DEEP PROBABILISTIC VIDEO COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a variational inference approach to deep probabilistic video compression. Our model uses advances in variational autoencoders (VAEs) for sequential data and combines it with recent work on neural image compression. The approach jointly learns to transform the original video into a lower-dimensional representation as well as to entropy code this representation according to a temporally-conditioned probabilistic model. We split the latent space into local (per frame) and global (per segment) variables, and show that training the VAE to utilize both representations leads to an improved rate-distortion performance. Evaluation on small videos from public data sets with varying complexity and diversity show that our model yields competitive results when trained on generic video content. Extreme compression performance is achieved for videos with specialized content if the model is trained on similar videos.

1 INTRODUCTION

The transmission of video content is responsible for up to 80% of the consumer internet traffic, and both the overall internet traffic as well as the share of video data is expected to increase even further in the future (Cisco, 2017). Improving compression efficiency is more crucial than ever.

Today, a variety of video codecs exists that have reached an impressive performance. The most commonly used standard is H.264 (Wiegand et al., 2003), also known as Advanced Video Coding (AVC). More recent codecs include H.265 (Sullivan et al., 2012), also known as High Efficiency Video Coding (HEVC), and VP9 (Mukherjee et al., 2015). All of these existing codecs follow the same block based hybrid structure (Musmann et al., 1985) and essentially emerged from engineering out and refining this concept over decades. From a very high level perspective, they differ in a huge number of smaller design choices and have grown to become more and more complex systems.

While there is room for improving the block based hybrid approach even further (Fraunhofer, 2018), the question is how much longer significant improvements can be obtained when following the same concept. Interestingly, in the context of image compression, deep learning approaches that are fundamentally different to existing codecs have already shown promising results (Ballé et al., 2018; 2016; Theis et al., 2017; Agustsson et al., 2017; Minnen et al., 2018), mostly within the past year.

Motivated by these successes for images, we propose a first step towards innovating beyond block-based hybrid codecs by framing video compression in a deep probabilistic context. To this end, we propose an unsupervised deep learning approach to encoding video. The approach simultaneously learns the optimal transformation of the video to a low-dimensional representation *and* a powerful predictive model that assigns probabilities to video segments, allowing us to efficiently entropy-code the discretized latent representation into a short code length.

Our end-to-end neural video compression scheme is based on sequential variational autoencoders (Bayer & Osendorfer, 2014; Chung et al., 2015; Li & Mandt, 2018) and the approach of Ballé et al. (2016) for discretizing and entropy coding a continuous latent representation. The transformations to and from the latent representation, known as the encoder and decoder, are parameterized by deep neural networks and are learned by unsupervised training on videos. We introduce both *local* latent variables, which are inferred from a single frame, and a *global* state, inferred from an entire segment, to efficiently store a video sequence. Furthermore, the trajectory of the latent variables is modeled stochastically by a deep probabilistic model. After training, the context-dependent predictive model is used to entropy code the latent variables into binary with an arithmetic coder.

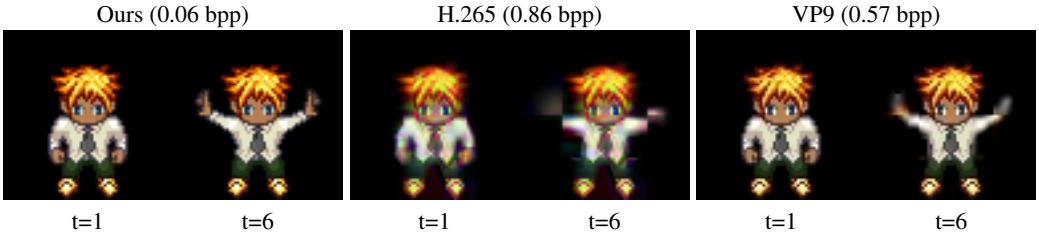


Figure 1: Reconstructed Sprite test video (bpp=0.06, PSNR=44.6 dB), H.265 (bpp=0.86, PSNR = 21.1 dB), and VP9 (bpp=0.57, PSNR = 26.0 dB), see Section 4. In contrast to our method, H.265 and VP9 show artifacts of block motion prediction. Our method uses a fraction of the bit rate.

As the first step towards a new approach, we focus on small resolution video (64×64) and aim to efficiently capture temporal correlations. Figure 1 shows a test example of the possible performance improvements using our approach if the model is trained on similar content. The plots show two frames of a video, compressed and reconstructed by our approach and classical video codecs. One sees that fine granular details, such as the hands of the cartoon character, are lost in the classical approach due to artifacts from block motion estimation (low bitrate regime), whereas our deep learning approach successfully captures these details with less than 10% of the file length.

Our contributions can be understood as follows:

1) Deep probabilistic video compression. To the best of our knowledge, this is the first work to employ a variational autoencoder (VAE) in conjuction with discretization and entropy coding to build an end-to-end trainable video codec.

2) Global inference. Temporal redundancy in a video can be taken into account by a temporal prior on each frame or through an architectural design that encodes an entire video segment into a global state. We propose a model which incorporates both global latent variables as well as per frame ones and show that this produces a shorter code length for a given image quality.

3) Small bit rates. We perform experiments on three large public data sets of varying complexity and diversity. Performance is evaluated by rate-distortion curves. Our method is competitive with traditional codecs on small videos after training and testing on a diverse set of videos. Extreme compression performance can be achieved for videos with specialized content if the model is trained on similar videos.

Paper Organization. In Section 2, we summarize important related works before describing our method for deep video compression in Section 3. Section 4 discusses our experimental results, including quantitative results and a qualitative discussion.

