Deep Generative Video Compression with Temporal Autoregressive Transforms

Ruihan Yang [1] **Yibo Yang** [1] **Joseph Marino** [2] **Yang Yang** [3] **Stephan Mandt** [1]

Abstract

State of the art learned methods for lossy video compression (Han et al., 2018; Liu et al., 2019; Lu et al., 2019; Habibian et al., 2019; Yang et al., 2020) build on sequential latent variable models like the sequential VAE (Chung et al., 2015). Recently, Marino et al. (2020) demonstrated improved density modeling performance by extending a sequential VAE with a masked autoregressive flow. We adapt this hybrid model to the task of video compression, allowing better modeling of lower-level video dynamics through learned autoregressive transforms. We train our model in an end-to-end fashion (Han et al., 2018) and evaluate it on low-resolution videos, achieving comparable or better rate-distortion performance compared to classical video codecs and a neural baseline (Han et al., 2018) that lacks the proposed autoregressive transform.

1. Introduction

Deep generative models have seen tremendous success in modeling high-dimensional sequential data such as video and audio (Lotter et al., 2016; Oord et al., 2016). There is growing interest in applying such models to video compression, which has the potential to reduce a sizable amount of global internet traffic (Cisco, 2017).

State-of-the-art neural methods for lossy video compression (Han et al., 2018; Liu et al., 2019; Lu et al., 2019; Habibian et al., 2019; Yang et al., 2020) are based on the common architecture design of a sequential variational autoencoder ("sequential VAE") (Chung et al., 2015), and employ recurrent or optical flow modules for modeling the low-level dynamics within video frames.

[1]University of California, Irvine [2]California Institute of Technology [3]Qualcomm AI Research, Qualcomm Technologies, Inc. Correspondence to: Ruihan Yang <ruihan.yang@uci.edu>, Yibo Yang <yibo.yang@uci.edu>.

Second workshop on *Invertible Neural Networks and Normalizing Flows* (ICML 2020).

Different from previous work, we consider the use of autoregressive invertible neural networks for modeling low-level video dynamics. We propose a hybrid model that combines a sequential VAE with autoregressive transforms, following the approach of (Marino et al., 2020). The combination of the two yields a powerful sequential model with the potential to capture more complex and structured dynamics compared to each one individually; we illustrate this concept in Figure 6 of Section 3. The autoregressive component is inspired by autoregressive normalizing flow (Kingma et al., 2016; Papamakarios et al., 2017), but we focus on a different task than the typical application of density estimation. Specifically, we apply autoregressive transforms along the time axis, and in a deterministic fashion that better aligns with the rate-distortion objective of lossy data compression. Our approach based on autoregressive transform has the additional advantage of avoiding the overhead of dedicated motion estimation in traditional frameworks.

We evaluate our method on two low-resolution video datasets, achieving competitive rate-distortion performance against widely-used traditional codecs H.265 and H.264. We also obtain performance improvement over a neural baseline (Han et al., 2018) that does not make use of autoregressive transforms, achieving better compression performance at higher bitrates and with better parameter efficiency.

2. Method

This section describes our proposed method. We first review and motivate the idea of combining autoregressive transforms with a sequential VAE, which our model builds on. We then describe the model architecture in detail.

2.1. Autoregressive Transform for Sequence Modeling

Let $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times D}$ be a sequence of video frames. Masked autoregressive flow (MAF) (Papamakarios et al., 2017) models the joint distribution $p(\mathbf{x}_{1:T})$ in terms of a simpler distribution of underlying "noise variables" $\mathbf{y}_{1:T} \in \mathbb{R}^{T \times D}$, through the following autoregressive transform:

$$\mathbf{x}_t = \exp(\alpha_t) \odot \mathbf{y}_t + \mu_t, \quad \text{where} \quad \alpha_t = f_{\alpha_t}(\mathbf{x}_{<t}) \text{ and } \mu_t = f_{\mu_t}(\mathbf{x}_{<t}), \tag{1}$$

