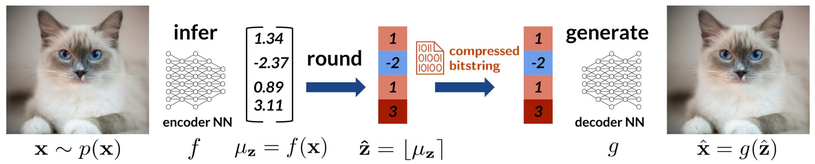
Improving Inference for Neural Image Compression

Yibo Yang, Robert Bamler, Stephan Mandt
Department of Computer Science, UC Irvine

Summary: better inference ⇨ better compression

- We identify common approximation gaps in existing neural image compression methods based on variational autoencoders (VAEs).
- Viewing data compression as *inference* with respect to a given decoder, we propose to close these approximation gaps with new algorithms based on iterative inference, stochastic discrete optimization, and bits-back coding.
- We dramatically improve the performance of an existing competitive compression model (Minnen et al., 2018) by changing only the inference method at test time, and achieve new state-of-the-art results in lossy image compression.

Background: neural lossy data compression through variational inference



- Current neural methods compress a data observation $\mathbf{x}$ with two steps:
  1. An encoder neural net computes its continuous latent representation $\mu_{\mathbf{z}} = f(\mathbf{x})$
  2. A discrete latent representation is obtained by rounding $\hat{\mathbf{z}} = \lfloor \mu_{\mathbf{z}} \rceil$, which can then be converted to a bit-string and decoded.
- This compression procedure incurs the following rate-distortion cost:

$$\mathcal{L}(\hat{\mathbf{z}}) = -\log_2 P(\hat{\mathbf{z}}) + \lambda \|\mathbf{x} - g(\hat{\mathbf{z}})\|^2$$

- To enable gradient-based optimization w.r.t. parameters of the encoder network, rounding is approximated by adding uniform noise, so the rate-distortion cost (approximately) corresponds to a negative ELBO (Evidence Lower BOund):

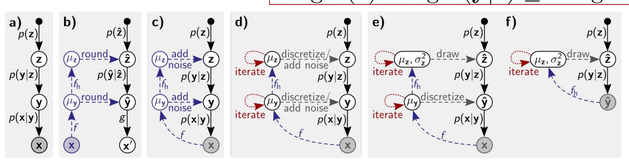
$$\tilde{\mathcal{L}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z}) + KL(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))]$$

where

$$q(\mathbf{z}|\mathbf{x}) = q(\mathbf{z}|f(\mathbf{x})) = \mathcal{U}(\mu_{\mathbf{z}} - 0.5, \mu_{\mathbf{z}} + 0.5)$$
$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|g(\mathbf{z}), 1/(2\lambda \log 2)\mathbf{I})$$

Motivation: three common approximation gaps

1. Amortization gap: amortized inference is too restrictive compared to iterative inference at test time: $\min_{\mu_{\mathbf{z}}} \tilde{\mathcal{L}}(\mu_{\mathbf{z}}) \leq \tilde{\mathcal{L}}(f(\mathbf{x}))$
2. Discretization gap: the negative ELBO is only a continuous approximation to the true rate-distortion cost: $\tilde{\mathcal{L}}(\lfloor \arg\min_{\mu_{\mathbf{z}}} \tilde{\mathcal{L}}(\mu_{\mathbf{z}}) \rceil) \neq \min_{\hat{\mathbf{z}} \in \mathbb{Z}^n} \mathcal{L}(\hat{\mathbf{z}})$
3. Marginalization gap: state-of-the-art compression models also incorporate a hyper-prior, where the transmission of hyper-latents incurs an overhead: $-\log P(\hat{\mathbf{z}}) - \log P(\hat{\mathbf{y}}|\hat{\mathbf{z}}) \geq -\log P(\hat{\mathbf{y}})$



Graphical illustration of a hierarchical VAE (a), baseline compression/training procedures (b, c), and proposed SGA (d) and lossy bits-back (e, f) algorithms.

