# Video Compression through Deep Bayesian Learning

**Jun Han*** **Salvator Lombardo*** **Christopher Schroers** **Stephan Mandt**

## 1 Introduction

The transmission of video content is responsible for up to 80% of the internet traffic [9]. Improving compression efficiency is more crucial than ever. Today, a variety of video codecs [25, 23, 20] exists that have reached an impressive performance. All of these codecs employ block based hybrid structure [21] and remain salient in progress for more than a decade.

Motivated by promosing results from neural image compression [1, 5, 4, 24, 19], we propose, to the best of our knowledge, a first step towards innovating beyond block-based hybrid codecs by framing video compression in a deep probabilistic context. Our end-to-end neural video compression scheme is based on sequential variational autoencoders [6, 8, 16] and the approach of Ballé et al. [4] for discretizing and entropy coding a continuous latent representation. Our approach simultaneously learns the optimal transform 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. We introduce both *local* latent variables, which are inferred from a single frame, and a *global* latent state, inferred from an entire segment, to efficiently store a video sequence.

As the first step towards a new approach, we focus on small resolution video ($64 \times 64$) 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. 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.

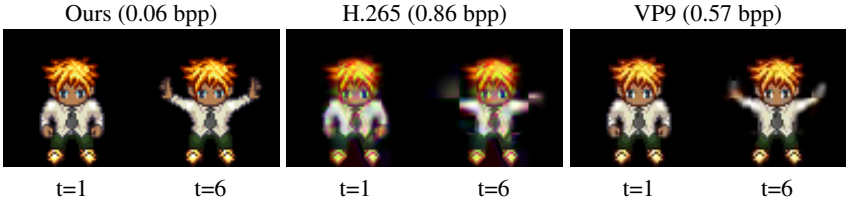



Figure 1: Reconstructed Sprite test video (bpp=0.06, PSNR=44.6 dB), H.265 (bpp=0.86, PSNR = 21.1 dB), and VP9 (bpp=0.57, PSNR = 26.0 dB), see Section 3. In contrast to our method, H.265 and VP9 show artifacts of block motion prediction. Our method uses a fraction of the bit rate.

## 2 Neural 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. 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

*Shared first authorship.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

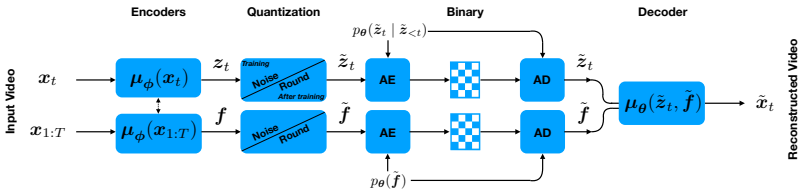



Figure 2: Operational diagram of our compression codec. A video segment is encoded into per-frame latent variables ($\boldsymbol{z}_t$) and per-segment global state ($\boldsymbol{f}$), which are then quantized and entropy encoded into binary according to a probabilistic model. To recover an approximation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

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 [12, 14]. Therefore, we propose to use a temporally-conditioned prior distribution parameterized by deep neural network 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. 2) 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.

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 [16], resulting in

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

where $\boldsymbol{\theta}$ 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, $\text{Laplace}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{f}), \lambda^{-1}\mathbf{1}\big)$. We employ amortized variational inference [7, 26, 17] 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 $\boldsymbol{x}_t$.

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 [13], and which we compare against. The variational objective is

$$-\mathbb{E}_{\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{z}}_{1:T} \sim q}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{z}}_{1:T})] - \beta\, \mathbb{E}_{\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{z}}_{1:T} \sim q}[\log p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{z}}_{1:T})] \tag{3}$$

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 $\beta$ to control the rate-distortion trade-off [2].

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 $H\big[q_{\boldsymbol{\phi}}(\boldsymbol{f}|\boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\boldsymbol{f})\big]$ and $H\big[q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}), p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T})\big]$.

## 3 Experiments

We train separately on three video datasets of increasing complexity with frame size $64 \times 64$: **1) Sprites**, used in [22, 18, 16], is generated from a script that samples the character action, skin color, clothing, and eyes from a collection of choices and has an inherently low-dimensional description; **2) BAIR** [11] with specialized content consisting of a robot pushing objects on a table, used in [3, 10, 15]; **3) Kinetics600**, a diverse set of YouTube videos depicting human actions, which is downsampled (removes compression artifacts) and cropped to $64 \times 64$. **Metrics:** we evaluate our method based on the compression rate in bits per pixel (bpp), and peak signal to noise ratio (PSNR).

