Deep Signal Compression with Feature Representation Learning

by

Bowen Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in The University of Michigan
2024

Doctoral Committee:

Associate Professor Hun-Seok Kim, Chair
Professor Jeffrey A. Fessler
Assistant Professor Qing Qu
Associate Professor Alanson Sample

Bowen Liu

bowenliu@umich.edu

ORCID iD: 0000-0002-4194-8119

© Bowen Liu 2024

All Rights Reserved

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who has helped me in the completion of this dissertation. First and foremost, I would like to thank my advisor and committee chair, Professor Hun-Seok Kim, for his invaluable guidance, unwavering support, and endless patience throughout my Ph.D. journey. His expertise, encouragement, and constructive feedback have been instrumental in shaping this work.

I extend my heartfelt appreciation to my esteemed committee members, Professor Jeffery A. Fessler, Professor Alanson Sample, and Professor Qing Qu, for their invaluable feedback, critical insights, and scholarly contributions that have enhanced the rigor and depth of this thesis.

I am deeply appreciative of the countless friends, colleagues, and mentors who have offered their guidance, support, and encouragement along this arduous yet rewarding journey.

Finally, I am indebted to my wife and parents for their unconditional love, unwavering support, and sacrifices they have made to provide me with opportunities to pursue my dreams. Their encouragement and belief in my abilities have been my guiding light.

This thesis is a testament to the collective efforts, support, and encouragement of all those mentioned above, and I am profoundly grateful for their contributions to my academic and personal growth.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Deep learning-based lossy signal compression methods have achieved substantial progress and significantly enriched signal compression methodologies in recent years. There are two major aspects that signal source coding can benefit from learned methods. Firstly, the data-intense nature of deep signal compression methods allows a good capture of the probabilistic distribution of feature representations, which leads to efficient entropy coding with proper modeling. Secondly, neural network architectures can provide powerful solutions to feature extraction and representation learning, therefore enabling the elimination of spatial and temporal redundancies by mapping the raw signal to compacter feature domains. This thesis presents four related works addressing the compression problem of different data formats, including speech audio, image, video, and point cloud.

The first work introduces a unified compression method that uses generative adversarial networks (GAN) to compress speech audio and images. The compressed signal is represented by a latent vector fed into a generator network, which is trained to produce high-quality signals that minimize a target objective function. The alternating direction method of multipliers (ADMM) based non-uniform quantization is incorporated to effectively discretize the resulting latent vectors.

The second work presents a deep video coding framework that predicts and compresses video sequences in the latent vector space. The proposed method first learns the efficient feature domain representation of each video frame and then performs inter-frame prediction in that lower-dimensional space. To exploit the temporal correlation among the feature space frames, it employs a convolutional long short-term memory (ConvLSTM) based network to predict the representation of the future frame. The transmitted bitstream is obtained by quantizing and entropy encoding the feature space residual. The application of the proposed video prediction scheme is studied in the anomaly detection task.

The third work aims to address the motion pattern adaptability issue that widely exists in video codecs by a block wise mode ensemble deep video compression framework. It selects the optimal mode for feature domain prediction adapting to different motion patterns. Proposed multi-modes include ConvLSTM-based feature domain prediction, optical flow conditioned feature domain prediction, and feature propagation to address a wide range of cases from static scenes without apparent motions to dynamic scenes with a moving camera. Guided by a binary density map, dense and sparse post-quantization residual blocks are coded in separate entropy coding schemes. On top of that, applying optional run-length coding to sparse residuals can further improve the compression rate.

The last work focuses on exploring methods to compress light detection and ranging (LiDAR) data, which extends the study of deep signal compression problems from 2D to 3D domain. LiDAR sensors are widely adopted in a number of applications in the autonomous navigation, virtual reality (VR), and augmented reality (AR) industries, where communication bandwidth is one of the top concerns but understudied. With point clouds and range images being two interchangeable LiDAR data representations, a hybrid framework is introduced to take the best of both worlds. The proposed pipeline mostly relies on a prediction-based approach to exploit spatial and temporal correlations in range images, while providing an octree-based path as an important fallback in certain cases to preserve the reconstruction quality. A content adaptive point cloud sampling technique is also introduced to promote extra compression gains while proving to have minimal impact on machine perceptual tasks.

CHAPTER I

Introduction

1.1 Big Data

The era of big data [99] has emerged as an evolutionary force in various industries, ranging from business and healthcare to scientific research and social media. Rapid technology advancements and the widespread adoption of digital platforms have led to an unprecedented generation of vast amounts of data from diverse sources such as text, image, video, and 3D representations. Big data is characterized not only by its volume but also by its variety, velocity, and veracity. This wealth of information has opened up new opportunities for data-driven decision-making and insights, but it also presents significant challenges that need to be addressed. No matter if we target all the data or just a little, we need to compress the data for less streaming bandwidth and memory requirements.

Recent studies have highlighted the dominance of streaming audio and video content in internet traffic for the year 2022, accounting for a significant majority of the total data volume (over 80% as reported by Cisco). As internet traffic continues to grow, the issue of signal compression becomes increasingly critical in people's daily lives. The size of compressed signals varies based on factors such as resolution, sampling rate, and compression method. As summarized in Table 1.1, the file size indeed increases progressively from audio to video, with point cloud having the largest stor-

Table 1.1: Averaged file size of various media contents, the data amount, resolution/sampling rate, and file format are picked based on their common use cases.

Media Content	Amount/Duration	Resolution/Sampling Rate	File Format	File Size
Audio	1 hour	24 bit & 96 kHz	FLAC	2.02 GB
	1 hour	256 kbps	MP3	0.12 GB
Image	1000	1024×768 pixels	PNG	0.44 GB
	1000	1024×768 pixels	JPG	0.08 GB
Video	1 hour	2160p / 30 fps	AVC	20.51 GB
	1 hour	2160p / 30 fps	HEVC	10.36 GB
Point Cloud	640 billion	1000×1250 m^2	ASCII XYZ	0.51 TB
	640 billion	1000×1250 m^2	Octree	0.28 TB

age requirement among all the types of signals. While audio signals generally require less storage space than more complex data types, the average file size for audio ranges from approximately 0.12 GB to 2.02 GB for a 1-hour duration, indicating considerable storage demands. Video data, in comparison, has even larger file sizes than audio, underscoring the growing importance of signal compression to manage data volumes effectively. To tackle this challenge, compression techniques are utilized to effectively reduce the size of raw data while retaining essential information, catering to both human perception and machine vision needs. Therefore, the subsequent section presents an overview of signal compression across various types of data.

1.2 Signal Compression and Its Challenges

Signal compression is a fundamental problem in the field of digital signal processing, aiming to efficiently represent and store data while minimizing information loss, as discussed in the paper [59]. Traditional compression techniques, such as transform coding and statistical modeling, have been widely used to achieve high compression ratios for multiple types of signals. However, these methods [118, 14, 136] often require handcrafted designs and may not fully exploit the underlying feature, leading to sub-optimal compression performance for complex signals.

Audio compression is possible due to the inherent redundancy and limitations in human auditory perception. In audio signals, there are often areas of redundancy, where certain frequencies or temporal patterns repeat, making the representation inefficient. Compression algorithms [56, 118] exploit this redundancy to remove unnecessary information while preserving essential perceptual features. Additionally, the human auditory system has limited sensitivity outside of the 20Hz-20kHz sound frequency range. By employing perceptual coding techniques, audio compression algorithms [130] can prioritize preserving perceptually important information, allowing for more efficient data representation without significant loss in quality. Overall, audio compression takes advantage of both signal properties and human perceptual characteristics to achieve efficient storage or transmission, while still maintaining satisfactory audio fidelity for most practical applications.

Image compression is made possible by the wide presence of spatial and spectral redundancies in images. Spatial redundancy refers to the repeating patterns or similar pixel values within neighbor regions in an image. Spectral redundancy occurs when the same information is present in multiple color channels or spectral bands. Image codecs [14, 126, 150] leverage these redundancies to efficiently represent images with reduced file size. By employing various compression techniques, such as transform coding [10], quantization [45, 17], and entropy coding [71, 124], image compression achieves significant data reduction while preserving essential visual information. This allows for faster data transfer, reduced storage requirements, and efficient handling of large-scale datasets without compromising the overall visual quality for various applications in computer vision and image processing domains.

Video compression is feasible given the presence of temporal and spatial redundancies. Video compression [160] techniques, such as inter-frame prediction and motion compensation, leverage the similarities between frames due to slow motion or gradual changes in objects and scenes. As a result, only the differences or motion information between them need to be encoded. Based on the above insights, compression methods such as the popular AVC and HEVC [136] standards, achieve substantial code size reduction while maintaining acceptable visual quality for various applications. This is crucial for video streaming services, video conferencing, video surveillance, and social media, where efficient data handling and transmission are essential for seamless user experiences.

Point cloud compression leverages spatial redundancies in 3D point cloud data. This comes from the extensive presence of adjacent 3D points sharing similar geometric attributes, such as coordinates and color information. Various point cloud compression techniques [132], such as octree-based encoding [128], geometry simplification, and predictive coding [47], leverage spatial correlations and inter-point relationships to compactly represent the point cloud while preserving the geometric features. Efficient point cloud compression is essential for a number of applications, including 3D modeling, AR/VR, and driving assistance, in that it facilitates real-time processing and enhances the overall performance of systems relying on 3D spatial information.

Signal compression poses several challenges that researchers and engineers continuously strive to overcome and achieve performance improvements [102, 80, 85, 57]. Finding a balanced solution between high compression efficiency and maintaining acceptable signal quality is a fundamental problem in data compression problems. Different applications may have varying requirements for signal fidelity, making it necessary to optimize the compression technique accordingly. Many advanced compression algorithms involve complex mathematical operations and transforms, leading to high computational requirements. Developing compression techniques that find a good feature representation for downstream tasks is crucial, especially for real-time and resource-constrained applications. In this way, it is possible to exercise various downstream tasks with more compact data representations in the lower-dimensional space.

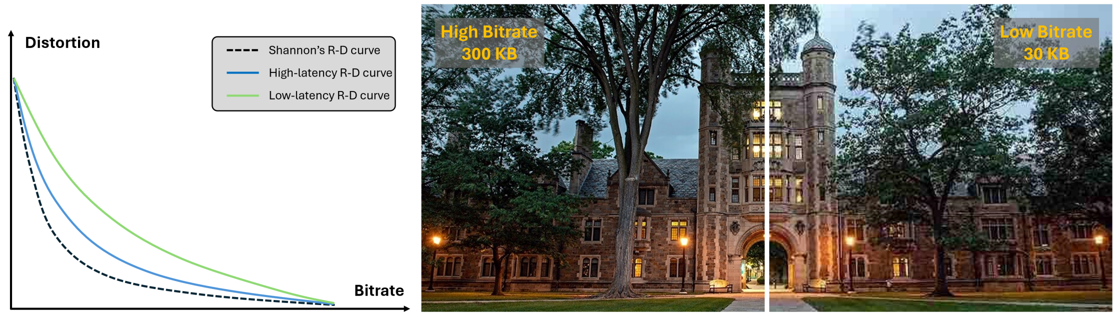

Figure 1.1: Left: Rate-distortion tradeoff visualized as a curve. Right: Image quality vs. file size.

In lossy signal compression, the information cost of the discretized representations and the quantization error are two competing costs that must be traded off [11]. This tradeoff describes the inherent relationship between the compression rate and the distortion introduced during the compression process. As visualized in Figure 1.1, Shannon's unconstrained R-D curve depicts a scenario where the code length expectation is equal to its theoretical lower bound (entropy). The actual achievable tradeoff gives a higher bit rate under the same level of distortion, or more distortion under the same bit rate. Signal compression methods aim to achieve a tradeoff that is as close to the theoretical one as possible. While the goal of signal compression is to reduce the data rate and maintain an acceptable level of reconstruction quality, the notion of latency is another key player that is worth studying. It generally requires more exhaustive searching or exploration to approach the optimal R-D tradeoff, which usually takes longer encoding time. For instance, different configurations of the presets in AVC/HEVC will lead to widely diverse encoding latencies. Therefore, achieving the right balance between rate, distortion, and time efficiency is key to developing effective signal compression systems that cater to the specific needs of various applications.

1.3 Deep Neural Networks and Representation Learning

Deep Neural Networks (DNN) emerged as a novel approach in the field of computer vision, ushering in a new era of intelligent data processing and analysis. By leveraging the power of deep learning, [51] has demonstrated remarkable capabilities in understanding and interpreting visual data, rivaling or even surpassing human performance in various vision tasks. From image classification [31] and object detection [120] to semantic segmentation and image generation, DNNs have become indispensable tools in addressing diverse challenges in the vision domain.

The success of DNNs in vision tasks can be attributed to their ability to abstract feature representations from raw data. By composing multiple layers of simple, nonlinear processing units, [139] progressively extracts features that capture intricate patterns and semantic information. The inherent capacity to learn from vast amounts of labeled data enables DNNs to generalize well across different visual domains and adapt to diverse imaging conditions.

There have been significant advancements in machine learning in recent years, showing revolutionary impact in numerous fields, including recommendation systems and content filtering on social networks. Popular algorithms like Logistic Regression [38, 107], Decision Trees [69], Support Vector Machines [129], and Gaussian Mixture Models [158] have been widely adopted. However, these conventional machine learning methods face limitations when dealing with data from a wide range of sources, given that they heavily rely on manually crafted feature extractors to transform the raw data into appropriate internal representations [76]. For instance, in diagnosing breast cancer, traditional machine learning algorithms require human-defined intermediate features, like tumor radius, texture, smoothness, and area, for prediction. Nonetheless, these algorithms may struggle with advanced tasks like visual understanding, making it non-trivial to encapsulate semantic meanings in these features.

Deep learning establishes a new paradigm by enabling the transformation of low-level raw inputs into high-level abstract representations through the composition of multiple non-linear modules. This approach allows DNNs to implicitly learn parameterized functions by chaining together these transformations. In recent years, the success of deep models has inspired the exploration of deep learning based compression methodologies. Specifically, deep learning allows for more complex and flexible mappings from data to feature space, hence encouraging more accurate estimations or predictions [55, 80, 152] to reduce the information entropy (the theoretical lower bound of bitrate). More importantly, building an end-to-end trainable deep framework that incorporates both the data compression model and the entropy estimation model [12, 105] fundamentally resolves the rate-distortion joint optimization problem. The above breakthroughs have now made learning-based codecs become the state-of-the-art methods for compression in a wide range of data formats, and further encourage studies in representation learning and probability modeling.

1.4 Dissertation Organization

This dissertation demonstrates the study of compression schemes targeting four signal types: image, speech, video, and point cloud. The proposed methodologies center on deep learning-based algorithms for lossy signal compression and their subsequent feature space applications. The signal compression process encompasses several stages, including signal pre-processing, decorrelation, quantization, entropy coding, and post-processing, grounded in the uttermost goal of reducing code size while ensuring acceptable quality degradation. With the focus on finding optimal compact feature representations and leveraging signal redundancies via different techniques in each of these stages, the details of each work are elucidated in the following chapters.

In Chapter II, a novel approach, BPGAN [84], is explored for signal compression using a unified framework for speech and image. In this GAN-based framework, the signal is compressed into a latent vector representation, which is then used as input to a generator network. The generator is trained to produce high-quality, realistic signals that minimize a specified target objective function. To efficiently quantize the compressed signal, BPGAN employs an iterative back-propagation method with an alternating direction method of multipliers optimization. This iterative process helps identify non-uniformly quantized latent vectors optimized for the best quality at target compression rates. The framework achieves a relatively high compression ratio while maintaining a high-quality reconstructed signal with intricate details. Performance of the proposed algorithm using various metrics, which collectively demonstrate the effectiveness and superiority of BPGAN compared to existing methods.

Chapter III presents LVC [85], a deep video coding framework designed to predict and compress video sequences in the latent vector space, effectively addressing spatial and temporal redundancies by obtaining optimal lower-dimensional representations of video frames. The proposed codec first learns efficient feature representations for each video frame and then performs inter-frame prediction in this domain. To leverage the temporal correlation within the video frame sequence, a convolutional long short-term memory network predicts the latent vector representation of future frames. The versatility and effectiveness of this approach are demonstrated through two key applications: video compression and abnormal event detection. Remarkably, both applications share the same latent frame prediction network. Comparative evaluations on the UVG and VTL datasets reveal that the proposed method achieves superior or competitive performance when compared to existing algorithms tailored for either video compression or anomaly detection. These promising results underscore the potential of implementing vision tasks in the latent domain with well-learned representations, signifying a significant step forward in enhancing the efficiency and accuracy of video compression and abnormal event detection algorithms.

A follow-up work in video compression is introduced in Chapter IV, where a dynamic mode selection-based deep video compression approach is attempted. The proposed framework, MMVC [86], adapts to different motion and contextual patterns with a *Skip* mode in the pixel domain, and multiple block-based prediction paths in the feature domain. To improve the residual sparsity without losing much quality while minimizing the bitrate, the framework involves a block wise channel removal scheme and a density-adaptive entropy coding strategy. Extensive experiments and a comparative study are performed to showcase that MMVC exhibits better performance compared to state-of-the-art learning-based methods and conventional codecs. The ablation study confirms the effectiveness of the proposed scheme by quantifying the utilization of different modes that vary by video content and scenes.

Chapter V concludes the dissertation with studies on LiDAR data compression in the 3D domain. LiDAR sensors are usually equipped on automobile and wearable devices. Managing the substantial volume of point cloud is essential for real-world use cases, posing a need for efficient compression algorithms. Similar to other data compression task domains, point cloud compression seeks to spatially and temporally decorrelate the information while preserving the reconstruction quality. There are existing methods that adopt the range image representation and use frameworks similar to the deep video coders to effectively achieve this goal. However, an obscured sensor pose retrieval practice or failures in motion estimation in real scenarios can result in considerable quality degradation in certain cases. Compensating for either kind of information loss in the range image space is non-trivial. In light of resolving the above issue, a hybrid point cloud compression pipeline, H-PCC, is proposed. H-PCC allows the fallback of the compression scheme to a new octree-based method supported by clustering. Further, the point cloud density can be adaptively reduced via fusion with the detected region of interest (RoI) in color images associated with the LiDAR data, which makes little impact on the performance of machine perception tasks.

CHAPTER II

Unified Signal Compression Using Generative Adversarial Networks

2.1 Introduction

[1] Image resolution and audio quality are continually increasing in today's data streams; however, limited communication bandwidth and storage space have posed substantial challenges to practical applications. Indeed, raw image and audio data can occupy significant storage space due to the inherent redundancy present in the data. In both images and audio, there are often repeating patterns and similarities between neighboring pixels or samples, leading to redundant information. It is impractical to store and communicate the large data streams generated in this information-rich world, and therefore signal compression is essential. In a typical image, lower spatial frequency components carry more information than higher spatial frequency ones. This property is exploited in a number of image compression algorithms, such as JPEG and BPG. For audio information, humans can hear frequencies ranging from 20 Hz to 20 kHz but are most sensitive to sounds between 1 kHz and 5 kHz while the speech signals have a strong correlation in the temporal domain allowing prediction of future signals. These properties are used by several algorithms for speech com-

[1] The content of this chapter is based on [84].

pression such as Code-excited linear prediction (CELP) [130] and adaptive multi-rate wideband (AMR-WB) [15].

A typical signal compression framework consists of three core components: an encoder, decoder, and quantizer. The encoder maps the target signals to the latent space, and the decoder reverses these latent representations back into target signals. The quantizer maps the signal representations to a stream of discretized symbols to reduce the bitrate of the compressed signals.

Inspired by the recent remarkable success of generative adversarial networks [44] in various applications, this work proposes a unified signal compression framework named back-propagated GAN (BPGAN). The compressed signal is represented by a latent vector fed into a generator network that is trained to produce high-quality realistic signals (either image or speech). The core idea of BPGAN is to 'search' for the optimal latent vector through iterative back-propagation during the encoding process for a given generator (with fixed weights) and compress the target signal. This process minimizes a loss function computed based on the generator output and the target signal, enabling a high-quality compressed signal represented by the latent vector input to the generator. This framework is generally applicable to different types of signals including speech and image as long as a GAN is trainable in that signal domain.

Recently, deep autoencoder structures have provided promising results in signal compression tasks. These methods have achieved better performance compared with many traditional (non-learning based) compression techniques. One of the key challenges of DNN-based signal compression is optimizing the bitrate of latent representation in the auto-encoder fashion, which requires a careful design of the quantizer. Such designs commonly employ either uniform or non-uniform quantization. Deep compression [49] first proposed the k-means non-uniform quantization in the DNN weight compression task, by updating the centroid of each cluster during the retraining process, to dramatically reduce the bit length of fixed point DNN weights. However, the cluster centroid values in this method vary for different compression targets, and the dynamic cluster centroid values are required during the decompression process, which adds a non-trivial overhead to the communication bandwidth. Extreme image compression [79] proposed a differentiable quantization module inserted in the bottleneck of an encoder-decoder style network, and it optimizes the quantization center and minimizes the bitrate in an end-to-end style during the training process. In the proposed quantization section, BPGAN formulates the quantization task as a manifold optimization problem with quantization constraints and puts forward an alternating direction method of multipliers solution.

2.2 Related Work

2.2.1 Image Compression

Pioneering deep autoencoder and neural network-based image compression methods include [11] and [122]. The focus of [11] is to optimize the mean squared error (MSE) and multi-scale structural similarity for image quality assessment (MS-SSIM) between decompressed images and the originals. In [141], the images are compressed through an encoder, and a traditional quantization method is applied to reduce the bitrate. GAN-based models are widely explored in image compression tasks. Prior works such as [100, 5] apply adversarial training on deep autoencoder networks to learn the underlying distribution of images and achieve extremely high compression rates with aesthetically pleasing details on the generated image. However, those ‘realistic’ details generated by the decoder often distort the ‘actual’ details of the original image. Recently, there has been a notable trend of using context-adaptive entropy models for image compression [105, 78, 26], where hyperpriors allocate additional bits to more complex and bit-consuming contexts, whereas autoregressive models are applied to the contexts that can be easily inferred. Unlike the aforementioned prior works, this proposed method uses a generator network for image/speech compression as well as decompression by iteratively searching the optimum latent input representation for the generator during the encoding process.

2.2.2 Speech Compression

Traditional speech codecs such as CELP [130], Opus [144], and AMR-WB [15] commonly employ hand-engineered encoder-decoder datapaths. These datapaths heavily rely on crafted audio representation features and therefore require high bitrates (typically higher than 16 kbps) for acceptable speech quality. Recent DNN-based approaches including [66] have demonstrated the feasibility of training an end-to-end speech codec that exhibits performance comparable to a hand-crafted AMR-WB codec at 9–24 kbps. In [20], paired phonological analyzers and synthesizers using deep and spiking neural networks report a 369 bps bitrate speech codec by only keeping the content and speaker identifier information to achieve a very low bitrate.

Another important strategy to realize a high-quality speech codec is to use a fine-tuned DNN-based vocoder such as Wavenet [111] or WaveRNN [64] to synthesize speech with high-quality. For example, a prior work [70] employs a learned Wavenet as an encoder to generate audio given a traditional parametric codec with audio quality that is on par with that produced by AMR-WB at 23.05 kbps. Furthermore, [146] demonstrates that the Wavenet vocoder can generate satisfactory speech given a discrete latent representation of audio generated by the vector-quantized variational autoencoder (VQ-VAE) framework. And [40] extends this work to reach 1.6 kbps. These methods, however, do not scale well to a very low bit-rate (*e.g.*, 1 kbps). The BPGAN for speech compression overcomes the limitations of the aforementioned prior works by achieving lower bitrates for the same quality while providing excellent scalability to tradeoff bitrate vs. signal quality.

The application of GAN has been studied for speech processing. In [113], speech enhancement GAN (SEGAN) is proposed, which demonstrates that GAN can achieve promising results in speech processing tasks. Furthermore, [34, 97, 36] show that it is possible for GAN to synthesize instrumental audio signals and simple speech with limited contexts. Nonetheless, it is still considerably difficult to generate high-quality speech audios with arbitrary latent input signals.

2.2.3 ADMM Optimization

The idea of the ADMM was first introduced in the 1970s to solve convex optimization problems with additional constraints. Later work found that ADMM can also be used for non-convex problems and can achieve sufficiently good, if not globally optimal, solutions. The principle of the ADMM method is splitting difficult problems into several sub-problems and solving these sub-problems efficiently by updating variables alternatively. Therefore, ADMM has been widely used in distributed optimization problems and neural network pruning tasks [17, 79, 171], achieving remarkable results.

2.3 Method

The proposed BPGAN compression framework is applicable to any signal type as long as it is possible to train a generative model that produces realistic outputs of that type. The overall framework of BPGAN compression is shown in Figure 2.1. The target signal x is encoded as an initialization of the compressed signal $z_0 = E(x)$ in the first step. Then, the latent vector z is updated and optimized to minimize a specific objective function $F(\cdot)$ via iterative back-propagation through a pre-trained generator $G(\cdot)$. A typical objective function is the similarity measure between the target signal x and the reconstructed signal $G(z)$. During the iterative back-propagation process, the optimal latent variable $\tilde{z}$ is discretized under the quantization scheme

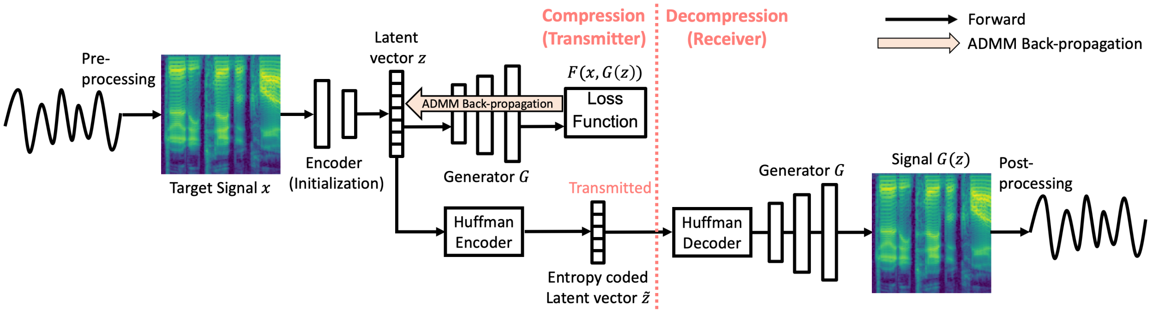

Figure 2.1: Overview of the unified compression framework. The target signal is first encoded as an initialization to the latent representation by an encoder. Then the latent vector z is optimized according to a reconstruction criterion through iterative ADMM back-propagation (BP).