Closing the discretization gap (and amortization gap) with Stochastic Gumbel Annealing (SGA)

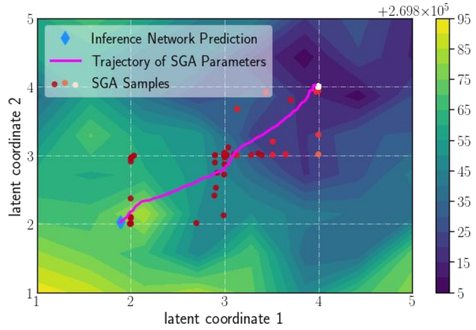
- Compressing data to a bit-string is an inherently *discrete* optimization problem:

$$\hat{\mathbf{z}}^* = \arg\min_{\hat{\mathbf{z}} \in \mathbb{Z}^n} \mathcal{L}(\hat{\mathbf{z}}) = \arg\min_{\hat{\mathbf{z}} \in \mathbb{Z}^n} \lambda \|\mathbf{x} - g(\hat{\mathbf{z}})\|^2 + (-\log_2 P(\hat{\mathbf{z}}))$$

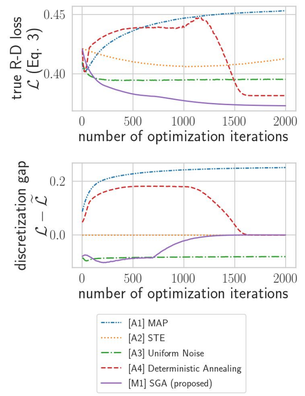
- SGA: introduces continuous proxy parameters $\mu_{\mathbf{z}}$ that get stochastically rounded to $\hat{\mathbf{z}} = r_{\mathbf{z}} \cdot (\lfloor \mu_{\mathbf{z}} \rfloor, \lceil \mu_{\mathbf{z}} \rceil)$ according to stochastic rounding directions $r_{\mathbf{z}}$.
- SGA: optimizes a variational upper bound on the original discrete problem

$$\tilde{\mathcal{L}}_{SGA}(\mu_{\mathbf{z}}) := \mathbb{E}_{q_\tau(r_{\mathbf{z}}|\mu_{\mathbf{z}})}\left[\mathcal{L}\left(r_{\mathbf{z}} \cdot (\lfloor \mu_{\mathbf{z}} \rfloor, \lceil \mu_{\mathbf{z}} \rceil)\right)\right] \geq \min_{\hat{\mathbf{z}} \in \mathbb{Z}^n} \mathcal{L}(\hat{\mathbf{z}})$$

- Discretization gap is closed by annealing the temperature $\tau \to 0$ throughout optimization



SGA optimization landscape; darker region = lower loss (better); SGA samples are colored by temperature (brighter color = lower temperature).



Comparing the true rate-distortion loss and discretization gap of SGA against alternative iterative inference methods, on Kodak images.

Closing the marginalization gap with lossy bits-back coding

- Key observation: in the hierarchical VAE (see Motivation: marginalization gap), the lower level latent representations $\hat{\mathbf{y}}$ are compressed losslessly.
- Therefore we apply bits-back coding to compress $\hat{\mathbf{y}}$ (using hyper-latents $\hat{\mathbf{z}}$ as the stochastic latent code), thus approximately coding $\hat{\mathbf{y}}$ under its *marginal* prior $P(\hat{\mathbf{y}})$.

Algorithm 1: Proposed lossy bits-back coding method (Section 3.3 and Figure 1e-f).

Global Constants: Trained hierarchical VAE with model $p(\mathbf{z}, \mathbf{y}, \mathbf{x}) = p(\mathbf{z})\, p(\mathbf{y}|\mathbf{z})\, p(\mathbf{x}|\mathbf{y})$ and inference networks $f(\,\cdot\,;\phi)$ and $f_{\mathrm{h}}(\,\cdot\,;\phi_{\mathrm{h}})$, see Figure 1e ($f$ is only used in subroutine `encode`).

Subroutine `encode`(image $\mathbf{x}$, side information $\boldsymbol{\xi}$) $\mapsto$ returns compressed bitstring $\mathbf{s}$