2 RELATED WORK

The approaches related to our method fall into three categories: deep generative video models, neural image compression, and neural video compression.

Deep Generative Video Models. Several works have applied the variational autoencoder (VAE) (Kingma & Welling, 2013; Rezende et al., 2014) to stochastically model sequences (Bayer & Osendorfer, 2014; Chung et al., 2015). Babaeizadeh et al. (2018); Xu et al. (2018) use a VAE for stochastic video generation. He et al. (2018) and Denton & Fergus (2018) apply a long short term memory (LSTM) in conjunction with a sequential VAE to model the evolution of the latent space across many video frames. Li & Mandt (2018) separate latent variables of a sequential VAE into local and global variables in order to learn a disentangled representation for video generation. Vondrick et al. (2016) generate realistic videos by using a GAN (Goodfellow et al., 2014) to learn to separate foreground and background, and Lee et al. (2018) combine variational and adversarial methods to generate realistic videos. This paper also employs a deep generative model to model the sequential probability distribution of frames from a video source. In contrast to other work, our work learns a continuous latent representation that can be discretized with minimal information loss,

required for further compression. Furthermore, our objective is to convert the original video into a short binary description rather than to generate new videos.

Neural Image Compression. There has been significant work on applying deep learning to image compression. In Toderici et al. (2016; 2017), a LSTM based codec is used to capture spatial correlations of pixel values and can achieve different bit-rates without having to retrain the model. Ballé et al. (2016) perform image compression with a VAE and demonstrate how to approximately discretize the VAE latent space by introducing noise during training. This work is refined by (Ballé et al., 2018) which improves the prior model (used for entropy coding) beyond the mean-field approximation by transmitting side information in the form of a hierarchical model. Minnen et al. (2018) consider an autoregressive model to achieve a similar effect. These image codecs encode each image independently and therefore their probabilistic models are stationary with respect to time. In contrast, our method performs compression according to a non-stationary, time-dependent probability density which has much lower entropy per pixel.

Neural Video Compression. The use of deep neural networks for video compression is relatively new. Wu et al. (2018) perform video compression through image interpolation between reference frames using a predictive model based on a deep neural network. Chen et al. (2017) and Chen et al. (2018) use a deep neural architecture to predict the most likely frame with a modified form of block motion prediction and store residuals in a lossy representation. Since these works are based on block motion prediction, they are similar in function and in performance to existing codecs. Our method is not based on block motion estimation, and the full inferred probability distribution over the space of plausible subsequent frames (rather than residuals) is used for entropy coding.

3 DEEP PROBABILISTIC VIDEO COMPRESSION

The objective of lossy video compression can be defined as finding the shortest description of a video while tolerating a certain level of information loss. Classical video codecs try to predict intermediate frames based on block motion estimates, since information which can be predicted does not need to be stored. The residual error is then stored in a lossy transform representation. An end-to-end machine learning approach to encoding video, however, should simultaneously learn the appropriate predictive model and the optimal lossy transformation to a discrete lower-dimensional representation. This allows both to transform the video into a low dimensional latent representation and to then use the jointly learned predictive model to remove the remaining redundancy in the latents by entropy coding them to a short binary representation (Huffman, 1952; Langdon, 1984).

The challenge over image compression is that the data is now sequences of images. These exhibit strong temporal correlations in addition to the spatial correlations already present in images. A naive approach to neural video compression would be to encode the video frame by frame, using the marginal distribution of images as done in VAE image compression. This distribution does not capture the temporal correlations and therefore tends to have high entropy, leading to long code lengths. On the other hand, treating an entire video segment as an independent data point in the latent representation leads to data sparseness and poor generalization performance.

Therefore, we propose to use a temporally-conditioned prior distribution parameterized by a deep generative model to efficiently code the latent variables associated with each frame. By conditioning on nearby frames in the sequence, the predictive model can be much more certain about the next frame, thus achieving a smaller entropy and code length. As detailed below, in addition to using a deep sequential probabilistic model, we propose an architecture that combines local and global information in the video. A global variable stores information that is common to the sequence of frames, while a local variable stores additional dynamical content.

In the following paragraphs, we describe our approach (see Fig. 3) in more detail. We describe the encoder and decoder models, the objective function, and the interplay between our deep probabilistic sequential model and entropy coding scheme.

Decoder. We propose a stochastic recurrent variational autoencoder to transform a sequence of frames $\boldsymbol{x}_{1:T} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$ into a compressed representation of latent variables $\boldsymbol{z}_{1:T} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_T)$. This model is refined to additionally include a global state $\boldsymbol{f}$ similar to Li & Mandt

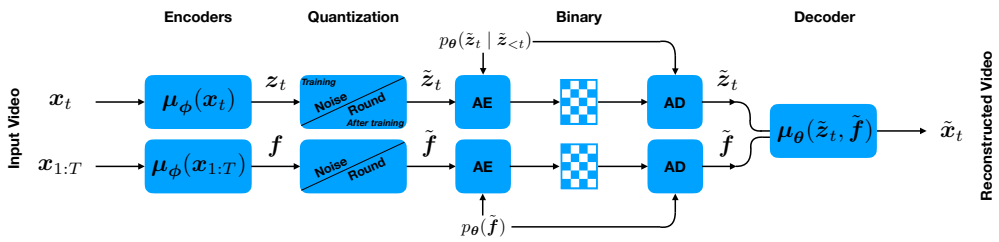


Figure 2: Operational diagram of our compression codec. A video segment is encoded into per-frame latent variables ($\boldsymbol{z}_t$) and per-segment global state ($\boldsymbol{f}$), which are then quantized and entropy encoded into binary according to a probabilistic model. To recover an approximation to the original video, the latent variables are entropy decoded from the binary and passed through the decoder.