$Q(\cdot)$. Before transmitting to the receiver, the compressed signal is entropy encoded for further code size reduction. In the proposed framework, the generator parameters are pre-trained (*i.e.*, fixed during the iterative back-propagation to find $\tilde{z}$) and shared between the transmitter and the receiver. At the receiver, the same generator decodes the latent signal $G(\tilde{z})$, and this reconstructed image/speech is converted back to its original format via post-processing. The overall compression process is summarized in Algorithm 1.

Unlike other GAN-based approaches [5] relying on an encoder to provide a compressed signal, compression in BPGAN is performed by iteratively searching and updating the generator input latent vector through back-propagation to minimize a target objective function at the output of the generator. In the proposed scheme, the main purpose of the encoder is to accelerate the back-propagation processing by initializing the latent vector using the output of the encoder. This initialization technique reduces the number of iterations significantly. Selecting a proper optimization criterion and progressively updating the lower-dimensional representation through back-propagation for that criterion is the key step to significantly improve the quality and/or compression ratio of the signal. As this compression framework allows the application of various optimization objectives for latent representation searching,

Algorithm 1 BPGAN Compression Algorithm

```
Require: well trained generator $G(\cdot)$, encoder $E(\cdot)$
    pre-defined quantization function $Q(\cdot)$
    signal to be compressed $x$
    objective function $F(\cdot)$
    quantized set $S$
Ensure: latent vector quantization $\tilde{z}$
    latent vector initialization $z_0 = E(x)$
    quantize latent elements into the discrete space $S$
    $z_1 = Q(z_0), z_1 \in S$
    repeat
        calculate the objective function: $F(x, G(z_k))$
        gradient descent: $z_{k+1} = z_k - \alpha \cdot \nabla F(z_k)$
        quantize latent elements into the discrete space $S$
        $z_{k+1} = Q(z_{k+1}), z_{k+1} \in S$
    until convergence to optimal latent variable $\tilde{z}$ or maximum iteration number satisfied
    apply Huffman Coding to $\tilde{z}$
```

it enables objective-aware signal compression to obtain the optimized compression results tailored for a target application such as signal classification and recognition.

2.3.1 Training Methodology

The proposed network training is in a two-stage manner: pre-compression training and post-compression fine-tuning. The detailed process and objectives are discussed as follows.

Stage one: Train a GAN with unquantized (floating-point) latent vectors for the target signal type. This step is similar to a typical GAN training procedure [155], where a generator $G(\cdot)$ and discriminator $D(\cdot)$ are adversarially trained. In addition, an encoder $E(\cdot)$ is cascaded by the generator to form an autoencoder structure, where the encoder is trained to learn mappings from the signal space to a latent space. The encoder and generator are optimized end-to-end under the following cost function:

$$\min_{E,G} \max_{D} \mathbb{E}[D(x)] - \mathbb{E}[D(G(E(x)))] + \lambda_1 \cdot \mathbb{E}[d(x, G(E(x)))], \tag{2.1}$$

where $G(\cdot)$, $E(\cdot)$, $D(\cdot)$ refer to the generator, encoder, and discriminator respectively, and $d(\cdot)$ is the similarity measure between the raw and reconstructed signals. Note that the similarity measure is different for image and speech compression and d_I and d_S are used to distinguish them in the following section respectively. The loss function is thus a weighted combination of the original GAN loss and the distortion loss.

For image compression, the similarity measure d_I is defined as a weighted combination of MSE and MS-SSIM loss [174]:

$$d_I(x, G(E(x))) = \mathcal{L}^{\text{MS-SSIM}}(x, G(z)) + \gamma \cdot \text{MSE}(x, G(z)). \tag{2.2}$$

And the $\mathcal{L}^{\text{MS-SSIM}}(x, y)$ is defined as:

$$\begin{aligned} \mathcal{L}^{\text{MS-SSIM}}(x, y) &= \frac{1}{N} \sum (1 - \text{MS-SSIM}(p)) \\ &= \frac{1}{N} \sum (1 - l_M^{\alpha}(p) \cdot \prod_{j=1}^{M} {cs_j}^{\beta_j}(p)), \end{aligned} \tag{2.3}$$

where

$$l_j(p) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{2.4}$$

$${cs_j}^{\beta_j}(p) = \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}. \tag{2.5}$$

In the above expressions, p denotes a pixel located in the reference patch x and the processed patch y, N represents the total number of pixels in x. $l_j(p)$ and $cs_j(p)$ are terms at the scale $j \in \{1, 2, ...M\}$. The means (μ) and standard deviations (σ) are computed with Gaussian filters centering at pixel p for each scale j. The standard deviation σ_G^j of each Gaussian filter corresponds to the level of the constructed Gaussian pyramid. Following [174], the parameter setting is $\gamma = 0.1$ and $\alpha = \beta_j = 1$, for $j \in \{1, 2, ...M\}$.

For speech compression, $d_S(\cdot)$ is defined as the MSE loss:

$$d_S(x, G(z)) = MSE(x, G(z)). \tag{2.6}$$

Stage two: Fine-tune the trained model with quantized latent vectors. After the first stage, signal compression is performed on the training set images through iterative back-propagation and quantization. Then the generator and discriminator models are retrained only using quantized/discretized latent vectors $\tilde{z}$ for model refinement in the quantized latent space. This procedure is formulated as follows:

$$\min_G \max_D \mathbb{E}[D(x)] - \mathbb{E}[D(G(\tilde{z}))] + \lambda_2 \cdot \mathbb{E}[d(x, G(\tilde{z}))], \tag{2.7}$$

where $\tilde{z}$ denotes the quantized latent vector generated by the proposed compression algorithm (using the first stage training result).

After the above two-stage training process, the network parameters are fixed for signal compression and decompression. The generator is viewed as a fixed mapping function from the abstract latent domain to the spatial/frequency domain for image/speech during the compression process for iterative latent space searching through back-propagation.

2.3.2 Objective Functions

Compared with prior compression methods, a noticeable advantage of this proposed compression lies in the fact that BPGAN can control the optimization direction of the optimal latent representations by applying different objective functions, so that the compression can be adjusted for various application tasks without the need of retraining. From this viewpoint, the trained generator is treated as a projector mapping the latent feature space to the target signal domain.

With a well-trained generator $G(\cdot)$, the signal compression process is expressed as an optimization problem:

$$\tilde{z} = \arg\min_{z} F(x, G(z)), \tag{2.8}$$

where $\tilde{z}$ is the optimal lower-dimensional representation of the original signal x and $F(\cdot)$ is the objective function. Under a selected objective, this work proposes to solve this problem with an iterative back-propagation process based on the gradient ∇F_z where z is initialized by the encoder and regularized by the reconstruction quality criterion.

In this work, similarity measures between the reconstructed signals and the target signals are used as the optimization objectives. The conventional choices of similarity metrics are typically L1 or L2 distances; however, they are not perfectly suited for the properties of some signals and are prone to poor performance in real-world applications. For instance, in image compression tasks, minimizing MSE would often produce blurry images and depress perceptual quality. Therefore, it is important to select/design a proper objective function that considers signal properties and back-end quality metrics.

According to the experiments, discriminator loss gives satisfactory results in the image compression task and feature loss is advantageous for the speech compression task.

1**Discriminator Loss for Image Compression:** After the adversarial training procedure for the generator and discriminator, the resulting discriminator can be regarded as a data-driven quality indicator function evaluated by a deep neural network. Therefore, the discriminator loss is combined in the compression objective function, which depicts the confidence of the discriminator in the quality of the (de)compressed signal. By introducing this loss to the compression back-propagation process, BP-GAN takes advantage of the complex quality measures evaluated by the discriminator for better reconstruction quality (at the cost of increased computational overhead).

The total loss function used in image compression can be defined as:

$$F(x, G(z)) = -D(G(z))) + \lambda_3 \cdot d_I(x, G(z)), \tag{2.9}$$

where the first term gives the discriminator objective, and the second term is the combination of MS-SSIM loss and MSE between the reconstructed image and the original image.

Feature Loss for Speech Compression: Inspired by the remarkable success of feature loss in image stylization and synthesis [61, 155], BPGAN adopts and designs the feature loss in the presented framework to improve the performance in speech signal compression. The core idea of feature loss is to apply a pre-trained classification network to the original signal and compressed signal, and compute internal activations in the network to be compared. This loss is defined in terms of the dissimilarity between those internal activations, and it is shown to yield state-of-the-art performance for various tasks without the need for prior expert knowledge or added complexity.

For better human sound perceptual quality and better back-end recognition performance, the proposed method adopts the speech feature loss to represent the content difference. Similar to image feature loss, speech feature loss measures the feature map (activation) distance between generated and real speech using a speech recognition neural network. For feature loss evaluation, a VGG-BLSTM [83] network is employed and trained for a phoneme recognition task on the TIMIT dataset using joint CTC-attention loss [67]. For faster inference speed of applying feature loss, the scheme only extracts feature maps of the convolution layers rather than the recurrent networks. The inputs for the VGG-BLSTM network are 40-dimensional log-mel spectrograms, therefore the loss is calculated in the spectral domain. The total objective function used in speech compression is defined as:

$$F(x, G(z)) = \mathcal{L}^{\mathbf{feat}}(x, G(z)) + \lambda_4 \cdot MSE(x, G(z)) \tag{2.10}$$

where $\mathcal{L}^{\mathbf{feat}}$ denotes the speech feature loss.

2.3.3 Quantization and Entropy Coding

To reduce the bitrate of the compressed signal, it is necessary to incorporate quantization in the compression strategy, which projects the original unquantized signal onto a discrete set of numeric values. There exists a fundamental tradeoff between the number of quantization levels (bitrate) and signal quality: reducing the number of quantization levels negatively affects the compression quality while reducing the required bitrate. This work adopts a non-uniform quantization strategy which requires transmitting/storing the non-uniform quantization center points. Non-uniform quantization in general allows for achieving higher signal quality given a fixed bitrate. However, the proper quantization centers must be chosen ahead of signal compression, which poses an additional challenge for BPGAN. To accommodate non-uniform quantization, the ADMM back-propagation algorithm is adopted, where the quantization problem is formulated as a constrained optimization task with two sub-problems. To quantify the performance of the ADMM-based quantization, the comparison with alternative algorithms such as the direct quantization method is evaluated.

ADMM Quantization: The latent vector search problem in the non-uniform quantized space could be formulated as an optimization problem with a quantization constraint solved by ADMM [17]. That is, given $G(\cdot)$, x, and $F(x, G(z))$, the problem of finding an optimal quantized latent vector z is formulated as follows:

$$\arg\min_{z,u} F(x, G(z) + I(u \in S) \quad s.t. \quad u = z \tag{2.11}$$

where S is a non-convex set whose elements are (non-uniformly) quantized vectors, u is an auxiliary variable, and $I(\cdot)$ is an indicator function. This optimization problem with non-convex constraints is difficult to solve directly, therefore it can be rewritten as an augmented Lagrangian function and apply ADMM to solve it.

The augmented Lagrangian of the above optimization problem is given by:

$$L(z, u, \eta, \mu) = F(x, G(z)) + I(u \in S) + \frac{\mu}{2}(\|z - u + \eta)\|_2^2 - \|\eta\|_2^2). \tag{2.12}$$

ADMM is designed to minimize $L(z, u, \eta, \mu)$ by updating variables z, u, η alternatively in every iteration to approach the optimal values that minimize the quantization error. The ADMM updating procedure for $k = 0, 1, 2, \ldots$ is given by:

$$z_{k+1} = \arg\min_{z} F(x, G(z)) + \frac{\mu}{2}\|z - u_k + \eta_k\|_2^2 \tag{2.13}$$

$$u_{k+1} = \arg\min_{u} I(u \in S) + \frac{\mu}{2}\|z_{k+1} - u + \eta_k\|_2^2 \tag{2.14}$$

$$\eta_{k+1} = \eta_k + z_{k+1} - u_{k+1}. \tag{2.15}$$

The optimization criteria for the compressed signal z described in Equation 2.12 is additionally regularized with an L_2 term. This can be easily translated to an updating rule with stochastic gradient descent (SGD) or Adam [68] optimizer.

For the updating objective of u depicted in the Equation 2.14, the analytical solution is $u_{k+1} = Q(z_{k+1} + \eta_k)$ where $Q(\cdot)$ is a non-uniform quantization function that directly projects into the set of quantized set S. The non-uniform quantization centers of S are obtained by K-means clustering based on the distribution of the unquantized value of the latent vectors corresponding to the entire training dataset. The quantization centers are obtained during the training and unchanged for each signal reconstruction to avoid separately transmitting the quantization codebook for each target signal. To ensure that the latent z is quantized after the alternating operations, this method projects the latent vector to the discrete set S directly as the final step of ADMM quantization.

Algorithm 2 ADMM-based BPGAN Compression Algorithm

```
Require: well trained generator $G(\cdot)$, encoder $E(\cdot)$
    pre-defined quantization function $Q(\cdot)$
    signal to be compressed $x$
    objective function $F(\cdot)$
    set hyperparameters $\mu, \alpha$
Ensure: quantized latent vector $\tilde{z}$
latent vector initialization $z_0 = E(x)$
auxiliary variables initialization: $u_0 = Q(z_0)$, $\eta_0 = 0$
repeat
    calculate the optimization objective:
    $O(z_k) = F(x, G(z_k)) + \frac{\mu}{2}\|z_k - u_k + \eta_k\|_2^2$
    gradient descent: $z_{k+1} = z_k - \alpha \cdot \nabla O(z)$
    $u$ update: $u_{k+1} = Q(z_{k+1} + \eta_k)$
    dual ascent: $\eta_{k+1} = \eta_k + z_{k+1} - u_{k+1}$
until convergence to $\tilde{z}$ or maximum iteration number satisfied
apply Huffman Coding to $\tilde{z}$
```

With ADMM quantization being specified, the proposed algorithm is refined as described in Algorithm 2. The idea behind ADMM back-propagation is that this algorithm searches for the optimal point or local optimal point in the discretized latent space to preserve signal fidelity after reconstruction.

Direct Quantization: A baseline method to be compared with the proposed ADMM-based approach is direct quantization, where an unquantized latent vector is first obtained by back-propagation and then each element is projected to the nearest numerical value in the quantized set. Then the iteration continues by updating the latent by back-propagation to a new unquantized vector. Applying this direct quantization one-time only to the final latent vector will degrade the fidelity of the reconstructed signal. Therefore, quantization and back-propagation steps are operated in an alternating order.

Entropy Coding: To further reduce the code size, the proposed scheme applies Huffman coding to the quantized latent vectors. The Huffman coding is a widely used prefix-free entropy coding method [147] that assigns different lengths of codes to encoded symbols depending on the relative frequency of the corresponding symbols. By assigning shorter codes to symbols with frequent occurrence, Huffman coding reduces the bitrate without losing any information.

2.4 Experimental Setup

The BPGAN compression is a unified signal compression method for different types of signals (*e.g.*, image and speech). To apply the unified approach, preprocessing, and post-processing are crucial for this algorithm during the training phase and the compression process. In this section, a detailed explanation of how to apply image and speech signals into the BPGAN framework is introduced and also includes the hyperparameter settings during training.

2.4.1 Latent Initialization with Trained Encoder

Finding sufficiently good latent vectors by back-propagation with random initialization could require many iteration steps and might consume excessive computation resources. To accelerate this process, this work proposes to train an encoder that learns mappings from target signals to (optimal) latent vectors. The output latent vectors of the encoder often cannot be exactly the same as the target optimal vectors but are expected to approximate them so that they provide good initialization for iterative back-propagation. The encoder is trained with the generator and discriminator in the training stage and it produces unquantized latent vectors to initiate the optimization process.

2.4.2 Image Compression Setup

The Adam optimizer is adopted for training and sets the batch size to 128. The network is trained for 100 epochs in total, with a fixed learning rate of 0.0005 in the first 50 epochs and a linear decayed learning rate in the following 50 epochs. To better evaluate the performance of the proposed algorithm on different datasets, images in the training dataset are resized to a pre-defined resolution (768×512) in RGB format.

2.4.3 Speech Compression Setup

Training Settings: The Adam optimizer and batch size follow the image compression setup. The framework is trained for 200 epochs in total, with a fixed learning rate of 0.0002 in the first 100 epochs and a linear decayed learning rate in the following 100 epochs.

Speech Pre-processing: For BPGAN compression, speech signals sampled in the time domain are converted to spectral domain representations using short-time Fourier transform (STFT) with 128-sample stride and 512-sample frame size, resulting in 75% frame overlap. BPGAN only uses the magnitude information of the STFT output (*i.e.*, spectrogram) for signal compression. The experiments show that a higher overlap ratio can lead to better audio quality with a higher bit rate, which supports the claim of [97]. To further reduce the bit rate of the compressed signal, spectrograms are transformed into mel spectrograms with 128 mel frequency bins for each frame. This algorithm collects 128 frame mel frequency bins to make up one 128×128 mel spectrogram, which corresponds to one second of speech audio at a sample rate of 16 kHz.

For human sound perception-oriented compression, the log-magnitude of mel-spectrograms is calculated and normalized for compression [97]. BPGAN first normalizes the magnitudes of mel-spectrograms to have a maximum value of 1, limiting the log-magnitude value to the range of $[-\infty, 0]$. Then the proposed method limits the dynamic range of the log magnitude into $[-r, 0]$ by truncating by $-r$. Finally, the compressed signal is scaled and shifted back to $[-1, 1]$ for the generator. In the experiments, r is set as 8.

Speech Post-processing for Phase Recovery: Inverse STFT requires magnitude and phase information to recover the time domain speech audio signals. However, phase information is removed in the pre-processing and the following back-propagated compression. Therefore the estimation of phase information given magnitudes is necessary, which can be regarded as an additional decoding process.

In this work, the Phase Gradient Heap Integration (PGHI) [116] method is employed to recover the phase from the amplitude-only signal. This technique relies on magnitude-phase gradient integration relations for phase reconstruction. PGHI often significantly outperforms Griffin-Lim [114] and shows superior robustness because it avoids integrating through phase instability areas.

2.4.4 Datasets and Metrics for Image Compression

Open Images Dataset V5: The proposed image compression framework is trained on Open Images Dataset V5, which contains 9 million images belonging to 600 categories. The images in this dataset are diverse and contain complex scenes with 8 objects per image on average. Comprising an abundant amount of various contents, it helps the generator model to learn the diversity of real-world objects. As the GAN network uses the same sized images for its training input, images are rescaled to 768 $\times$ 512 pixels prior to training.

Kodak Dataset: The Kodak dataset comprises 25 landscape or portrait RGB images of size 768×512 pixels. It is widely used in image compression tasks and serves as a generic benchmark data collection. The Kodak dataset is used to evaluate PSNR and MS-SSIM of (de)compressed images using the generator trained with Open Images Dataset V5. The Kodak dataset is not used during the training of the generator, discriminator, and encoder.

ImageNet Dataset: The ImageNet 2012 classification dataset consists of images belonging to 1,000 classes. It is split into three sets: training (1.3M images), validation (50K images), and testing (100K images with held-out class labels). ImageNet dataset is used to quantify the image quality by performing ImageNet classification on (de)compressed images using a VGG-16 network pre-trained with raw images.

Evaluation Metrics: The quantitative performance of the proposed image compression method is primarily measured by PSNR and MS-SSIM on the Kodak dataset which is a widely used generic benchmark data collection to evaluate image compression. A significant drawback of Kodak is the limited amount of data which may introduce biases to the quality measurement. To alleviate this issue, the image compression method is evaluated from the machine perspective by comparing the classification accuracy on ImageNet using the original test dataset and the dataset decompressed by BPGAN. For subjective image quality evaluation, the sample reconstructed images are presented in Figure 2.2 and Figure 2.5.

2.4.5 Datasets and Metrics for Speech Compression

TIMIT Dataset: The speech audio experiments of this chapter are performed on the TIMIT dataset, which contains a total of 6,300 sentences spoken by 630 speakers from 8 major dialect regions of the United States at a sample rate of 16 kHz. The training and testing subsets are mutually exclusive.

PESQ Metrics: The perceptual speech quality is evaluated by PESQ [125], which is an objective metric designed to predict the mean opinion score (MOS) for speech quality by an algorithm. It is adopted by ITU-T as a recommended standard metric. PESQ values range from -0.5 to 4.5, of which larger is better. PESQ is proposed for predicting the perceptual quality with acceptable accuracy for waveform-based compression and CELP codec for bit rates larger than 4 kilo-bits per second (kbps).

Subjective Evaluation: The subjective quality of the (de)compressed speech is quantified with an experiment involving 20 users guided by Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) [148]. The users are asked to listen to 5 sets of audio clips. Each set of clips is presented with a labeled reference audio, a set of test samples, a hidden version of the reference, and one anchor. Based on their perceptual evaluation, the users are asked to score these test samples from 0 to 100.

Phoneme Recognition Evaluation: In addition, this work performs phoneme recognition tests to measure the phoneme error rate (PER, the lower the better). Phoneme recognition was performed on (de)compressed speech signal using the combination of SGMM and Dans deep neural networks in Kaldi [115], and also using MLP and LSTM networks [119]. These network models take MFCC as the input and are trained with raw (uncompressed) audio data.

2.5 Experimental Results

In this section, the proposed approach is compared with conventional and learning-based image/speech compression methods from the perspective of various evaluation metrics. To quantify the impact of various techniques incorporated in this scheme, the ablation studies on ADMM quantization, latent representation optimization objectives, and latent initialization with a trained encoder are involved in the following subsections.

2.5.1 Image Compression Results

This section compares the proposed method with conventional codecs such as BPG [14] and JPEG, and an existing GAN-based method [5]. Notably for extremely low-bpp (bit per pixel) regimes, PSNR and MS-SSIM are usually inadequate to reflect the perceptual image fidelity. Thus, the classification error rate metric is applied to quantify the image quality from the machine's perspective. To fit the input size of

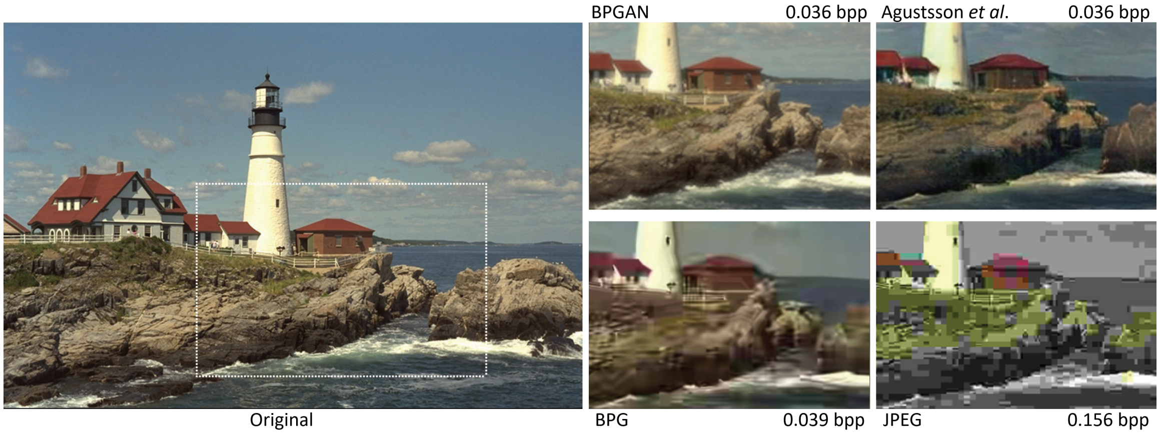

Figure 2.2: Visualization of decompressed image from Kodak dataset. Compared with other existing methods, the lossy reconstruction of the proposed method exhibits higher fidelity with a similar or lower bitrate.

the ImageNet dataset, the generator is modified and trained to match the dimension of ImageNet images (256×256 in RGB format).

The experimental results for image compression are shown in Figure 2.2 and summarized in Table 2.1. PSNR and MS-SSIM evaluations are performed on the Kodak dataset with 768×512 resolution images in RGB format (24 bit per pixel, bpp). A compressed image of 0.286 bpp is obtained by using a latent vector z of 20,000 dimensions and non-uniform quantization with 256 levels for each element before Huffman coding.

Table 2.1: Image compression performance comparison.

Method	**Bitrate** (bpp)	**PSNR**	**MS-SSIM**	**ImageNet Top-1 error rate %**	**ImageNet Top-5 error rate %**
Original	24	-	-	23.7	6.8
BPGAN	**0.286**	**32.9**	**0.968**	**23.7**	**6.8**
Agustsson et al. [5]	0.305	28.2	0.922	26.0	7.9
JPEG	0.306	26.9	0.864	42.5	16.6
BPG	0.298	32.3	0.961	25.8	7.4

In Table 2.1, this method adopts an extremely high compression ratio (84×) to show the performance comparison between different methods. Results measured by the classification accuracy metric (error rate) are listed in the last two columns.

Table 2.1 indicates that the proposed compression scheme produces higher quality images measured by PSNR and MS-SSIM with a compressed rate of ≈ 0.3 bpp compared to other existing compression methods. This approach manages to maintain the same classification accuracy before and after the compression of the ImageNet test images. The result shows that BPGAN compression, unlike other approaches, successfully preserves the perceptual features of the target images after compression without significantly affecting deep visual learning.

The compressed images generated by BPAGN have higher subjective quality for the same/similar bitrate compared to other methods. In Figure 2.2, the compressed image comparison of BPG, JPEG, and [5] are visualized with a high compression ratio (0.036 bpp) using *Kodim21* in the Kodak dataset. In the region bounded by the white box, the proposed approach exhibits the best quality in terms of preserving the color and shape of objects at a very low bit per pixel (bpp) regime. With severe block artifacts, JPEG fails to compress this image at the target bpp of 0.036. BPG handles the high-frequency contents better (*e.g.*, the fence in the image), however, it loses some of the features of the rocks and waves. The image generated by the method described in [5] provides smoother features, especially in regions containing clouds. It is worth noting that a GAN-based method [5] sometimes 'creates' realistic details in the compressed image that are non-existent in the original. The actual details are better preserved in the BPGAN compression output because of the joint loss (discriminator score, MSE, and MS-SSIM) used to find the optimal compressed latent in the compression stage.

