Deep Generative Video Compression

Jun Han
with Salvator Lombardo, Christopher Schroers, Stephan Mandt

accepted at NeurIPS 2019

November 5, 2019
Background & Motivation

Traditional codecs
- H.264/H265; VP9

Figure: Key frame $x_1$; Reference frames $x_{>1}$
Background & Motivation

1. Motion $v_t$ from $x_t$ and $\hat{x}_{t-1}$
2. Pred $\bar{x}_t$ and Resid $r_t = x_t - \bar{x}_t$
3. Transf and Quantize $r_t$ to $\hat{y}_t$
Background & Motivation

1. Motion $v_t$ from $x_t$ and $\hat{x}_{t-1}$

2. Pred $\bar{x}_t$ and Resid $r_t = x_t - \bar{x}_t$

3. Transf and Quantize $r_t$ to $\hat{y}_t$

4. Inverse $\hat{y}_t$ to residual $\hat{r}_t$

5. Entropy coding $v_t$ and $\hat{y}_t$

6. Reconstruct $\hat{x}_t = \bar{x}_t + \hat{r}_t$
Comparison

H.265 (0.86 bpp)  
VP9 (0.57 bpp)  
Ours (0.06 bpp)  

bpp: bit per pixel; original: $8 \times 3 = 24$ bits per pixel.
Variational Autoencoder

**Encoder**

\[ \mathbf{x} \rightarrow \mu_{\phi}(\mathbf{x}) \rightarrow \mathbf{z} \rightarrow \mu_{\theta}(\mathbf{z}) \rightarrow \tilde{\mathbf{x}} \]

**Decoder**

**Input Image**

**Reconstructed Image**

**Figure:** \( \mathbf{x} \): image; \( \tilde{\mathbf{x}} \): reconstructed image or sample. \( \mathbf{z} \): latent variable.

Let \( q_{\phi} \): inference model; \( p_{\theta} \): generative model; \( p(\mathbf{z}) \): prior,

\[
\mathcal{L}(\phi; \theta) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[\log p_{\theta}(\mathbf{x} | \mathbf{z})] - D_{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})),
\]
Variational Autoencoder

Let $q_\phi$: inference model; $p_\theta$: generative model; $p(z)$: prior,
\[
\mathcal{L}(\phi; \theta) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x | z)] - D_{KL}(q_\phi(z | x) \parallel p(z)),
\]

**Problem for coding**
- Inference model $q_\phi$ is Gaussian and $z$ is continuous.
- $z$ should be discrete for **coding** (arithmetic coding; Huffman coding).

Figure: $x$: image; $\tilde{x}$: reconstructed image or sample. $z$: latent variable.
Figure: \( \tilde{z} \): adding noise (training) or rounding (after training).
Deep Image Compression (Balle, ICLR 2017)

Figure: \( \tilde{z} \): adding noise (training) or rounding (after training). Prior \( p_\theta(\tilde{z}) \): parametric form to fit data distribution for entropy coding.
Deep Image Compression (Balle, ICLR 2017)

Figure: $\tilde{z}$: adding noise (training) or rounding (after training). Prior $p_\theta(\tilde{z})$: parametric form to fit data distribution for entropy coding.

- Inference model: $\tilde{z} \sim q_\phi(z \mid x) = \mathcal{U}(z - \frac{1}{2}, z + \frac{1}{2})$
- Loss: $\mathcal{L}(\phi; \theta) = \mathbb{E}_{\tilde{z} \sim q}[\log p_\theta(x \mid \tilde{z})] - \beta \left( 0 - \mathbb{E}_{\tilde{z} \sim q}[\log p_\theta(\tilde{z})]\right)$, where $\beta$ adjusts rate-distortion ratio.
### Entropy Coding

**Arithmetic Coding**

<table>
<thead>
<tr>
<th>symbols</th>
<th>probs</th>
<th>c.m.f</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.12</td>
<td>[0, 0.12)</td>
</tr>
<tr>
<td>E</td>
<td>0.42</td>
<td>[0.12, 0.54)</td>
</tr>
<tr>
<td>I</td>
<td>0.09</td>
<td>[0.54, 0.63)</td>
</tr>
<tr>
<td>O</td>
<td>0.3</td>
<td>[0.63, 0.93)</td>
</tr>
<tr>
<td>U</td>
<td>0.07</td>
<td>[0.93, 1.0)</td>
</tr>
</tbody>
</table>

**IOU**: $0.37630 \leq C < 0.37819$.  
**Binary**: $0.0110000001$ (9 bits).
**Entropy Coding**

**Arithmetic Coding**

<table>
<thead>
<tr>
<th>symbols</th>
<th>probs</th>
<th>c.m.f</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.12</td>
<td>[0, 0.12)</td>
</tr>
<tr>
<td>E</td>
<td>0.42</td>
<td>[0.12, 0.54)</td>
</tr>
<tr>
<td>I</td>
<td>0.09</td>
<td>[0.54, 0.63)</td>
</tr>
<tr>
<td>O</td>
<td>0.3</td>
<td>[0.63, 0.93)</td>
</tr>
<tr>
<td>U</td>
<td>0.07</td>
<td>[0.93, 1.0)</td>
</tr>
</tbody>
</table>

