End-to-End Learned Image Compression with Augmented Normalizing Flows

Yung-Han Ho[1] Chih-Chun Chan[1] Wen-Hsiao Peng[1,3] Hsueh-Ming Hang[2,3]

{hectorho0409.cs04g@, dororojames.cs07g, wpeng@cs., hmhang@}nctu.edu.tw

[1]Computer Science Dept., [2]Electronics Engineering Dept.,
[3]Pervasive AI Research (PAIR) Labs, National Chiao Tung University, Taiwan

Abstract

This paper presents a new attempt at using augmented normalizing flows (ANF) for lossy image compression. ANF is a specific type of normalizing flow models that augment the input with an independent noise, allowing a smoother transformation from the augmented input space to the latent space. Inspired by the fact that ANF can offer greater expressivity by stacking multiple variational autoencoders (VAE), we generalize the popular VAE-based compression framework by the autoencoding transforms of ANF. When evaluated on Kodak dataset, our ANF-based model provides 3.4% higher BD-rate saving as compared with a VAE-based baseline that implements hyper-prior with mean prediction. Interestingly, it benefits even more from the incorporation of a post-processing network, showing 11.8% rate saving as compared to 6.0% with the baseline plus post-processing.

1. Introduction

End-to-end learned image compression [1, 2, 3, 4, 6, 9, 10] has recently made great progress in terms of compression efficiency, showing comparable rate-distortion performance to the state-of-the-art hand-crafted codecs, such as the Versatile Video Coding (VVC) standard under the All Intra configuration. This line of research started with Balle et al. [1] proposing a variational autoencoder (VAE)-based analysis and synthesis architecture together with a learned global prior for entropy coding the image latents. The global prior is later replaced with a hyper-prior structure in [2], which encodes additional side information to enhance the density estimation of the image latents for better entropy coding. Minnen et al. [10] further combines the hyper-prior with an autoregressive prior model to make best use of causal context information. Based on the same VAE-based framework, Cheng et al. [4] improve the representation learning and the density estimation by introducing attention modules in the autoencoder and Gaussian mixture priors, respectively.

Deviating from these prior works, this paper makes a new attempt at using normalizing flow (NF) models [5, 8] for learning image representations and their prior distributions. The NF model is deep invertible models, which transform input data into their latent representations via a bijective mapping, often implemented by an invertible network composed of affine coupling layers (Fig. 1). The bijective mapping allows the NF model to be trained by maximizing the exact data likelihood. In this paper, we turn to a specific type of NF models, called augmented normalizing flows (ANF) [7] (Fig. 1). Unlike the ordinary NF model, the ANF augments the input with an independent noise. It is argued in [7] that the augmented input space allows a smoother transformation to the latent space. More importantly, VAE is shown to be a special case of ANF, which can have greater expressivity by stacking multiple VAEs. These insights motivate our use of ANF for generalizing the popular VAE-based image compression framework.

Experimental results on Kodak dataset show that our ANF-based approach outperforms an enhanced VAE-based baseline (hyper-prior with mean prediction). In terms of BD-rate saving with BPG as anchor, it achieves 3.3% more rate saving than the baseline. Interestingly, it benefits even more from the incorporation of a post-processing enhancement network, showing 11.8% rate saving as compared to 6.0% with the baseline plus post-processing.

The remainder of this paper is organized as follows: Section 2 overviews the VAE-based image compression and augmented normalizing flows. Section 3 details our proposed method. Section 4 presents experimental results. Section 5 concludes this work.

2. Related Work

2.1. VAE-based Image Compression

End-to-end learned image compression usually includes three major elements: analysis, entropy coding, and synthesis. The analysis of an image x is done by converting it through an encoding distribution $q_{\pi}^{enc}(z|x)$ into a discretized latent representation z, the process of which in-

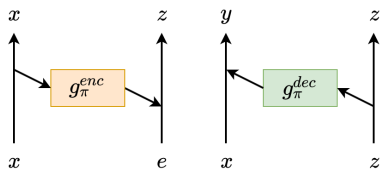

Figure 1. The autoencoding transform of one-step ANF: the encoding transform (left) and the decoding transform (right).