2.5.2 Speech Compression Results

The speech compression evaluation results are summarized in Table 2.2 and Figure 2.3. In this experiment, the original speech is sampled at 16k samples per second with 16-bit per sample, thus the original bit rate is 256 kbps. The compressed speech rate

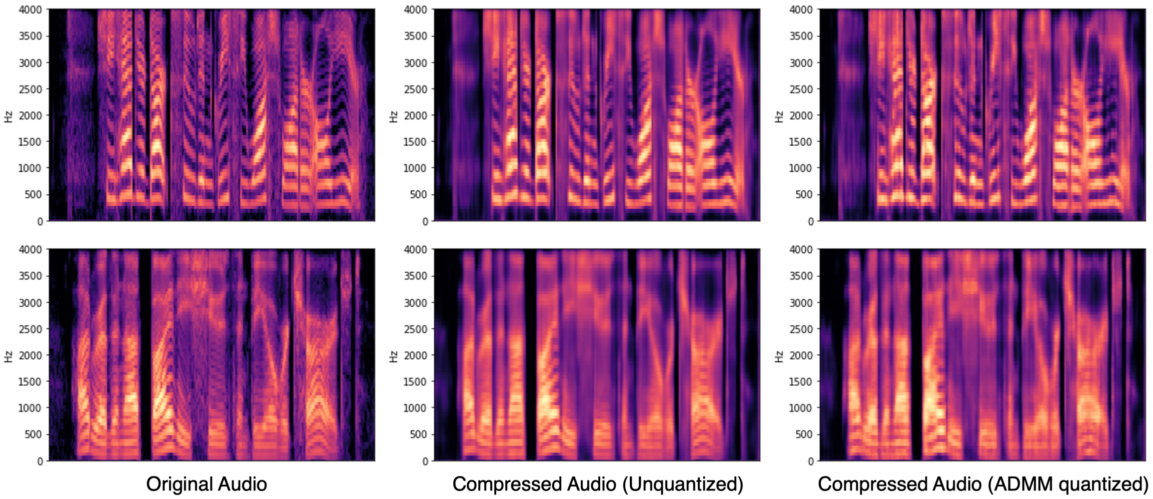

Figure 2.3: Compressed audio on Timit dataset in the form of spectrogram. With unquantized latent vectors, the reconstructed audio (middle) preserves most of the detailed information. ADMM quantization (right) shows negligible quality degradation after quantization.

of 2 kbps is achieved by using a latent vector z size of 512 and 16 non-uniform quantization levels for each element. While the proposed BPGAN-based compression provides the lowest data rate of 2 kbps, it exhibits a better MUSHRA subjective quality score than the other methods with higher data rates except for Opus at 9 kbps (4.5× higher data rate than ours). It is worth pointing out that while the PESQ scores are similar among multiple methods, they do not accurately predict the (subjective) quality of the speech when the bit rate is low. The PER measured by phoneme recognition tests indicates that the error rate for the proposed audio compression is superior (less PER degradation) to other compression methods while providing the lowest data rate.

Table 2.2: Speech compression performance comparison.

Method	**Bitrate (bps)**	**PESQ**	**MUSHRA**	**Kaldi Percentage %**	**MLP Percentage %**	**LSTM Percentage %**
Original	256k	4.50	95.0	18.7	18.6	15.4
BPGAN	**2k**	**3.25**	**64.1**	**20.9**	**20.8**	**18.6**
CELP	4k	2.54	32.0	28.2	27.6	27.3
CELP	8k	3.39	59.4	23.0	23.6	21.2
Opus	9k	3.47	79.3	22.7	23.7	21.2
AMR-WB	6.6k	3.36	58.9	22.6	23.6	22.3

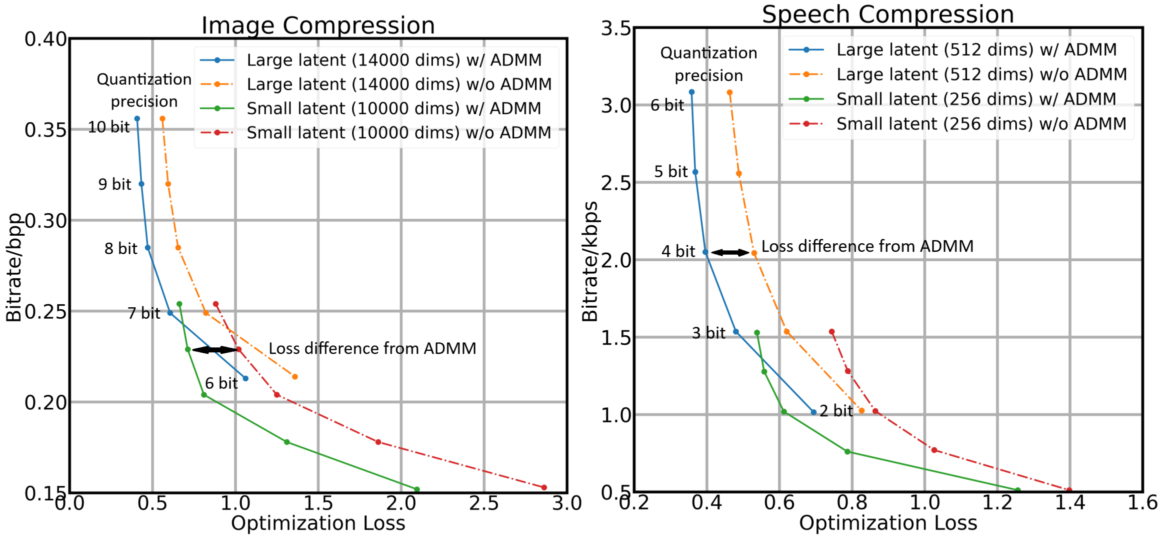

Figure 2.4: Image and audio compression parameter sensitivity evaluation: rate-distortion tradeoff is obtained by adjusting the latent vector size and number of quantization levels. Compared with direct quantization, ADMM leads to substantially lower optimization loss under the same latent dimension and quantization level.

The experimental results in Table 2.2 show that the proposed compression method significantly outperforms traditional codecs in both objective and subjective evaluations. For phoneme recognition tasks, this method demonstrates a clear advantage for all inference models even with at least 2× smaller bit rate. This is because this method compresses audio in the spectral domain, preserving more context information with optimized feature loss through BP unlike other methods working in the time domain without an explicit objective function.

2.5.3 Ablation Studies

Quantization Analysis: Figure 2.4 visualizes the tradeoff between bitrate (without Huffman coding, lower is better) and optimization (quality) loss. BPGAN offers a wide rate-distortion tradeoff space. The gain of ADMM-based non-uniform quantization is shown in the same figure. In all investigated cases, ADMM quantization manages to reduce the quality loss compared to direct quantization. One can notice that along the pareto-optimal line, the combination of the optimal latent vector

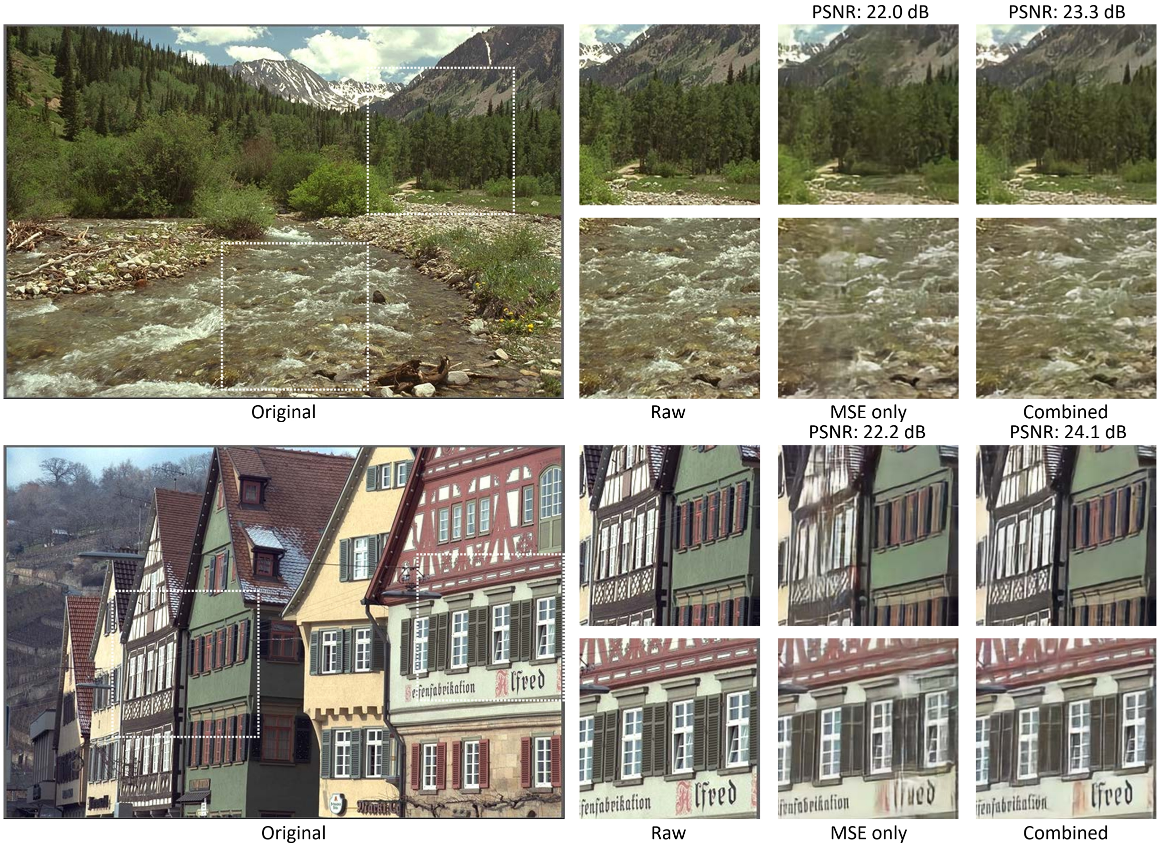

Figure 2.5: Visualization of decompressed images optimized according to MSE and the joint objective that combines MSE with discriminator output.

dimension and the number of quantization levels per element changes for different bitrate targets.

Effect of Different Latent Optimization Objectives: For image compression, this work investigates the impact of using the combined objective function that includes the quality measure provided by the discriminator compared to the MSE-only objective function. The results are visualized in Figure 2.5. Images on the right are substantially better than the middle ones with higher PSNR and better visual quality. In both cases (top and bottom), using the joint objective facilitates sharper and better-defined details in the reconstructed images. The MSE-only objective often induces blurry effects in the decompressed images leading to the loss of details. The observation indicates that incorporating perceptual objectives with the per-pixel similarity measure enables the search of latent representations that can capture more

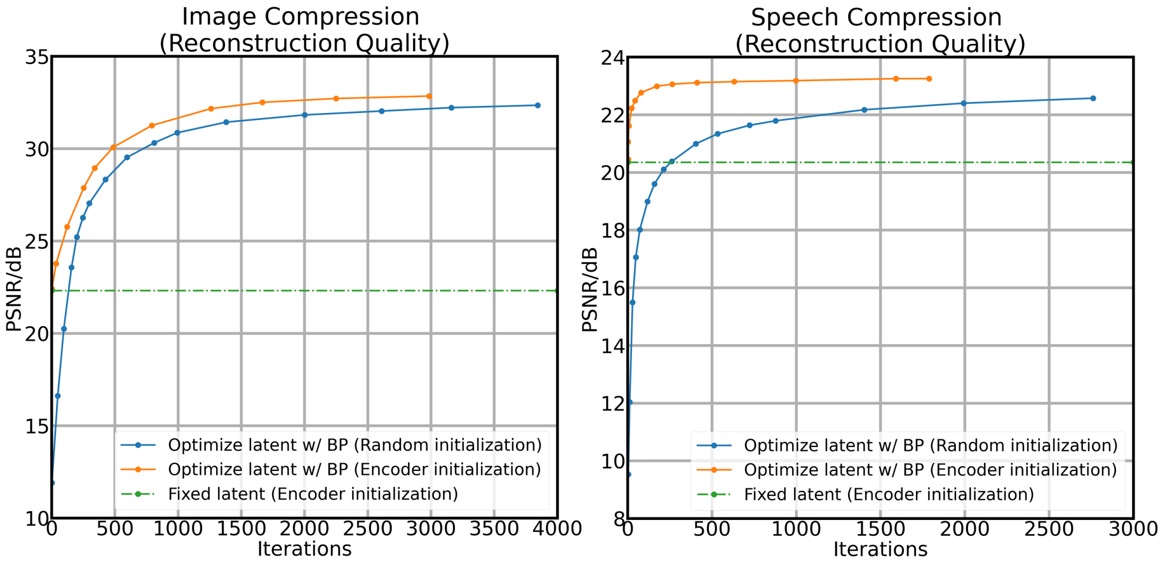

Figure 2.6: Signal quality measured by PSNR vs. number of back-propagation iterations. Left: image compression, right: speech compression.

accurate features for visually pleasing reconstructions.

To study the influence of speech feature loss during the compression process, an ablation experiment on different back-propagation objectives (Equation 2.10) is evaluated. The decompressed speech quality/performance in PESQ and PER for the same 2 kbps rate with feature-loss-only, MSE-loss-only, and a combination of MSE and feature losses are tested. Table 2.3 shows that adding feature loss can significantly improve the performance on speech recognition tasks compared with MSE-loss-only, and the combination loss achieves a good tradeoff between PESQ and phoneme recognition error rate. It is worth noting that the feature loss is calculated using VGG-BLSTM neural network [83] that does not share the same structure in the phoneme recognition task evaluation networks, implying that the gain from the feature loss is not because of overfitting.

Latent Initialization with Trained Encoder: This work evaluates the effectiveness of using an encoder for latent initialization by showing the compressed signal quality vs. the number of back-propagation iterations during image/speech compression tasks in Figure 2.6. Initializing latent vectors with an encoder before back-

Table 2.3: Speech compression results with various loss functions during optimal latent vector search back-propagation.

Method	PESQ	Kaldi PER (%)	MLP PER (%)	LSTM PER (%)
Original	4.50	18.7	18.6	15.4
Feature Loss	2.92	21.0	20.6	18.1
MSE Loss	3.29	21.5	21.7	19.9
Combined Loss	**3.25**	**20.9**	**20.8**	**18.6**

propagation (BP) can effectively reduce the number of iterations needed to find a good latent representation and achieve better quality. The same figure also illustrates that the proposed iterative back-propagation brings significant gain compared to the one-shot encoder output.

2.6 Conclusion

This chapter introduces the BPGAN compression scheme, a GAN-based unified signal compression method that produces compressed data with high fidelity at low bitrate by searching the optimal latent vector through iterative back-propagation. To improve the compression ratio, the proposed method applies ADMM with non-uniform quantization to search for an optimal latent representation of the signal. The proposed method first trains the generator network model in a GAN setup, and then it iteratively updates and discretizes the optimal latent code through the pre-trained generator for each signal input for compression. Experiment results demonstrate that BPGAN-compressed signal exhibits significantly lower data rate and/or better signal quality compared to other methods evaluated with various metrics including neural network based image classification and speech phoneme recognition.

CHAPTER III

Deep Learning in Latent Space for Video Prediction and Compression

3.1 Introduction

[1] Video data transmission occupies the majority of the internet data traffic nowadays. With the trend of extensive mobile device usage worldwide, video data streaming is extensively used for productivity tools and entertainment platforms that assist people's work and life in various aspects. On top of the ubiquitous video engagement, superior video quality standards such as 4k UHD, and VR 360 became more widely available, which makes high-performance video compression even more critical. Traditional video coding standards such as MPEG, AVC/H.264 [160], HEVC/H.265 [136], and VP9 [108] have achieved impressive performance on video compression tasks. However, as their primary applications are human perception driven, those hand-crafted codecs are likely suboptimal for machine-related tasks such as deep learning-based video analytics.

In recent years, a growing trend of employing DNNs for image compression tasks has been witnessed. Prior works [143, 11, 105] have provided a theoretical basis for the application of deep autoencoders (AEs) on image codecs that attempt to optimize the

[1] The content of this chapter is based on [85].

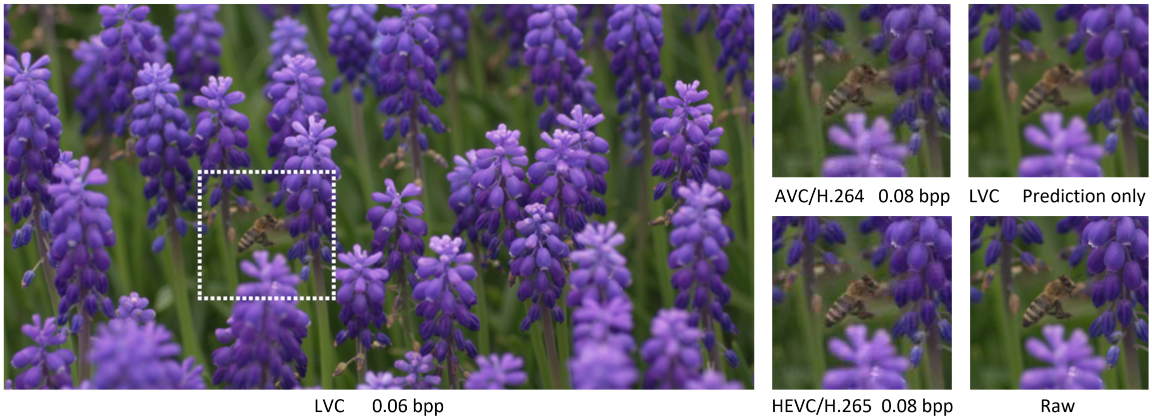

Figure 3.1: Reconstructed frame with the conventional codecs (H.264, H.265) and LVC approach. Information and details are well preserved in the frame generated from a purely prediction-based latent representation (top right).

rate-distortion tradeoff, and they have shown the feasibility of latent representation as a format of compressed signal.

While image compression reduces the redundancy only in the spatial domain, video compression exploits the temporal correlation among consecutive frames as well. Using learned video prediction to substitute traditional block-based motion prediction/estimation methods has become a critical part of deep learning based video compression. Related recent works [32, 77, 89] address the uncertainties of real-world videos with stochastic video prediction networks using autoencoders and/or GAN structures in recurrent settings. Learned video compression is a relatively recent topic. Early works [21, 162] either directly interpolate the key-frames or emulate the functional units in hand-crafted codecs with neural networks. Later proposed deep neural video codecs [94, 90, 48, 33, 123, 4, 165, 42, 93, 53] mainly target on learning data-driven algorithms that take advantage of the end-to-end trainability of DNNs. Most of them [90, 48, 33, 123, 4, 165, 42] adopt autoencoder style structures that encode frame and residual representations in latent space.

This chapter presents a novel end-to-end deep learning video codec, LVC, that benefits from video prediction in latent space. The proposed method obtains the compressed frames in latent space by searching for the optimal latent representations

[84], and then it learns temporal correlation within the latent space sequential data under a recurrent network setting. As opposed to previous approaches, the training and inference processes of the proposed prediction network are entirely performed in the latent domain. The presented video coding method shares the same predictor between the sender and receiver, and only transmits (stores) the quantized and entropy-coded prediction error (residual). The residual corrected latent frames are fed back to the prediction network for progressive estimation on consecutive latent representations of the data sequence.

Video compression evaluation results validate that this technique achieves superior performance compared to the state-of-the-art video codes. The proposed prediction method demonstrates its application in abnormal event detection, which is triggered when the prediction error exceeds a predefined threshold that represents a normal event. Anomaly detection evaluation results confirm the superiority/competitiveness of the proposed method compared to recent algorithms specifically designed for that task.

3.2 Related Work

3.2.1 Learning-based Image Compression

There has been extensive study on applying DNNs to image compression tasks. Most approaches typically seek compression gain from translating images to lower-dimensional representations through either recurrent neural networks (RNNs) [142, 8, 62, 143] or autoencoder style networks [11, 141, 12, 100, 105]. Recent approaches use GAN-based structures for image compression [127, 5, 84] aiming to enhance the subjective quality of image reconstructions from deep encoder-decoder pairs and take advantage of the qualification feedback provided by a discriminator. These approaches often target optimizing distortion indicators such as mean squared error

(MSE), PSNR, and MS-SSIM between the raw and reconstructed image, or the hybrid objective function including the perceptual loss. This work adopts a GAN-based structure to search for the optimal latent vectors that minimize distortion via back-propagation through a pre-trained generator (decoder).

3.2.2 Learning-based Video Compression

Video coding benefits from exploiting the temporal correlation between subsequent frames. Similar to the conventional codecs, learned video compression leverages the temporal correlation through inter-frame prediction. Chen *et al.* [21] first predicts a frame and then encodes the residual (error between the prediction and actual) with a CNN. This approach shares great similarities with block-based codecs. Arguably, however, DNNs are less efficient to learn from small image blocks. To overcome that issue, Wu *et al.* [162] propose a codec that captures temporal redundancy through hierarchical interpolation between keyframes. The method uses a non-learning based optical flow to generate motion information, and it is not jointly optimized with the rest of the model.

Lu *et al.* [94] construct a DNN-based video compression pipeline close to the conventional codecs and optimize compression rate in conjunction with distortion. Lombardo *et al.* [90] presents a learning-based video codec that performs end-to-end optimization on rate-distortion tradeoff, quantization, and entropy coding. The framework is built with sequential variational autoencoder (VAE) where the encoded global state based prediction is used to tackle the temporal redundancy. Similarly, Rippel *et al.* [123] proposes to represent all prior memory as a generic and learnable state that will continuously be updated during its propagation. The flow-residual information between two consecutive frames is generated from the state representation. Habibian *et al.* [48] use a 3D spatiotemporal autoencoder network that temporally decorrelates the latent vectors. Based on the encoder-decoder pair proposed in [11] for image compression, Djelouah *et al.* [33] encode displacement and blending coefficients into latent space representations. To address the failure cases typically observed in the flow-residual paradigm, Agustsson *et al.* [4] propose a scale-space flow that trilinearly warps the frame stack constructed by the previous frame as well as its variations obtained by applying different levels of Gaussian blurring. Following the hierarchical prediction approach, Yang *et al.* [165] design a framework that encodes video frames with different quality levels, and refines the coarsely predicted frames by leveraging the temporal correlation contained in the high quality frames. In this work, unlike the prior works mentioned above, spatial redundancy is primarily exploited by finding the optimal low dimensional latent vectors to represent each video frame. Then LVC performs temporal predictions on successive frames in latent space. The residuals between the directly compressed frames (*i.e.*, latent vectors) and the predicted ones are quantized, entropy coded, and transmitted to the receiver.

3.2.3 Video Prediction and Motion Compensation

The study on deep neural video prediction has led to a number of design choices. Early works are usually devoted to predicting small frame patches. To reduce the blurry reconstruction effect, Mathieu *et al.* [98] train a multi-scale network in an adversarial setting. Whereas Finn *et al.* [37] present an LSTM-based network to learn the motion dynamics and to construct motion information with the content mask to form a predicted frame. Other approaches such as [7, 32, 77] propose variational methods to address the embedded stochasticity in real-world videos. Motion compensated prediction is an essential sub-task of video compression. Chen *et al.* [21] present a DNN-based implementation that resembles block motion estimation in traditional codecs while others [94, 48] incorporate an optical flow encoder network into the compression system. Unlike prior approaches, the proposed framework employs ConvLSTM based frame prediction in latent space for motion compensation. With a well-learned prediction network, the experimental results demonstrate that very sparse residuals can be obtained in latent space to produce extremely compressed video sequences.

3.2.4 Anomaly in the Scene Detection

Anomaly detection can be treated as an application of video prediction. A network structure in [27] includes cascaded convolutional LSTM networks in the autoencoder to learn the spatio-temporal features of the video frames. Liu *et. al* [89] is the first to introduce a video prediction framework adversarially trained under a temporal constraint for anomaly detection. Park *et. al* [112] proposes to further enhance the performance by adopting a memory module to record representative normal patterns. Different from these works, this approach targets predicting the next frame in the latent domain. This work demonstrates that the video representation learned from latent space temporal redundancy can be adopted to perform reliable abnormal event detection.

3.3 Method

3.3.1 Video Compression Framework

Figure 3.2 depicts the proposed video compression framework. The flow first encodes each frame to an optimal latent representation using the technique in [84]. This image compression technique searches for an optimal latent representation through the frame-by-frame back-propagation using a pre-trained generator network. The LVC scheme trains the generator (which serves as the decoder in the proposed video coding framework) such that it can reconstruct a close-to-original frame from a latent representation. Once the optimal latent representation is produced for each frame, the end-to-end video compression framework learns temporal correlation among the

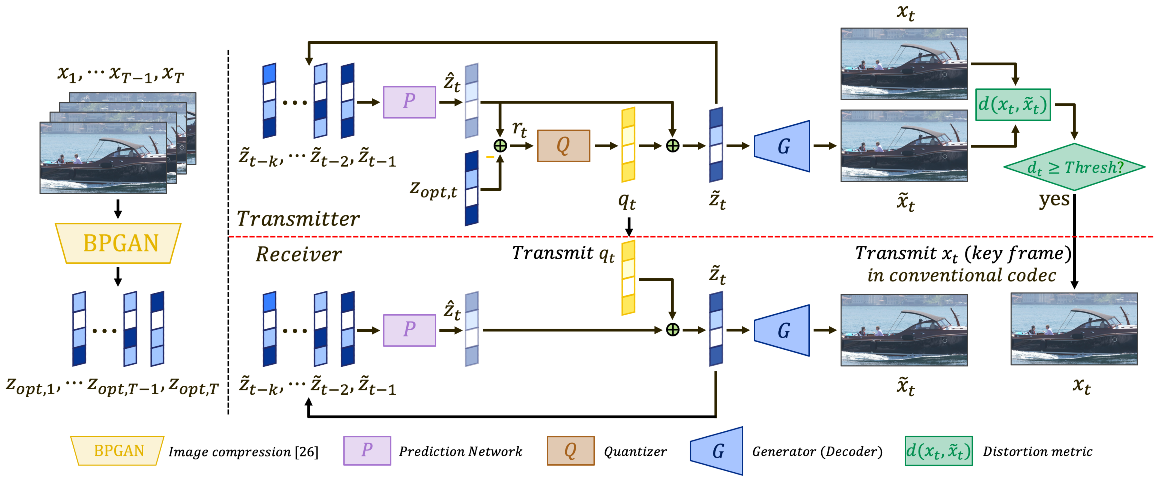

Figure 3.2: The operational flow of the proposed codec. Each frame of the video sequence is first individually compressed to a latent representation by the method in [84]. The predicted next-frame latent (output of the prediction network) is conditioned on the former reconstructed latent representations.

latent space representations of consecutive video frames. To achieve this goal, this method predicts the next frame's latent representation based on the sequence of latent of previous frames using a ConvLSTM. The prediction network takes the optimal latent vectors of each frame as the input and it is trained to predict the latent vector for the next frame as close as possible to the actual one. The element-wise difference between the predicted and actual latent is stored as the residual. Given a successfully trained latent space predictor, the residual is sparse with low entropy. Hence LVC attains the inter-frame compression gain from prediction on top of the intra-frame compression of compact latent representations.