**Comparisons.** We compare our proposed local-global architecture with LSTM prior (LSTMP-LG) with 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 [13] (KFP-LG). For the last variation, we introduce a simpler model which only has local latent variables with the LSTM predictive model (LSTMP-L). We also provide the performance of H.264, H.265, and VP9 codecs [25, 23, 20] using the open source FFMPEG in constant rate mode. Traditional codecs are not optimized for small resolution videos. Unless otherwise stated, performance is tested on videos with 4:4:4 chroma sampling and on test videos with $T = 10$ frames.

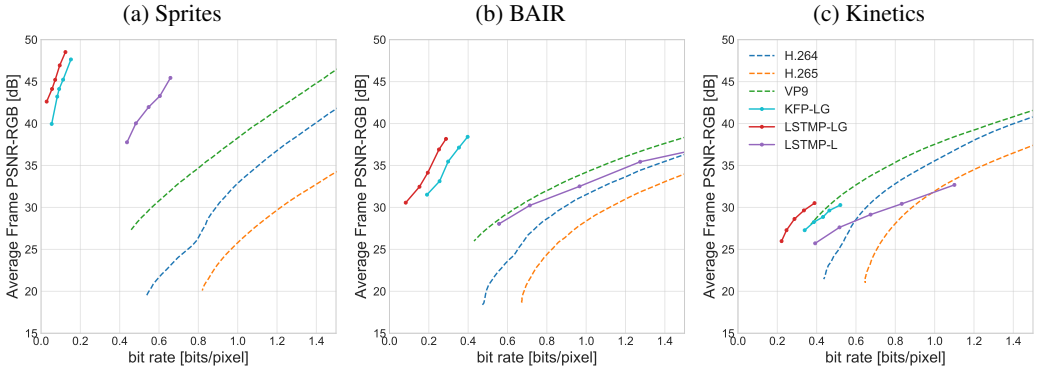



Figure 3: Rate-distortion curves on three datasets measured in PSNR (higher corresponds to lower distortion). Legend shared. Solid lines correspond to our models, with LSTMP-LG proposed.

**Rate-distortion trade-off.** The rate-distortion curves measured in PSNR in Fig. 3 is generated by varying $\beta$ for our method and trained on three datasets. 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.

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. 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) on Appendix B. Our method tends to produce blurry video in the low bit-rate regime but does not suffer from the block artifacts present in the H.265/VP9 compressed video.

## 4 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.

## References


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

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

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

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

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

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

[7] Blei, David M, Kucukelbir, Alp, and McAuliffe, Jon D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

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

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

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

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

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

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

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

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

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

[17] Marino, Joseph, Yue, Yisong, and Mandt, Stephan. Iterative amortized inference. In *International Conference on Machine Learning*, 2018.

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

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

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

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

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

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

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

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

[26] Zhang, Cheng, Butepage, Judith, Kjellstrom, Hedvig, and Mandt, Stephan. Advances in variational inference. *arXiv preprint arXiv:1711.05597*, 2017.

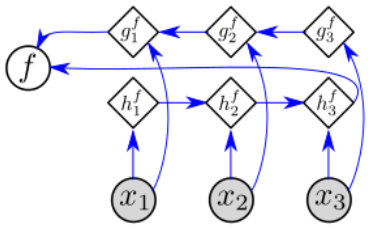



Figure 4: Inference network diagram for the global state $\boldsymbol{f}$. The features from the video segment are processed by a bi-directional LSTM (with hidden states $\boldsymbol{g^f}$, $\boldsymbol{h^f}$) which is used to infer the global state.

# A Model Architecture

The specific implementation details of our model are now described. We describe the two baseline models, LSTMP-L and KFP-LG, and the best-performing LSTMP-LG model.