volves quantization for lossy compression. The latent z is then entropy encoded into a bitstream using a learned prior $p_\pi(z)$. The synthesis part decodes the bitstream and reconstructs approximately the input x by a decoding distribution $p_\pi^{dec}(x|z)$. All the network parameters, encapsulated in π, are trained end-to-end by minimizing

$$\mathcal{L}(\pi; x) = \underbrace{-E_{q_\pi^{enc}(z|x)}[\log p_\pi^{dec}(x|z)]}_{D} \underbrace{-E_{q_\pi^{enc}(z|x)}[\log p_\pi(z)]}_{R}, \tag{1}$$

where the first term, denoted by D, aims to minimize the negative log-likelihood of x and the second term to minimize the rate R needed for signaling z. Minimizing Eq. (1) admits of the interpretation of maximizing the evidence lower bound (ELBO) of a latent variable model specified by $p_\pi(z)$ and $p_\pi^{dec}(x|z)$, with $q_\pi^{enc}(z|x)$ taking a uniform distribution that models the effect of uniform quantization. In a more general setting, a hyper-parameter λ is introduced to balance D against R, yielding $\mathcal{L}(\pi; x) = D + \lambda R$.

2.2. Augmented Normalizing Flows

The ANF model is an invertible latent variable model. It is composed of multiple *autoencoding* transforms, each of which comprises a pair of the encoding and decoding transforms as depicted in Fig. 1. Consider the example of ANF with one autoencoding transform (i.e. one-step ANF). It converts the input x coupled with an independent noise e into a latent representation (y, z) with one pair of encoding and decoding transforms:

$$g_\pi^{enc}(x, e) = (x, s_\pi^{enc}(x) \odot e + m_\pi^{enc}(x)) = (x, z) \tag{2}$$

$$g_\pi^{dec}(x, z) = ((x - \mu_\pi^{dec}(z))/\sigma_\pi^{dec}(z), z) = (y, z) \tag{3}$$

where π is the network parameters. VAE is seen to be a special case of one-step ANF by letting $e \sim \mathcal{N}(0, I)$, the encoding distribution $q_\pi^{enc}(z|x) = \mathcal{N}(m_\pi^{enc}(x), (s_\pi^{enc}(x))^2)$, the decoding distribution $p_\pi^{dec}(x|z) = \mathcal{N}(\mu_\pi^{dec}(x), (\sigma_\pi^{dec}(x))^2)$, with the priors y and z following $\mathcal{N}(0, I)$, respectively.

From Fig. 1, the encoding g_π^{enc} or decoding g_π^{dec} transform implements an invertible affine coupling layer. Stacking pairs of these coupling layers leads also to an invertible network. As such, ANF can be trained by maximizing the

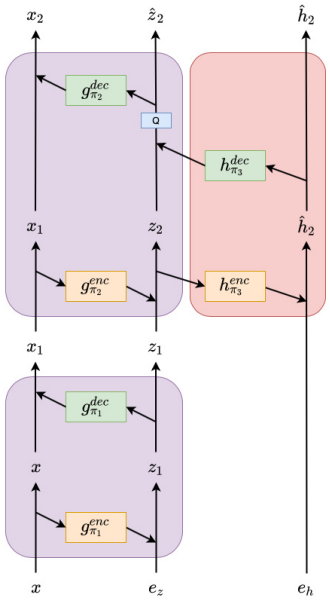

Figure 2. The proposed ANF-based image compression framework: the autoencoding transforms for feature extraction (purple); and the autoencoding transform of the hyper-prior (pink). Q denotes quantization with the nearest-integer rounding.

augmented joint likelihood, i.e. $\arg\max_\pi p_\pi(x, e)$:

$$p_\pi(x, e) = p(G_\pi(x, e)) \left| det \frac{\partial G_\pi(x, e)}{\partial (x, e)} \right|, \tag{4}$$

where $G_\pi = g_{\pi_N}^{dec} \circ g_{\pi_N}^{enc} \circ \ldots \circ g_{\pi_1}^{dec} \circ g_{\pi_1}^{enc}$ is the alternate composition of the encoding and decoding transforms with $\pi = \{\pi_1, \cdots, \pi_N\}$ and $p(G_\pi(x, e))$ represents the specified or learned prior distribution over the latent (y, z).

Huang et al. [7] proves that maximizing the augmented joint likelihood $p_\pi(x, e)$ in ANF amounts to maximizing a lower bound on the marginal likelihood $p_\pi(x)$, with the gap attributed to the model's incapability of modeling e independently of x.