To further reduce the video code size, the proposed algorithm encodes the residual with quantization and entropy coding. A desired compression rate is controlled by the size of the latent dimension in the image compression stage as well as the number of quantization levels used in the residual encoding. The quantized and entropy-coded residuals are sent from the transmitter to the receiver as the compressed representation of the video.

The reconstructed latent is obtained by adding the compressed residual to the predicted latent. The transmitter and the receiver share the same prediction network, which produces the identical reconstructed latent for the next frame using the previously reconstructed latent frames. At the beginning stage of the proposed video compression flow, the predictor on both sides is initialized with the latent vectors of several *initial frames* ($z_{\text{opt},1:k}$, with $k = 6$ in the experiment) that are generated without prediction. This ensures the prediction for successive latents on both sides starts with the same recurrent state. Using the same generator (decoder) adopted in the image compression stage, the reconstructed latent ($\tilde{z}_t$) is translated to the spatial domain video frame, x_t.

The image compression problem can be formulated as a joint model of a raw image x and its discrete latent representation z with θ representing model parameters:

$$p_\theta(x, z) = p_\theta(z) \cdot p_\theta(x|z) \tag{3.1}$$

In the above formula, $p_\theta(x|z)$ is the prior model, and $p_\theta(z)$ is the likelihood. Under the scheme of video compression, the proposed ConvLSTM prediction network exploits temporal correlation, resulting in a likelihood model given the former latent representations in the sequence. Therefore, the prior model and the likelihood expression can be redefined as

$$p_\theta(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t|z_{<t}) \cdot p_\theta(x_t|z_t) \tag{3.2}$$

where t is the time index for a frame. The following sections address the main functional units of the proposed framework.

3.3.2 Video Prediction

The adopted image compression method [84] provides an optimal lower-dimensional representation z_{opt} in latent space for each image by minimizing a distortion function.

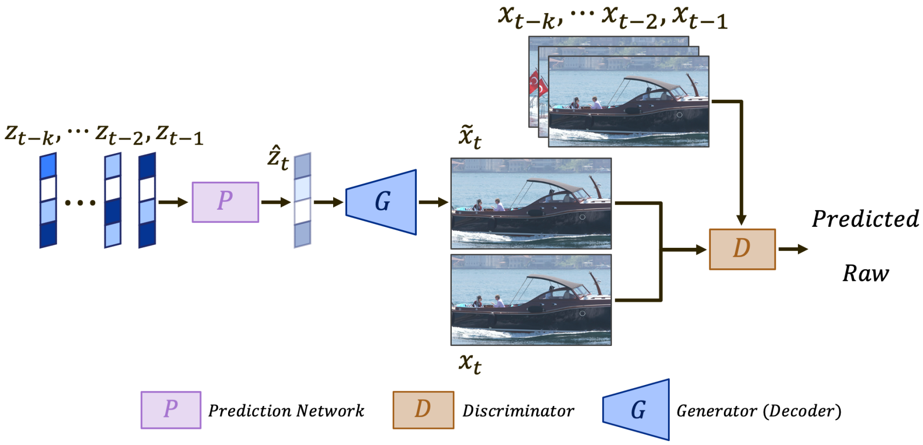

Figure 3.3: The latent space video prediction network estimates the next-frame latent with the understanding of latent space temporal correlation. A discriminator conditioned on the preceding frames provides feedback to the predictor so that it learns the mapping from the latent domain to the image space as well as the correlation of the previous and current frames.

As such, the image compression model learns a lossy transformation from spatial domain to latent space. Temporal correlation between subsequent frames in the latent domain is exploited with a ConvLSTM-based predictive model. An accurate inter-frame prediction model is a critical component in time-series data compression to capture the temporal correlation in the frame sequence and thereby achieve small cross-entropy between the original and compressed data. A well-trained predictor learns the capability to predict the normal inter-frame content of the video such as the movement of an object and the translation of the camera. This characteristic of the predictor allows abnormal event detection as a byproduct of video compression as discussed in Section 4.5.

Similar to existing video codecs, the introduced approach only encodes and transmits the residual between the predicted latent vector and the optimal latent z_{opt} obtained from the image compression process. Next frame prediction hinges on the conditional prior model learned by the prediction network, whose cells retain the memory of previous data distribution. The generic prior model is defined as

$$p_\theta(z_T|z_{<T}) = \prod_{t=2}^{T} \frac{p_\theta(z_{1:t})}{p_\theta(z_{1:t-1})}. \tag{3.3}$$

Given the complexity of stochastic data distribution in videos and the possible rapid transition between frames, training a good prior model is a main challenge to obtain satisfying compression performance. This method proposes a GAN-based adversarially trained ConvLSTM network to make predictions $\hat{z}$ on z_{opt} for video frame reconstruction.

As opposed to previously proposed methods, the prediction model $P(\cdot)$ produces estimated latent vectors instead of frames. This method requires significant attention to define a proper reconstruction objective function since element-wise error in latent space is often insufficient to measure spatial domain image reconstruction fidelity. The proposed prediction network is trained under an adversarial setup to exploit the complex similarity metric implicitly learned by a discriminator [75], which plays a critical role under the latent space prediction regime. Involving the discriminator objective [106] in the cost function improves the LSTM cells to establish more effective memory in terms of learning the temporal correlation. The loss function in the framework is expressed as

$$\begin{aligned} \mathcal{L} = \lambda \cdot \Big\{ & \mathbb{E}_{z\sim p_z}\big[\log\left(1 - D(G(P(z_{<t}|x_{<t})))\right)\big] \\ & + \mathbb{E}_{z\sim p_{\text{opt}}}\big[\log\left(D(G(z_{\text{opt}}|x_{<t}))\right)\big]\Big\} \\ & + (1-\lambda) \cdot \mathbb{E}_{p(z<t)}\big[\log p(z_t|z_{<t})\big], \end{aligned} \tag{3.4}$$

where $G(\cdot)$ is the generator network that reconstructs a frame x from a latent z and $D(\cdot)$ is the discriminator network that judges whether a frame belongs to a valid frame sequence as shown in Figure 3.3. The first term of the above loss function refers to the cross entropy of the discriminator cost function and the second term indicates the prediction error. Crucially, the image and video compression frameworks share the same generator (decoder) to reconstruct a frame x from a latent z. The generator parameters are inherited from image compression and thus fixed when training the prediction network. As shown in Figure 3.2, the reconstructed latent vectors, $\tilde{z} = \hat{z} + Q(r)$, inevitably lose some information from the optimal ones, $z_{\text{opt}} = \hat{z} + r$, due to discretization $Q(r)$. Note that the quantization distortion metric is not presented in the prediction network loss, thus a well-trained generator $G(z)$ is crucial to minimize the loss between the target frame x_t and $\tilde{x}_t$ synthesized/generated from $\tilde{z}_t$.

3.3.3 Quantization and Entropy Coding

The quantization and entropy coding of the residual r_t from latent prediction incorporated in the proposed video compression framework are deconstructed below. **Quantization:** This scheme applies ADMM [121] quantization to find discretized vectors with minimal degradation of quality compared to the original frames. A generic quantization problem can be described as follows:

$$\min_{r} f(r) \quad \text{subject to } r \in S, \tag{3.5}$$

where S is a quantized set and f is a loss function. In the context of residual quantization, the loss function (Equation 3.6) is defined as

$$f(r) = d_l(z_{\text{opt}}, (\hat{z} + r)) + d_s(x, G(\hat{z} + r)) \quad \text{subject to } r \in S. \tag{3.6}$$

The optimization problem above is given by a combination of a) the latent space distortion d_l given the optimal z_{opt}, and b) the distortion $d_s(\cdot)$ in spatial domain reconstruction. It is non-convex and not solvable with stochastic gradient descent (SGD) due to the quantization constraint. Moreover, direct quantization is likely to cause gradient vanishing. Recent works address these issues with ADMM using an indicator function and combining it with a differentiable loss function. During iterations, the ADMM method projects all elements of residual latent vectors to different quantized levels and minimizes the loss function in parallel. This guarantees that all elements are quantized in the process to find the optimal level.

In ADMM quantization, the problem (3.5) is redirected to optimizing the cost function $\min_r f(r) + g(u)$ subject to $r = u$ by introducing u as an auxiliary variable. The indicator function $g(u)$ is 0 if $u \in S$ or ∞ otherwise.

$$\min_r f(r) + g(u) \quad \text{subject to } r = u \tag{3.7}$$

$$g(u) = \begin{cases} 0 & \text{if } u \in S \\ +\infty & \text{otherwise} \end{cases} \tag{3.8}$$

Then, the augment Lagrangian method decomposes the dual variable optimization problem into two partial updating tasks performed iteratively and separately as described in (Equations 3.9, 3.10, and 3.11). With the addition of a convex and differentiable regularization term, the optimal solution is iteratively approximated through SGD with Adam optimizer.

$$r_{k+1} = \arg\min_r f(r) + \frac{\mu}{2} \cdot \|r - u_k + \eta_k\|_2^2 \tag{3.9}$$

$$u_{k+1} = \arg\min_u g(u) + \frac{\mu}{2} \cdot \|r_{k+1} - u + \eta_k\|_2^2 \tag{3.10}$$

$$\eta_{k+1} = \eta_k + r_{k+1} - u_{k+1} \tag{3.11}$$

Equation 3.10 is solved by the Euclidean projection of $r_{k+1} + \eta_k$ onto the quantized set S, which is formulated as $u_{k+1} := \Pi_S\left(r_{k+1} + \eta_k\right)$ where Π_S is the projection function.

The proposed method adopts an adaptive non-uniform quantization scheme where the quantization set is determined specifically for each video clip and transmitted together with the compressed data. Non-uniform quantization is implemented by selecting a subset of uniformly quantized levels and the proposed codec only stores/transmits the entropy-coded indices of the selected quantization levels instead of the original value.

Entropy Coding: A well-trained prediction model produces very sparse (with few non-zero elements) residual latent representations after quantization. The quantized residual vector is reshaped and stored in the compressed sparse row format and finally entropy coded with Adaptive Arithmetic Coding [161]. After this final step, the proposed codec achieves extremely high compression ratios with superior/competitive reconstruction quality compared to conventional codecs.

3.3.4 Rate-distortion Control

In order to allow a wide range of rate-distortion tradeoffs, the compression rate (or bit-per-pixel, bpp) of LVC is controlled by changing the number of elements in the latent vector and the number of quantization levels for the residuals. The transmitter in this approach continuously monitors the quality of the reconstructed (decompressed) frame using a distortion metric $d(x_t, \tilde{x}_t)$ shown in Figure 3.2 to adjust the compression method on-the-fly. Note that the prediction based residual encoding can occasionally fail when the scene changes abruptly. In rare occasions, the generator (decoder) may yield an inferior reconstructed frame $\tilde{x}_t$ due to limitations of the trained generator model. The frames that cause these issues are defined as *key-frames* and LVC encodes key-frames using a conventional image codec BPG [14]. These key frames are equivalent to intra-coded frames in conventional video codecs. This adaptive encoding prevents catastrophic failures in the proposed method that is designed to cover a wide range of rate-distortion tradeoff space.

3.4 Experimental Setup

The previous section introduces the loss function (Equation 3.4) that combines the loss of the latent reconstruction and the discriminator loss to enhance the quality of the reconstructed image from the prediction network. This work empirically chooses $\lambda = 0.1$ during the training to balance the first and second terms. Adding the discriminator loss term does not necessarily improve the PSNR/SSIM metric but it does enhance the subjective quality of video/image. The number of iterations typically needed for ADMM quantization in compression tasks is 50. The structure of DNNs (including the number of layers and kernel sizes, etc.) used in the experiment is specified in the supplementary material.

3.4.1 Datasets

The proposed framework is trained with the Kinetics dataset [19] and the UGC dataset [156]. In the Kinetics dataset, roughly 98,000 videos each lasting for around 10 seconds with a resolution higher than 720p are used for training. The UGC dataset has a rich collection of content such as lectures, animation, and music videos with more than 1,500 clips for an average length of 20 seconds. Training and evaluation datasets are mutually exclusive. The evaluation is tested on Video Trace Library (VTL) [2] and Ultra Video Group (UVG) [103] datasets. The VTL dataset contains 20 videos with around 40,000 frames of resolution 352×288. The UVG dataset has 16 videos, and the evaluation is tested on the original 8 videos with an overall 3,900 frames of resolution 1920×1080 to compare with other existing methods.

3.4.2 Metrics

This work quantitatively compares the experimental results with the most prevailing hand-crafted video codecs as well as the learning-based codecs recently proposed. The quality distortion is measured by PSNR and MS-SSIM of decompressed frames.

For the conventional codecs, H.264 and H.265, the experiment applies the *ffmpeg* very-fast mode with a GOP of 10/12 for the VTL/UVG dataset.

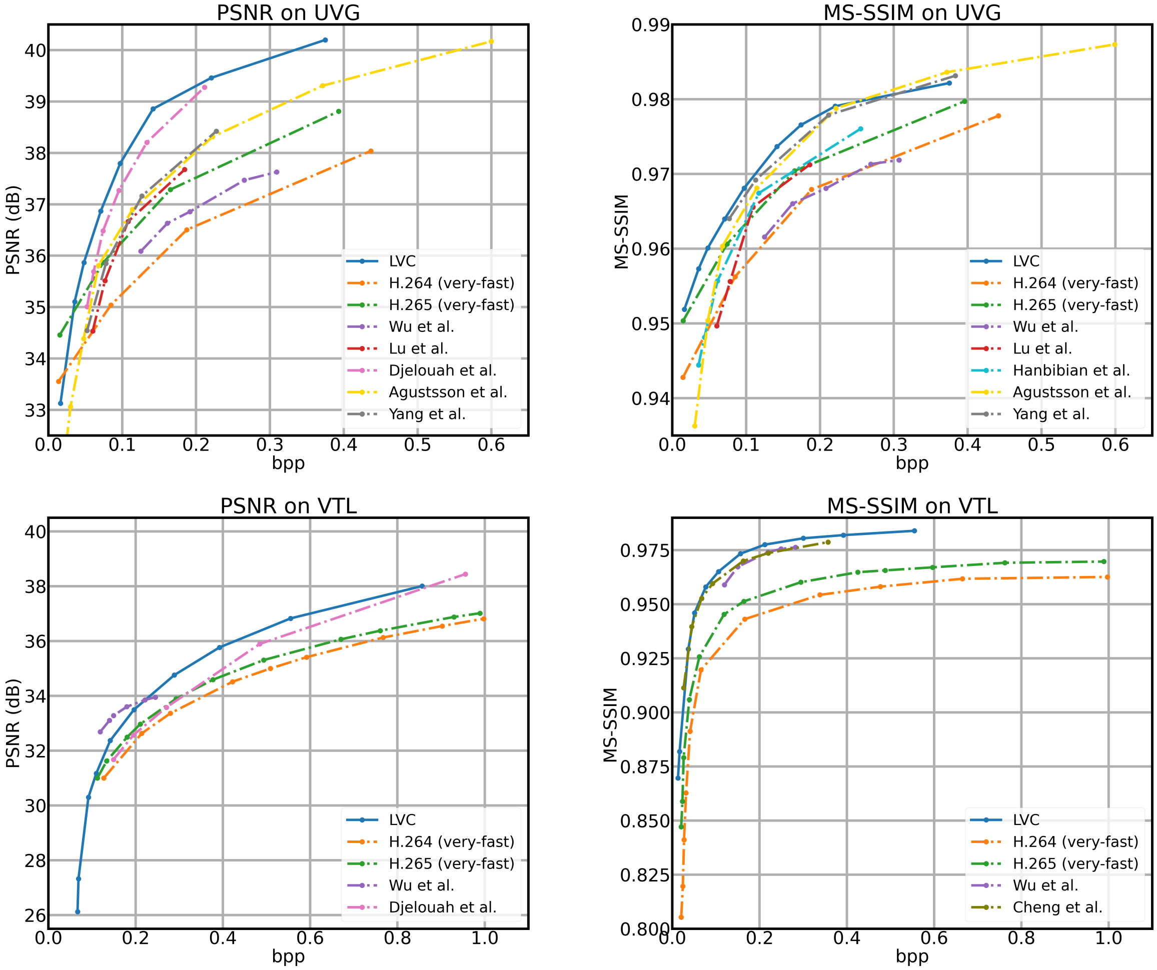

Figure 3.4: Comparison of the proposed approach with H.264, H.265, and learning-based codecs [162, 94, 33, 165, 4] for PSNR, and [162, 94, 48, 25, 165, 4] for MS-SSIM on UVG and VTL datasets. LVC video codec is not optimized specifically for PSNR or MS-SSIM.

3.5 Experimental Results

3.5.1 Video Compression Performance Analysis

The experimental results on VTL and UVG datasets in Figure 3.4 show that this method outperforms AVC/H.264, HEVC/H.265, and the state-of-the-art DNN-based codecs for most of the tested bits-per-pixel rates in terms of PSNR and MS-SSIM.

Figure 3.1 is the visualization of a frame from the UVG dataset showing that the introduced approach gives equally if not more visually appealing reconstructed frames under a lower/same bpp compared with other codecs. The experimental result indicates that incorporating discriminator loss in training can provide more complex details and realistic quality. Although this subjective quality gain is not always captured by the commonly adopted pixel-wise distortion metrics such as PSNR and MS-SSIM, this method still achieves better results in the rate-distortion tradeoff measured by PSNR and MS-SSIM compared to other video codecs. Note that this scheme is flexible to adopt a different set of $z_{\text{opt},t}$ that minimizes an application-specific target metric (instead of generic MSE) such as a CNN feature loss or discriminator loss if the decompressed video is intended for machine learning based applications. The result shows that LVC's PSNR in a very low bpp regime can be lower than that of some other codecs (Figure 3.4 second left). It is mainly because of two factors: 1) there is a resolution mismatch between training and evaluation datasets, and 2) the target z is minimized by a combined loss as shown in Equation 3.4 for superior subjective quality on reconstructed frames although it may result in slightly degraded PSNR.

3.5.2 Ablation Studies

Latent Space Video Prediction: This work presents an example video frame directly generated from the predicted latent $\hat{z}_t$ (without residual compensation) in Figure 3.1 and Figure 3.5. The result shows that the proposed approach produces effective next-frame prediction to achieve high compression rates. Figure 3.6 right shows the quality of video entirely generated from the predicted latent $\hat{z}_t$ in terms of PSNR on UVG [103] dataset. For this experiment, the prediction-only mode produces the video sequences from predicted next-frame latent $\hat{z}_t$ without residual compensation whereas the input to the predictor is the reconstructed latent $\tilde{z}_{<t}$. The result in Figure 3.6 right shows that this approach provides high quality next-frame prediction

while the average prediction quality saturates as the bit rate increases. Note that the bit rate for the prediction-only mode is overstated because it is mostly dominated by residuals that are unused in the prediction-only mode. The observation implies that while the reconstructed latents, $\tilde{z}_t$, can learn the spatial domain correlation within a frame very well, there remain non-negligible inter-frame temporal prediction errors caused by complex motion dynamics. In the video compression task, this work alleviates this issue and achieves significant quality improvements by saving and transmitting keyframes and residuals. For the majority of *normal* video frame sequences that have strong temporal correlations, the proposed prediction method provides reliable and accurate prediction for compression. For occasional abnormal sequences, the limitation of the prediction network is exploited for the task of abnormal event detection.

Compressed Data Size Distribution: As described in the framework, the codec first transmits $k = 6$ *initial frames* without prediction in their latent representation z_{opt}, and they are fed to the prediction ConvLSTM network to initialize the latent sequence. During the regular codec operation after initialization, the proposed scheme keeps monitoring the quality of reconstructed frames $\tilde{x}_t$ compared with the raw frame by evaluating the MSE distortion $d(x_t, \tilde{x}_t) = \|x_t - \tilde{x}_t\|^2$. When the distortion exceeds a predefined threshold, the scheme declares this frame is a *key-frame* and transmits the BPG-coded frame directly to the receiver (without using the latent domain residual). To further inspect the implication of this proposed approach, analysis of the distribution of code length for different bit-rates is shown in Figure 3.6. The result shows that residual latent vectors dominate the overall compressed data as expected. The proportion of key-frames is dependent on the video content (abrupt scene changes incur more key-frames) but it is less significant, especially for very low or high compression rates.

A lower distortion threshold is set for the higher image quality target. Thus it

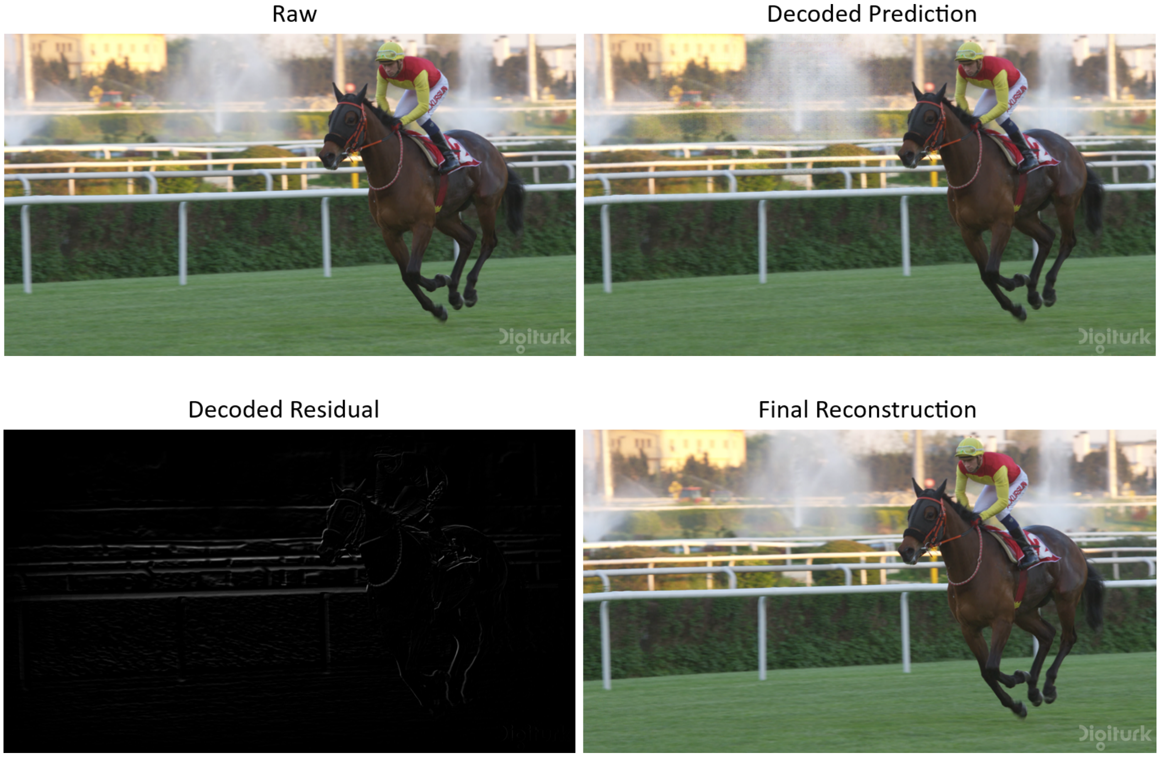

Figure 3.5: Visualization of LVC model results. The prediction network predicts the next-frame latent according to several preceding latent. The residual in the latent space is added to the predicted latent for final reconstruction via the generator.

is likely to encounter more intra-coded key-frames for higher quality video compression. On the other hand, allowing more bits per pixel reduces the distortion, thereby decreasing the number of bits for key-frame transmission. Because of these two counteracting effects, the bit allocation for the key-frames shown in Figure 3.6 left shows a non-monotonic pattern with relatively fewer bits at two extremes (lowest and highest rates), while more bits are allocated to key-frames when the compression is at a medium level.

The evaluation of the UVG dataset confirms that the key-frames occupy on average only 8.73% of the total code length. The bit streams for prediction residual encoding account for 84.8% of the total compressed bit sequence on average, while the rest 6.48% contributes to the initial frames. According to the inspection of the composition of the compressed signal, the residual r_t is 8× more sparse (fewer non-zero values)

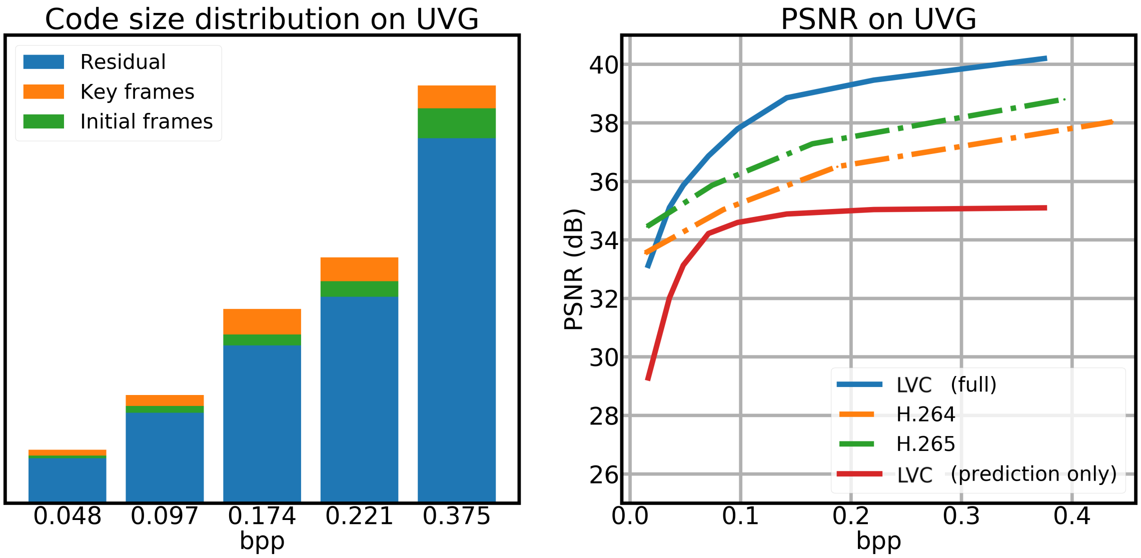

Figure 3.6: Code length distribution for different compression ratios (left) and evaluation of next-frame prediction performance (right). The prediction-only method generates the video sequences from predicted next-frame latent $\hat{z}_t$ without residual compensation whereas the input to the predictor is the reconstructed latent $\tilde{z}_t$. Note that bpp values for the prediction-only curve do not reflect the actual code size (which should be very close to 0 as residuals are discarded). They are just proportional to the latent space dimension adopted for the experiment.

than the target latent representation $z_{\text{opt},t}$ on average. This validates that generating accurate predictions is a key enabler for LVC codec.