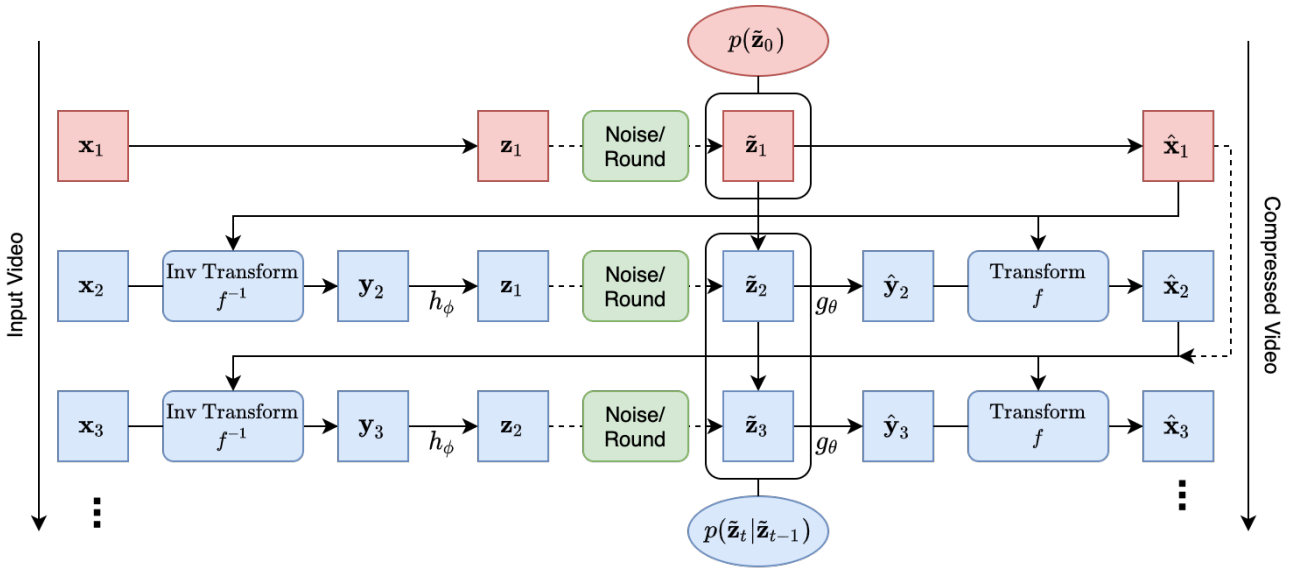


Figure 1. Computational diagram illustrating encoding and decoding with our proposed model. We used the image compression model of Ballé et al. (2016) for the initial frame (highlighted in red), and a sequential VAE with an autoregressive transform for the remaining frames. Given a sequence of video frames $\mathbf{x}_{1:T}$, an autogressive transform f decorrelates them across time into higher-level dynamics $\mathbf{y}_{1:T}$, which are then encoded by inference network h_ϕ into latent representations $\tilde{\mathbf{z}}_{1:T}$. $\tilde{\mathbf{z}}_{1:T}$ are then entropy-coded by a Markovian prior model, then decoded to $\hat{\mathbf{y}}_{1:T}$, which are finally transformed back to video reconstruction $\hat{\mathbf{x}}_{1:T}$ by the inverse transform f^{-1}. The dashed arrow originating from $\hat{\mathbf{x}}_1$ indicates the possibility of having an autoregressive dependence on more than one previous frame, although we simply used a single frame $\hat{\mathbf{x}}_{t-1}$ in our experiments.

