

ANFIC: Image Compression Using Augmented Normalizing Flows

Yung-Han Ho, Chih-Chun Chan, Wen-Hsiao Peng, *Senior Member, IEEE*, Hsueh-Ming Hang, *Fellow, IEEE*, and Marek Domański, *Senior Member, IEEE*

This paper introduces an end-to-end learned image compression system, termed ANFIC, based on Augmented Normalizing Flows (ANF). ANF is a new type of flow model, which stacks multiple variational autoencoders (VAE) for greater model expressiveness. The VAE-based image compression has gone mainstream, showing promising compression performance. Our work presents the first attempt to leverage VAE-based compression in a flow-based framework. ANFIC advances further compression efficiency by stacking and extending hierarchically multiple VAE's. The invertibility of ANF, together with our training strategies, enables ANFIC to support a wide range of quality levels without changing the encoding and decoding networks. Extensive experimental results show that in terms of PSNR-RGB, ANFIC performs comparably to or better than the state-of-the-art learned image compression. Moreover, it performs close to VVC intra coding, from low-rate compression up to perceptually lossless compression. In particular, ANFIC achieves the state-of-the-art performance, when extended with conditional convolution for variable rate compression with a single model. The source code of ANFIC can be found at <https://github.com/dororojames/ANFIC>.

Index Terms—Learning-based image compression, flow-based image compression, augmented normalizing flows, perceptually lossless image compression, variable rate image compression

I. INTRODUCTION

Image compression has been a thriving research area for decades due to the storage and transmission requirements in various applications that underpin our modern digital life. Image compression also appears in the form of intra-frame coding for video compression [1]. The rapid advances in inter-frame prediction make efficient intra-frame coding become increasingly important because intra-coded frames often predominate over the bit rate of a compressed video. Therefore, it is much desirable to achieve even higher image compression efficiency.

The state-of-the-art image compression methods, e.g. BPG and VVC intra coding, usually involve block-based intra prediction, block-based transform coding of residuals, and context-adaptive binary arithmetic coding. Over the years, tremendous research effort has been invested to better every component in a way that seeks higher compression efficiency at the expense of an acceptable complexity increase. These hand-crafted codecs, although achieving a good balance between compression efficiency and complexity, lacks the opportunity to optimize all the components jointly in a seamless, end-to-end manner.

The rising of deep learning recently spurred a new wave of developments in image compression, with end-to-end learned systems attracting lots of attention. Among them, the variational autoencoder (VAE)-based methods [2], [3], [4], [5] have achieved compression performance very close to the latest VVC intra coding. Different from traditional hand-crafted

codecs, the VAE-based methods usually implement an image-level non-linear transform that converts an input image into a compact set of latent features, the dimensions of which are much smaller than the input image. Ever since the advent of the first VAE-based scheme [6], several improvements have been made on the expressiveness [4], [5], [7] of the autoencoder and the efficiency of entropy coding [2], [3], [4], [5], [8], [9], [10]. Up to now, the VAE-based methods have become the mainstream approach to end-to-end learned image compression.

However, one issue with most VAE-based schemes is that the autoencoder is generally lossy. There is no guarantee that its non-linear transform can reconstruct the input image losslessly even without quantizing the latent features of the image. This is unlike the traditional transforms, such as Discrete Cosine Transform and Wavelet Transform, which have the desirable property of perfect reconstruction and allow the codec to offer a wide range of quality levels by merely changing the quantization step size.

Recently, the flow-based models [11], [12] emerged as attractive alternatives. These models have the striking feature of realizing a bijective and invertible mapping between the input image and its latent features via the use of reversible networks composed of affine coupling layers [13], [14]. This invertibility is utilized to develop lossless image compression in [15], while the affine coupling layers are used in place of the lossy autoencoder in [11], [12] to achieve both lossy and lossless (or perceptually lossless) compression with a single unified model. The reversible networks, however, are quite distinct from the commonly used autoencoders, making these two types of compression systems not compatible with each other.