3.5.3 Application: Anomaly Detection

The ablation study and experiment are extended to anomaly detection on surveillance video clips, which mostly contain homogeneous contents with relatively small changes between scenes. With the assumption that the proposed prediction model reliably predicts the next frame in normal scenes, an anomalous event is detected when the difference between the predicted latent vector and the target z_{opt} of the frame is substantial. The error of prediction caused by the abnormal event is quantified by computing the Euclidean distance between the predicted and target latent vector.

Following the regularity score proposed in [27], $e(t)$ is the L_2 distance between the prediction and target latent at frame index t. The score $S(t)$ reflects the normality

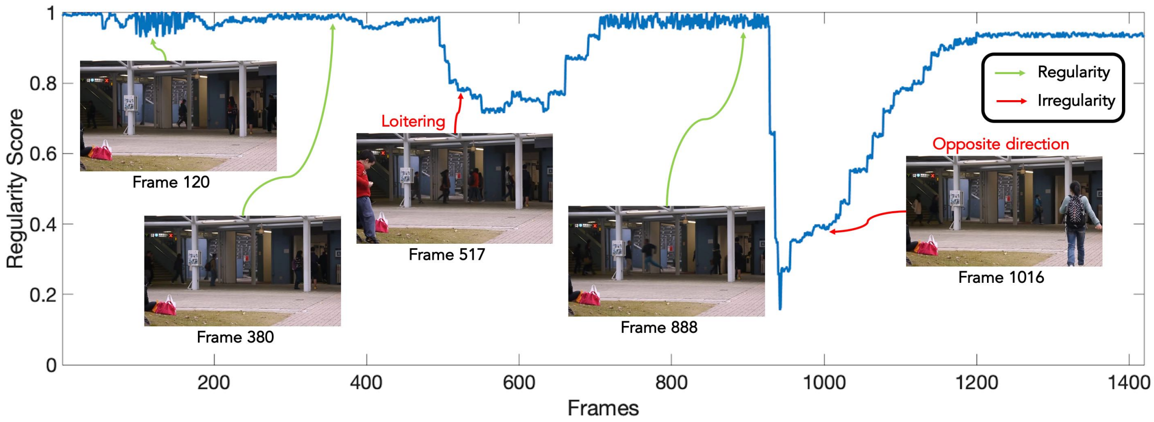

Figure 3.7: Regularity score (blue curve in the figure) along the frame sequence. Normal scenes typically score > 0.9 whereas abnormal events cause steep score degradation.

of the frame within the sequence of time duration T.

$$
\begin{aligned}
e(t) &= \|z_{\text{opt}} - \tilde{z}_t\|_2 \\
S(t) &= 1 - \tfrac{e(t) - \min_\tau(e(\tau))}{\max_\tau(e(\tau))}, \quad {\scriptstyle \tau = t-T, t-T+1, \cdots, t}
\end{aligned}
\tag{3.12}
$$

The regularity score (Equation 3.12) indicates that a relatively small prediction error produces a score close to 1, while it will drop significantly when a large prediction error is encountered because of an abnormal (*i.e.*, unseen during the prediction training) event in the scene. Figure 3.7 visualizes the change of the regularity score in a video clip.

This work tests the anomaly detection performance on several widely used datasets with distinctive features. UCSD [96] Ped1 contains 40 abnormal events in 70 video clips, and UCSD Ped2 has 12 abnormal events in 28 videos. The Subway entrance/exit dataset [3] has 96/43 minutes of video with 66/19 abnormal events. Avenue dataset [92] comprises overall 47 abnormal events. The proposed predictor network is solely trained for the video compression task, and it has not been retrained for anomaly detection. This approach is evaluated on the test sequences of those datasets in terms of area under the Receiver Operation Characteristic (ROC) curve (AUC), which

Table 3.1: Anomaly detection performance evaluated by area under ROC curve (AUC %).

Methods	**UCSD Ped 1**	**UCSD Ped 2**	**Subway Entrance**	**Subway Exit**	**CUHK Avenue**
Wang *et al.* [154]	72.7	87.5	81.6	84.9	–
Hasan *et al.* [50]	81.0	90.0	**94.3**	80.7	70.2
Chong *et al.* [27]	89.9	87.4	84.7	94.0	80.3
Liu *et al.* [89]	83.1	**95.4**	–	–	84.9
Gong *et al.* [43]	–	94.1	–	–	83.3
LVC	**90.9**	93.6	88.2	**94.5**	**85.4**

cumulatively reflects the ROC metric. Generally, a higher AUC indicates better performance.

In this scheme, normal scenes with learned patterns mostly do not trigger false alarms as they create small fluctuations in the regularity score $S(t)$. However, when an abnormal event happens, the regularity score significantly drops with a high probability. This abnormal event detection is just a byproduct of the compression algorithm. Nonetheless, the event detection performance shown in Table 3.1 exhibits comparable/superior accuracy compared with others specifically designed for the task.

3.6 Conclusion

This chapter proposes a GAN-based framework that accomplishes video prediction and compression. The method simultaneously learns a transform of the original video into a lower-dimensional latent representation as well as a temporally-conditioned probabilistic model. Performance evaluations show that this work achieves superior/competitive results compared to other (learning-based) codecs for a wide range of rate-distortion tradeoffs. The LVC performance gain is majorly attributed to the approach that reduces both spatial and temporal redundancy by combining image compression and video prediction in latent space. The study additionally applies the proposed video prediction scheme to abnormal event detection, showing competitive accuracy compared to algorithms specifically optimized for that task.

CHAPTER IV

MMVC: Learned Multi-Mode Video Compression with Block-based Prediction Mode Selection and Density-Adaptive Entropy Coding

4.1 Introduction

[1] Over the past several years, with the emergence and booming of short videos and video conferences across the world, video has become the major container of information and interaction among people on a daily basis. Consequently, we have been witnessing a vast demand increase on transmission bandwidth and storage space, together with the vibrant growth and discovery of handcrafted codecs such as AVC/H.264 [136], HEVC [136], and the recently released VVC [135], along with a number of learning based methods [162, 94, 123, 48, 4, 55, 85, 167, 166, 80, 101, 81].

Prior works in deep video codecs have underlined the importance of utilizing and benefiting from deep neural network models, which can exploit complex spatial-temporal correlations and have the ability of 'learning' contextual and motion features. The main objective of deep video compression is to predict the next frame from previous frames or historical data, which results in the reduction of the amount of residual information that needs to be encoded and transmitted. This has so far

[1] The content of this chapter is based on [86].

led to two directions: (1) to build efficient prediction or estimation models, and extract motion information by leveraging the temporal correlation across the frames [94, 4, 167, 55]; (2) to make an accurate estimation of the distribution of residual data and push down the information entropy statistically by appropriate conditioning [48, 166, 80]. The existing works usually fall in one or a combination of the above two realms. In the light of the learning capability that deep neural networks can offer, the results in this work argue and demonstrate that some measures of adaptively selecting the right mode among different available models in the encoding path can be advantageous on top of the existing schemes, especially when the adaptive model selection is applied at the block level in the feature space.

Drawing wisdom from conventional video codec standards that typically address various types of motions (including the unchanged contextual information) in the unit of macroblocks, this work presents a learning-based, block wise video compression scheme that applies content-driven mode selection on the fly. The proposed method consists of four modes targeting different scenarios:

- *Skip* mode (S) aims to utilize the frame buffer on the decoder and find the most condensed representation to transmit unchanged blocks to achieve the best possible bitrate. This mode is particularly useful for static scenes where the same backgrounds are captured by a fixed-view camera.

- *Optical Flow Conditioned Feature Prediction* mode (OFC) leverages the temporal locality of motions. This mode captures the optical flow [140] between the past two frames, and the warped new frame is treated as a preliminary prediction of the current frame. This warping serves as the condition to provide guidance to the temporal prediction DNN model.

- *Feature Propagation* mode (FPG) applies to blocks where changes are detected, but there is no better prediction mode available. This mode copies the previously reconstructed feature block as the prediction, and encodes the residual from there.

- For other generic cases, MMVC proposes the *Feature Prediction* mode (FP) for feature domain inter-frame prediction with a ConvLSTM network to produce a predicted current frame (block).

Prior to the mode selection step, The transmitter produces the optimal low-dimensional representation of each frame using a learned encoder and decoder pair based on the image compression framework in [84] for the mapping from pixel to feature space. The block by block difference between the previous frame and the current frame represents the block wise motion. Unlike some state-of-the-art video compression frameworks that separately encode motions and residuals, the presented method does not encode the motion as it is automatically generated by the prediction using the information available on both the transmitter and receiver. To adapt to different dynamics that may exist even within a single frame for different blocks, this method evaluates multiple prediction modes that are listed above at the block level. As a result, this codec can always obtain residuals that have the highest sparsity thereby the shortest code length per block. Furthermore, MMVC proposes a residual channel removing strategy to mask out residual channels that are inessential to frame reconstruction, exploiting favorable tradeoffs between noticeably higher compression ratio and negligible quality degradation.

4.2 Related Work

4.2.1 Learned Video Compression

Pioneering works in learned video compression generally inherit the concept and methodology in conventional codecs. Wu *et al.* [162] propose to hierarchically interpolate the frames between a predefined interval, where both the forward and backward motions are represented by block motion vectors. DVC [94] and Agustsson *et al.* [4] adopt optical flow based motion estimation and warping schemes. Habibian *et al.* [48] map a patch of frames with 3D spatial-temporal convolutions to a lower-dimensional space and make temporal predictions on the prior distribution through GRU. With the progress in designing autoencoder-based feature extraction and reconstruction as a stepping stone, recent works have achieved better performance by performing motion prediction or estimation in the feature domain (as opposed to the pixel domain), which naturally represents both motion and residual in a more information-dense form to benefit compression. FVC [55] learns a feature space offset map as motion representation, and the motion compensation step is accomplished by deformable convolution [30].

4.2.2 Mode Selection

As a widely adopted tool in the conventional video coding standards, the idea of mode selection intends to evaluate different schemes on the fly to address the inter-frame temporal correlations in a context-dependent manner. Based on this concept, Ladune *et al.* [72] proposes a network that learns the pixel wise weighting to determine whether or not to skip encoding the respective pixel. Hu *et al.* [54] presents a hyper-prior guided mode selection scheme that compresses motion in different resolutions, and it uses a learned mask to skip the encoding of some residual features.

4.2.3 Entropy Coding

Most existing works in learned video coding adopt a learned entropy coding scheme as presented in [11] originally for image compression to facilitate end-to-end rate-distortion optimization. This method then evolved to provide more flexible and accurate entropy modeling by learning the distribution parameters [12, 105]. Additionally for video sequences, incorporating temporal cues to obtain more accurate entropy estimation can lead to higher compression gains. RLVC [166] proposed a probability model that approximates the distribution of encoded residuals to a parameterized logistic distribution, conditioned on the feature of previous frames propagated under a recurrent setup to establish a richer temporal prior. With a similar insight, DCVC by Li *et al.* [80] proposes building an entropy model with temporal conditions. Recently, Mentzer *et al.* [101] introduces a novel transformer-based framework that establishes a temporally conditioned entropy model and abstracts all decorrelation efforts by one-shot model execution.

4.2.4 Quantization and Channel Removal

Adjusting the bin size or quantization granularity is a technique to address the uneven redundancy between channels, allowing the content of greater importance to occupy more quantized bits and the rest with fewer bits for the maximum quality under a similar bitrate. Cui *et al.* [29] propose scaling the residual feature with learned channel wise factors before quantization and inverse-scaling before reconstruction as an effective way of rate adaptation. In this work, to simplify the datapath but still benefit from the same concept, a channel wise binary masking (0 or 1 scaling) is applied to remove disposable channels in the feature blocks when it offers a favorable tradeoff in the achievable rate vs. quality.

4.3 Method

The original temporal sequence of raw frames is denoted as the set $X = \{x_1, x_2, \cdots, x_{t-1}, x_t, \cdots\}$. Correspondingly, on the receiver (and also on the transmitter), the reconstructed previous frames are available as $\hat{x} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{t-1}\}$. The whole system flow of the proposed video compression scheme, MMVC, is shown in Figure 4.1. The system can be divided into four main parts: feature extraction, temporal prediction, channel removal, and quantization plus entropy coding.

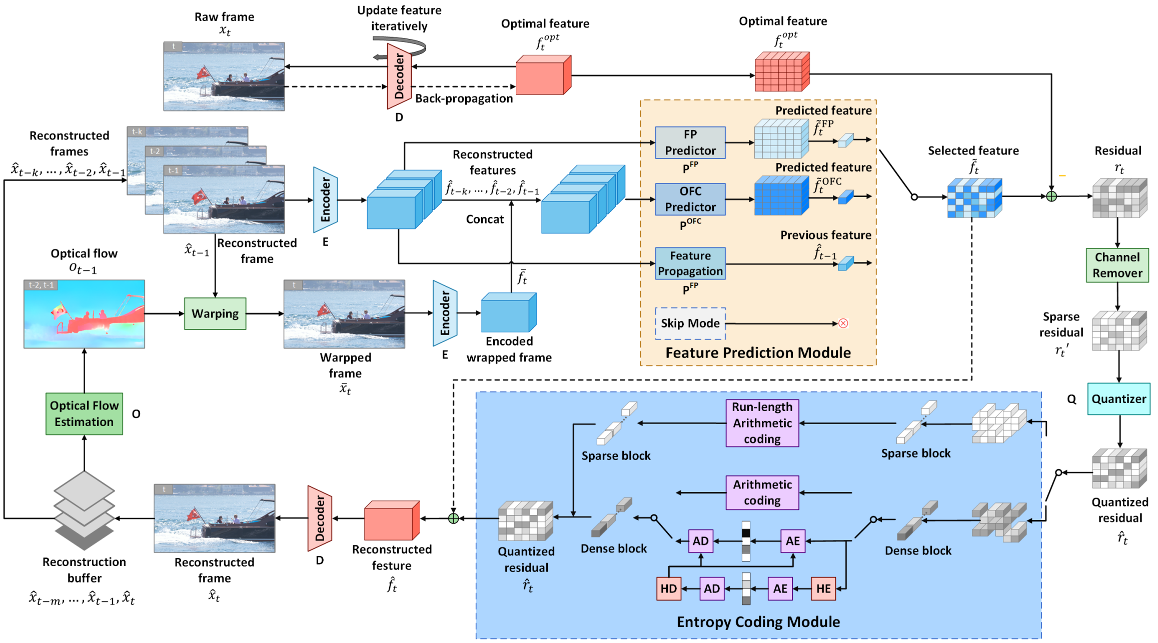

Figure 4.1: Overview of the proposed multi-mode video coding method. The current and previous frames are fed into the feature extractor and then go through branches of prediction modes followed by residual channel removal, quantization, and entropy coding process. The proposed codec then selects the optimal prediction and entropy coding schemes for each block that lead to the smallest code size.

4.3.1 Pixel Space Preprocessing

As an initial step, MMVC partitions two consecutive frames into $k \times k$ blocks and calculate the block wise differences to form $n = \{n_{1,2}, n_{2,3}, \cdots, n_{t-1,t}, \cdots\}$, where

$$n_{t-1,t}^{i,j} = \left\| x_{t-1}^{i,j} - x_t^{i,j} \right\|_2^2, \qquad i, j \in \{1, \cdots, k\}, \tag{4.1}$$

and the superscripts i, j denote the 2D position of a block. At each time step, the MMVC scheme examines the numerical values in the block wise difference. The blocks with all-(near)zero differences are encoded using the *Skip* mode, where only the block (positional) index information is recorded and transmitted. With an algorithm parameter $\epsilon \approx 0$, this step generates a binary mask $m = \{m_{1,2}, m_{2,3}, \cdots, m_{t-1,t}, \cdots\}$

each of $k \times k$ elements, where

$$m_{t-1,t}^{i,j} = \begin{cases} 0, & n_{t-1,t}^{i,j} < \epsilon, \\ 1, & \text{otherwise.} \end{cases} \tag{4.2}$$

The unchanged blocks are masked out, indicating that no prediction will be performed and no residual will be stored. Instead, they can be recovered directly from the previously reconstructed frames by copying their pixel blocks at the corresponding positions.

To capture the pixel level motion information and utilize its temporal locality, the proposed scheme obtains the optical flow [140] $O(\cdot)$ between the previous two reconstructed frames (*e.g.*, $\hat{x}_{t-2}, \hat{x}_{t-1}$) and warp it to the latter one to form a set $\{\bar{x}_2, \cdots, \bar{x}_t\}$ that is commonly available on both the transmitter and the receiver. This step can be described as:

$$\bar{x}_t = \mathit{warp}(O(\hat{x}_{t-2}, \hat{x}_{t-1}), \hat{x}_{t-1}). \tag{4.3}$$

4.3.2 Feature Extraction

To this point, the pixel space preparation and optical flow based warping are accomplished. To extract rich features across the frames in a compact representation, a set of auto-encoders is trained together with an entropy modeling network to achieve different rate-distortion tradeoff points. Given a trained encoder $E(\cdot)$ and decoder pair $D(\cdot)$, each raw frame x_t is encoded to an optimal feature set f_t^{opt} using a back-propagation based iterative scheme [84] that refines the one-shot encoding output $f_t = E(x_t)$ through a coupled decoder $D(\cdot)$ to obtain f_t^{opt} by minimizing the MSE distortion:

$$f_t^{opt} = \arg\min_{f_t} d(x_t, D(f_t)). \tag{4.4}$$

Note that f_t^{opt} is only available at the transmitter since the raw frame x_t is not available at the receiver. Previously reconstructed frames $\{\hat{x}_{t-k}, \cdots, \hat{x}_{t-1}\}$, as well as the warped frame $\bar{x}_t$ are encoded (without iterative optimization) to corresponding feature sets $\hat{f}_t = E(\hat{x}_t)$ and $\bar{f}_t = E(\bar{x}_t)$ using the same encoder. These are commonly available on both the transmitter and receiver.

4.3.3 Feature Prediction

Figure 4.1 depicts the feature space prediction and mode selection strategy. Prior to the mode selection step, this work assumes the optimal low-dimensional representation of current frame f_t^{opt} and the binary mask $m_{t-1,t}$ for *Skip* mode indication are ready at the transmitter. The proposed method replaces conventional motion estimation, compression, and compensation steps with feature domain prediction optionally conditioned by pixel-domain optical flow. Residual is the difference between the optimal feature f_t^{opt} and the predicted feature. However, this might lead to large residuals when the prediction is not accurate. To accommodate the rich variety of motions, a prediction method consisting of three mode branches is introduced, and mode selections for *each block* (not for the entire frame) are selected based on the entropy of residuals.

The *Feature Prediction* branch is implemented with a ConvLSTM network, where the predictor P^{FP} takes the reconstructed features $\{\hat{f}_{t-k}, \cdots, \hat{f}_{t-1}\}$ as inputs, and produces a predicted current frame feature representation, capturing temporal correlation in the feature domain:

$$\tilde{f}_t^{FP} = P^{\text{FP}}(\hat{f}_{<t}), \text{ with } r_t^{\text{FP}} = f_t^{opt} - \tilde{f}_t^{\text{FP}}. \tag{4.5}$$

To augment this prediction process with more contextual cues from certain scenes, MMVC forms another prediction path, called *Optical Flow Conditioned Feature Pre-*

diction. It uses the optical flow warped feature $\bar{f}_t$ as a conditional input for a prediction network P^{OPC}:

$$\tilde{f}_t^{\mathrm{OFC}} = P^{\mathrm{OFC}}(\hat{f}_{<t}|\bar{f}_t), \text{ with } r_t^{\mathrm{OPC}} = f_t^{opt} - \tilde{f}_t^{\mathrm{OFC}}. \tag{4.6}$$

The experimental results indicate that in some cases neither of the above modes can outperform directly copying/propagating the respective block in reconstructed features at time $t-1$ such that $\hat{f}_t = \hat{f}_{t-1}$. Hence, this *Feature Propagation* is adopted as the third prediction type described as:

$$\tilde{f}_t^{\mathrm{FPG}} = \hat{f}_{t-1}, \text{ with } r_t^{\mathrm{FPG}} = f_t^{opt} - \tilde{f}_t^{\mathrm{FPG}}. \tag{4.7}$$

After having the predicted feature representations under the above modes for all non-skip blocks, the residuals $(r_t^{\mathrm{FP}}, r_t^{\mathrm{OFC}}, r_t^{\mathrm{FPG}})$ are partitioned to equal-sized residual blocks $(r_t^{\mathrm{FP},i,j}, r_t^{\mathrm{OFC},i,j}, r_t^{\mathrm{FPG},i,j})$ so that each block indexed by i and j has a set of residuals from different modes. The residual block partition side is determined to keep the number of blocks unchanged from that of pixel domain blocks (*i.e.*, $k \times k$). To determine the optimal prediction mode, this codec quantizes and entropy encodes (introduced in Section 4.3.5) each of the block partitioned residuals respectively, and proceeds with the one that has the shortest code length. Therefore, the output of this step is a block-based residual map $r_t^{i,j}$ constructed by block wise optimal prediction mode selection. This process is described as:

$$\begin{aligned} r_t^{i,j} &= \arg\min_{\hat{r}_t^{i,j}}(R(Q(\hat{r}_t^{i,j}))), \text{ with} \\ \hat{r}_t^{i,j} &\in \{r_t^{\mathrm{FP}\,i,j}, r_t^{\mathrm{OFC}\,i,j}, r_t^{\mathrm{FPG}\,i,j}\} \text{ and } i,j \in \{1, \cdots, k\}, \end{aligned} \tag{4.8}$$

where $Q(\cdot)$ and $R(\cdot)$ represent the quantization step and the bitrate after entropy coding respectively.

4.3.4 Block Wise Channel Removal

An adaptive residual channel removal technique is adopted to ensure that more bits are allocated to quality-critical residual elements. Carefully designed channel removal criteria can guarantee the reconstruction quality while reducing the number of bits consumed by unimportant residual feature channels. In an effort to only preserve feature channels carrying essential residuals and maintain the reconstruction quality after channel removal, the proposed method inspects each channel in a block separately. For one residual block and the predicted block along with it, the least important channel is selected by evaluating the PSNR degradation per channel removal. Channels are evaluated and removed iteratively in this manner as long as the quality degradation is within a predefined limit.

4.3.5 Density Adaptive Entropy Coding

The entropy coding is highlighted with the blue background in Figure 4.1. This datapath operates in accordance with the adaptive channel removal strategy, where pruned residual channels are set to zeros. It also considers sparse non-zero residuals as a result of efficient prediction. The proposed density-adaptive entropy coding method consists of a block wise sparse path and a dense path as shown in Figure 4.1. The density of each non-zero residual block is first evaluated, and the block is fed into the sparse path when the density is under a predefined threshold. Otherwise, the block is fed into the dense path. This mode selection is recorded as a block wise binary density map.

In the sparse residual path, the non-zero residual positions are run-length coded prior to conventional arithmetic encoding, and non-zero residuals are gathered together for separate arithmetic encoding. The dense path consists of two options: (1) a learned entropy codec model guided by the hyperpriors followed by direct quantization [105], and (2) a conventional arithmetic coding method coupled with ADMM

[17] quantization trained to non-uniformly discretize the residuals and optimized for minimal quantization error. This work proceeds with the option that leads to a lower rate and records the corresponding block wise entropy coding mode map w_t for the receiver as side information.

Note that MMVC maintains the same entropy coding path across all channels in each block to limit the cost of bits to encode w_t. To further reduce the bitrate, the binary density map (sparse vs. dense path) is also entropy-coded with Huffman coding. This method incurs additional bits for conveying the side information (density map and w_t), the experiments in Section 4.5.3 confirm that the overall bitrate reduction offsets the side information overhead.

4.3.6 Model Training Strategy and Losses

The mode selection scheme requires a uniform feature space for different prediction modes. Therefore both predictors (P^{FP} and P^{OFC}) need to be optimized under the same pixel-feature space mapping. To get this mapping regularized under different rates, the *Optical Flow Conditioned Feature Prediction* model P^{OFC} is optimized by:

$$\begin{aligned}\min_{\gamma,\eta,\phi,\varphi} R_\gamma(f_t^{opt} - P_\eta^{\text{OFC}}(E_\phi(\hat{x}_{<t})|E_\phi(\bar{x}_t)))+ \\ \lambda \cdot d(D_\varphi(\tilde{f}_t^{\text{OFC}} + \hat{r}_t^{\text{OFC}}), x_t),\end{aligned} \tag{4.9}$$

where $E(\cdot)$ and $D(\cdot)$ are the auto-encoder pair, and $d(\cdot)$ is the distortion calculated by MSE.

After this mapping is fixed (*i.e.*, the weights of encoder/decoder pair are trained) for P^{OFC} at a specific rate point, then the *Feature Prediction* model P^{FP} is optimized to minimize distortion between the predicted and optimal features, measured by both

MSE and discriminator loss. This optimization process is expressed as:

$$\begin{aligned}\min_{\theta}\max_{\psi}\ (1-\alpha)\cdot\{&\mathbb{E}_{f\sim p_{\mathrm{opt}}(f)}[\log S_{\psi}(D(f_t^{opt}))]+\\ &\mathbb{E}_{f\sim p_f(f)}[\log\Big(1-S_{\psi}(D(P_{\theta}^{\mathrm{FP}}(\hat{f}_{<t})))\Big)]\}+\\ &\alpha\cdot d(D(P_{\theta}^{\mathrm{FP}}(\hat{f}_{<t})),x_t),\end{aligned} \tag{4.10}$$

where $S(\cdot)$ is the discriminator network optimized in the GAN setting together with the prediction model to judge whether the reconstructed frame from the feature set is original (*i.e.*, raw frame) or not. This discriminator model makes the training of the prediction model converge faster.

4.4 Experimental Setup

4.4.1 Training Details

Two prediction modes are trained separately so that the training procedure can be divided into two stages. First, the encoder-decoder pair, the context and entropy model, together with the *Optical Flow Conditioned Feature Prediction* model P^{OFC} (Feature Predictor path is disabled) are end-to-end trained till convergence. The model involved in the experiments is only optimized with MSE as the distortion loss. To achieve different bitrates for rate-distortion tradeoffs, this work curates a set of Lagrange multipliers as $\lambda = \{2, 64, 256, 1024, 2048, 4096\}$. The model is optimized for 10M steps with a batch size of 16. The learning rate is initialized to be 10^{-4}, which is scaled to half every 2M steps. As each individual value λ_i leads to a unique bitrate along with the quality, MMVC ends up having a set of trained encoder-decoder pairs for various rates. During the second stage in the training procedure, the parameters are fixed in the trained encoder-decoder pair, and the optimal feature representations f_t^{opt} are obtained from iterative back-propagation through the decoder. These obtained optimal features serve as the input to train the *Feature Prediction* model P^{FP}, which is optimized by discriminator loss in addition to the MSE loss for enhanced visual quality. This work trains P^{FP} model at an initial learning rate of 5×10^{-4} for 20M steps with a batch size of 8, and the learning rate is decayed by half for every 2M steps after training 10M steps.