and the inverse transform:

$$\begin{aligned} &\mathbf{y}_t = \exp(-\alpha_t) \odot (\mathbf{x}_t - \mu_t), \\ &\text{where} \quad \alpha_t = f_{\alpha_t}(\mathbf{x}_{<t}) \text{ and } \mu_t = f_{\mu_t}(\mathbf{x}_{<t}), \end{aligned} \tag{2}$$

where f_{α_t} and f_{μ_t} are generic functions parameterized by neural networks. Often the base distribution $p(\mathbf{y}_{1:T})$ is taken to be a standard normal distribution, and is related to the distribution $p(\mathbf{x}_{1:T})$ of video frames through the change-of-variables formula:

$$p(\mathbf{x}_{1:T}) = p(\mathbf{y}_{1:T}) \left| \det \left(\frac{\partial f^{-1}}{\partial \mathbf{x}_{1:T}} \right) \right|$$

Originally in the context of density modeling, Marino et al. (2020) proposed using such temporal autoregressive flows for modeling the dynamics within sequential latent variable models. Specifically, a sequential VAE (Chung et al., 2015) with latent variables $\mathbf{z}_{1:T}$ was introduced to model the base distribution $p(\mathbf{y}_{1:T})$ via

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{z}_{\leq t}) p(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t}), \tag{3}$$

where at each time step, $\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{z}_{\leq t})$ is transformed to $\mathbf{x}_t$ via the transform in Eq. 1. Alternatively, the inverse autoregressive transform in Eq. 2 can be seen as a pre-processing step that decorrelates the raw input $\mathbf{x}_{1:T}$ across time, providing the sequential VAE with a learned reference frame. The sequential VAE can then focus model capacity on the structure of higher-level dynamics $\mathbf{y}_{1:T}$, instead of the highly temporally correlated video signal $\mathbf{x}_{1:T}$.

2.2. Variational Compression Model

We adapt the hybrid sequential VAE + autoregressive transform idea to video compression. We base our architecture on that of (Han et al., 2018), taking it as the sequential VAE that models the base distribution of higher-level video dynamics, and extend it with an autoregressive transform (Eqs 1, 2) that ultimately generates the raw video frames. See Figure 1 for a flowchart overview of our architecture.

Decoder. A direct application of MAF to video frames like in the density modeling task of (Marino et al., 2020) is incompatible with the variational approach (Han et al., 2018; Ballé et al., 2018) to compression, which we explain below. To match that of a rate-distortion loss of lossy compression, the (negative) log-likelihood of the data $\mathbf{x}_{1:T}$ must correspond to some reconstruction error $D(\mathbf{x}_{1:T}, \hat{\mathbf{x}}_{1:T})$, where $\hat{\mathbf{x}}_{1:T}$ are reconstructed data. The sequential VAE + MAF model of (Marino et al., 2020), however, results in likelihood terms of the form $p(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t})$ which breaks this correspondence, as the reconstruction of $\mathbf{x}_t$ relies on previous *ground truth* frames $\mathbf{x}_{<t}$, which the decoder does not

have access to.

To resolve this issue, we employ a deterministic version of MAF, where we take the transforms in Eqs. 1 and 2 simply as invertible deterministic transforms between the raw video frames and "inverse" frames without any probabilistic interpretation. This allows us to use the model's reconstructed frames $\hat{\mathbf{x}}_{<t}$ as input to the forward transform in Eq. 1. The decoder is then described by the following generative model,

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p_\theta(\mathbf{z}_{1:T}) \prod_t p_\theta(\mathbf{x}_t | \hat{\mathbf{x}}_{<t}, \mathbf{z}_{\leq t}), \quad (4)$$

where $p_\theta(\mathbf{z}_{1:T})$ is the prior distribution over the latents discussed separately below, and $\hat{\mathbf{x}}_{<t}$ is a deterministic function of $\mathbf{z}_{<t}$. Specifically, we take $p_\theta(\mathbf{x}_t | \hat{\mathbf{x}}_{<t}, \mathbf{z}_{\leq t})$ to be a isotropic Gaussian distribution with mean $\hat{\mathbf{x}}_t$ and variance β, with $\hat{\mathbf{x}}_t$ computed by a deterministic version of Eq. 1:

$$\hat{\mathbf{x}}_t = \exp(f_{\alpha_t}(\hat{\mathbf{x}}_{<t})) \odot \hat{\mathbf{y}}_t + f_{\mu_t}(\hat{\mathbf{x}}_{<t}) \quad (5)$$

where $\hat{\mathbf{y}}_t = g_\theta(\mathbf{z}_t)$ is the output of the VAE's deconvolutional generative network g_θ. As we do not give $\hat{\mathbf{y}}_{1:T}$ any probabilistic interpretation, the above transform in Eq. 5 can be seen as part of a convolutional decoder of the extended sequential VAE defined by Eq. 4.