**LSTMP-L.** Our proposed baseline model LSTMP-L, which is introduced to study the efficiency of the global state for capturing temporal redundancy, contains only local latent variables $\boldsymbol{z}_t$ (the global state $\boldsymbol{f}$ is omitted). The local state for each frame $\boldsymbol{z}_t$ is inferred from each frame $\boldsymbol{x}_t$. LSTMP-L employs the same encoder and decoder architectures from Ballé et al. [4]. The encoder $\boldsymbol{\mu_\phi}(\boldsymbol{x}_t)$ infers each $\boldsymbol{z}_t$ independently by a five-layer convolutional network. For layer $\ell = 1$, the stride is 4, while a stride of 2 is used for layer $\ell = 2, 3, 4, 5$. The padding is 1 and the kernel size is $4 \times 4$ for all layers. The number of filters used for the Sprites video, for $\ell = 1, 2, 3, 4, 5$, are 192, 256, 512, 512 and 1024, respectively. For the more realistic video (BAIR and Kinetics video), the number of filters used at layer $\ell = 1, 2, 3, 4, 5$ are 192, 256, 512, 1024 and 2048, respectively. The decoder $\boldsymbol{\mu_\theta}(\boldsymbol{z}_t)$ is symmetrical to the encoder $\boldsymbol{\mu_\phi}(\boldsymbol{x}_t)$. With this architecture, the dimension of the latent state $\boldsymbol{z}_t$ is 1024 for Sprites and 2048 for BAIR and Kinetics video. The prior for the latent state corresponding to the first frame, $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1)$, is parametrized by the same density model defined on Appendix 6.1 of Ballé et al. [5]. The conditional prior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})$ is parameterized by a normal distribution convolved with uniform noise. The means and (diagonal) covariance of the normal distribution are predicted by an LSTM with hidden state dimension equal to the dimension of the latent state $\boldsymbol{z}_t$.

**LSTMP-LG.** LSTMP-LG is our proposed model in this paper which uses an efficient latent representation by splitting latent states into both global states and local states as well as the use of an effective LSTM predictive model for entropy coding. Now we describe the inference network. The two encoders $\boldsymbol{\mu_\phi}(\boldsymbol{x}_{1:T})$ and $\boldsymbol{\mu_\phi}(\boldsymbol{x}_t)$ begin with a convolutional architecture to extract feature information. The global state $\boldsymbol{f}$ is inferred from all frames by processing the output of the convolutional layers over $\boldsymbol{x}_{1:T}$ with a bi-directional LSTM architecture (note this LSTM is used for inference not entropy coding), shown diagrammatically in Fig. 4. This allows $\boldsymbol{f}$ to depend on features from the entire segment. For the local state, the individual frame $\boldsymbol{x}_t$ is passed through the convolutional layers of $\boldsymbol{\mu_\phi}(\boldsymbol{x}_t)$ and a two-layer MLP infers $\boldsymbol{z}_t$ from the feature information of the individual frame. The decoder $\boldsymbol{\mu_\theta}(\boldsymbol{z}_t, \boldsymbol{f})$ first combines $(\boldsymbol{z}_t, \boldsymbol{f})$ with a multilayer perceptron (MLP) and then upsamples with a deconvolutional network. The prior models $p_{\boldsymbol{\theta}}(\boldsymbol{f})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1)$ are parametrized by the density model defined in Appendix 6.1 of Ballé et al. [5]. The conditional prior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})$ in the LSTMP-LG architecture is modeled by a normal distribution which is convolved with uniform noise. The means and covariance of the normal distribution are predicted by an additional LSTM with hidden state $\boldsymbol{h}$.

Both encoders $\boldsymbol{\mu_\phi}(\cdot)$ have 5 convolutional (downsampling) layers. For layer $\ell = 1, 2, 3, 4$, the stride and padding are 2 and 1, respectively, and the convolutional kernel size is $4 \times 4$. The number of channels for layer $\ell = 1, 2, 3, 4$ are 192, 256, 512, 1024. Layer 5 has kernel size 4, stride 1, padding 0, and 3072 channels. The decoder architecture $\boldsymbol{\mu_\theta}$ is chosen to be asymmetric to the encoder with convolutional layers replaced with deconvolutional (upsampling) layers. For the Sprites toy video, the dimensions of $\boldsymbol{z}$, $\boldsymbol{f}$, and hidden state $\boldsymbol{h}$ are 64, 512 and 1024, respectively. For less sparse videos (BAIR and Kinetics600), the dimensions of $\boldsymbol{z}$, $\boldsymbol{f}$, and hidden state $\boldsymbol{h}$ are 256, 2048 and 3072, respectively.