(2018), resulting in the following probabilistic deep generative model:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}, \boldsymbol{f}) = p_{\boldsymbol{\theta}}(\boldsymbol{f}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T}) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{z}_t, \boldsymbol{f}) \tag{1}$$

Above, $\boldsymbol{\theta}$ is shorthand for the parameters. Each frame $\boldsymbol{x}_t$ at time t depends on the corresponding latent variables $\boldsymbol{z}_t$ and global variables $\boldsymbol{f}$. The frame likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{f}, \boldsymbol{z}_t)$ for reconstruction is the Laplace distribution, $\text{Laplace}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{f}), \lambda^{-1}\mathbf{1}\big)$. The reason is that its log likelihood results in an ℓ_1 regularized loss, which typically outperforms the ℓ_2 loss for autoencoding images. The decoder $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\cdot)$ is a function parameterized by neural networks. The prior distributions $p_{\boldsymbol{\theta}}(\boldsymbol{f})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T})$ will be discussed separately below.

After training, the reconstructed frame in image space is obtained from taking the most likely frame $\tilde{\boldsymbol{x}}_t = \text{argmax}\ p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{f}, \boldsymbol{z}_t) = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{f})$. Crucially, the decoder is conditioned both on global code $\boldsymbol{f}$ and time-local code $\boldsymbol{z}_t$.

Encoder. We employ amortized variational inference to predict a distribution over latent codes given the input video,

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}, \boldsymbol{f} \mid \boldsymbol{x}_{1:T}) = q_{\boldsymbol{\phi}}(\boldsymbol{f} \mid \boldsymbol{x}_{1:T}) \prod_{t=1}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t \mid \boldsymbol{x}_t). \tag{2}$$

The global variables $\boldsymbol{f}$ are inferred from all video frames in a sequence and may thus contain static information, while $\boldsymbol{z}_t$ is only inferred from a single frame $\boldsymbol{x}_t$.

As will be explained in the paragraph on model-based entropy coding below, modifications to standard variational inference are required for compression. Instead of sampling from Gaussian distributions with learned variances, here we employ uniform distributions centered at their means:

$$\tilde{\boldsymbol{f}} \sim q_{\boldsymbol{\phi}}(\boldsymbol{f} \mid \boldsymbol{x}_{1:T}) = \mathcal{U}\big(\hat{\boldsymbol{f}} - \frac{1}{2}, \hat{\boldsymbol{f}} + \frac{1}{2}\big); \qquad \tilde{\boldsymbol{z}}_t \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_t \mid \boldsymbol{x}_t) = \mathcal{U}\big(\hat{\boldsymbol{z}}_t - \frac{1}{2}, \hat{\boldsymbol{z}}_t + \frac{1}{2}\big). \tag{3}$$

The means are predicted by two additional encoder neural networks $\hat{\boldsymbol{f}} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_{1:T})$, $\hat{\boldsymbol{z}}_t = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)$ with parameters ϕ. This corresponds to adding random noise $\epsilon_i \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ to $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{\mu}}$ in the inference process. The encoder and decoder neural networks are described in more detail in Appendix B.

Prior Models. The models that parameterize the learned prior distributions are ultimately used for entropy coding. Each dimension of the latent space has its own density model:

$$p_{\boldsymbol{\theta}}(\boldsymbol{f}) = \prod_{i}^{\dim(\boldsymbol{f})} p_{\boldsymbol{\theta}}(f^i) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}); \quad p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T}) = \prod_{t}^{T} \prod_{i}^{\dim(\boldsymbol{z})} p_{\boldsymbol{\theta}}(z_t^i \mid \boldsymbol{c}_t) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}). \tag{4}$$

Above, indices refer to the dimension index of the latent variable and $\boldsymbol{c}_t$ is a time-dependent context. The convolution with uniform noise is to allow the priors to better match the true marginal distribution when working with the box-shaped approximate posterior in Eq. 3 (see Ballé et al. (2018) Appendix 6.2). This convolution has an analytic form in terms of the cumulative probability density.

The stationary density $p_{\boldsymbol{\theta}}(f^i)$ is parameterized by a flexible non-parametric, fully-factorized model from (Ballé et al., 2018). The density is defined by its cumulative and is built out of compositions of nonlinear probability densities, similar to the construction of a normalizing flows (Rezende & Mohamed, 2015).

Two dynamics models are considered to model the sequence $\boldsymbol{z}_{1:T}$. We propose a LSTM prior architecture which conditions on all previous frames in a segment: $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t^i \mid \boldsymbol{c}_t) \equiv p_{\boldsymbol{\theta}}(z_t^i \mid \boldsymbol{z}_{<t})$. We also considered a simpler model with a single frame context: $p_{\boldsymbol{\theta}}(z_t^i \mid \boldsymbol{z}_{t-1})$ which is essentially a deep Kalman filter (Krishnan et al., 2015), and which we compare against. A discussion of the connection between the amount of conditional context and the entropy is given in appendix A.