In this paper, we propose a novel end-to-end lossy image compression system, termed ANFIC, based on Augmented Normalizing Flows (ANF) [16]. ANF is a new type of flow models that work on augmented input space to offer greater transformation ability than the ordinary flow models. Our scheme ANFIC is motivated by the fact that ANF is a gener-

Manuscript received July 1, 2021; revised September 29, 2021; accepted October 15, 2021. This work was supported by National Center for High-Performance Computing, Taiwan.

Yung-Han Ho, Chih-Chun Chan, and Wen-Hsiao Peng are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan (e-mail: wpeng@cs.nctu.edu.tw).

Hsueh-Ming Hang is with the Department of Electronics Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan (e-mail: hmhang@nctu.edu.tw).

Marek Domański is with the Institute of Multimedia Telecommunications, Poznań University of Technology, Poznań, Poland (e-mail: marek.domanski@put.poznan.pl).

alization of VAE that stacks multiple VAE's as a flow model. In a sense, this allows ANFIC to extend any existing VAE-based compression system in a flow-based framework to enjoy the benefits of both approaches. ANFIC is novel and unique in that (1) it distinguishes from flow-based compression by operating in augmented input space, being able to leverage the representation power of any VAE-based image compression, and that (2) it is more general than the VAE-based compression by allowing VAE to be stacked and/or extended hierarchically.

Extensive experimental results on Kodak, Tecnick, and CLIC validation datasets show that ANFIC performs comparably to or better than the state-of-the-art end-to-end image compression in terms of PSNR-RGB. It performs close to VVC intra over a wide range of quality levels from low-rate compression up to perceptually lossless compression. In particular, ANFIC achieves the state-of-the-art performance among the competing methods, when extended with conditional convolutional layers [17] for variate rate compression with a single model.

Our main contributions are three-fold:

- We propose ANFIC, which uses augmented normalizing flows for image compression, as the first work that leverages VAE-based image compression in a flow-based framework.
- We offer extensive ablation studies to understand and visualize the inner workings of ANFIC.
- Extensive experimental results show that ANFIC is competitive with the state-of-the-art image compression, VAE-based and flow-based, over a wide range of quality levels and performs close to VVC intra coding.

This work improves on our previous publication [18] by (1) replacing the affine coupling layers with additive coupling layers to improve the training stability and avoid degrading the performance, (2) introducing the Gaussian mixture model along with the autoregressive module for better entropy coding, and (3) providing more comprehensive ablation studies of ANFIC.

The remainder of this paper is organized as follows: Section II reviews VAE-based image compression and the basics of ANF. Section III elaborates the design of ANFIC. Section IV compares ANFIC with the state-of-the-art methods in terms of objective compression performance and subjective image quality. Section V presents our ablation studies. Finally, we provide concluding remarks in Section VI.

II. RELATED WORK

In this paper, we propose an ANF-based image compression. It can be viewed as an extension of VAE-based image compression. Hence, this section focuses on the recent developments of VAE-based image compression and introduces the fundamentals of ANF to ease the understanding of our scheme.

A. VAE-based Image Compression

VAE-based image compression [2], [3], [4], [5], [6], [8], [9] is the most popular approach to end-to-end learned image compression. Its training framework includes three major components: the analysis transform, the prior distribution, and

the synthesis transform. These components are implemented by neural networks.

The analysis transform g_a encodes the raw image x through an encoding distribution $q_\phi^{g_a}(y|x)$ with the latent representation y uniformly quantized as $\hat{y}$. The $\hat{y}$ is then entropy encoded into a bitstream using a learned prior $p_\pi(\hat{y})$ implemented by a network π . Finally, the synthesis transform g_s reconstructs approximately the input x from $\hat{y}$ by a decoding distribution $p_\theta^{g_s}(x|\hat{y})$.