4.4.2 Datasets

Training Datasets: The Vimeo-90k dataset, Kinetics dataset, and UGC dataset are used for training purposes. The *Optical Flow Conditioned Feature Prediction* mode is trained with the Vimeo-90k Septuplet [164], which contains 89,800 short video sequences, with each sequence having 7 consecutive frames of size 488×256 pixels. To enlarge the training set, random cropping of each original sequence to four 256×256-pixel aggregated sequences is implemented. The *Feature Prediction* mode is trained with part of the Kinetics dataset and the UGC dataset. The Kinetics dataset has 98,000 videos, each of 10 seconds with a resolution higher than 720p. The UGC dataset is composed of clips each lasting for 20 seconds. The videos in Kinetics and UGC that have resolutions higher than 1080p are collected and cropped to 1024×1024 pixels for the training process.

Testing Datasets: To evaluate the performance of the presented method quantitatively and qualitatively, this work performs experiments on three datasets: the UVG dataset [104], the MCL-JCV dataset [151] and the HEVC class B dataset [136]. All testing videos that are chosen have the same 1080p resolution. To showcase the benefit of the *Optical Flow Conditioned Feature Prediction* mode, the presented work also adopts part of the Kinetics dataset for ablative experiments. Video frames used for testing are not included in the training dataset.

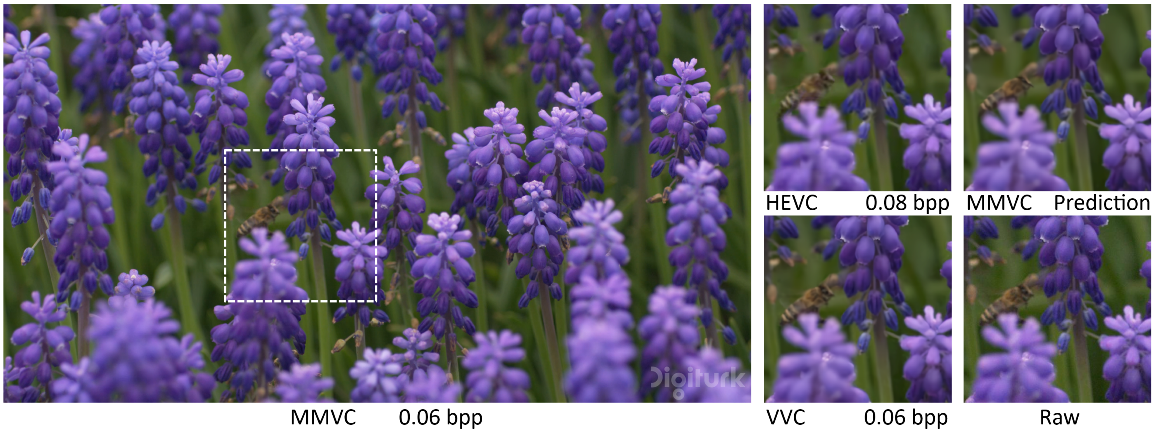

Figure 4.2: Reconstruction with standard codecs (HEVC, VVC) and MMVC method. Details of the static background and dynamic objects are well preserved in the frame generated from the predicted features and entropy-coded residuals that are block wise selected from multiple modes.

4.4.3 Metrics

PSNR and MS-SSIM are used as the quantitative evaluation metrics in the experiments. PSNR is a standard way to reflect the degree of distortion in reconstruction whereas MS-SSIM often serves as a proxy indicator for perceptual quality.

4.5 Experimental Results

4.5.1 Quantitative Results

To demonstrate the performance of the proposed codec, this work evaluates the rate and distortion tradeoff curves and PSNR-based BD-bitrate measurements with the state-of-the-art learned video compression algorithms published in recent years. Specifically, the comparison includes results from DVC [94], FVC [55], Liu *et al.* [85], DCVC [80], C2F [54], VCT [101], Li *et al.* [81], HLVC [165], M-LVC [82], Agustsson *et al.* [4], and RLVC [166]. The measurements from traditional codec standards: AVC [160], HEVC [136], and the latest VVC [135] are also included.

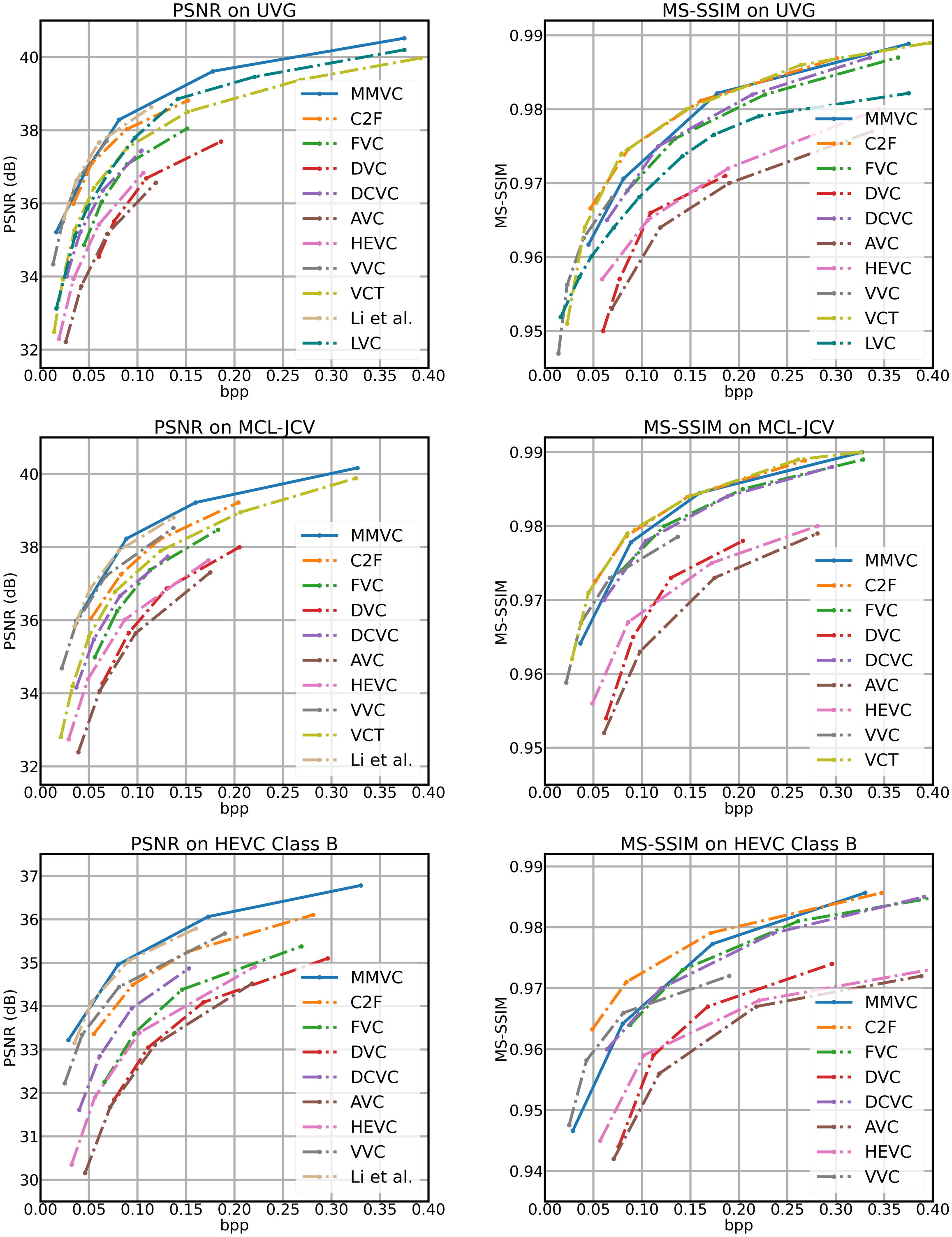

Figure 4.3: Rate-distortion curves measured on UVG, MCL-JCV, and HEVC Class B datasets in terms of PSNR and MS-SSIM.

Table 4.1: Performance results evaluated by BD-Bitrate (BDBR) with PSNR metric (%). The VVC (*low delay P* mode) is used as the anchor (*i.e.*, BDBR = 0 for VVC). Negative values imply bitrate saving compared to VVC, while positive values imply the opposite.

Dataset	MMVC	AVC	HEVC	C2F	FVC	DVC	DCVC	VCT	Li *et al.*	Liu *et al.*	HLVC	M-LVC	Agustsson *et al.*	RLVC
UVG	0.81	246.70	171.98	12.59	112.14	210.75	98.19	65.49	**-2.58**	73.97	183.90	161.38	175.24	148.72
MCL-JCV	**-16.73**	188.00	124.56	19.44	72.45	150.08	67.71	44.92	-11.25	–	–	–	118.78	–
HEVC Class B	**-28.28**	198.79	127.10	14.59	108.01	176.82	66.14	–	-20.29	–	–	63.72	–	–

Following the prior work [80], this work chooses to encode AVC and HEVC under *veryslow* mode with a GoP of 12/10. Compared with *veryfast*, the *veryslow* mode compresses video frames to a lower bitrate at the cost of longer encoding time, which aligns better with the target to generate high quality frames with the lowest bitrate but potentially longer latency. For VVC encoding, the *low delay P* mode is applied with a GoP size of 100 and set the IntraPeriod to be 4.

Figure 4.3 presents the rate-distortion curves measured with PSNR and MS-SSIM on UVG, MCL-JCV, and HEVC Class B datasets. The performance curves demonstrate that the proposed method outperforms the state-of-the-art learning-based approaches and conventional codecs in terms of PSNR for most of the bitrates covered. Particularly at 0.1 bit per pixel, this approach achieves 2dB quality improvement on average compared with HEVC (*veryslow*) for all testing datasets. Although not specially trained or fine-tuned for MS-SSIM, the presented method achieves comparative performance under the MS-SSIM metric consistently across all test datasets, especially in higher bpp regions. Table 4.1 shows the BD-Bitrate (BDBR) results in terms of PSNR anchored to VVC. The evaluation is based on UVG, MCL-JCV, and HEVC Class B datasets. The proposed method demonstrates competitive or superior performance compared to other schemes in Table 4.1.

4.5.2 Qualitative Analysis

Figure 4.2 shows example reconstructed frames from the UVG dataset. This approach exhibits similar quality (if not more visually appealing) with a bpp comparable to other codecs. The details of dynamic motions are well preserved at less than 0.1

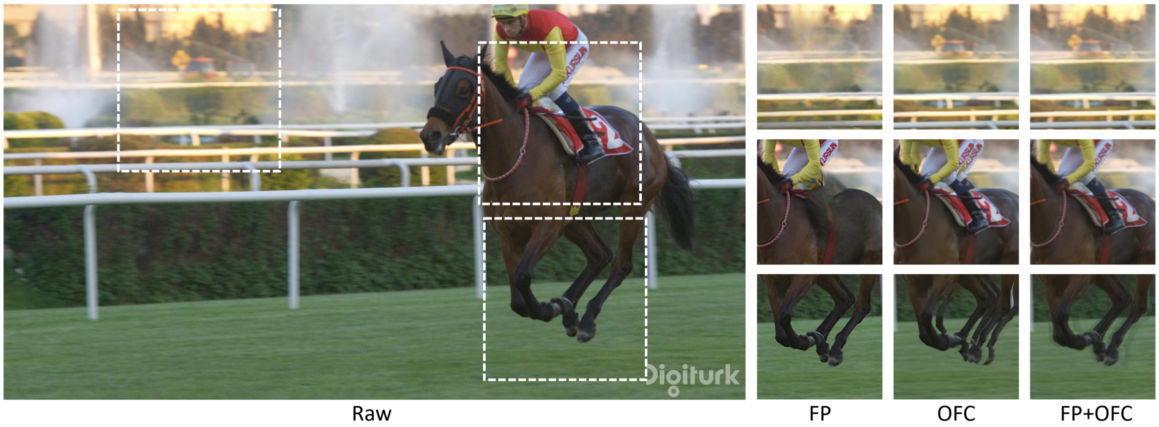

Figure 4.4: Reconstruction of multiple prediction modes without using the information from residual, where FP and OFC stand for *Feature Prediction* and *Optical Flow conditioned Feature Prediction* respectively.

bpp, demonstrating that MMVC can accurately predict the next frame by a combination of different modes. For the background field, the sharpness of the reconstructed frame is (subjectively) better than other standard codecs. The complementary nature of different prediction modes in the proposed scheme is visualized in Figure 4.4, which shows the decoded scenes directly obtained from the predicted features using a specific mode for the entire frame without residual compensation. Using FP mode only leads to sharper details in general, but it loses some contents such as the vehicle behind the fountain and the rider's leg. On the contrary, applying OFC mode only results in unfaithful reconstructions near the horse's legs. By adopting multiple prediction modes that complement each other (FP + OFC), the prediction is able to cover content variety in the original frame. The resulting residual is sparse and can be condensed to a shorter bitstream.

4.5.3 Ablation Studies

Mode Utilization: The result of mode utilization is summarized for the UVG dataset and 20 selected video sequences from the Kinetics dataset at relatively low bitrates of 0.081 and 0.088, respectively. As presented in Table 4.2, this analysis involves evaluating the separate performance of each prediction mode, examining

Table 4.2: Mode utilization and performance on the UVG dataset and Kinetics dataset, where FP, OFC, FPG, and S stands for *Feature Prediction*, *Optical Flow Conditioned Feature Prediction*, *Feature Propagation*, and *Skip* mode respectively.

Dataset	**UVG**				**Kinetics**			
Prediction mode	**PSNR (dB)**	**Removed channels**	**Bpp**	**Bitrate saving**	**PSNR (dB)**	**Removed channels**	**Bpp**	**Bitrate saving**
FP	38.0	23%	0.146	0%	37.7	29.8%	0.136	0%
OFC	36.9	47%	0.118	19.2%	37.4	49.3%	0.106	22.1%
FP+OFC	38.1	27%	0.096	34.3%	37.7	41.6%	0.099	27.2%
FP+OFC+FPG	38.2	27%	0.084	42.5%	37.7	43.5%	0.096	29.4%
FP+OFC+FPG+S	38.2	44%	0.081	44.5%	37.8	50.8%	0.088	35.3%
Mode utilization	**FP**	**OFC**	**FPG**	**S**	**FP**	**OFC**	**FPG**	**S**
	78.1%	10.6%	6%	5.3%	37.6%	38.3%	12%	11.6%

the impact of utilizing multiple prediction modes, and quantifying gains provided by skipping the encoding of unchanged blocks (*i.e.*, *Skip* mode).

As shown in Table 4.2, the *Feature Prediction* (FP) and *Optical Flow Conditioned Feature Prediction* (OFC) modes achieve comparable performance. For the UVG dataset, FP slightly outperforms OFC with 1dB higher PSNR and only 20% higher bitrate. Meanwhile, OFC is more favorable than FP for the Kinetics dataset. By adopting the ensemble of both modes (FP + OFC), the quality is preserved with an even lower bitrate, indicating that these two prediction modes can complement each other by capturing different motion patterns. Including the *Feature Propagation* (FPG) mode as an alternative prediction path further reduces the bitrate without degrading the quality. The compression ratio improves noticeably by introducing the *Skip* (S) mode as the final additional mode.

For sequences in the UVG dataset, the results showcase that usage of FP surpasses other modes significantly. However, the Kinetics dataset where the scenes are captured mostly by a fixed-view camera showcases higher utilization of the other modes. Fixed backgrounds in Kinetics sequences enable higher utilization of the *Skip* mode for significant bitrate reduction. In general, introducing S mode reduces the required bitrate for the same quality.

Table 4.3: Percentage of additional bitrate saving from density-adaptive entropy coding module compared to the baseline of using FP mode and dense path only on the Kinetics dataset. Note that the S mode is not included because it does not involve any residual coding.

Prediction mode	**Dense path only**	**Dense + sparse paths**
FP	0%	4.1%
FP+OFC	6.3%	21.2%
FP+OFC+FPG	6.5%	23.9%

Channel Removal and Adaptive Entropy Coding: One column in Table 4.2 shows the percentage of removed residual channels for various prediction modes. It shows that the percentage of removed residual channels is generally higher as the prediction becomes more accurate with mode selections from the full ensemble of available prediction modes (FP+OFC+FPG+S) compared to the single mode case (FP only).

Table 4.3 summarizes the additional bitrate saving by the density-adaptive entropy coding compared to the baseline of using FP mode and dense-path only. The evaluation is based on the Kinetics dataset at a relatively low bpp of 0.165, where the utilization of each prediction mode is well balanced. Allowing more prediction modes generally reduces the density of the residual. Thus the proposed density-adaptive entropy coding provides more significant savings (an additional 23.9% saving) when it is combined with the full ensemble of available prediction modes (FP+OFC+FPG). This saving includes the overhead of sending the density map and mode selection side information.

4.6 Conclusion

A dynamic mode selection-based video coding scheme, MMVC, is presented in this chapter. It can dynamically switch between multiple prediction paths adapting to distinct motion patterns that appear on different blocks within a frame. To further reduce the required bitrate for the prediction residual encoding, this work proposes a channel removal approach together with a density-adaptive entropy coding scheme to attain more compact residual representations when the residual entropy and density significantly vary block wise. Evaluations with various test datasets confirm that this method can attain outstanding rate-distortion tradeoffs.

CHAPTER V

H-PCC: Point Cloud Compression with Hybrid Mode Selection and Content Adaptive Down-sampling

5.1 Introduction

Light detection and ranging (LiDAR) sensor measures the travel time of laser beams to calculate the distances from the sensor to targets. It has been widely used in autonomous driving and augmented/virtual reality industries to provide a rich 3D understanding of the scene. However, as the resolution of LiDAR sensors keeps rising, the size of LiDAR data scales up to millions of points per 'frame', which poses a significant challenge for both memory and computation bandwidth. Thus, it is of great importance to develop an efficient LiDAR data compression algorithm to facilitate real-world 3D applications.

LiDAR data is commonly represented as 3D point clouds and can be directly used for various downstream tasks such as 3D object detection [73, 133, 22] and semantic segmentation [157, 172, 169]. Several works [57, 39, 117, 1] have been proposed to directly compress the LiDAR 3D points with octree structures. Despite the impressive performance of octree-based methods to compress a single frame, it is non-trivial to eliminate the temporal redundancy of consecutive LiDAR frames as the points in

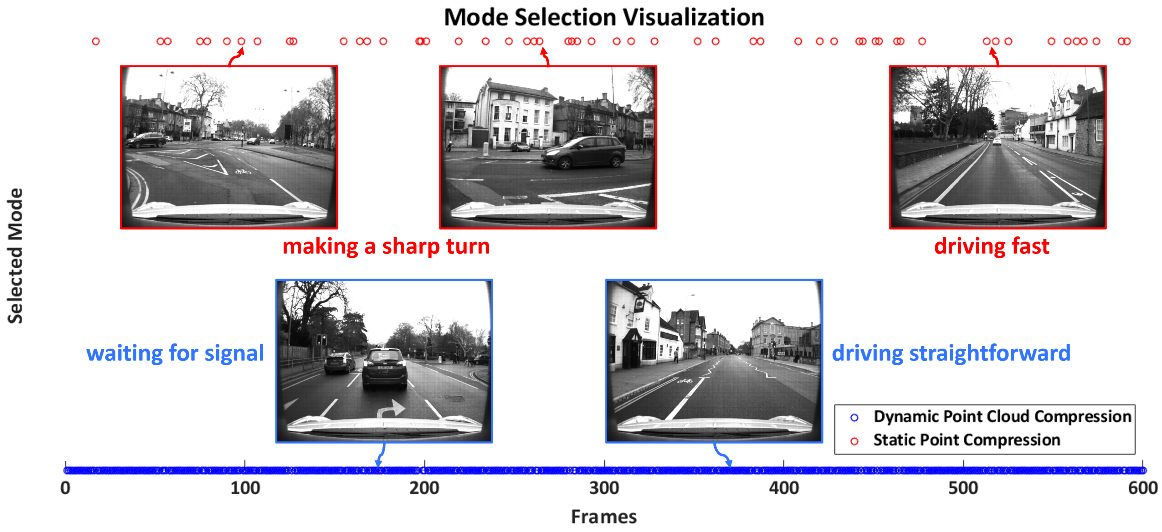

Figure 5.1: Mode selection results in a sequence from the Oxford dataset. The octree-based static point cloud compression path backs up in situations where sensor poses are not accurately measured or dynamic motion is not well estimated.

adjacent frames lie in different coordinate systems. To deal with the coordinate misalignment, Que *et al.* [117] utilizes the ground truth vehicle poses to map points to the same coordinate system for multiple frames. However, such pose information is not always available in real-world scenarios.

A straightforward representation for raw LiDAR measurements is through range images, where each pixel corresponds to a laser beam and stores the depth information. Compared with the 3D point cloud representation, range images are naturally more suitable for modeling the temporal correlation since they can be treated as a sequence of depth frames. Motivated by recent learning-based image and video compression frameworks [84, 85, 86], prior work [138, 153, 87] adopt similar techniques to compress LiDAR data as range image sequences. These methods mainly rely on the keyframe interpolation between past and future frames to handle sequential data, which is not directly applicable in a streaming setup where future frames are generally not available. Moreover, their keyframe selection policy is fixed to certain intervals regardless of the content of the scene.

A notable limitation in the majority of existing LiDAR data compression methods is their emphasis on reconstructing *all* LiDAR points and upholding consistent scanning density across the entire scene to comply with human visual perception and/or classical quality metrics. Yet, this focus overlooks the fact that point clouds are mostly meant to be consumed for machine vision tasks in practical usage, making it often unnecessary to retain all points in realistic applications. Recently, several works have delved into adaptive point cloud sampling strategies for various downstream tasks such as classification [24, 35] and 3D object detection [149]. Such adaptive down-sampling strategies can potentially enhance LiDAR data compression but remain unexplored in this field.

This paper presents a hybrid learning-based framework for sequential LiDAR data compression that benefits from both the point cloud octree-based static method and the range image-based dynamic method. Static LiDAR data compression involves a novel octree-based method, where 3D points are clustered and compressed using separate octrees. Dynamic LiDAR data compression adopts the range image representation and introduces an optical flow-based range image prediction network leveraging the approaches in recent video compression works [85, 86], where the residuals between the raw and the predicted range images are encoded. Besides, a novel image-guided point cloud adaptive sampling scheme is proposed to further enhance compression efficiency by reducing the point density in less important regions for downstream tasks. The method exhibits state-of-the-art performance compared to other baselines across multiple metrics. Additionally, the ablation study highlights the efficacy of combining the down-sampling with the hybrid compression scheme. The contributions are summarized as follows:

- This work introduces H-PCC, a hybrid LiDAR data compression framework adopting both point cloud (static) and range image (dynamic) representations. It primarily relies on a predictive method on the dynamic path to better exploit inter-frame correlations while engaging a quality-driven fallback to the more robust static path when needed (illustrated in Figure 5.1).

- Meanwhile there are existing works dedicated to either the static or dynamic form of LiDAR data compression methods adopted in H-PCC, this proposal improves both with new approaches. The clustering method on the static path can meaningfully reduce the depth of individual octree and also allow a parallelized execution scheme for a shorter decoding time. H-PCC dynamic path is the first work in this field that adopts a predictive model to deal with sequential range images to enable real-time streaming applications.
- To enhance compression for point clouds used in machine applications, a content-adaptive point cloud sampling approach is presented, which effectively removes redundant points in regions less critical to downstream tasks.

5.2 Related Work

5.2.1 LiDAR Point Cloud Compression

Octree based Methods: The octree structure has been widely adopted to compress any quantized point clouds not limited to LiDAR measurements. Various methods [41, 128, 58, 65, 46] have been proposed to combine the octree structure with hand-crafted entropy models. Recently, with the success of deep learning in image and video compression, several works [57, 16, 117, 134] propose to replace the hand-crafted entropy model with learning-based entropy estimation networks and have achieved impressive performance gains. Among these approaches, OctSqueeze [57] is the first learning-based method that uses a neural network to model the conditional entropy for each node based on its parent nodes. Following this method, MuSCLE [16] explores the spatial-temporal correspondence in LiDAR data streams. The VoxelContext-Net [117] further expands the context of each node in the entropy model to its local voxel, which can also be extended to LiDAR sequences by combining the information of temporally adjacent voxels. OctAttention [39] introduced an auto-regressive entropy model that utilizes a self-attention mechanism to explore dependencies in the context. Lately, EHEM [134] proposes a hierarchical attention structure that replaces the autoregressive contextual network with a grouped structure that allows linear complexity to the context scale.

Range Image based Methods: For LiDAR point clouds, another popular representation converts 3D points to the polar system and forms the range image. Since range images have regular grids, classical image compression methods (*e.g.*, JPEG, and PNG) and video compression methods (*e.g.*, H.264 [131], and H.265 [136]) have been explored to compress static range images [6, 145, 52] or sequential range images [109]. Rather than relying on classical codecs, Sun *et al.* [137] and R-PCC [152] utilize clustering algorithms to segment instance-based regions and encode the residual between the ground truth and averaged depth for each region. Recently, learning-based methods have also been explored. RIDDLE [175] proposes to train a pixelCNN-style decoder for autoregressive point predictions and encode the residual between ground truth and predictions. BIRD-PCC [87] mimics modern deep video compression systems to encode range image streams.