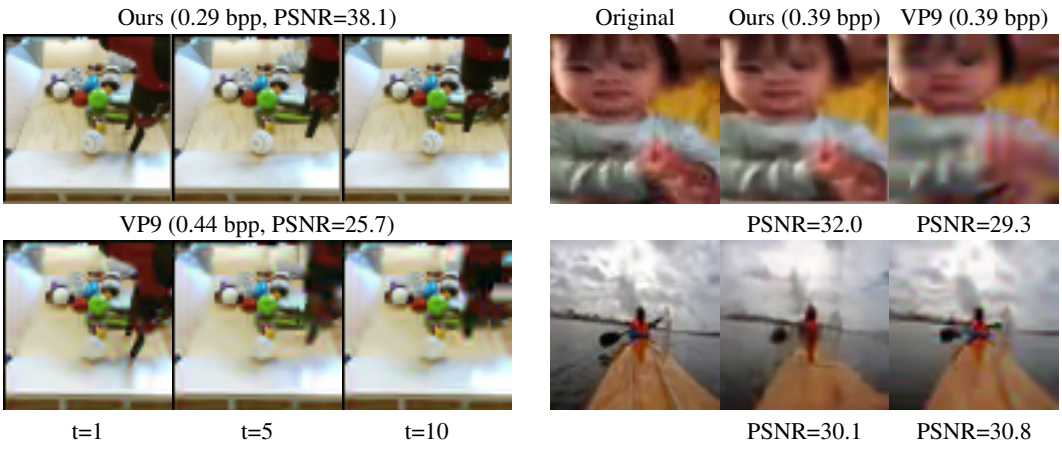



Figure 5: Compressed videos by our LSTMP-LG model and VP9 in the low bit rate regime. Our approach achieves better quality on specialized content (BAIR, left) and comparable visual quality on generic video content (Kinetics, right) compared to VP9.

**KFP-LG.** KFP-LG is also a proposed baseline model which incorporates both the global state $\boldsymbol{f}$ and local latent $\boldsymbol{z}_t$ but uses a weaker predictive model $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$ for entropy coding. The main purpose of the KFP-LG model is to compare to the LSTMP-LG model which has a longer memory. The conditional prior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$ in KFP-LG is described by a deep Kalman Filter parametrized by a three-layer MLP. The dimension at each layer of MLP is the same as the dimension of the latent state $\boldsymbol{z}_t$. KFP-LG has the same encoder and decoder structures as the proposed LSTMP-LG model aforementioned. The only difference between KFP-LG and LSTMP-LG is that they employ different prior models for conditional entropy coding.

## B Qualitative Results

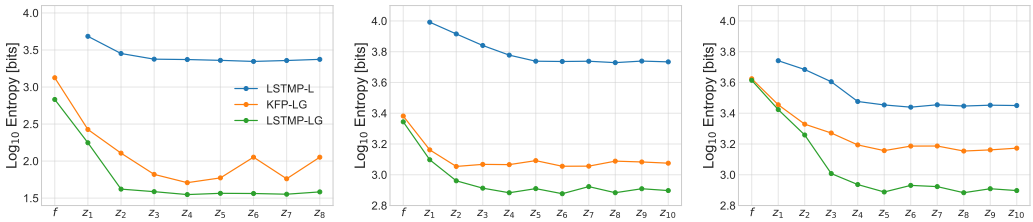
Now we discuss the qualitative performance of our method. We have shown that a deep neural approach to encode video (LSTMP-LG architecture) can outperform traditional codecs with respect to PSNR metrics overall on low-resolution videos. Test videos from the Sprites and BAIR datasets after compression 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 D).

## C Latent Variable Entropy Visualization

**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. 6. 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 $\boldsymbol{z}_1$ has a

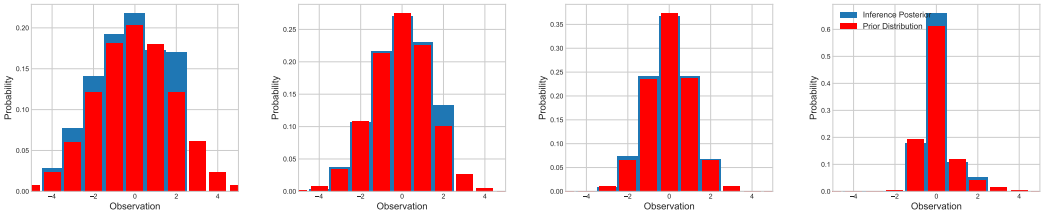
cold start. Note that our approach relies on sequential decoding, prohibiting a bi-directional LSTM for the local state.




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

Figure 6: 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.

## D Latent Variable Distribution 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 7: Empirical distributions of the posterior of the inference model estimated from BAIR data and the 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 the inference model estimated from data. In Fig. 7, 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 regime. From Fig. 7, 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 $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})$ is high dimensional, we do not visualize the distribution.