All the network parameters are trained end-to-end by minimizing

$$\mathcal{L}(\phi, \theta, \pi) = \underbrace{-E_{q_\phi^{g_a}(\hat{y}|x)}[\log p_\theta^{g_s}(x|\hat{y})]}_D - \underbrace{E_{q_\phi^{g_a}(\hat{y}|x)}[\log p_\pi(\hat{y})]}_R, \quad (1)$$

where the first term, denoted by D , aims to minimize the negative log-likelihood of x and the second term minimizes the rate R needed for signaling $\hat{y}$. In particular, it is shown that minimizing Eq. (1) amounts to maximizing the evidence lower bound (ELBO) of a latent variable model [19], which is specified by $p_\pi(\hat{y})$ and $p_\theta^{g_s}(x|\hat{y})$, with $q_\phi^{g_a}(\hat{y}|x)$ taking a uniform distribution that models the effect of uniform quantization. In a more general setting, a hyper-parameter λ is introduced to balance between D and R , yielding $\mathcal{L} = \lambda D + R$.

Balle et al. [6] are the first to introduce the aforementioned VAE framework together with a learned factorized prior to image compression. In entropy coding the image latents, they assume the prior distribution $p_\pi(\hat{y})$ over $\hat{y}$ to be factorial and learn the distribution by the network π . Their analysis and synthesis transforms are composed of convolutional neural networks and the general division normalization (GDN) layers, which originate from [20].

Even since the advent of the VAE-based compression framework, several efforts have been made to advance its coding efficiency. In particular, some [2], [3], [4], [5], [8], [9], [10] improve the prior estimation for better entropy coding while others [4], [5], [7] address the analysis and syntheses transforms (referred collectively to as the autoencoding transform). We summarize briefly these efforts as follows.

Enhanced Prior Estimation: The prior distribution $p_\pi(\hat{y})$ crucially determines the number of bits (i.e. the rate) needed to signal the quantized image latents $\hat{y}$. Recognizing the suboptimality of the factorized prior $p_\pi(\hat{y})$, where feature samples in every channel of $\hat{y}$ are independently and identically distributed, *Balle et al.* [8] propose the notion of hyperprior to model every feature sample separately by a Gaussian distribution. To this end, additional side information $\hat{z}$ is extracted from the image latent y and sent to the decoder, making the density estimation of $\hat{y}$ dependent on the input x . The $\hat{y}$ and $\hat{z}$ form the latent representation of the input x . The hyperprior thus bears the interpretation of factorizing the joint distribution $p(\hat{y}, \hat{z})$ as $p(\hat{y}|\hat{z})p(\hat{z})$, where $p(\hat{y}|\hat{z})$ and $p(\hat{z})$ are assumed to be Gaussian and factorial, respectively. *Hu et al.* [3], [10] extend the idea to include more than one layer of hyperprior, leading to a factorization of $p(\hat{y}, \hat{z}_1, \hat{z}_2, \dots, \hat{z}_n) = p(\hat{y}|\hat{z}_1)p(\hat{z}_1|\hat{z}_2), \dots, p(\hat{z}_n)$, where $\hat{z}_1, \hat{z}_2, \dots, \hat{z}_n$ form a multi-layer hyperprior. In addition to the use of hyperprior, *Minnen et al.* [2], *Lee et al.* [9], *Chen*

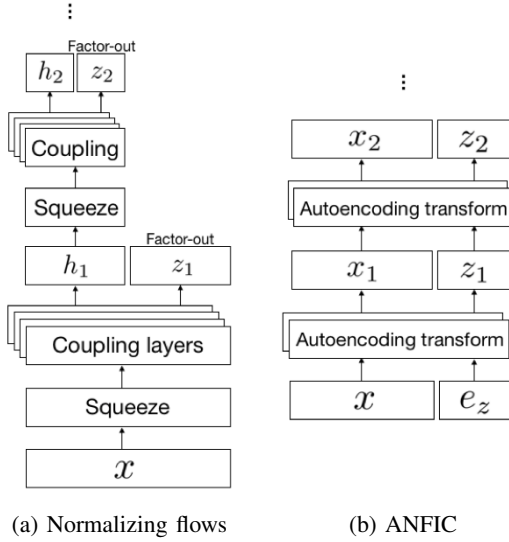


Fig. 1: Flow-based image compression with (a) normalizing flows [11] and (b) augmented normalizing flows (ours).