5.2.2 Point Cloud Down-Sampling

Point cloud down-sampling aims at reducing memory and computation costs for large-scale point cloud processing and analysis. Previous works [35, 110, 74, 24, 149, 173, 159, 163] focus on exploiting geometric and topological features directly from the point cloud itself, with the down-sampling results validated by downstream tasks such as classification and 3D object detection. Rather than abiding by this rationale of using LiDAR data only, there are works arguing that LiDARs often co-exist with other sensors such as a camera on typical real-world platforms, encouraging the use of fused information for point cloud compression. Jovanov *et al.* [63] come up with

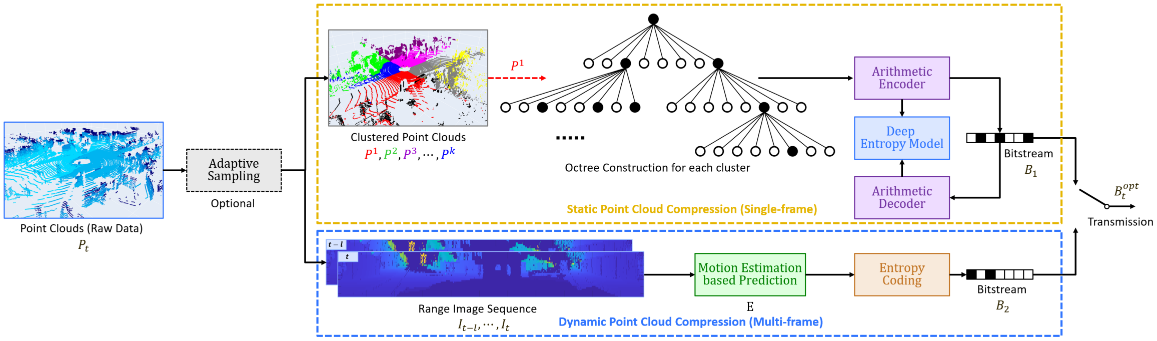

Figure 5.2: H-PCC framework with hybrid modes. It first down-samples the point cloud based on the object bounding boxes detected from the corresponding camera image. The compression process undergoes either: (a) octree-based single point cloud frame compression, or (b) point cloud to range image conversion followed by a prediction-based encoder adopted to reduce spatial and temporal redundancies. Mode selection is performed by examining the resulting bitrate and quality of the reconstruction.

a LiDAR sensor scanning scheme that picks up hints from the segmented camera images. In alignment with this idea, this work proposes an approach to detect RoIs in corresponding images and adaptively down-sample LiDAR points outside the RoI. This adaptive down-sampling strategy is demonstrated to have substantially improved the proposed point cloud compression framework by achieving a lower bitrate with almost no performance degradation in downstream tasks.

5.3 Method

5.3.1 H-PCC Framework

The H-PCC framework takes the point cloud data sequence $\{P_t\}_{t=1}^{N}$ captured by the LiDAR sensor as input, and produces a compressed bitstream B output that conveys essential information with minimal storage requirement. H-PCC is versatile, supporting both lossy and lossless configurations.

As illustrated in Figure 5.2, the proposed pipeline has two distinctive pathways side by side, both culminating in the selection of the better mode. The upper path is designed to compress single-frame point cloud data based on 3D geometrical analytics. The other is the dynamic compression path that exploits temporal and spatial correlations among sequential range images.

Based on the fact that point cloud data are mostly consumed by machine perception systems, the H-PCC pipeline is augmented with a content-adaptive point cloud down-sampling method. This strategy promotes point sparsity in less critical regions while maintaining the original density across areas pertinent to RoI. As evaluated in various experiments, the proposed down-sampling approach retains essential information with reduced data amount, attuning to the diverse requirements of machine-oriented vision applications.

5.3.2 Static Point Cloud Compression

The static point cloud compression module serves as a reliable fallback in scenarios where substantial errors arise in the alternative dynamic compression approach that operates based on a prediction model in the range image space to exploit the temporal redundancy. Unlike the dynamic counterpart, the static mode independently compresses a point cloud frame directly in the point cloud domain without using any previous frames.

Existing approaches such as OctSqueeze [57] employ an octree structure that uses positional correlation for storing point cloud data. However, such approaches may suffer information loss from quantization when the octree depth is insufficient, and adding more octree levels leads to an exponential increase in compressed data space and decoding time. The proposed approach differs from existing approaches as points are grouped to form smaller clusters and it explores correlations in relevant local contexts to offer a more bit-efficient solution.

As an initial step, the points are voxelized to foster a 3D geometry representation. With P representing the global point cloud, k clusters are formed and denoted by $\mathcal{P}^1$, $\mathcal{P}^2$, $\cdots$, $\mathcal{P}^k$. The process commences by clustering points at time step t into k classes using the iterative self-organizing data analysis technique Algorithm (ISODATA) [9]. ISODATA solves the classification problem iteratively by merging clusters when their separation in the multispectral feature space is below a certain threshold. It also outlines rules for splitting a single cluster into two, enhancing adaptability to complex multispectral patterns.

All points in each cluster $\mathcal{P}^i, i \in \{1, 2, \cdots, k\}$ are then enclosed in a cube, with the centroid c^i of the cluster and the X, Y, and Z dimensions of the cube are recorded. This cluster-based representation achieves a balance between the tree sparsity and voxel quantization resolution, therefore mitigating information loss while efficiently managing the data space.

Following OctSqueeze [57], H-PCC translates each point cluster $\mathcal{P}^i$ to an octree by progressively partitioning the corresponding cube space to finer octants until all points are represented in this hierarchy or the maximum number of tree levels is reached. To represent this spatial data structure, every internal node is assigned an 8-bit value to denote children's occupancy (each node in an octree has 8 children and each bit indicates the occupancy of a child node). The proposed approach incorporates learning-based entropy models to encode each node's 8-bit occupancy code, which is conditioned on the grid coordinates, level, and parent's occupancy. The occupancy data bitstream is separately encoded for each cluster and it is arranged for the level-order (*i.e.*, breadth-first) octree traversal.

To overcome octree depth limitations and promote proximity within each cluster, a two-stage global-to-local filtering step is adopted to sort out the outliers. Those outliers do not belong to any cluster and they are encoded separately so that the tree depth of each cluster does not grow excessively to include those outliers. The first global filtering stage identifies all points whose distance to the nearest neighbor exceeds a specified threshold t_g. This outlier collection can be represented as

$$\mathcal{P}^g = \{p \mid d(p, q) > t_g,\ \forall q \in P\ (p \neq q)\}, \tag{5.1}$$

where $d(\cdot, \cdot)$ is the Euclidean distance. The second stage performs cluster-level filtering to identify and remove outliers from each cluster to prevent the over-expansion of voxels. The outliers in this stage are determined by evaluating the point-to-centroid distance with a threshold t_l. The set of points obtained by this process is described as

$$\mathcal{P}^l = \bigcup_{i=1}^{k} \{p \mid d(p, c^i) > t_l,\ \forall p \in \mathcal{P}^i\}. \tag{5.2}$$

The outlier sets are combined with $\mathcal{P}^g$ and $\mathcal{P}^l$, and compressed together using the LZ77 method, separately from the 'regular' (non-outlier) points in the clusters $\mathcal{P}^i, i \in \{1, 2, \cdots, k\}$.

5.3.3 Dynamic Point Cloud Compression

Exploring the temporal correlation of consecutive LiDAR frames in the domain of point clouds is a complicated problem due to the sparsity of the point cloud in the 3D coordinate space. This challenge is addressed by adopting range images $I_t \in \mathbb{R}^{H \times W}$ as LiDAR data representations over time, where W refers to the width (*i.e.*, the number of beam columns in one sweep/frame) and H denotes the height (*i.e.*, number of beams in one column) of the range image. Each pixel in a range image stores the corresponding depth value of the laser beam. In reality, however, many datasets lack the raw sensor data in the range image format, posing a unique challenge for the implementation of the proposed algorithm. H-PCC works this around with a pre-processing procedure to convert point cloud data into range images, ensuring algorithm compatibility across diverse datasets.

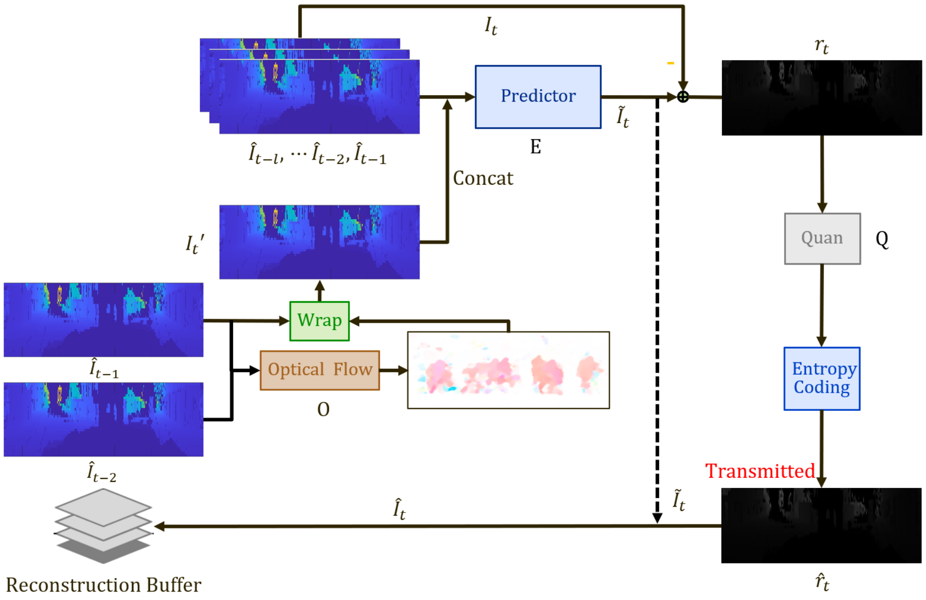

Figure 5.3: The range image prediction network operates by estimating the future frame based on the optical flow-warped frame and previously reconstructed frames. The prediction and reconstruction process is recurrently applied across the frame sequence.

Each valid pixel at position (i, j) with a finite depth in the range image of resolution $h \times w$ corresponds to a reflected laser shot with beam angle (θ, γ) in the point cloud. This relationship can be represented as:

$$i = \left[\frac{\delta_{max} - \gamma}{\delta_{max} - \delta_{min}} \cdot h \right], \; j = \left[\frac{\theta}{2\pi} \cdot w \right], \tag{5.3}$$

where the leaser beam diverges from δ_{min} to δ_{max}. Pixels that do not correspond to any points in the point cloud are set to zero. The conversion between a point $p = (x, y, z)$ in the 3D space point cloud and a range value ρ with laser beam angles (θ, γ) for the range image in the azimuth and elevation system is formulated as follows, where the mapping from a range image pixel value $I_t(i, j)$ to 3D space is denoted as $f(\cdot)$:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \rho \cdot \cos\theta \cdot \cos\gamma \\ \rho \cdot \sin\theta \cdot \cos\gamma \\ \rho \cdot \sin\gamma \end{bmatrix} \tag{5.4}$$

The mismatch between the point cloud and range image arises when there are fewer points in the point cloud than the total number of pixels in the range image due to unreflected laser beams. A binary mask M is employed to remark this, where 0 denotes a valid pixel belonging to the raw data $\mathcal{P}_t$, and 1 indicates an invalid point in order to address the data disparity.

$$m_t(i,j) = \begin{cases} 0, & f(I_t(i,j)) \in P_t \\ 1, & \text{otherwise} \end{cases} \tag{5.5}$$

The proposed dynamic method leverages the schemes in deep video compression which have shown significantly enhanced compression performance. In recent learning-based video compression methods [55], each frame is initially encoded into a lower-dimensional feature domain, and motion prediction compensation is performed in the feature domain. However, the aspect ratio of LiDAR range images often significantly differs from that of regular camera video frames as LiDAR range images usually have a much lower vertical resolution of only 32 or 64. This gap is closed by referring to pixel-level motion information to capture temporal locality. The optical flow $O(\cdot)$ between consecutive range image frames (I_{t-2}, I_{t-1}) is computed using [140], and then warped to the latter frame, leading to a preliminary estimation of the current frame t following Equation 5.6:

$$\bar{I}_t = Warp\left(O(\hat{I}_{t-2}, \hat{I}_{t-1}), \hat{I}_{t-1}\right). \tag{5.6}$$

Figure 5.3 visualizes the proposed dynamic compression method using an optical flow and prediction model in the range image domain. The predictor neural network $E(\cdot)$ forecasts the current range image frame using information from the previous l reconstructed frames $\hat{I}_{<t}$ together with the optical flow warped frame $\bar{I}_t$. As an outcome, a refined estimation of the current range image frame $\tilde{I}_t$ is attained as the

output of the predictor. The prediction process is written as follows where r_t is the residual error after prediction.

$$\tilde{I}_t = E(\hat{I}_{<t}, \bar{I}_t), \text{ with } r_t = I_t - \tilde{I}_t. \tag{5.7}$$

An accurate prediction yields a sparse residual r_t between the prediction and the raw frame. The invalid pixels are masked out to indicate that no residual is stored after the prediction for those pixels. Quantization and entropy coding are then performed to further reduce the size of the encoded residual $\hat{r}_t$.

Following [12], H-PCC incorporates an adaptive entropy coding model regulated by learned distribution parameters. Initially, the residual undergoes quantization to the nearest bin, replacing the values with the bin centroids. An arithmetic encoder then entropy codes the residual, guided by the estimated probabilities. Note that the binary point mask is also quantized and entropy coded under a similar procedure, albeit with distinct parameters for mask coding compared to residual image coding.

Equation 5.8 defines the network training process. The first term corresponds to the residual bitrate estimated by the parameterized entropy model network R. The second term signifies the weighted distortion between the reconstruction and the original range image frame. The distortion metric can be either MSE or PSNR. By adjusting the weight term λ, the models can be trained to achieve different bitrates. In Equation 5.8, ϕ and φ represent the model parameters for networks R and E, respectively.

$$\min_{\phi,\varphi} R_\phi\left(I_t - E_\varphi(\hat{I}_{<t}, \bar{I}_t)\right) + \lambda \cdot d\left((\tilde{I}_t + \hat{r}_t), I_t\right), \tag{5.8}$$

In the final step, the quantized and entropy-coded residual undergoes decoding, and it is added to the prediction result to form a reconstructed range image frame. Afterward, the reconstructed frame is converted back to the point cloud in 3D coordinates using Equation 5.4, allowing for quality evaluation using various metrics or downstream tasks.

5.3.4 Static vs. Dynamic Mode Selection

The dual path compression scheme allows the choice of either the static or dynamic mode to proceed with at each time step. However, finding a strategy to reliably identify a *better* mode for coding efficiency, rate, and distortion tradeoff is challenging, especially when the quality and bitrate are proportional to each other. This work presents three strategies for choosing a preferred mode for each frame.

Ideal RD-Tradeoff (Optimized) Strategy: This strategy weighs achieving a better rate-distortion tradeoff than coding efficiency. To facilitate a reasonable comparison between rate and distortion per frame, a tradeoff from the dynamic path is established by running multiple passes on every frame with different target bpps. The rate and distortion result obtained from a single run in the static path is then compared against this tradeoff to determine the best compressed bitstream to proceed with.

High-Quality (HQ) Driven Strategy: This strategy first collects the average performance of the octree-based static compression method acquired from a few samples in the test set and establishes a baseline bitrate vs. PSNR tradeoff from the static mode. The bitrate and PSNR results from the dynamic mode are compared to this baseline to encode each frame. If the dynamic mode turns out to provide a strictly better tradeoff to attain higher quality (*i.e.*, higher PSNR at equal or lower bitrate) for a particular frame compared to the baseline tradeoff curve, it selects the dynamic mode for that frame. The framework falls back to the static mode otherwise. This strategy generally improves the rate-distortion tradeoff compared to the baseline.

Low-Bitrate (LB) Driven Strategy: This strategy targets aggressive bitrate saving. Hence in this mode, all frames are compressed using the dynamic compression path, except for frames whose reconstruction quality from the dynamic mode is lower than a pre-defined threshold. The static compression path serves as a fallback method for those frames where the prediction method in the dynamic mode does not perform

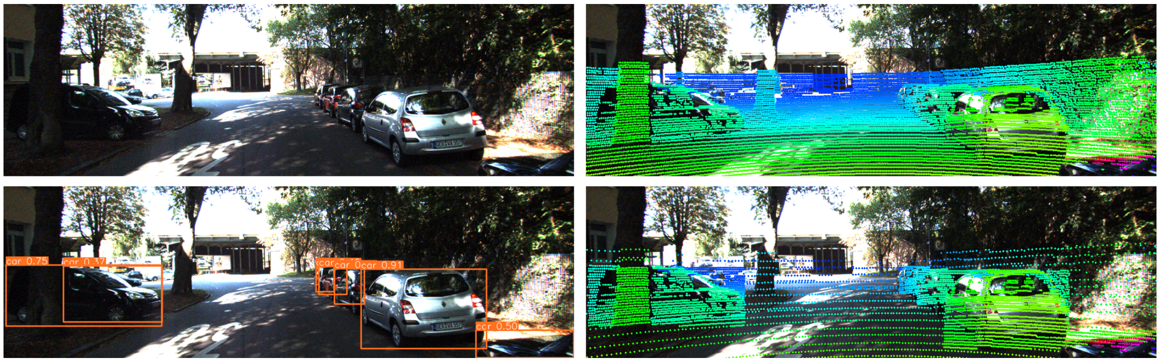

Figure 5.4: Top left: original image. Bottom left: detection results from YOLOv8x [60]. Top right: original point cloud projected onto the image. Bottom right: point cloud down-sampled with the proposed approach projected onto the image.

well, leading to significant quality degradation compared to the average case.

Note that the first strategy can achieve the best quality overall but sacrifices runtime efficiency, it is therefore undesired in practice and only used for reference in experiments. Both HQ and LB strategies always execute the dynamic mode for each frame, but the static mode is optionally executed only when a particular condition specified in each strategy is met.

5.3.5 Content Adaptive Point Cloud Down-Sampling

Point cloud is a fundamental 3D representation used in a variety of vision tasks, such as 3D object detection [73, 168], 3D segmentation [176], and simultaneous localization and mapping (SLAM) [170]. However, not all points within the point cloud contribute significantly to these tasks. Take 3D object detection as an example, the points situated on and near the objects of interest are of paramount importance but points outside are not. There exists an opportunity to lower the point cloud density across regions with no interest. Subsequently, the H-PCC framework has an optional pre-processing block as shown in Fig. 5.2 to adaptively down-sample the input point cloud before compression with a goal to achieve a reduced bitrate without compromising the performance of downstream tasks. To assist the adaptive down-sampling process, the proposed method detects the RoI from camera images and integrates this information into the point cloud compression flow. This LiDAR-camera fusion technique forges a more informed down-sampling approach, contributing to the coding effectiveness of the H-PCC framework.

RoI Detection: Both 2D object detection [120, 60, 88] and semantic segmentation [91] are capable of identifying camera image RoIs. Learning-based 2D object detection is well-studied with numerous state-of-the-art algorithms publicly available, and it is proven to be sufficient for the intended purpose. In the experiments, the YOLOv8x model [60] is employed to detect object bounding boxes, as illustrated in the bottom left quarter of Figure 5.4. The 2D image RoIs are represented by bounding boxes surrounding the detected objects.

Point Cloud Down-sampling: All 3D points are projected onto the corresponding camera image domain, preserving the ones that are within a detected RoI while uniformly down-sampling those that are falling outside the ROIs. This process is facilitated by converting all points into polar coordinates before compression, leading to a straightforward implementation of sampling of points outside the RoI. Figure 5.4 illustrates the outcome of the proposed down-sampling strategy (bottom right) in contrast with the projected original point cloud (top right).

5.4 Experimental Setup

5.4.1 Implementation and Training Details

The static point cloud compression path incorporates the ISODATA approach for point clustering. The parameters required by ISODATA are listed as follows: the initial number of clusters (N_c), the minimum number of points in each cluster (T_n), the maximum standard deviation in each cluster (T_e), and the lower bound of the distance between two clusters (T_c). The parameters are set empirically as

$\{N_c, T_n, T_e, T_c\} = \{5, 100, 2.16, 8\}$ to produce effective clustering results. Note that K-means clustering is employed to replace ISODATA at extremely low bitrate configurations, where the number of clusters is constrained to a small K (< 5). The octree in each cluster is parameterized by the depth level D to govern the tradeoff between bitrate and precision, with $D \in \{6, 8, 10, 12, 14, 16\}$. The entropy model network and hyperparameter settings are aligned with those detailed in [57].

For the dynamic compression mode, H-PCC adopts an optical flow-based predictor and a learning-based entropy model for range image sequence compression. The two-stage training flow is initiated by training the predictor independently on regular video frames. Subsequently, the predictor is fine-tuned using a pre-trained optical flow model [140] on range images. In the second stage, both the predictor and the entropy model are trained jointly to minimize Equation 5.8. The tradeoff between bitrate and quality is balanced by adjusting the weight term λ from $\{16, 64, 256, 1024, 2048, 4096\}$. The predictor is optimized for 2 million steps with a batch size of 16, followed by fine-tuning for an additional 1 million steps. The initial learning rate is 10^{-4} and is decayed by half every 0.1 million steps.

Evaluation of the proposed adaptive point cloud down-sampling technique is conducted with the PointPillars [73] model. This experiment selects 3D object detection as the downstream task for performance benchmarking, and all models are trained and evaluated with the MMDetection3D toolbox [28].

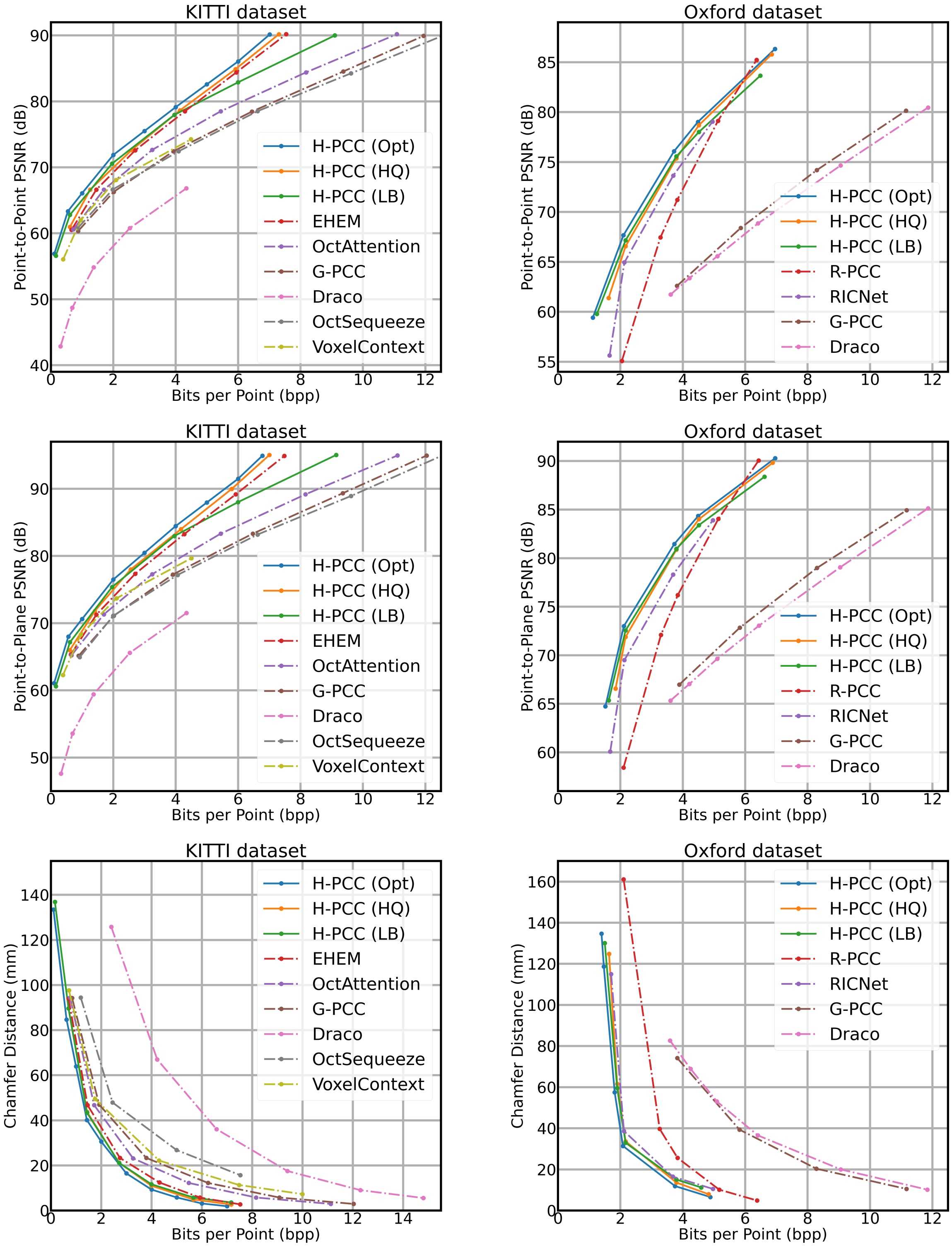

Figure 5.5: Rate-distortion performance of different methods on the KITTI dataset (top) and the Oxford dataset (bottom).

5.4.2 Datasets

The performance of H-PCC compression flow is assessed with the SemanticKITTI [13] and Oxford [95] datasets. The SemanticKITTI dataset comprises 23,311 LiDAR scans from 22 sequences collected in vehicle driving scenes. The Velodyne LiDAR on the KITTI vehicle captures 64 beam scans per frame. This dataset is a subset of the KITTI dataset with semantic labels. For a fair comparison, sequences 00 to 10 are designated for training and the remaining sequences for evaluation as in previous approaches [23, 134].

The Oxford Robotcar dataset features raw LiDAR data gathered from dual Velodyne HDL-32E sensors in driving scenes around Oxford, UK. The raw data is stored in a range image format with depth information, reflection intensity, and timestamps stored in separate channels. The range image in the Oxford dataset has an original resolution of 1080×32, covering over 1,000 km of recorded driving.

For evaluation based on downstream tasks, the Pointpillars 3D object detection model is trained using the original point cloud data from all training samples. The adaptive point cloud down-sampling strategy is then assessed using all validation samples.

5.4.3 Metrics

Three metrics are selected to gauge the reconstruction quality: point-to-point PSNR, point-to-plane PSNR, and Chamfer distance, all designed to measure the distortion of the reconstruction. To guarantee a fair comparison, the PSNR calculation method is similar to previous works [57, 16, 134], which sets the peak value to be 59.7 for evaluation on the KITTI dataset. However, for the Oxford dataset evaluation, there is no reference peak value reported in prior works. Hence this work dynamically sets the peak value for each frame as the maximum value in that frame. Additionally, BDBR is employed to reflect the relative bitrate savings under similar quality

Table 5.1: Performance results evaluated by BDBR (%). G-PCC is set as the anchor (*i.e.*, BDBR = 0) for evaluation on both datasets. Negative values imply relative bitrate savings.

