometric Modeling and Imaging*, London, England: IEEE Press, 2006, pp. 203–208.

[288] K. Oka, Y. Sato, Y. Nakanishi, and H. Koike, "Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control.," in *MVA*, 2005, pp. 586–589.

[289] M. Bray, E. Koller-Meier, and L. Van Gool, "Smart particle filtering for 3D hand tracking," in *Proceedings of the 2004 IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, South Korea: IEEE Press, 2004, pp. 675–680.

[290] M. Gordon, J. P. Bigham, and W. S. Lasecki, "LegionTools: A Toolkit + UI for Recruiting and Routing Crowds to Synchronous Real-Time Tasks," in *Proceedings of the 2015 Annual ACM Symposium on User Interface Software & Technology*, Daegu, Kyungpook, South Korea: ACM Press, 2015, pp. 81–82.

[291] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, Dec. 1997.

[292] Y.-J. Tsai, A. Bousse, M. J. Ehrhardt, *et al.*, "Fast Quasi-Newton Algorithms for Penalized Reconstruction in Emission Tomography and Further Improvements via Preconditioning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 4, pp. 1000–1010, Apr. 2018.

[293] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele, "Building statistical shape spaces for 3D human modeling," *Pattern Recognition*, vol. 67, pp. 276–286, Jul. 2017.

[294] A. Harakeh, M. Smart, and S. L. Waslander, "BayesOD: A Bayesian Approach for Uncertainty Estimation in Deep Object Detectors," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*, Virtual: IEEE Press, 2020, pp. 87–93.

[295] S. J. Lemmer, J. Y. Song, and J. J. Corso, "Crowdsourcing More Effective Initializations for Single-Target Trackers Through Automatic Re-querying," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Virtual: ACM Press, May 2021, 391:1–391:13.

[296] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the *EM* Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, Sep. 1977.

[297] M. Ester, H.-P. Kriegel, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 1996 International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, USA: AAAI Press, 1996, pp. 226–231.

[298] D. Widmann, F. Lindsten, and D. Zachariah, "Calibration tests in multi-class classification: A unifying framework," in *Proceedings of the 2019 Conference on Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada: Curran Associates, 2019, pp. 12 236–12 246.

[299] H. Cen, K. Koedinger, and B. Junker, "Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement," in *Proceedings of the 2006 International Conference on Intelligent Tutoring Systems*, Jhongli, Taiwan: Springer, 2006, pp. 164–175.

[300] K. Rivers, E. Harpstead, and K. Koedinger, "Learning Curve Analysis for Programming: Which Concepts do Students Struggle With?" In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, Melbourne, Victoria, Australia: ACM Press, Aug. 2016, pp. 143–151.

[301] G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz, "Updates in Human-AI Teams: Understanding and Addressing the Performance/Compatibility Tradeoff," in *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA: AAAI Press, Jul. 2019, pp. 2429–2437.

[302] M. Haghighat and M. A. Razian, "Fast-FMI: Non-reference image fusion metric," in *Proceedings of the 2014 IEEE International Conference on Application of Information and Communication Technologies*, Paris, France: IEEE Press, Oct. 2014, pp. 1–3.

[303] M. R. Ganesh, J. J. Corso, and S. Y. Sekeh, "MINT: Deep Network Compression via Mutual Information-based Neuron Trimming," in *Proceedings of the 2020 International Conference on Pattern Recognition*, Virtual: Springer, Jan. 2021, pp. 8251–8258.

[304] M. Noshad, Y. Zeng, and A. O. Hero III, "Scalable Mutual Information Estimation using Dependence Graphs," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, United Kingdom: IEEE Press, 2019, pp. 2962–2966.

[305] O. Gascuel and G. Caraux, "Distribution-free performance bounds with the resubstitution error estimate," *Pattern Recognition Letters*, vol. 13, no. 11, pp. 757–764, Nov. 1992.

[306] A. E. Pollard and J. L. Shapiro, *Visual Question Answering as a Multi-Task Problem*, Jul. 2020.

[307] S. Amershi, D. Weld, M. Vorvoreanu, *et al.*, "Guidelines for Human-AI Interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow Scotland Uk: ACM, May 2019, pp. 1–13.

[308] A. Rosenfeld, R. Zemel, and J. K. Tsotsos, *The Elephant in the Room*, [cs], Aug. 2018.

[309] J. Wei, Y. Tay, R. Bommasani, *et al.*, "Emergent Abilities of Large Language Models," *Transactions on Machine Learning Research*, 2022.

[310] L. F. W. Anthony, B. Kanding, and R. Selvan, *Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models*, Jul. 2020.

[311] D. Gurari, Q. Li, C. Lin, *et al.*, "VizWiz-Priv: A Dataset for Recognizing the Presence and Purpose of Private Visual Information in Images Taken by Blind People," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE Press, Jun. 2019, pp. 939–948.

[312] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women Also Snowboard: Overcoming Bias in Captioning Models," in *Proceedings of the 2018 European Conference on Computer VIsion*, Munich, Germany: Springer International Publishing, 2018, pp. 793–811.

[313] K. Yang, K. Qinami, L. Fei-Fei, J. Deng, and O. Russakovsky, "Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the ImageNet hierarchy," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Barcelona Spain: ACM Press, Jan. 2020, pp. 547–558.

[314] V. Manjunatha, N. Saini, and L. S. Davis, "Explicit Bias Discovery in Visual Question Answering Models," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE PRess, Jun. 2019, pp. 9554–9563.

[315] M. Hall, L. van der Maaten, L. Gustafson, M. Jones, and A. Adcock, *A Systematic Study of Bias Amplification*, arXiv:2201.11706 [cs], Oct. 2022. (visited on 02/16/2023).

[316] P. Stock and M. Cisse, "ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases," in *Proceedings of the 2018 European Conference on Computer Vision*, Series Title: Lecture Notes in Computer Science, Munich, Germany: Springer International Publishing, 2018, pp. 504–519.

[317] Y. Li and N. Vasconcelos, "REPAIR: Removing Representation Bias by Dataset Resampling," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA: IEEE Press, Jun. 2019, pp. 9564–9573.

[318] A. Moss, *Demographics of People on Amazon Mechanical Turk*, Jun. 2020. [Online]. Available: https://www.cloudresearch.com/resources/blog/who-uses-amazon-mturk-2020-demographics/ (visited on 02/17/2023).

[319] J. Uijlings, K. Konyushkova, C. H. Lampert, and V. Ferrari, "Learning Intelligent Dialogs for Bounding Box Annotation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE Press, Jun. 2018, pp. 9175–9184.

[320] B. Zhang, E. M. Provost, and G. Essl, "Cross-Corpus Acoustic Emotion Recognition with Multi-Task Learning: Seeking Common Ground While Preserving Differences," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 85–99, Jan. 2019.