et al. [4], and *Cheng et al.* [5] incorporate an autoregressive prior by 2D [2], [5], [9] or 3D [4] masked convolution [21], in order to utilize causal contextual information for better density estimation. In particular, *Cheng et al.* [5] model $p(\hat{y}|\hat{z})$ with a Gaussian mixture distribution instead of a Gaussian.

Enhanced Autoencoding Transform: The capacity of the autoencoding transform determines its expressiveness. *Chen et al.* [4] add residual blocks to the autoencoder along with several non-local attention modules (NLAM). NLAM is shown to facilitate spatial bit allocation among coding areas of varied texture complexity. Unlike most of the VAE-based systems, which operate at image level, the block-based autoencoder in [7] divides the input image into non-overlapping macroblocks, each of which contains multiple sub-blocks coded sequentially using recurrent-based analysis and synthesis transforms. It has the striking feature of allowing high degree of computational parallelism at macroblock level. In general, most autoencoders are not guaranteed to reconstruct the input perfectly even when no quantization is involved.

B. Flow-based Image Compression

Recently, flow-based models [13], [14] emerge as an attractive alternative to VAE [19] or other autoencoders. They are characterized by the bijective mapping between the input and its latent representation, ensuring that the input can be perfectly reconstructed from its latent in the absence of quantization. *Ma et al.* [12] make an interesting attempt to introduce lifting-based coupling layers, which are a specialized implementation of additive coupling layers [13], [14] often used to construct a flow model, as the analysis and synthesis backbone. In particular, they split an input image, first row-wise and then column-wise, into latent subbands, the resulting decomposition being similar to 2D wavelet transform. *Helming et al.* [11] also use additive coupling layers but with the factor-out splitting to generate a multi-scale image representation as shown in Fig. 1a. Their work extends the notion of integer discrete flows for lossless compression [15] to

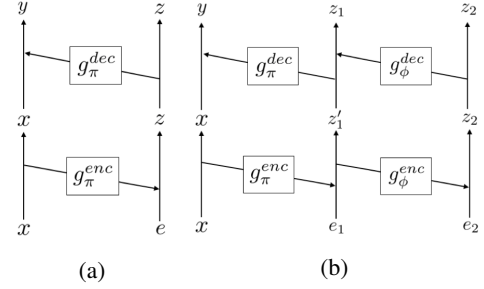


Fig. 2: The architectures of ANF: (a) one-step ANF, composed of the encoding g_{π}^{enc} and the decoding g_{π}^{dec} transforms, and (b) one-step hierarchical ANF.

lossy compression. In common, these works show the potential of flow-based models to offer a wide range of quality levels ranging from low-rate compression to nearly-lossless or even lossless compression.

Our work aims to leverage the developments of VAE-based schemes in a flow-based framework to enjoy the benefits of both (see Fig. 1b). For this purpose, we resort to augmented normalizing flows [16], the basics of which are presented next.

C. Augmented Normalizing Flows (ANF)

The ANF model [16] 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. 2a. 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 their latent representation (y, z) with one pair of encoding and decoding transforms:

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

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

where s_{π}^{enc} , m_{π}^{enc} , μ_{π}^{enc} , and σ_{π}^{enc} are element-wise affine transformation parameters. These learnable parameters are driven by the encoding and decoding neural networks, the weights of which are referred collectively to as π . Compared with ordinary flow models, ANF augments the input with an independent noise. It is shown in [16] that the augmented input space allows a smoother transformation to the required latent space.