Variational Objective. The encoder (variational model) and decoder (generative model) can be learned jointly by minimizing the VAE loss function which consists of the KL divergence between the approximate and true posterior. In compression applications, however, one needs to adjust the trade-off between the bit rate and distortion. This is achieved by the β-VAE loss which takes the form (up to constant terms)

$$\begin{aligned} &-\mathbb{E}_{\tilde{\boldsymbol{f}},\tilde{\boldsymbol{z}}_{1:T}\sim q}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\tilde{\boldsymbol{f}},\tilde{\boldsymbol{z}}_{1:T})] - \beta\, \mathbb{E}_{\tilde{\boldsymbol{f}},\tilde{\boldsymbol{z}}_{1:T}\sim q}[\log p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{f}},\tilde{\boldsymbol{z}}_{1:T})] \\ &= \mathbb{E}_{\tilde{\boldsymbol{f}},\tilde{\boldsymbol{z}}_{1:T}\sim q} \sum_{t=1}^{T} \|\tilde{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_1 + \beta H\left[q_{\boldsymbol{\phi}}(\tilde{\boldsymbol{z}}_{1:T},\tilde{\boldsymbol{f}} \mid \boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{f}},\tilde{\boldsymbol{z}}_{1:T})\right] \end{aligned} \tag{5}$$

where the reconstructed frame $\tilde{\boldsymbol{x}}_t = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_t), \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_{1:T}))$ and we have introduced a parameter β to control the rate-distortion trade-off (Alemi et al., 2018). The Laplace parameter λ was set to one.

The first term corresponds to the distortion and the second term is the cross entropy between the approximate posterior and the prior. The latter has the interpretation of the expected code length when using the prior distribution $p(\boldsymbol{f}, \boldsymbol{z}_{1:T})$ to entropy code the latent variables. This term is minimized for $p(\boldsymbol{f}, \boldsymbol{z}_{1:T}) = \mathbb{E}_{\boldsymbol{x}_{1:T}}[q(\boldsymbol{f}, \boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T})]$, that is, when the empirical distribution of codes matches the prior model. For our choice of generative model, the cross entropy separates into two terms $H\left[q_{\boldsymbol{\phi}}(\boldsymbol{f}|\boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\boldsymbol{f})\right]$ and $H\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T})\right]$.

Model-based Entropy Coding. Sequential VAEs can be used to reduce the dimensionality of a video by performing a lossy transformation. However, the reduction in dimensionality does not mean the video is optimally compressed since there still exists redundancy in the form of temporal correlations in the sequence of the latent space variables. Sequential VAEs may include a temporal prior to model this redundancy, but to actually remove this redundancy and achieve a compact binary representation, the latent space has to be entropy coded. This is the distinguishing element between variational autoencoders and compression algorithms.

Crucially, the entropy coder needs a discrete vocabulary, which is obtained by rounding the latent state ($\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{z}}_{1:T}$) after training. Care must be taken such that the quantization is approximated in a differentiable way during training. Ballé et al. (2016) address the problem of discretization of a continuous latent space in the context of VAE image compression by introducing noise in the inference process. By adding uniform noise to the most-likely inferred latent variables, the VAE is prevented from storing information in the latent space on length scales smaller than the discretization bin size. As such, rounding (after training) does not significantly affect the image reconstruction.

Our VAE framework exactly leads to such injection of noise with width one; it corresponds to our choice of a box-shaped approximate posterior distribution, centered at the maximally-likely values for the latent variables. Besides dealing with quantization, for efficient entropy coding we also need an estimate of the frequency of atoms in order to obtain short file sizes. We can obtain this from our prior model.

4 EXPERIMENTS

In this section, we present the experimental results of our work. We first describe the video datasets, performance metrics, and competing methods in Section 4.1. This is followed by a quantitative analysis of our performance in terms of rate-distortion curves in Section 4.2 and then finally qualitative results in Section 4.3. We report that our method can achieve extreme compression ratios on

videos with specialized content if it is trained on similar videos and that the performance is content-dependent. Our method is also comparable with modern codecs when trained on videos with diverse content. We find that the inclusion of the global state is more efficient than the using solely local variables.

4.1 Datasets, Metrics, and Methods

In this work, we train separately on three video datasets of increasing complexity with frame size 64×64. **1) Sprites.** The simplest dataset consists of videos of Sprites characters from an open-source video game project, which is used in (Reed et al., 2015; Mathieu et al., 2016; Li & Mandt, 2018). The videos are generated from a script that samples the character action, skin color, clothing, and eyes from a collection of choices and have an inherently low-dimensional description (*i.e.* the script that generated it). **2) BAIR.** BAIR action-free robot pushing dataset (Ebert et al., 2017) consists of a robot pushing objects on a table, which is also used in (Babaeizadeh et al., 2018; Denton & Fergus, 2018; Lee et al., 2018). The video is more realistic and less sparse, but the content is specialized since all scenes contain the same background and robot, and the depicted action is simple since the motion is described by a limited set of commands sent to the robot. The first two datasets are uncompressed and no preprocessing is performed. **3) Kinetics600.** The last dataset is the Kinetics600 dataset (Kay et al., 2017) which is a diverse set of YouTube videos depicting human actions. The dataset is downsampled, which removes compression artifacts, and cropped to 64×64.

Metrics. We evalute our method based on the compression rate in bits per pixel (bpp), and peak signal to noise ratio (PSNR), which is related to mean square error and is measured in decibels. In the appendix, we also report on multi-scale structural similarity (MS-SSIM) (Wang et al., 2004) which is a perception-based metric that approximates the perceived change in structural information.

Comparisons. We wish to study the performance of our proposed local-global architecture with LSTM prior (LSTMP-LG) by comparing to other approaches. To study the effect of the prior model, we show a variation of our method which utilizes the same local-global representation but with the LSTM prior replaced by a deep Kalman filter (Krishnan et al., 2015) (KFP-LG). For the last variation, we introduce a simpler latent represenation which excludes the global variables (LSTMP-L). We also provide the performance H.264, H.265, and VP9 codecs (Wiegand et al., 2003; Sullivan et al., 2012; Mukherjee et al., 2015). Traditional codecs are not optimized for small resolution videos. However, their performance is far superior to neural or classical image compression methods (applied to compress video frame by frame), so their performance is presented for comparison. Codec performance is evaluated using the open source FFMPEG implementation in constant rate mode and distortion is varied by adjusting the constant rate factor. Unless otherwise stated, performance is tested on videos with 4:4:4 chroma sampling and on test videos with $T = 10$ frames.