**Figure:** Symbol **IOU**

**IOU:** $0.37630 \leq C < 0.37819$.

**Binary:** $0.0110000001$ (9 bits).

- Known probability models and length of symbols to decode
- Average of code lengths approximates the cross entropy
Proposed Baseline Video Compression Model

Figure: Two choices of **predictive models** for entropy coding: $p_\theta(\tilde{z}_t \mid \tilde{z}_{<t})$ Kalman Filter; $p_\theta(\tilde{z}_t \mid \tilde{z}_{t-1})$ LSTM.
**Improved Video Compression Model**

**Figure:** Global state $f$: per-segment, inferred from a segment $T$ of video by LSTM after encoder $\mu_\phi$; local state $z_t$: per-frame, inferred from $x_t$ after encoder $\mu_\phi$. 
Encoder:

\[ q_\phi(z_{1:T}, f | x_{1:T}) = q_\phi(f | x_{1:T}) \prod_{t=1}^{T} q_\phi(z_t | x_t). \]

\[ \tilde{f} \sim q_\phi(f | x_{1:T}) = \mathcal{U}(\hat{f} - \frac{1}{2}, \hat{f} + \frac{1}{2}); \quad \tilde{z}_t \sim q_\phi(z_t | x_t) = \mathcal{U}(\hat{z}_t - \frac{1}{2}, \hat{z}_t + \frac{1}{2}). \]
Choice of probability models

**Encoder:**

$$ q_\phi(z_{1:T}, f \mid x_{1:T}) = q_\phi(f \mid x_{1:T}) \prod_{t=1}^{T} q_\phi(z_t \mid x_t). $$

$$ \tilde{f} \sim q_\phi(f \mid x_{1:T}) = \mathcal{U}(\hat{f} - \frac{1}{2}, \hat{f} + \frac{1}{2}); \quad \tilde{z}_t \sim q_\phi(z_t \mid x_t) = \mathcal{U}(\hat{z}_t - \frac{1}{2}, \hat{z}_t + \frac{1}{2}). $$

**Prior:**

$$ p_\theta(f) = \prod_{i}^{\dim(f)} p_\theta(f^i) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}); $$

$$ p_\theta(z_{1:T}) = \prod_{t}^{T} \prod_{i}^{\dim(z)} p_\theta(z^i_t \mid z_{<t}) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}). $$
Experiments

Datasets:

- **Sprites**: characters from video game \((64 \times 64)\).
- **BAIR**: a robot pushing objects on a table \((64 \times 64)\).
- **Kinetics**: diverse set of human actions (downsampled and cropped to \(64 \times 64\)).

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

Datasets:
- **Sprites**: characters from video game (64 × 64).
- **BAIR**: a robot pushing objects on a table (64 × 64).
- **Kinetics**: diverse set of human actions (downsampled and cropped to 64 × 64).

Results
- Verification experiments
- Comparison with codecs
Latent Variable Distribution Visualization

Figure: Empirical distributions of the latent states from inference posteriors and learned prior model in one BAIR example.
Latent Variable Entropy Visualization

(a) Sprites

(b) BAIR

(c) Kinetics

Figure: Average bits of information stored in $f$ and $z_{1:T}$ for LSTMP-L (LSTM prior with only local states), KFP-LG (Kalman Filter prior with local and global states), and LSTMP-LG (LSTM prior with local and global states).
Quantitative Comparison

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

Figure: Rate-distortion curves on three datasets measured in PSNR (higher corresponds to lower distortion). Solid lines correspond to our models.
Qualitative Comparison (BAIR)

Ours (0.29 bpp, PSNR=38.1)

VP9 (0.44 bpp, PSNR=25.7)

$t=1$  $t=5$  $t=10$

Figure: Compressed videos by our LSTMP-LG and VP9 in low-bit rate regime.
Qualitative Comparison (Kinetics)

Original | Ours (0.39 bpp) | VP9 (0.39 bpp)

PSNR=32.0 | PSNR=29.3

PSNR=30.1 | PSNR=30.8

Figure: Compressed videos by our LSTMP-LG and VP9 in low-bit rate regime.
Summary

- first end-to-end trainable video codec in VAE framework.
- global and local states with predictive model for efficient entropy coding.
- small rate in specialized content and competitive rate in generic content.
Summary

- first end-to-end trainable video codec in VAE framework.
- global and local states with predictive model for efficient entropy coding.
- small rate in specialized content and competitive rate in generic content.

Future Work

- Extend to full-resolution video (Conv LSTM)
- Incorporate motion compensation
- Include hierarchical structure

Thank You