Multi-step ANF and Hierarchical ANF: From Fig. 2a and according to Eqs. (2) and (3), the encoding g_{π}^{enc} or decoding g_{π}^{dec} transform implements an invertible affine coupling layer. Stacking pairs of these coupling layers leads to an invertible network, termed multi-step ANF, with much improved capacity than one-step ANF. Another way to increase the model capacity is to augment more noise inputs as hierarchical ANF (see Fig. 2b). Particularly, these two approaches can be combined in a flexible way for even higher model capacity.

Training ANF: Like the ordinary flow models, ANF can be trained by maximizing the *augmented joint likelihood*, i.e. $\arg \max_{\pi} p_{\pi}(x, e)$:

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

where $G_\pi = g_{\pi_N}^{dec} \circ g_{\pi_N}^{enc} \circ \dots \circ g_{\pi_1}^{dec} \circ g_{\pi_1}^{enc}$ is the alternate composition of the encoding and decoding transforms with $\pi = \{\pi_1, \dots, \pi_N\}$ and $p(G_\pi(x, e))$ represents the specified or learned prior distribution over the latents (y, z) . It is shown in [16] 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 .

VAE as One-step ANF: Notably, VAE can be viewed as one-step ANF by (1) letting $e \sim \mathcal{N}(0, I)$ be a Gaussian noise, (2) transforming e into z via re-parameterizing the VAE's encoding distribution $q_\pi^{enc}(z|x)$ of the form $\mathcal{N}(m_\pi^{enc}(x), (s_\pi^{enc}(x))^2)$, and (3) normalizing x as $y = (x - \mu_\pi^{dec}(z))/\sigma_\pi^{dec}(z)$ via the VAE's decoding distribution $p_\pi^{dec}(x|z) = \mathcal{N}(\mu_\pi^{dec}(z), (\sigma_\pi^{dec}(z))^2)$. The resulting y then follows $\mathcal{N}(0, I)$ and so does the aggregated distribution of z from various inputs x . Maximizing Eq. (4) for such an one-step ANF is shown in [16] to be identical to maximizing the ELBO of VAE [19].

III. PROPOSED METHOD

Inspired by the fact that most learned image compression is VAE-based and that VAE is equivalent to one-step ANF, we propose an ANF-based image compression framework, termed ANFIC. We first outline the ANFIC framework in Section III-A, with a focus on how to extend VAE-based image compression with hyperprior by multi-step and hierarchical ANF. This is followed by discussions on the entropy coding of the latent representation (Section III-B), the modeling of the prior distribution in ANFIC (Section III-A), and the training objective (Section III-C).

To the best of our knowledge, ANFIC is the first work that combines VAE and flow models in a unified framework. It distinguishes from flow-based compression in that it operates on augmented input space (see Fig. 1b), being able to leverage the representation power of any existing VAE-based image compression. Moreover, ANFIC is more general than the VAE-based scheme by allowing it to be stacked and/or extended hierarchically (see Fig. 2).

A. ANFIC Framework

Fig. 3a describes the framework of ANFIC. From bottom to top, it stacks two autoencoding transforms (i.e. two-step ANF), with the top one extended further to the right to form a hierarchical ANF [16] that implements the hyperprior. More autoencoding transforms can be added straightforwardly to create a multi-step ANF. In particular, the g_π^{enc} and g_π^{dec} in the autoencoding transform follow Eqs. (2) and (3), except that we make them purely additive by removing $s_\pi^{enc}(x)$ and $\sigma_\pi^{dec}(z)$ for better convergence as with some other flow-based schemes [11], [12].