Dataset	Method	Point-to-point PSNR	Point-to-plane PSNR	Chamfer Distance
KITTI	Draco	133.17	134.18	167.90
	OctSqueeze	0.79	3.52	25.64
	VoxelContext	-13.83	-24.22	-2.63
	OctAttention	-16.36	-16.30	-9.69
	EHEM	-32.82	-32.84	-25.83
	H-PCC Static	-26.73	-26.18	-19.53
	H-PCC Dynamic	-33.96	-33.57	-26.82
	H-PCC LB	**-38.58**	**-38.14**	**-32.49**
	H-PCC HQ	**-37.35**	**-38.80**	**-30.92**
Oxford	Draco	5.88	8.20	4.86
	R-PCC	-53.96	-44.55	-39.57
	RICNet	-56.26	-54.13	-58.59
	H-PCC Static	-46.23	-46.87	-47.87
	H-PCC Dynamic	-55.18	-55.69	-57.63
	H-PCC LB	**-58.27**	**-59.97**	**-61.51**
	H-PCC HQ	**-57.43**	**-59.18**	**-61.89**

within the tested range. For the evaluation of 3D object detection, mean average precision (mAP) is reported on all instances of moderate difficulty for three classes (cars, pedestrians, and cyclists) in the KITTI dataset.

5.5 Experimental Results

5.5.1 Quantitative Results

Figure 5.5 presents quantitative results for R-D tradeoffs. The H-PCC hybrid mode outperforms existing conventional and learning-based algorithms. Across various measurements such as point-to-point PSNR, point-to-plane PSNR, and Chamfer distance, H-PCC consistently achieves superior performance over a wide range of bitrates measured in bits per point (bpp), manifesting the flexibility of the hybrid approach.

The H-PCC framework achieves a 38.58% gain in BDBR (in terms of point-to-point PSNR) compared to G-PCC in the KITTI dataset, indicating a remarkable

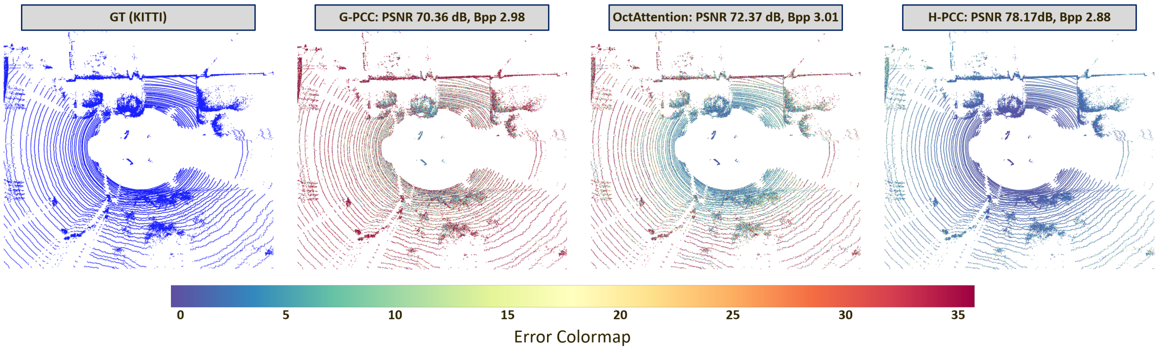

Figure 5.6: Visualized compression results of OctAttention [39], G-PCC [46], and the proposed method on the SemanticKITTI dataset. The unit of error distance is mm.

average bitrate saving under similar quality. The state-of-the-art performance on the KITTI and Oxford datasets underscores the effectiveness of the proposed compression method, especially from the dynamic mode, and the advantages of encoding point cloud temporal correlation in the range image domain.

5.5.2 Qualitative Analysis

Figure 5.6 visualizes the reconstruction of a compressed point cloud from the KITTI dataset, and it color-codes each point based on the reconstruction error. For a valid comparison, the reconstruction results from other compression schemes operating under a similar bitrate are also included. The visualization reveals that H-PCC attains higher quality with less bitrate, showcasing the effectiveness of the proposed H-PCC.

5.5.3 Runtime Analysis

Table 5.2 presents the average encoding and decoding time for components in the H-PCC pipeline, which runs with an Nvidia A40 GPU and an Intel Xeon Gold 6226R CPU. It is worth noting that the average decoding time on the static path is significantly shorter than that of other existing octree compression methods, due to simplifications and tree-depth reduction contributed by the clustering-based approach. The run time of this hybrid strategy is shorter than the sum of the run times of the static and dynamic modes because it executes both modes only when the dynamic mode fails to attain the target quality. This experiment also confirms that the execution rate of the static mode is relatively low (11.5% on average) per frame. Therefore, the overall complexity of the hybrid scheme does not significantly increase beyond that of the dynamic mode.

Table 5.2: Average runtime of encoder/decoder components.

	Static	Dynamic	Hybrid	Down-sample	Overall
Encoder	67.5 ms	57.9 ms	65.4 ms	3.5 ms	68.9 ms
Decoder	496.8 ms	71.3 ms	113.9 ms	N/A	113.9 ms

5.5.4 Ablation Studies

Mode Selection: For both selection strategies, H-PCC predominantly relies on the dynamic mode, given that the range image based predictive approach performs well in the majority of the testing frames. While the static path is not as effective from the average performance perspective, it supplements the quality loss appearing in the dynamic mode from time to time to obtain the enhanced combined performance reported in Table 5.1.

Content Adaptive Down-Sampling: Various adaptive down-sampling ratios (2× or 3× both horizontally and vertically) are explored for the regions that are not RoIs. This experiment compares the 3D object detection performance against the raw point cloud without down-sampling or with 2× uniform down-sampling. Figure 5.7 reports the bitrate vs. mAP results. The same set of H-PCC models is adopted with different compression ratios to compress and reconstruct all examined point clouds (remarked as *H-PCC* in Figure 5.7) regardless of setups. Note that on average the numbers of remaining points in the adaptive 3×3 case and the uniform 2×2 case are roughly equal. The reconstructed point clouds exhibit similar mAP with adaptive down-sampling to achieve 1.7-2.5× higher compression ratios compared to the vanilla H-PCC without adaptive down-sampling.

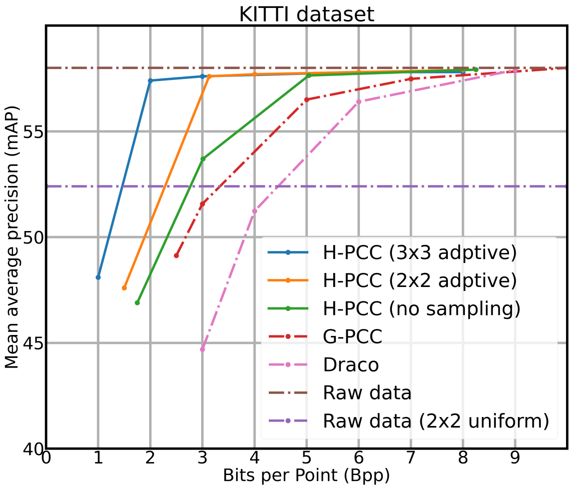

Figure 5.7: 3D object detection mAP for Pointpillars with the uncompressed and reconstructed point cloud, along with the corresponding compression gain on the KITTI dataset.

5.6 Conclusion

This work introduces a hybrid LiDAR data compression scheme that dynamically switches between static point cloud compression and a dynamic compression path, adapting to different scenarios. Considering that machine vision is the primary consumer of LiDAR data in realistic situations, a content-guided point cloud sampling strategy is proposed to enhance the hybrid mode compression framework. Experiments with various test datasets consolidate the proposed method from several practical perspectives.

CHAPTER VI

Conclusions and Future Work

6.1 Summary

This thesis presents a set of deep networks and methods for signal representation learning and compression. It covers the investigation of various signal types, including speech audio, image, video, and 3D LiDAR data, which almost span the majority of web and media traffic nowadays. Driven by the fact that machine/downstream neural networks would be the main consumer of signal in certain contexts, some of these works not only dedicate to finding the high fidelity solution from the human perspective but also explore ways to preserve the reconstruction quality in downstream applications, and treat the later as a crucial offering of the learned compression frameworks.

Chapter II introduces a GAN-based speech and image compression algorithm that produces compressed data by iteratively searching for the optimal representation through back-propagation. ADMM is incorporated into this flow to perform non-uniform quantization progressively. This method unifies speech audio and image by transforming the audio signal from temporal to frequency domain, and it guarantees the reconstruction quality by including the discriminator loss term in the loss function. Extensive studies validate that this approach is efficacious under a wide collection of subjective and objective metrics, as well as the metrics induced from the learning-based phoneme recognition and image classification tasks.

Chapter III presents a deep framework that accomplishes video prediction and compression in feature domain. The method simultaneously learns a transform of the spatial domain video frames into a lower-dimensional representation as well as a temporally-conditioned probabilistic model. In-depth experiments and analysis consolidate that prediction in feature domain can well capture the temporal correlations without the loss of generality, which can be leveraged to accomplish meaningful perceptual tasks such as anomaly detection without the need to obtain the spatial domain reconstructions. This finding can potentially reduce the latency, bandwidth, and power requirements with the presence of the proposed codec as part of the system.

A dynamic mode selection-based video coding scheme, MMVC, is presented in Chapter IV. It switches between multiple prediction paths on the fly to fit different motion patterns that appear on each block within a frame. To further reduce the required bitrate for the prediction residual encoding, this work proposes a channel removal approach together with a density-adaptive entropy coding scheme to attain more compact residual representations when the residual entropy and density significantly vary block wise.

Chapter V proposes a deep LiDAR data compression pipeline that addresses the weakness in existing works with several innovative designs. Given that LiDAR data naturally has a point cloud representation and a range image representation, this framework uses a dual path construction to leverage both the robustness offered by point clouds and the high compression ratio offered by range images. On the static path that deals with point clouds, the proposed approach significantly improves decoding efficiency by clustering. On the dynamic path that deals with range images, a predictive method is presented to be applicable to a streaming setup. An adaptive point cloud sampling unit is introduced to aggressively reduce the point density while retaining the quality in machine perception when sensor fusion is at hand.

6.2 Future Work

In recent years, learning-based image and video compression techniques have achieved significant advancements. However, one disadvantage of these approaches compared to conventional codecs is the encoding and decoding time efficiency. It is crucial and intriguing to address this issue by streamlining network architectures in the future. Techniques such as network distillation and Neural Architecture Search (NAS) have shown promising achievements in overcoming this challenge.

The growing interest in 3D representation has led to exploration across various types, including Signed Distance Functions (SDF), Neural Radiance Fields (NeRF), and Mesh data. Investigating the spatial and temporal correlations within these domains holds the potential for developing applications for VR/AR devices and enhancing the reconstruction of real-world scenes with improved speed and quality.

As Large Language Models (LLM) gain popularity, translating images or videos into prompts offers the possibility of achieving extremely high compression ratios. Utilizing LLM, a single frame or video sequence can be condensed into a concise set of words. Identifying crucial latent representations is pivotal in this process.

Recent research indicates that video content generation heavily relies on learning feature representations, with optimal mapping from the latent domain to the pixel domain. Methods like Sora [18] leverage diffusion models to generate video clips, demonstrating the strong temporal correlations present in the feature domain. Thus, effective feature representation learning significantly enhances the quality of generated videos.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Draco. `https://github.com/google/draco`. Accessed: 2021-09-28.

[2] *Video Trace Library*. http://trace.eas.asu.edu/index.html.

[3] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008.

[4] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 221–231, 2019.

[6] Jae-Kyun Ahn, Kyu-Yul Lee, Jae-Young Sim, and Chang-Su Kim. Large-scale 3d point cloud compression using adaptive radial distance prediction in hybrid coordinate domains. *IEEE Journal of Selected Topics in Signal Processing*, 9(3):422–434, 2014.

[7] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018.

[8] Mohammad Haris Baig, Vladlen Koltun, and Lorenzo Torresani. Learning to inpaint for image compression. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1246–1255. Curran Associates, Inc., 2017.

[9] Geoffrey H Ball, David J Hall, et al. *ISODATA, a novel method of data analysis and pattern classification*, volume 4. Stanford research institute Menlo Park, CA, 1965.

[10] J Ballé, V Laparra, and E P Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *Proc. 32nd Picture Coding Symposium*, Nuremberg, Germany, Dec 2016. IEEE Signal Processing Society.

[11] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.

[12] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[13] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[14] Fabrice Bellard. *BPG Image Fromat*. https://bellard.org/bpg/.

[15] Bruno Bessette, Redwan Salami, Roch Lefebvre, Milan Jelinek, Jani Rotola-Pukkila, Janne Vainio, Hannu Mikkola, and Kari Jarvinen. The adaptive multirate wideband speech codec (amr-wb). *IEEE transactions on speech and audio processing*, 10(8):620–636, 2002.

[16] Sourav Biswas, Jerry Liu, Kelvin Wong, Shenlong Wang, and Raquel Urtasun. Muscle: Multi sweep compression of lidar using deep entropy models. *Advances in Neural Information Processing Systems*, 33:22170–22181, 2020.

[17] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 01 2011.

[18] Tim Brooks, Bill Peebles, Connor Homes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Wing Yin Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[19] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[20] Milos Cernak, Alexandros Lazaridis, Afsaneh Asaei, and Philip N Garner. Composition of deep and spiking neural networks for very low bit rate speech coding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2301–2312, 2016.

[21] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma. Deepcoder: A deep neural network based video compression. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2017.

[22] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9775–9784, 2019.

[23] Zhili Chen, Zian Qian, Sukai Wang, and Qifeng Chen. Point cloud compression with sibling context and surface priors. In *European Conference on Computer Vision*, pages 744–759. Springer, 2022.

[24] Ta-Ying Cheng, Qingyong Hu, Qian Xie, Niki Trigoni, and Andrew Markham. Meta-sampler: Almost-universal yet task-oriented sampling for point clouds. In *European Conference on Computer Vision*, pages 694–710. Springer, 2022.

[25] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learning image and video compression through spatial-temporal energy compaction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[27] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. *CoRR*, abs/1701.01546, 2017.

[28] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020.

[29] Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai. Asymmetric gained deep image compression with continuous rate adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10532–10541, June 2021.

[30] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[32] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1174–1183, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[33] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. Neural inter-frame compression for video coding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[34] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[35] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2760–2769, 2019.

[36] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[37] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 64–72. Curran Associates, Inc., 2016.

[38] David A Freedman. *Statistical models: theory and practice.* cambridge university press, 2009.

[39] Chunyang Fu, Ge Li, Rui Song, Wei Gao, and Shan Liu. Octattention: Octree-based large-scale contexts model for point cloud compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 625–633, 2022.

[40] Cristina Gârbacea, Aäron van den Oord, Yazhe Li, Felicia SC Lim, Alejandro Luebs, Oriol Vinyals, and Thomas C Walters. Low bit-rate speech coding with vq-vae and a wavenet decoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 735–739. IEEE, 2019.

[41] Diogo C Garcia and Ricardo L de Queiroz. Intra-frame context-based octree coding for point-cloud geometry. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1807–1811. IEEE, 2018.

[42] Adam Golinski, Reza Pourreza, Yang Yang, Guillaume Sautiere, and Taco S. Cohen. Feedback recurrent autoencoder for video compression. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[43] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[44] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[45] Robert M. Gray and David L. Neuhoff. Quantization. *IEEE transactions on information theory*, 44(6):2325–2383, 1998.

[46] Danillo Graziosi, Ohji Nakagami, Shinroku Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. An overview of ongoing point cloud compression standardization activities: Video-based (v-pcc) and geometry-based (g-pcc). *APSIPA Transactions on Signal and Information Processing*, 9:e13, 2020.

[47] Stefan Gumhold, Zachi Kami, Martin Isenburg, and Hans-Peter Seidel. Predictive point-cloud compression. In *ACM SIGGRAPH 2005 Sketches*, pages 137–es. 2005.

[48] Amirhossein Habibian, Ties Van Rozendaal, Jakub Tomczak, and Taco Cohen. Video compression with rate-distortion autoencoders. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[49] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[50] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, and Larry S. Davis. Learning temporal regularity in video sequences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[52] Hamidreza Houshiar and Andreas Nüchter. 3d point cloud compression using conventional image compression for efficient data transmission. In *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*, pages 1–8. IEEE, 2015.

[53] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu. Improving deep video compression by resolution-adaptive flow coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 193–209, Cham, 2020. Springer International Publishing.

[54] Zhihao Hu, Guo Lu, Jinyang Guo, Shan Liu, Wei Jiang, and Dong Xu. Coarse-to-fine deep video coding with hyperprior-guided mode prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5921–5930, June 2022.

[55] Zhihao Hu, Guo Lu, and Dong Xu. Fvc: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 1502–1511, June 2021.

[56] H Huang, H Shu, and R Yu. Lossless audio compression in the new ieee standard for advanced audio coding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6934–6938. IEEE, 2014.

[57] Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. Octsqueeze: Octree-structured entropy model for lidar compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1313–1323, 2020.

[58] Yan Huang, Jingliang Peng, C-C Jay Kuo, and M Gopi. A generic scheme for progressive point cloud coding. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):440–453, 2008.

[59] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1385–1422, 1993.

[60] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, jan 2023.

[61] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[62] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[63] Ljubomir Jovanov, Wei-Yu Lee, and Wilfried Philips. Adaptive point cloud acquisition and upsampling for automotive lidar. *Applied Optics*, 62(17):F8–F13, 2023.

[64] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*, 2018.

[65] Julius Kammerl, Nico Blodow, Radu Bogdan Rusu, Suat Gedikli, Michael Beetz, and Eckehard Steinbach. Real-time compression of point cloud streams. In *2012 IEEE international conference on robotics and automation*, pages 778–785. IEEE, 2012.

[66] Srihari Kankanahalli. End-to-end optimized speech coding with deep neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2521–2525. IEEE, 2018.

[67] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE, 2017.

[68] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[69] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature biotechnology*, 26(9):1011–1013, 2008.

[70] W Bastiaan Kleijn, Felicia SC Lim, Alejandro Luebs, Jan Skoglund, Florian Stimberg, Quan Wang, and Thomas C Walters. Wavenet based low rate speech coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 676–680. IEEE, 2018.

[71] Donald E Knuth. Dynamic huffman coding. *Journal of algorithms*, 6(2):163–180, 1985.

[72] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Modenet: Mode selection network for learned video coding. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2020.

[73] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[74] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[75] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[76] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[77] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018.

[78] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *International Conference on Learning Representations*, 2019.

[79] Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin. Extremely low bit neural network: Squeeze the last bit out with admm. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[80] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[81] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 1503–1511, New York, NY, USA, 2022. Association for Computing Machinery.

[82] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3554, 2020.

[83] Alexander Liu, Hung-yi Lee, and Lin-shan Lee. Adversarial training of end-to-end speech recognition using a criticizing language model. In *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[84] Bowen Liu, Ang Cao, and Hun-Seok Kim. Unified signal compression using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3177–3181. IEEE, 2020.

[85] Bowen Liu, Yu Chen, Shiyu Liu, and Hun-Seok Kim. Deep learning in latent space for video prediction and compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 701–710, June 2021.

[86] Bowen Liu, Yu Chen, Rakesh Chowdary Machineni, Shiyu Liu, and Hun-Seok Kim. Mmvc: Learned multi-mode video compression with block-based prediction mode selection and density-adaptive entropy coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18487–18496, June 2023.

[87] Chia-Sheng Liu, Jia-Fong Yeh, Hao Hsu, Hung-Ting Su, Ming-Sui Lee, and Winston H Hsu. Bird-pcc: Bi-directional range image-based deep lidar point cloud compression. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[88] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[89] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection - A new baseline. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6536–6545. IEEE Computer Society, 2018.

[90] Salvator Lombardo, Jun Han, Christopher Schroers, and Stephan Mandt. Deep generative video compression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9287–9298. Curran Associates, Inc., 2019.

[91] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[92] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *2013 IEEE International Conference on Computer Vision*, pages 2720–2727, 2013.

[93] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. Content adaptive and error propagation aware deep video compression. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, pages 456–472. Springer, 2020.

[94] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[95] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[96] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981, 2010.

[97] Andrés Marafioti, Nicki Holighaus, Nathanaël Perraudin, and Piotr Majdak. Adversarial generation of time-frequency features with application in audio synthesis. *arXiv preprint arXiv:1902.04072*, 2019.

[98] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[99] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[100] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[101] Fabian Mentzer, George Toderici, David Minnen, Sung-Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. Vct: A video compression transformer, 2022.

[102] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020.

[103] A. Mercat, M. Viitanen, and J. Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. *ACM Multimedia System Conference*, 2020.

[104] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, MMSys '20, page 297–302, New York, NY, USA, 2020. Association for Computing Machinery.

[105] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10771–10780. Curran Associates, Inc., 2018.

[106] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[107] Carina Mood. Logistic regression: Why we cannot do what we think we can do, and what we can do about it. *European sociological review*, 26(1):67–82, 2010.

[108] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, and Ronald S Bultje. The latest open-source video codec vp9 - an overview and preliminary results. 2013.

[109] Fabrizio Nenci, Luciano Spinello, and Cyrill Stachniss. Effective compression of range data streams for remote robot operations using h. 264. In *2014 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pages 3794–3799. IEEE, 2014.

[110] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. Adaptive hierarchical down-sampling for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12956–12964, 2020.

[111] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[112] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[113] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[114] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. A fast griffin-lim algorithm. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013.

[115] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[116] Zdeněk Průša, Peter Balazs, and Peter Lempel Søondergaard. A noniterative method for reconstruction of phase from stft magnitude. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5):1154–1164, 2017.

[117] Zizheng Que, Guo Lu, and Dong Xu. Voxelcontext-net: An octree based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6042–6051, 2021.

[118] Rassol Raissi. The theory behind mp3. *MP3'Tech*, 2002.

[119] Mirco Ravanelli, Titouan Parcollet, and Yoshua Bengio. The pytorch-kaldi speech recognition toolkit. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6465–6469. IEEE, 2019.

[120] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[121] Ao Ren, Tianyun Zhang, Shaokai Ye, Jiayu Li, Wenyao Xu, Xuehai Qian, Xue Lin, and Yanzhi Wang. ADMM-NN: an algorithm-hardware co-design framework of dnns using alternating direction method of multipliers. *CoRR*, abs/1812.11677, 2018.

[122] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2922–2930. JMLR. org, 2017.

[123] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G. Anderson, and Lubomir Bourdev. Learned video compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[124] Jorma Rissanen and Glen G Langdon. Arithmetic coding. *IBM Journal of research and development*, 23(2):149–162, 1979.

[125] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001.

[126] GREG Roelofs. Png lossless image compression. *Lossless Compression Handbook*, pages 371–390, 2002.

[127] S. Santurkar, D. Budden, and N. Shavit. Generative compression. In *2018 Picture Coding Symposium (PCS)*, pages 258–262, 2018.

[128] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. *PBG@ SIGGRAPH*, 3, 2006.

[129] Bernhard Schölkopf, Christopher JC Burges, and Alexander J Smola. *Advances in kernel methods: support vector learning*. MIT press, 1999.

[130] Manfred Schroeder and BS Atal. Code-excited linear prediction (celp): High-quality speech at very low bit rates. In *ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 937–940. IEEE, 1985.

[131] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the scalable video coding extension of the h. 264/avc standard. *IEEE Transactions on circuits and systems for video technology*, 17(9):1103–1120, 2007.

[132] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. Emerging mpeg standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148, 2018.

[133] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10529–10538, 2020.

[134] Rui Song, Chunyang Fu, Shan Liu, and Ge Li. Efficient hierarchical entropy model for learned point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14368–14377, 2023.

[135] Gary Sullivan. Versatile video coding (vvc) arrives. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–1, 2020.

[136] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[137] Xuebin Sun, Han Ma, Yuxiang Sun, and Ming Liu. A novel point cloud compression algorithm based on clustering. *IEEE Robotics and Automation Letters*, 4(2):2132–2139, 2019.

[138] Xuebin Sun, Sukai Wang, Miaohui Wang, Zheng Wang, and Ming Liu. A novel coding architecture for lidar point cloud sequence. *IEEE Robotics and Automation Letters*, 5(4):5637–5644, 2020.

[139] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.

[140] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[141] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017.

[142] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

[143] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[144] Jean-Marc Valin, Koen Vos, and Timothy Terriberry. Definition of the opus audio codec. *IETF, September*, 2012.

[145] Peter Van Beek. Image-based compression of lidar sensor data. *Electronic Imaging*, 31:1–7, 2019.

[146] Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[147] Jan Van Leeuwen. On the construction of huffman trees. In *ICALP*, pages 382–410, 1976.

[148] Emmanuel Vincent, Maria Jafari, and Mark Plumbley. Preliminary guidelines for subjective evalutation of audio source separation algorithms. 2006.

[149] Niclas Vödisch, Ozan Unal, Ke Li, Luc Van Gool, and Dengxin Dai. End-to-end optimization of lidar beam configuration for 3d object detection and localization. *IEEE Robotics and Automation Letters*, 7(2):2242–2249, 2022.

[150] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[151] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C.-C. Jay Kuo. Mcl-jcv: A jnd-based h.264/avc video quality assessment dataset. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1509–1513, 2016.

[152] Sukai Wang, Jianhao Jiao, Peide Cai, and Lujia Wang. R-pcc: A baseline for range image-based point cloud compression. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10055–10061. IEEE, 2022.

[153] Sukai Wang and Ming Liu. Point cloud compression with range image-based entropy model for autonomous driving. In *European Conference on Computer Vision*, pages 323–340. Springer, 2022.

[154] T. Wang and H. Snoussi. Histograms of optical flow orientation for visual abnormal events detection. In *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pages 13–18, 2012.

[155] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[156] Yilin Wang, Sasi Inguva, and Balu Adsumilli. Youtube UGC dataset for video compression research. *CoRR*, abs/1904.06457, 2019.

[157] Yuan Wang, Tianyue Shi, Peng Yun, Lei Tai, and Ming Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv preprint arXiv:1807.06288*, 2018.

[158] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16249–16258, 2021.

[159] Cheng Wen, Baosheng Yu, and Dacheng Tao. Learnable skeleton-aware 3d point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17671–17681, 2023.

[160] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[161] Ian Witten, Radford Neal, and John Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30:520–540, 1987.

[162] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. In *ECCV*, 2018.

[163] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2023.

[164] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

[165] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[166] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):388–401, 2021.

[167] Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Hierarchical autoregressive modeling for neural video compression. In *International Conference on Learning Representations*, 2021.

[168] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[169] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 644–663. Springer, 2020.

[170] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[171] Tianyun Zhang, Shaokai Ye, Kaiqi Zhang, Jian Tang, Wujie Wen, Makan Fardad, and Yanzhi Wang. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199, 2018.

[172] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020.

[173] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022.

[174] Hang Zhao et al. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2016.

[175] Xuanyu Zhou, Charles R Qi, Yin Zhou, and Dragomir Anguelov. Riddle: Lidar data compression with range image deep delta encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17212–17221, 2022.

[176] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021.