Initialize $\mu_{\mathbf{y}} \leftarrow f(\mathbf{x};\phi)$ and $(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2) \leftarrow f_{\mathrm{h}}(\mu_{\mathbf{y}};\phi_{\mathrm{h}})$. ▷ *Figure 1e (blue)*
Optimize over $\hat{\mathbf{y}}$ using SGA (Section 3.2), and over $\mu_{\mathbf{z}}$ and $\sigma_{\mathbf{z}}^2$ using BBVI. ▷ *Figure 1e (red)*
Reset $(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2) \leftarrow$ `reproducible_BBVI`$(\hat{\mathbf{y}})$. # to ensure the encoder and decoder find the same $q(\mathbf{z}|\hat{\mathbf{y}}) = \mathcal{N}(\mathbf{z}|\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2)$
Decode side information $\boldsymbol{\xi}$ into $\hat{\mathbf{z}}$ using $Q(\hat{\mathbf{z}}|\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^{\varepsilon})$ as entropy model.
Encode $\hat{\mathbf{z}}$ and $\hat{\mathbf{y}}$ into $\mathbf{s}$ using $P(\hat{\mathbf{z}})$ and $P(\hat{\mathbf{y}}|\hat{\mathbf{z}})$ as entropy models, respectively.

Subroutine `decode`(compressed bitstring $\mathbf{s}$) $\mapsto$ returns (lossy reconstruction $\mathbf{x}'$, side info $\boldsymbol{\xi}$)

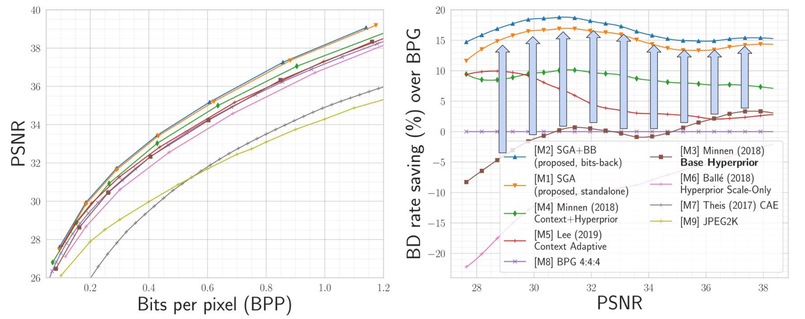
Decode $\mathbf{s}$ into $\hat{\mathbf{z}}$ and $\hat{\mathbf{y}}$; then get reconstructed image $\mathbf{x}' = \arg\max_{\mathbf{x}} p(\mathbf{x}|\hat{\mathbf{y}})$.
Set $(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2) \leftarrow$ `reproducible_BBVI`$(\hat{\mathbf{y}})$. # to ensure the encoder and decoder find the same $q(\mathbf{z}|\hat{\mathbf{y}}) = \mathcal{N}(\mathbf{z}|\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2)$
Encode $\hat{\mathbf{z}}$ into $\boldsymbol{\xi}$ using $Q(\hat{\mathbf{z}}|\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2)$ as entropy model.

Subroutine `reproducible_BBVI`(discrete latents $\hat{\mathbf{y}}$) $\mapsto$ returns variational parameters $(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2)$

Initialize $(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2) \leftarrow f_{\mathrm{h}}(\hat{\mathbf{y}};\phi_{\mathrm{h}})$; seed random number generator reproducibly. ▷ *Figure 1f (blue)*
Refine $\mu_{\mathbf{z}}$ and $\sigma_{\mathbf{z}}^2$ by running BBVI for fixed $\hat{\mathbf{y}}$. ▷ *Figure 1f (red)*

Results (code repo: https://github.com/mandt-lab/improving-inference-for-neural-image-compression)

- Improved compression performance by improving inference at *compression* time, using pre-trained baseline models (Minnen et al., 2018).
- New state-of-the-art lossy image compression performance on Kodak (SGA gave 15% avg rate savings over neural baseline (Minnen et al., 2018) and BPG) and Tecnick (SGA gave 19% avg rate savings over neural baseline (Minnen et al., 2018), 21% over BPG); lossy bits-back gave an additional 1~2% avg rate savings.



Compression performance comparisons on Kodak against existing baselines. Left: rate-distortion curves. Right: average rate savings (%) relative to BPG (state-of-the-art traditional codec). Legend shared; higher values are better in both.

Qualitative comparison between the compressed reconstruction of baseline method (left; Minnen et al., 2018) and ours (right) with SGA (using the same generative model), for an example Kodak image.

Acknowledgements and References

We thank Yang Yang for valuable feedback, and the Hasso Plattner Foundation, DARPA (Contract No. HR001120C0021), National Science Foundation (Grants 1928718, 2003237 and 2007719), and Qualcomm for support.

[Minnen et al., 2018] "Joint autoregressive and hierarchical priors for learned image compression." In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.