3. Proposed Method

3.1. ANF-based Compression Framework

Fig. 2 depicts our ANF-based image compression framework. It includes two autoencoding transforms (i.e. two-step ANF), with the upper one extended further to the right to form a hierarchical ANF [7] that implements the hyper-prior. The g_π^{enc} and g_π^{dec} in the autoencoding transforms follow Eqs. (2) and (3). The encoding and decoding transforms of the hyper-prior are defined as:

$$h_{\pi_3}^{enc}(z_2, e_h) = (z_2, e_h + m_{\pi_3}^{enc}(z_2)) = (z_2, \hat{h}_2) \tag{5}$$

$$h_{\pi_3}^{dec}(z_2, \hat{h}_2) = (\lfloor z_2 - \mu_{\pi_3}^{dec}(\hat{h}_2) \rceil, \hat{h}_2) = (\hat{z}_2, \hat{h}_2) \tag{6}$$

where $\lfloor \cdot \rceil$ denotes the nearest-integer rounding.

Our ANF model operates by passing the augmented input (x, e_z, e_h) through the autoencoding and hyper-prior transforms, i.e. $G_\pi = g^{dec}_{\pi_2} \circ h^{dec}_{\pi_3} \circ h^{enc}_{\pi_3} \circ g^{enc}_{\pi_2} \circ g^{dec}_{\pi_1} \circ g^{enc}_{\pi_1}$, to obtain the latent representation $(x_2, \hat{z}_2, \hat{h}_2)$. In particular, x represents the input image, $e_z \sim \mathcal{N}(0, I)$ denotes the augmented Gaussian noise, and $e_h \sim \mathcal{U}(-0.5, 0.5)$ simulates the additive quantization noise of the hyper prior. To reconstruct approximately the input x, we apply the inverse mapping function G^{-1}_π to the quantized latent $(0, \hat{z}_2, \hat{h}_2)$, where x_2 is set to zero as will be explained next.

3.2. Latent Representation and Prior Distribution

In our scheme, z_2 serves as the latent representation of x and $m^{enc}_{\pi_3}(z_2)$ is the corresponding hyper prior. Both are quantized element-wise to arrive at $\hat{z}_2$ and $\hat{h}_2$. Similar to most VAE-based image compression, our scheme adopts an additive noise for modeling quantizaiton. To this end, we have $\hat{h}_2 = m^{enc}_{\pi_3}(z_2) + e_h, e_h \sim \mathcal{U}(-0.5, 0.5)$ and $\hat{z}_2 = \lfloor z_2 - \mu^{dec}_{\pi_3}(\hat{h}_2) \rceil$ follow a distribution given by the convolution of $\mathcal{N}(0, (\sigma^{dec}_{\pi_3}(\hat{h}_2))^2)$ and $\mathcal{U}(-0.5, 0.5)$. To have the latent $\hat{z}_2$ and the hyper prior $\hat{h}_2$ capture most of the information of the input x, we require the latent x_2 to follow a zero-mean Gaussian with an extremely small variance σ^2. This justifies the use of zero for x_2 in our decoding process.

To sum up, the joint prior distribution $p(x_2, \hat{z}_2, \hat{h}_2)$ of our ANF model factorizes as:

$$p(x_2, \hat{z}_2, \hat{h}_2) = p(x_2)p(\hat{z}_2|\hat{h}_2)p(\hat{h}_2) \tag{7}$$

with

$$\begin{aligned} p(x_2) &= \mathcal{N}(0, \sigma^2) \\ p(\hat{z}_2|\hat{h}_2) &= \mathcal{N}(0, (\sigma^{dec}_{\pi_3}(\hat{h}_2))^2) * \mathcal{U}(-0.5, 0.5) \\ p(\hat{h}_2) &= \mathcal{P}_{\hat{h}_2|\psi} * \mathcal{U}(-0.5, 0.5) \end{aligned} \tag{8}$$

where $*$ denotes convolution and $P_{\hat{h}_2|\psi}$ is a learned distribution parameterized by ψ. We have tacitly assumed that $p(x_2), p(\hat{z}_2|\hat{h}_2), p(\hat{h}_2)$ are factorial over the elements of $x_2, \hat{z}_2, \hat{h}_2$, respectively.

3.3. Training Objective

Training of our ANF model can be achieved by minimizing the negative augmented log-likelihood, i.e. $\arg\min_\pi -\log p_\pi(x, e_z, e_h)$. This leads to the following loss function:

$$\begin{aligned} \mathcal{L}(x, e_z, z_h; \pi) = &-\log p(\hat{h}_2) - \log p(\hat{z}_2|\hat{h}_2) + \lambda_1 \|x_2 - 0\|^2 \\ &- log \left| det \frac{\partial G_\pi(x, e_z, e_h)}{\partial(x, e_z, e_h)} \right|, \end{aligned} \tag{9}$$

The Jacobian log-determinant in Eq. (9) prevents the collapse of the latent space. In our implementation, we replace it with a reconstruction loss $\lambda_2 \|x - \hat{x}\|^2$, which exerts a similar effect on the latent space but focuses more on the reconstruction quality.