Alternatively, we could keep the sequential VAE + flow architecture of (Marino et al., 2020), and train the model with ground truth context frames $\mathbf{x} \leq t$ for the flow in Eqs. **??**, and replace them with reconstructed frames $\hat{\mathbf{x}} \leq t$ at compression time (similar to sampling in this generative model); however we obtained worse performance compared to our modified decoder with MSE likelihood, which is unsurprising given the mismatch between the training and compression objective.

Encoder Similarly, a deterministic inverse transform is recursively applied to an input video $\mathbf{x}_{1:T}$ to obtain "inverse" frames $\mathbf{y}_{1:T}$:

$$\mathbf{y}_t = \exp(-f_{\alpha_t}(\hat{\mathbf{x}}_{<t})) \odot (\mathbf{x}_t - f_{\mu_t}(\hat{\mathbf{x}}_{<t})) \quad (6)$$

Note that $\mathbf{y}_t$ is simply a deterministic transform of $\mathbf{x}_t$, and depends stochastically on $\mathbf{z}_{<t}$ through $\hat{\mathbf{x}}_{<t}$. We then encode latent representations $\mathbf{z}_{1:T}$ with a mean-field distribution:

$$q_\phi(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{x}_t).$$

Following the same amortized variational inference framework as in (Ballé et al., 2018; Han et al., 2018), we let $q_\phi(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{x}_t)$ be a uniform distribution, whose mean is computed from $\mathbf{y}_t$ by the VAE's inference network h_ϕ: $q_\phi(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{x}_t) = \mathcal{U}(h_\phi(\mathbf{y}_t) - \frac{1}{2}, h_\phi(\mathbf{y}_t) + \frac{1}{2})$. We sample $\tilde{\mathbf{z}}_t \sim q_\phi(\mathbf{z}_t | \mathbf{y}_t)$ during training, and let $\hat{\mathbf{z}}_t = \mathrm{round}(h_\phi(\mathbf{y}_t))$ at compression time; the resulting integer latents $\hat{\mathbf{z}}_t$ can then be entropy-coded into a bit-stream using the prior $p_\theta(\mathbf{z}_{1:T})$ described below.

Prior Models and Entropy Coding We use the same form of prior model $p_\theta(\mathbf{z}_{1:T})$ for entropy coding as in (Han et al., 2018), and limit the context of each conditional model $p_\theta(\mathbf{z}_t | \mathbf{z}_{<t})$ to a single frame for simplicity, i.e.,

$$p_\theta(\mathbf{z}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{z}_t | \mathbf{z}_{t-1})$$

Initial Frame Model In this work, we consider autoregressive transforms implemented by temporal convolutions with receptive field of size one. We thus independently parameterize the decoding and encoding distributions of the initial frame $\mathbf{x}_1$, implementing $p(\mathbf{x}_1 | \mathbf{z}_1)$ and $q(\mathbf{z}_1 | \mathbf{x}_1)$ with the established image compression VAE architecture of (Ballé et al., 2016).

Variational Objective We train our model end-to-end by minimizing the Negative Evidence Lower Bound (NELBO) w.r.t. parameters (θ, ϕ) of the sequential VAE and of the flow transform f:

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

corresponding to a rate-distortion trade-off (Alemi et al., 2017; Ballé et al., 2016), where the first term corresponds to the reconstruction error, and the second term corresponds to entropy estimate of the latents $\mathbf{z}_{1:T}$, with the trade-off controlled by β.