4.2 Quantitative Analysis: rate-distortion trade-off

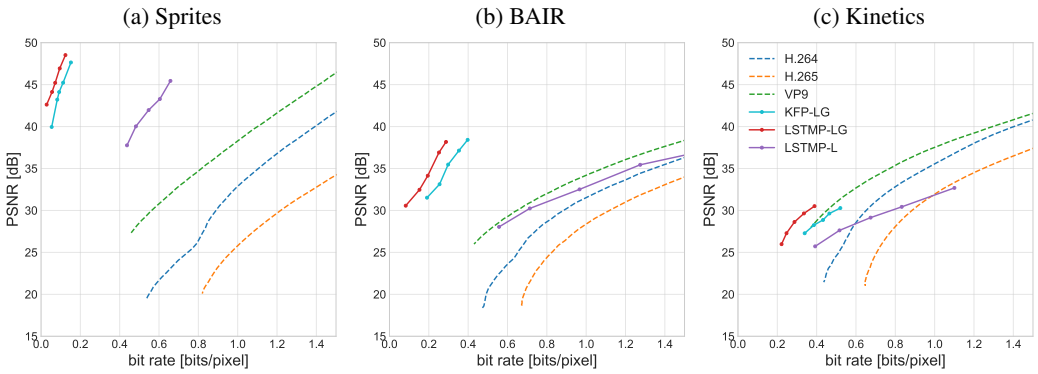


Figure 3: Rate-distortion curves on three datasets measured in PSNR (higher corresponds to lower distortion). Legend shared. Solid lines correspond to our models, with LSTMP-LG proposed.

Quantitative compression performance is evaluated in terms of rate-distortion theory. This characterizes the trade-off between the binary representation size and average distortion. For a fixed image

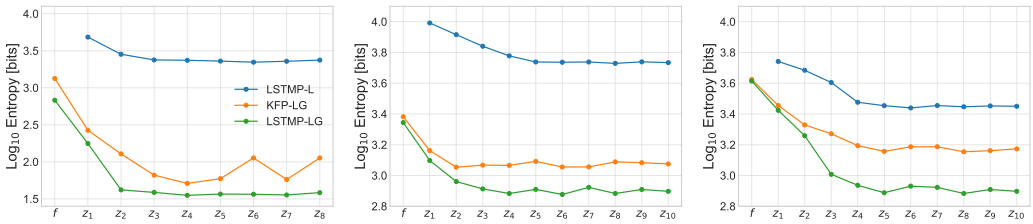
quality setting, a video codec produces an average bit rate on a given dataset. By varying the image quality setting, a curve is traced out in the rate-distortion plane. The curves for our method are generated by varying β (Section 3).

The rate-distortion curves for our method, trained on three datasets and measured in PSNR, is shown in Fig. 3. Higher curves indicate better performance. From the Sprites and BAIR results, one sees that our method has the ability to drastically outperform traditional codecs when focusing on specialized content. By training on videos with a fixed content, the model is able to learn an efficient representation for such content. The results from training on the more diverse Kinetics videos also outperform or are competitive with standard codecs and better demonstrate the performance of our method on general content videos. Results measured in MS-SSIM are provided in Appendix D and show similar behavior.

The first observation is that the LSTM prior outperforms the deep Kalman filter prior in all cases. This is because the LSTM model has a longer memory, allowing the predictive model to be more certain about the trajectory of the local latent variables. This, in turn, results in shorter code lengths.

Global Variables. The VAE encoder has the option to store information in local or global variables. The local variables are modeled by a temporal prior and can be efficiently stored in binary if the sequence $\boldsymbol{z}_{1:T}$ can be sequentially predicted with relative certainty from the context. The global variables, on the other hand, provide an architectural approach to removing temporal redundancy since the entire segment is stored in one global state without temporal structure. We find that the local-global architecture (LSTMP-LG) outperforms the local architecture (LSTMP-L) on all datasets, demonstrating the usefulness of a hybrid approach which partially encodes the entire video segment in a global state along with extra frame-by-frame information stored as a sequence.

During training, the VAE learns to utilize the global and local information in the optimal way. The utilization of each variable can be visualized by plotting the average code length of each latent state, which is shown in Fig. 4. The VAE learns to significantly utilize the global variables even though $\dim(\boldsymbol{z})$ is sufficiently large to store the entire content of each individual frame. This provides further evidence that it is more efficient to incorporate global inference over several frames. Notice that entropy in the local variables initially tends to decrease as a function of time since the first $\boldsymbol{z}_0$ has a cold start. Note that our approach relies on sequential decoding, prohibiting a bi-directional LSTM for the local state.



(a) Sprites (b) BAIR (c) Kinetics

Figure 4: Average bits of information stored in $\boldsymbol{f}$ and $\boldsymbol{z}_{1:T}$ with PSNR 43.2, 37.1, 30.3 for different models in (a, b, c). Entropy drops with the frame index as the models adapt to the video sequence.