The autoencoding transform of the hyperprior, which assumes each sample in the latent representation z_2 is a Gaussian, is defined as

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

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

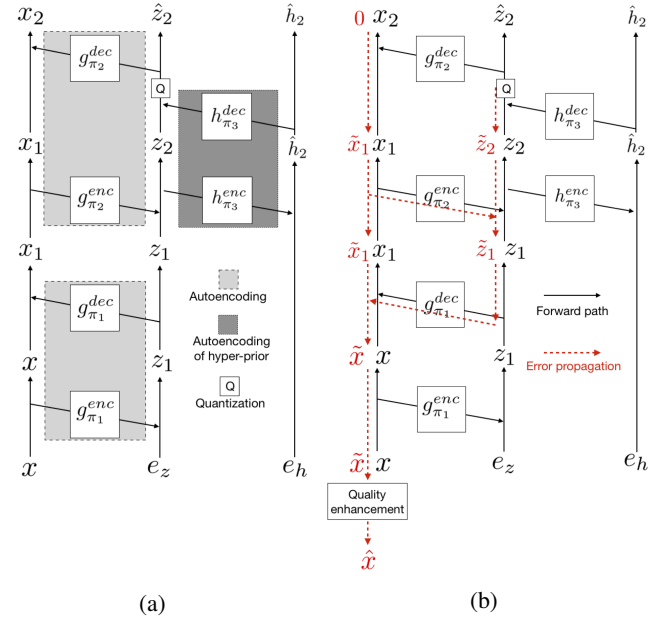


Fig. 3: (a) The overall architecture of our proposed ANF-based image compression (ANFIC). (b) Error propagation due to the quantization of the image latents z_2 . To alleviate propagation errors, we place a quality enhancement (QE) network at the end of the reverse path (the red dotted line).

where $\lfloor \cdot \rfloor$ (depicted as Q in Fig. 3a) denotes the nearest-integer rounding for quantizing the residual between z_2 and the predicted mean $\mu_{\pi_3}^{dec}(\hat{h}_2)$ of the Gaussian distribution from the hyperprior $\hat{h}_2$. This part implements the autoregressive hyperprior in [2], with z_2 denoting the image latents whose distributions are signaled as the side information $\hat{h}_2$.

The encoding of ANFIC proceeds by passing the augmented input (x, e_z, e_h) through the autoencoding and hyperprior transforms, i.e. $G_\pi = g_{\pi_2}^{dec} \circ h_{\pi_3}^{dec} \circ h_{\pi_3}^{enc} \circ g_{\pi_2}^{enc} \circ g_{\pi_1}^{dec} \circ g_{\pi_1}^{enc}$, to obtain the latent representation $(x_2, \hat{z}_2, \hat{h}_2)$. In particular, x represents the input image, $e_z = 0$ denotes the augmented input, and $e_h \sim \mathcal{U}(-0.5, 0.5)$, another augmented input, simulates the additive quantization noise of the hyperprior during training. To achieve lossy compression, we want $\hat{z}_2$ and $\hat{h}_2$ to capture most of the information about the input x and regularize x_2 during training to approximate noughts. As such, only $\hat{z}_2$ and $\hat{h}_2$ are entropy coded into bitstreams. Note that due to the volume-preserving property of ANF (or any flow model), x_2 has the same dimensionality as the input x while that of $\hat{z}_2$ and $\hat{h}_2$ is usually much smaller depending on the design choice. This flexibility allows us to incorporate any existing VAE-based compression scheme as one specific realization of the autoencoding transform in ANFIC. For example, the encoder of any VAE-based compression can be used to implement $m_\pi^{enc}(x)$ for the encoding transform in Eq. (2); likewise, its decoder can realize $\mu_\pi^{dec}(x)$ for the decoding transform in Eq. (3). Note that we have assumed the use of additive coupling layers.

To decode the input x , we apply the inverse mapping function G_π^{-1} to the quantized latents $(0, \hat{z}_2, \hat{h}_2)$, where x_2 is set to noughts. In ANFIC, there are two sources of distortion that cause the reconstruction to be lossy: the quantization

error of z_2 and the error of setting x_2 to noughts during the inverse operation. Essentially, ANFIC is an ANF model, which is bijective and invertible. The errors between the encoding latents (x_2, z_2) and their quantized version $(0, \hat{z}_2)$ will introduce distortion to the reconstructed image, as shown in Fig. 3b.