3. Experiment

We demonstrate our model's competitive performance on two low resolution datasets: BAIR robot pushing dataset (Finn et al., 2016) and the rescaled Vimeo-90K (Xue et al., 2019) dataset, with resolutions 64x64 and 128x128 respectively. On the specialized BAIR dataset, we significantly outperform classical codecs, and improve over a sequential VAE baseline (Han et al., 2018) that lacks the proposed autogressive transform. Encouragingly, we also outperform classical codecs on the more diverse Vimeo-90K dataset.

The dimension of latent variable of each frame is fixed to 256 in all experiments, and we trained models with $\beta \in \{0.02, 0.01, 0.005, 0.001\}$ in our results. We found it beneficial to pre-train the initial frame model by itself, then train the entire model jointly. We use video lengths of 10 and 5 frames to report the results on BAIR and Vimeo-90k dataset respectively.

3.1. Specialized video compression

As BAIR dataset only contains video frames for a specific task – a moving robot arm pushing objects, we primarily compare with a neural baseline architecture (Han et al., 2018) that lacks the proposed autoregressive transform. As we show (Figure 2 right), traditional codecs perform much worse than the neural methods on this specialized content. Figure 2 shows the resulting rate-distortion curves, where

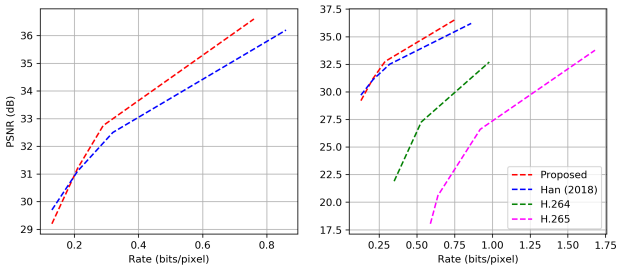


Figure 2. Rate-Distortion curves on the BAIR dataset. Left: comparison between neural methods, proposed (red) with autoregressive transform and baseline(blue) without. Right: additional comparison with classical codecs. Legend shared.

we used Peak Signal-to-Noise Ratio (PSNR) as video quality metric, and averaged PSNR and bit-rate (BPP) across all videos and time frames. The right subplot shows much worse performance by classical codecs as expected. The left subplot shows that our model produces a *steeper* curve than the baseline, with slight under-performance in the very low bit-rate regime. Moreover, our model is more parameter-efficient than the baseline (Han et al., 2018), using only 18M instead of 28M parameters. It is also worth noting that model proposed in (Han et al., 2018) represents a non-causal codec for which the compression of a frame depends on both past and future frames; whereas in our model, the compression of a frame does not rely on any future context.

For additional insight, we compare the bit-rate of our method against the baseline across time in Figure 3, where both methods spent the same average number of bits (0.13BPP) on the 10 frames. As shown, our model dedicates higher bit-rate to compressing the initial frame, but saves around 50% bits on the remaining frames compared to the baseline, which potentially indicates better performance on longer sequences. We remark that our method can likely be further improved by designing a more powerful model for the initial frame.

3.2. General video compression

We now compare our proposed method with classical codes H.265 and H.264. on more general video content from the Vimeo-90K dataset, containing 90 thousand clips of diverse real-world scenes and actions. The R-D results are presented

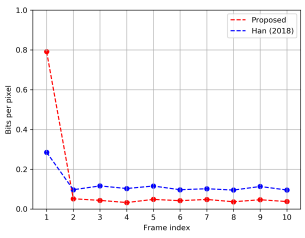


Figure 3. Bit-rate per frame on 10 frames of the BAIR dataset.

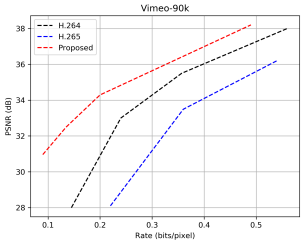


Figure 4. Rate-Distortion curves on the Vimeo-90k dataset.

in Figure 4, showing that our model clearly outperforming traditional codecs on a wide range of bit-rates, and especially successful in the low-rate regime.