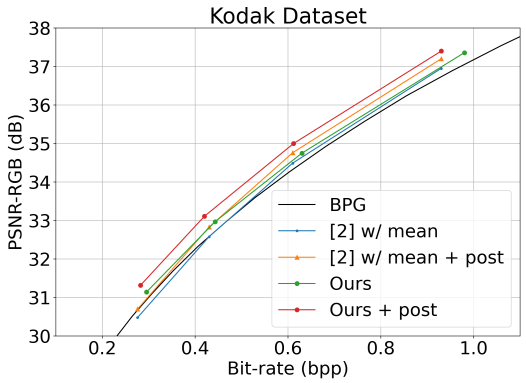

Figure 3. Rate-distortion comparison on Kodak dataset.

Table 1. Comparison of model size and BD-rate (BPG as anchor).

Codec	[2] w/ mean	[2] w/ mean + post	Ours	Ours + post
Parameters	13.4M	13.8M	11.4M	11.8M
BD-rate	-1.2%	-6.0%	-4.8%	-11.8%

4. Experiments

4.1. Settings and Implementation Details

Network Architectures: Our autoencoding and hyper-prior transforms share similar architectures to that in [2]. All the autoencoding transforms in our model have separate network weights. To maintain an overall model size comparable to that of [2], we reduce the number of channels in every convolutional layer to 96. Table 1 compares our model size with [2]. We also experiment with a post-processing quality enhancement network (denoted as + post), the architecture of which is shared between our method and the baseline [2] for a fair comparison. Note that the network weights may vary for end-to-end optimization.

Training: For training, we use *vimeo-90k* dataset. It contains 91,701 training videos, each having 7 frames. During a training epoch, we randomly choose one frame from each video and crop it to 256×256. We choose the Adam optimizer with a batch size of 96. The learning rate is fixed at $1e^{-4}$. The two hyper-parameters (Section 3.3) are chosen such that $\lambda_1 = 0.01 * \lambda_2$ and λ_2 is one of the values from $\{0.05, 0.02, 0.01, 0.005\}$. In particular, we first train our model for the highest rate point. It is then fine tuned with few epochs to obtain the models for lower rate points.

Evaluation: We evaluate our model on *Kodak* dataset, which includes 24 uncompressed 768 × 512 images. To evaluate the rate-distortion performance, we report rates in bits per pixel (bpp) and quality in PSNR-RGB. Our baselines include BPG and the enhanced version of [2], which incorporates the hyper-prior mean prediction.

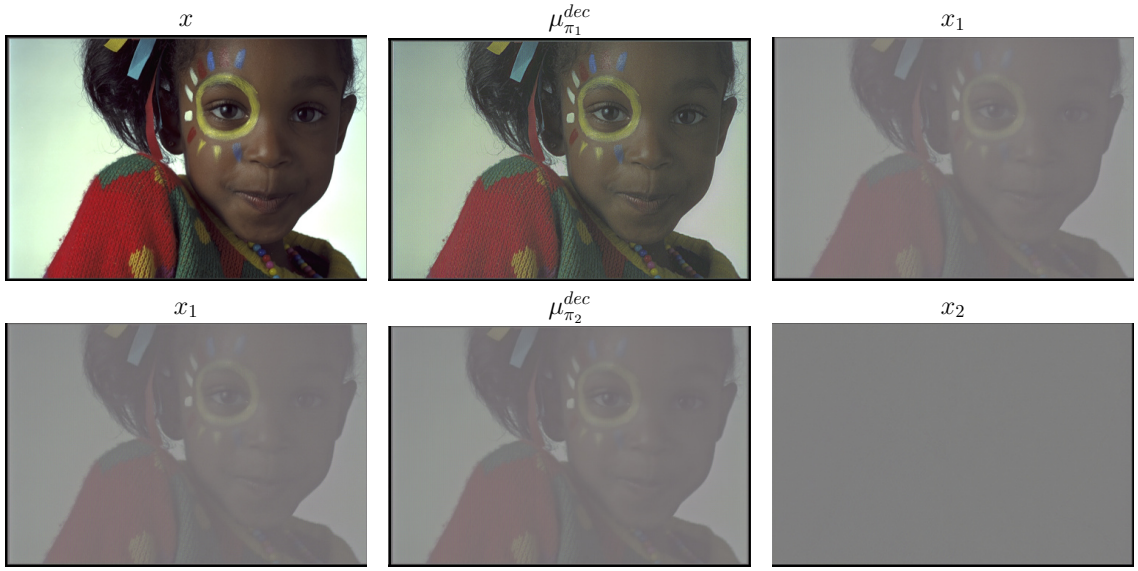

Figure 4. Visualization of our ANF transformation.