4.3 Qualitative Results

Now we discuss the qualitative performance of our method. We have shown that a deep neural approach to encode video (LSTMP-LG architecture) can outperform traditional codecs with respect to PSNR or MS-SSIM (Appendix D) metrics overall on low-resolution videos. Test videos from the Sprites and BAIR datasets after compression with our method are shown in Fig. 1 and Fig. 5 (left), respectively, and compared to modern codec performance. Our method achieves a superior image quality at a significantly lower bit rate than H.264/H.265 and VP9 on these specialized content datasets. This is perhaps expected since traditional codecs cannot learn efficient representations for specialized content and the learned priors capture the empirical data distribution well (Appendix C). Furthermore, fine-grained motion is not accurately predicted with block motion estimation. The artifacts from our method are more clearly displayed in Fig. 5 (right). Our method tends to produce

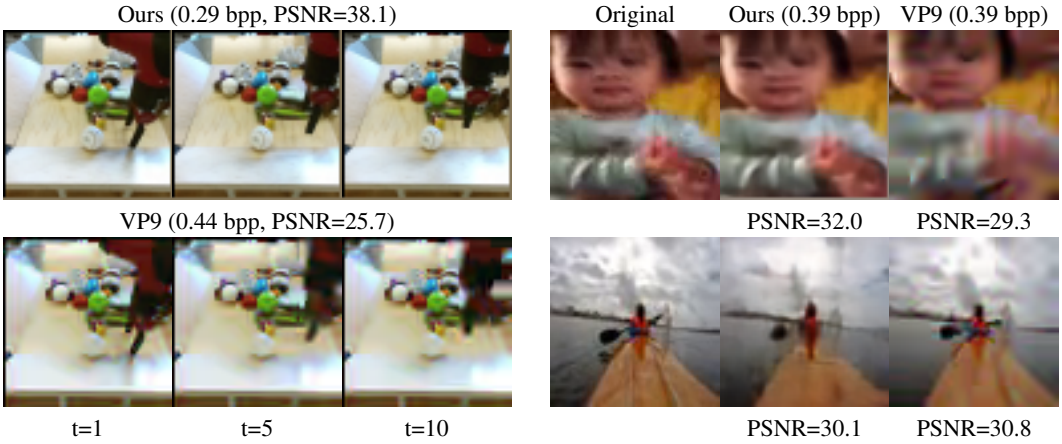


Figure 5: Compressed videos by our LSTMP-LG model and VP9 in the low bit rate regime. Our approach achieves better quality on specialized content (BAIR, left) and comparable visual quality on generic video content (Kinetics, right) compared to VP9.

blurry video in the low bit-rate regime but does not suffer from the block artifacts present in the H.265/VP9 compressed video.

Limitations. When training on specialized content videos, our method tends to fit to the type of content, leading to artifacts which can be more abstract and perhaps undesirable for a general purpose codec. For example, at a low bit-rate setting the codec tends to leave out new kinds of objects which were not present in the training data, or to misinterpret content. This is a big departure from the behavior of traditional codecs. Such undesirable defects can be avoided, however, by training on a general content training set or keeping a larger local latent state.

Our current paper focused on small scale videos. One future avenue is to extrapolate our method to full resolution videos, where the dimension of the latent representation must scale with the resolution of the video in order to achieve good reconstruction preformance. Currently, the GPU memory limits the maximum size of the latent dimension for the local/global architecture due to the presence of fully-connected layers to infer global and local states. While we showed that this architecture was very efficient for small videos in the strongly compressed regime, a future architecture may focus more on convolutional structures and avoidance of fully connected networks.

5 Conclusions

We have proposed a deep probabilistic modeling approach to video compression. Our method simultaneously learns to transform the original video into a lower-dimensional representation as well as the temporally-conditioned probabilistic model for entropy coding. The best performing proposed architecture splits up the latent code into global and local variables and yields competitive results on low resolution videos. For video sources with specialized content, deep probabilistic video coding allows for a significant increase coding performance. This could be interesting for transmitting specialized content such as teleconferencing or sports broadcasting.

We have showed viability on low resolution videos and in future work it will be interesting to efficiently scale our deep probabilistic coding framework to longer sequences and high resolution videos. Furthermore, there is much potential in investigating how additional "side information" could aid the predictive model. Thus we think that our work is a first step into a new direction for video coding which opens up many exciting avenues for future work.

REFERENCES

Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, 2017.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, 2018.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *ICLR*, 2018.

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *International Conference on Learning Representations*, 2016.

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

Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

Tong Chen, Haojie Liu, Qiu Shen, Tao Yue, Xun Cao, and Zhan Ma. Deepcoder: A deep neural network based video compression. In *Visual Communications and Image Processing (VCIP), 2017 IEEE*. IEEE, 2017.

Zhibo Chen, Tianyu He, Xin Jin, and Feng Wu. Learning for video compression. *arXiv preprint arXiv:1804.09869*, 2018.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pp. 2980–2988, 2015.

Visual Network Index Cisco. Forecast and methodology, 2016-2021. *White Paper*, 2017.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *International Conference on Machine Learning*, 2018.

Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *Conference on Robot Learning*, 2017.

Fraunhofer. Institute Website. *https://jvet.hhi.fraunhofer.de*, 2018.

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*, 2014.

Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. *European Conference on Computer Vision*, 2018.

David A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 1952.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Advances in Neural Information Processing Systems*, 2013.

Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Glen G Langdon. An introduction to arithmetic coding. *IBM Journal of Research and Development*, 1984.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. *International Conference on Machine Learning*, 2018.

Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pp. 5040–5048, 2016.

David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances In Neural Information Processing Systems*, 2018.

Debargha Mukherjee, Jingning Han, Jim Bankoski, Ronald Bultje, Adrian Grange, John Koleszar, Paul Wilkins, and Yaowu Xu. A technical overview of vp9the latest open-source video codec. *SMPTE Motion Imaging Journal*, 2015.