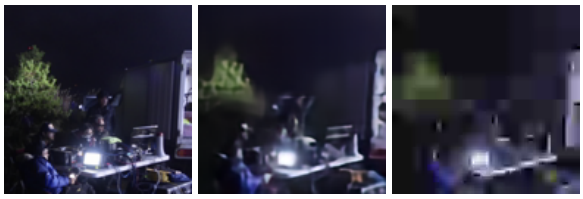
3.3. Qualitative analysis

Figure 5 demonstrates that our model can automatically extract higher-level dynamics from video in a *disentangled* way, so that the model mainly encodes the moving objects captured by the "inverse" sequence and keeps the unmoving background static. Figure 6 compares the compression quality of our proposed model with a classical codec H.264. Our model can clearly recover the structure of the original frame, whereas H.264 produces blocky artifacts.

4. Conclusion

We presented a hybrid model based on combining a sequential VAE with invertible autoregressive transforms. Our approach efficiently captures video dynamics, and substantially improves the compression performance compared to traditional codec and a previously proposed learned method (Han et al., 2018) without the proposed transform. Our model can be further improved by using more sophisticated invertible transforms, such as GLOW (Kingma & Dhariwal, 2018), neural autoregressive transform (Huang et al., 2018), etc., or by incorporating non-causal context (e.g., future frames). Furthermore, we can implement more evaluation experiment on high-resolution videos and improve the model by utilizing future context.

Figure 5. Visualization of our proposed hybrid approach on BAIR. Top row: reconstructed frames $\hat{\mathbf{x}}_{1:T}$; bottom row: underlying "inverse" sequence $\mathbf{y}_{1:T}$ computed by the inverse transform Eq. 6. The "inverse" sequence $\mathbf{y}_{1:T}$ can be seen to clearly separate the robot arm in motion from the static background, simplifying the compression task of the sequential VAE.



(a) Source (b) Proposed; PSNR: 27.84dB (0.143 BPP) (c) H.264; PSNR: 21.56dB (0.159 BPP)

Figure 6. Qualitative comparison of compression performance on a frame from Vimeo-90K. The bit-rate (BPP) is averaged across the 5-frame sequence containing the frame being compared.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0021. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA). Furthermore, this work was supported by the National Science Foundation under Grants 1928718 and 2003237, and by Qualcomm.

References

Alemi, A. A., Poole, B., Fischer, I., Dillon, J. V., Saurous, R. A., and Murphy, K. Fixing a broken elbo. *arXiv preprint arXiv:1711.00464*, 2017.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Cisco, V. N. I. Forecast and methodology, 2016-2021. *White Paper*, 2017.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pp. 64–72, 2016.

Habibian, A., Rozendaal, T. v., Tomczak, J. M., and Cohen, T. S. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7033–7042, 2019.

Han, J., Lombardo, S., Schroers, C., and Mandt, S. Deep probabilistic video compression. *arXiv preprint arXiv:1810.02845*, 2018.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. *arXiv preprint arXiv:1804.00779*, 2018.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pp. 4743–4751, 2016.

Liu, H., Huang, L., Lu, M., Chen, T., Ma, Z., et al. Learned video compression via joint spatial-temporal correlation exploration. *arXiv preprint arXiv:1912.06348*, 2019.

Lotter, W., Kreiman, G., and Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., and Gao, Z. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11006–11015, 2019.

Marino, J., Chen, L., He, J., and Mandt, S. Improving sequential latent variable models with autoregressive flows. In *Symposium on Advances in Approximate Bayesian Inference*, pp. 1–16, 2020.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.

Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

Yang, Y., Sautière, G., Ryu, J. J., and Cohen, T. S. Feedback recurrent autoencoder. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3347–3351. IEEE, 2020.