To mitigate the effect of quantization errors on the decoded image quality, we incorporate a quality enhancement (QE) network at the end of the reverse path, as illustrated in Fig. 3b. This enhancement network is an integral part of ANFIC, which is constrained by the fact that the analysis and the synthesis transforms must share the same autoencoding transforms (i.e. invertible coupling layers). This constraint makes it difficult to learn a synthesis transform that can effectively compensate for quantization errors while maintaining the invertibility. The same observation was made in [12]. In this paper, we adopt the same lightweight quality enhancement network as [12].

B. Prior Distribution

The prior distribution of ANFIC refers to the joint distribution $p(x_2, \hat{z}_2, \hat{h}_2)$ of the latents $(x_2, \hat{z}_2, \hat{h}_2)$, which like VAE-based schemes plays a crucial role in determining the rate needed to signal the image latents. Rather than manually specifying the prior distribution, we adopt a parametric approach to learn $p(x_2, \hat{z}_2, \hat{h}_2)$, for the sake of balancing between rate and distortion. As noted previously, our ANFIC has the latent $\hat{z}_2$ and the hyperprior $\hat{h}_2$ capture most of the information of the input x . We thus regularize the latent x_2 to follow a zero-mean Gaussian with a small variance σ^2 and to be independent of $\hat{z}_2, \hat{h}_2$. That is, $p(x_2, \hat{z}_2, \hat{h}_2)$ factorizes as:

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

with

$$p(x_2) = \mathcal{N}(0, \sigma^2), \quad (8)$$

and the remaining terms, $p(\hat{z}_2|\hat{h}_2)$ and $p(\hat{h}_2)$, learned from data by neural networks.

Similar to VAE-based schemes [8], we assume $p(\hat{h}_2)$ to be a non-parametric distribution and $p(\hat{z}_2|\hat{h}_2)$ to be a conditional Gaussian. Recall that $\hat{z}_2$ and $\hat{h}_2$ are the quantized version of the primary image latent z_2 and its hyperprior $m_{\pi_3}^{enc}(z_2)$ (see Eq. (5)), which is output by the encoding transform of $h_{\pi_3}^{enc}$ (see Fig. 3a). We follow the additive noise model for quantization during training. As a result, we have $\hat{h}_2 = m_{\pi_3}^{enc}(z_2) + e_h, e_h \sim \mathcal{U}(-0.5, 0.5)$ and $\hat{z}_2 = \lfloor z_2 - \mu_{\pi_3}^{dec}(\hat{h}_2) \rfloor$ follow a distribution given by the convolution of $\mathcal{N}(0, (\sigma_{\pi_3}^{dec}(\hat{h}_2))^2)$ and $\mathcal{U}(-0.5, 0.5)$. In symbols, $p(\hat{h}_2)$ and $p(\hat{z}_2|\hat{h}_2)$ have the forms of

$$\begin{aligned} p(\hat{z}_2|\hat{h}_2) &= \mathcal{N}(0, (\sigma_{\pi_3}^{dec}(\hat{h}_2))^2) * \mathcal{U}(-0.5, 0.5) \\ p(\hat{h}_2) &= \mathcal{P}_{\hat{h}_2|\psi} * \mathcal{U}(-0.5, 0.5) \end{aligned} \quad (9)$$

where $*$ denotes convolution and $\mathcal{P}_{\hat{h}_2|\psi}$ is a learned distribution parameterized by ψ . Note that unless otherwise specified, $p(x_2), p(\hat{z}_2|\hat{h}_2), p(\hat{h}_2)$ are all assumed to be factorial over the elements of $x_2, \hat{z}_2, \hat{h}_2$, respectively.