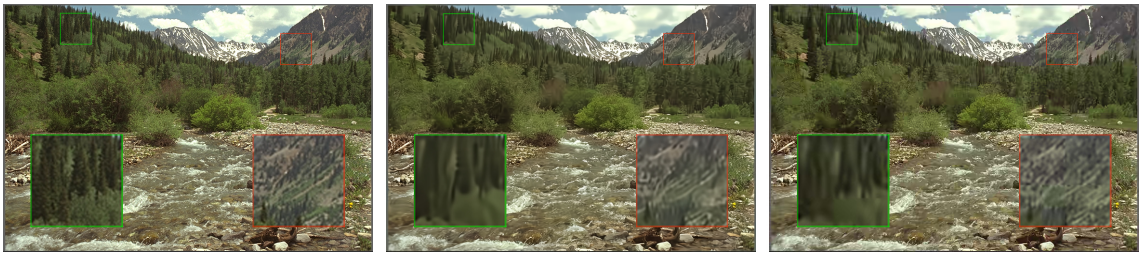

Figure 5. Subjective quality comparison between the original (left), ours + post (middle), and BPG (right). Ours + post: 27.29dB / 0.69bpp. BPG: 26.83dB / 0.68bpp.

4.2. Rate-distortion Performance

From Fig. 3, we see that without the post-processing network, our ANF model shows slightly better performance than the hyper-prior scheme with mean prediction, especially at low rates. It further improves the BD-rate saving by around 3.4% (Table 1). Interestingly, our scheme can benefit more from the incorporation of the post-processing network, showing much more significant improvement over the competing baseline in terms of BD-rate savings (11.8% vs. 6.0%). Moreover, the improvement can be seen across the entire bit rate range.

4.3. Qualitative Results

Fig. 4 visualizes how our ANF model (Fig. 2) transforms the input image from x into x_2 and sheds light on its inner workings. Recall that the latent x_2 should be a zero image, whose pixel values are mostly zero (Section 3.2). For better visualization, zero values are indicated by gray color. From Fig. 2, the first ANF layer encodes the input x into a latent representation z_1, which is decoded to give an estimate of the mean and variance of x (**top left**). It is seen that the mean image (**top middle**) contains enhanced high-frequency details of x. As a result, the centered and normalized output x_1 (**top right**) becomes a low-pass filtered signal of x. In a sense, the first ANF layer acts as a low-pass filter. Similarly, the second ANF layer operates as another low-pass filter to remove the remaining high-frequency details in x_1, yielding a nearly zero image as x_2 (**the bottom row** of Fig. 4). Fig. 5 presents a subjective quality comparison between decoded images.

5. Conclusion

This paper introduces an ANF-based image compression system. It is motivated by the facts that VAE, which forms the basis of end-to-end learned image compression, is a special case of ANF and that ANF can offer greater expressivity by stacking multiple VAEs. Experimental results confirm the superiority of our ANF-based scheme over a VAE-based baseline when they are operated under the same setting. In addition, we demonstrate that the autoencoding transforms of ANF act as progressive low-pass filters.

Acknowledgements

We are grateful to Chin-Wei Huang for discussions and helpful feedback on the manuscript. This work is supported by Qualcomm technologies, Inc. (NAT-439543) and National Center for High-Performance Computing, Taiwan.

References

[1] J Ballé, V Laparra, and E P Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.

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

[3] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021.

[4] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[5] Leonhard Helminger, Abdelaziz Djelouah, Markus Gross, and Christopher Schroers. Lossy image compression with normalizing flows. In *International Conference on Learning Representations Workshop*, 2021.

[6] Yueyu Hu, Wenhan Yang, and Jiaying Liu. Coarse-to-fine hyper-prior modeling for learned image compression. In *AAAI Conference on Artificial Intelligenc*, 2020.

[7] Chin-Wei Huang, Laurent Dinh, and Aaron C. Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *CoRR*, 2020.

[8] I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[9] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *International Conference on Learning Representations*, 2019.

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