Hans Georg Musmann, Peter Pirsch, and H-J Grallert. Advances in picture coding. *Proceedings of the IEEE*, 73(4), 1985.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in neural information processing systems*, pp. 1252–1260, 2015.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *International Conference on Machine Learning*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning*, 2014.

CE Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 2001.

Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, et al. Overview of the high efficiency video coding(hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 2012.

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

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. *International Conference on Learning Representations*, 2016.

George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. 2017.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pp. 613–621, 2016.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 2003.

Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. *European Conference on Computer Vision*, 2018.

Qiangeng Xu, Hanwang Zhang, Peter N Belhumeur, and Ulrich Neumann. Stochastic video long-term interpolation. *arXiv preprint arXiv:1809.00263*, 2018.

A ENTROPY CODING

Predictive modeling is crucially important in the entropy coding stage. Theoretical compression ratios can be computed from the cross entropy between the true probability distribution (measured empirically) and a model for the probabilities. A better model that more accurately captures the true certainty about the next symbol can achieve a lower entropy and thus a smaller bit rate. Entropy coding is a solved problem since there exist coding schemes, *e.g.* arithmetic coding, which approximately achieve the the cross entropy, up to extra information that is sent through the stream for practical reasons, and this bit rate is optimal for long messages (Shannon, 2001). Modeling, however, is difficult and leaves much room for improvement.

In this work, after the latent representation has been quantized to N finite symbols, the variables are losslessly compressed into binary. The global latent variables $\boldsymbol{f}$ are static are entropy coded according to a stationary distribution. The optimal lossless compression algorithm encodes the i-th symbol $\boldsymbol{f}^i$ with a number of bits set by $\log \tilde{p}(\boldsymbol{f}^i)$ on average, where $\tilde{p}$ is the true probability for $\boldsymbol{f}^i$. The theoretical code length in bits after entropy coding the global variables with the prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{f})$ is computed from the cross entropy:

$$H\left[q_{\boldsymbol{\phi}}(\boldsymbol{f}|\boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\boldsymbol{f}^i)\right] = -\mathbb{E}_{\boldsymbol{f}\sim q} \sum_{i=1}^{N} \log_2 p_{\boldsymbol{\theta}}(\boldsymbol{f}^i). \tag{6}$$

The cross entropy is minimized on average in the limit that the $p_{\boldsymbol{\theta}}(\boldsymbol{f}) = \mathbb{E}_{\boldsymbol{x}_{1:T}} q_{\boldsymbol{\phi}}(\boldsymbol{f}|\boldsymbol{x}_{1:T}) \equiv \tilde{p}(\boldsymbol{f})$. In this work, we quote theoretical compression rates by computing the cross entropy between the variational distribution and the prior. However, we have explicitly checked that this performance is achieved by an arithmetic coder implementation which uses the probabilities from the prior.

For the case of the sequence $\boldsymbol{z}_{1:T}$, the theoretical description length of the sequence is given by the following cross entropy:

$$H\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T})\right] = -\mathbb{E}_{\boldsymbol{z}_{1:T}\sim q} \sum_{t=1}^{T} \sum_{i=1}^{N} \log_2 p_{\boldsymbol{\theta}}(\boldsymbol{z}_t^i \mid \boldsymbol{c}_t) \tag{7}$$

where $\boldsymbol{c}_t$ is the context which may in principle depend on the previous appearing symbols in the sequence, the future symbols, or the global variables.

Consider the true probability distribution $\tilde{p}(\boldsymbol{z}_t|\neg\boldsymbol{z}_t)$, conditioned on all other $\boldsymbol{z}_t$ in the sequence both forward and backward in time. Averaging over all future elements of the sequence leads to a simplified distribution which only captures the conditional distribution based on a backwards in time context $\tilde{p}(\boldsymbol{z}_t|\boldsymbol{z}_{<t})$. One can average over long-distance context to arrive at an even simpler condition disribution $\tilde{p}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$, or average out all context to obtain the marginal distribution $\tilde{p}(\boldsymbol{z}_t)$. We are guaranteed that the Shannon entropies are progressively larger since

$$H\left[\tilde{p}(\boldsymbol{z}_t)_{t=1:T}\right] \geq H\left[\tilde{p}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})_{t=1:T}\right] \geq H\left[\tilde{p}(\boldsymbol{z}_t|\boldsymbol{z}_{<t})_{t=1:T}\right] \geq H\left[\tilde{p}(\boldsymbol{z}_t|\neg\boldsymbol{z}_t)_{t=1:T}\right]. \tag{8}$$

Equality holds when the mutual information between $\boldsymbol{z}_{1:T}$ and the additional context vanishes.[1] The left-most entropy corresponds to treating each frame independently as in image compression. By improving the amount of context in the probabilistic model, the entropy and code length can be potentially reduced. We illustrate this point in the experiment section by demonstrating that the LSTM prior model, which has a longer contextual scope, produces a shorter code length than the deep Kalman filter prior.

B MODEL ARCHITECTURE

The specific structure of our model is now described.

LG-VAE. The encoder $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_{1:T})$ begins with a convolution architecture over $\boldsymbol{x}_{1:T}$ and $\boldsymbol{x}_t$, respectively to extract feature information. After the convolutional layers, an LSTM depending on all feature of per segment of video will infer the global state $\boldsymbol{f}$ and two-layer MLP will infer $\boldsymbol{z}_t$ which

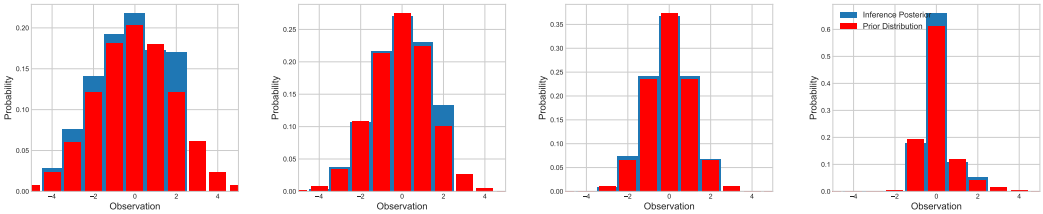
[1] This is equivalent to the fact from statistical physics that entropy of a system cannot decrease when averaging over a subset of the degrees of freedom.

only depends on the feature information from per frame $\boldsymbol{x}_t$. The decoder $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{f})$ first combines $(\boldsymbol{z}_t, \boldsymbol{f})$ with a multilayer perceptron (MLP) and then upsamples with a deconvolutional network. For prior model $p_{\boldsymbol{\theta}}(\boldsymbol{f})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1)$ is parametrized by the model defined on Appendix 6.1 of Ballé et al. (2018). $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})$ in LSTMP-LG is an LSTM with hidden state $\boldsymbol{h}$. $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$ in KFP-LG is a deep Kalman Filtering with three-layer MLP.

Our encoder $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ involves 5 convolutional layers. For layer $\ell = 1, 2, 3, 4$, the number of stride and padding are 2 and 1, respectively, and the convolutional kernel size is 4. The number of channels for layer $\ell = 1, 2, 3, 4$ are 192, 256, 512, 1024. Layer 5 has kernel size 4, stride 1 and padding 0 and number of channels 3072. The decoder architecture $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is chosen to be symmetric to the encoder with convolutional layers replaced with deconvolutional (upsampling) layers. For toy videos Sprites, the dimensions of $\boldsymbol{z}$, $\boldsymbol{f}$, and hidden state $\boldsymbol{h}$ are 64, 512 and 1024, respectively. For realistic videos BAIR and Kinetics600, the dimensions of $\boldsymbol{z}$, $\boldsymbol{f}$, and hidden state $\boldsymbol{h}$ are 256, 2048 and 3072, respectively.

L-VAE. L-VAE employs the same encoder and decoder architectures from Ballé et al. (2016). The encoder $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)$ infers each $\hat{\boldsymbol{z}}_t$ independently by a convolutional network over $\boldsymbol{x}_t$. The dimension of the latent variable $\boldsymbol{z}_t$ is 1024. $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1)$ is parametrized by the model defined on Appendix 6.1 of Ballé et al. (2018) and $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})$ is an LSTM with the dimension of hidden state 2048.

C LATENT SPACE VISUALIZATION



(a) $p_{\boldsymbol{\theta}}(f^i)$ (b) $p_{\boldsymbol{\theta}}(f^i)$ (c) $p_{\boldsymbol{\theta}}(z_1^i)$ (d) $p_{\boldsymbol{\theta}}(z_1^i)$

Figure 6: Empirical distributions of the posterior of inference model estimated from BAIR data and ground truth prior model in one specific rate-distortion example.

In this appendix, we visualize the distribution of our prior model and empirical distribution of the posterior of inference model estimated from data. In Fig. 6, we show the learned priors and the empirically observed prior over two dimensions of the latent global variable $\boldsymbol{f}$ and $\boldsymbol{z}$ in order to demonstrate that the prior is capturing the empirical distribution in low-bit rate of our model. From Fig. 6, we can see that the learned priors $p_{\boldsymbol{\theta}}(\boldsymbol{f})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1)$ match the empirical data distributions well, which leads to low-bit rate encoding of the latent variables. As the conditional probability model $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})$

D QUANTITATIVE ANALYSIS: RATE-DISTORTION TRADE-OFF

We also plot the MS-SSIM w.r.t. the bit rate to quantitatively compare our models with traditional codecs. From Fig. 7, we can see that our LSTMP-LG saves significantly more bits when trained on specialized content dataset and achieves competitive result when trained on general content dataset.

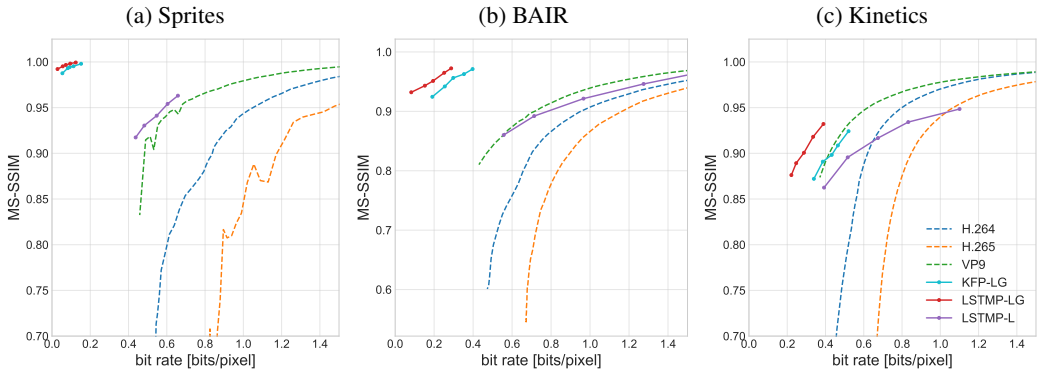


Figure 7: Rate-distortion curves on three datasets measured in MS-SSIM (higher corresponds to lower distortion). Legend shared. Solid lines correspond to our models, with LSTMP-LG proposed.