Algorithm 1 and 2 present the encoding and decoding procedures of ANFIC, respectively, where the $\tilde{x}_1, \tilde{z}_2$, and

Algorithm 1 The encoding procedure of ANFIC

- 1: Input: The image x and the augmented inputs e_z, e_h
 - 2: Output: The bitstream of $\hat{z}_2$ and $\hat{h}_2$
 - 3: $z_1 = m_{\pi_1}^{enc}(x) + e_z$
 - 4: $x_1 = x - \mu_{\pi_1}^{dec}(z_1)$
 - 5: $z_2 = z_1 + m_{\pi_2}^{enc}(x_1)$
 - 6: $\hat{h}_2 = m_{\pi_3}^{enc}(z_2) + e_h$ (replaced with the nearest-integer rounding of $m_{\pi_3}^{enc}(z_2)$ at inference time)
 - 7: Encode $\hat{h}_2$ using $p(\hat{h}_2)$ in Eq. (9)
 - 8: $\hat{z}_2 = \lfloor z_2 - \mu_{\pi_3}^{dec}(\hat{h}_2) \rfloor$
 - 9: Encode $\hat{z}_2$ using $p(\hat{z}_2|\hat{h}_2)$ in Eq. (9)
 - 10: $x_2 = x_1 - \mu_{\pi_2}^{dec}(\hat{z}_2)$
-

Algorithm 2 The decoding procedure of ANFIC

- 1: Input: The bitstream of $\hat{z}_2$ and $\hat{h}_2$
 - 2: Output: The reconstructed image $\hat{x}$
 - 3: Set x_2 to 0
 - 4: Decode $\hat{h}_2$ using $p(\hat{h}_2)$ in Eq. (9)
 - 5: Decode $\hat{z}_2$ using $p(\hat{z}_2|\hat{h}_2)$ in Eq. (9)
 - 6: $\tilde{x}_1 = \mu_{\pi_2}^{dec}(\hat{z}_2)$
 - 7: $\tilde{z}_2 = \hat{z}_2 + \mu_{\pi_3}^{dec}(\hat{h}_2)$
 - 8: $\tilde{z}_1 = \tilde{z}_2 - m_{\pi_2}^{enc}(\tilde{x}_1)$
 - 9: $\tilde{x} = \tilde{x}_1 + \mu_{\pi_1}^{dec}(\tilde{z}_1)$
 - 10: $\hat{x} = QE(\tilde{x})$
-

$\tilde{x}$ stand for the reconstructed version of x_1, x_2 , and x , respectively (See Fig. 3b).

Gaussian Mixtures Extension: ANFIC is flexible in accommodating more sophisticated modeling of $p(\hat{z}_2|\hat{h}_2)$, such as Gaussian mixture models. Unlike the single Gaussian model, the mixture model requires to estimate the mixing probabilities $w_{(k)}, k = 1, 2, \dots, K$ for K components as well as the corresponding mean $\mu_{(k)}$ and variance $\sigma_{(k)}$. All these parameters are functions of the hyperprior $\hat{h}_2$. In the present case, the decoding transform $h_{\pi_3}^{dec}$ (see Eq. (6)) is changed to $h_{\pi_3}^{dec}(z_2, \hat{h}_2) = (\lfloor z_2 \rfloor, \hat{h}_2) = (\hat{z}_2, \hat{h}_2)$ —namely, an identity transform followed by the quantization of z_2 . This change is necessary because with the mixture model, the subtraction of a single predicted mean from z_2 is not feasible. In addition, $p(\hat{z}_2|\hat{h}_2)$ follows a distribution given by

$$\begin{aligned} p(\hat{z}_2|\hat{h}_2) &= \left(\sum_{k=1}^K w_{(k)}(\hat{h}_2) \mathcal{N}(\mu_{\pi_3(k)}^{dec}(\hat{h}_2), (\sigma_{\pi_3(k)}^{dec}(\hat{h}_2))^2) \right) \\ &\quad * \mathcal{U}(-0.5, 0.5) \end{aligned} \quad (10)$$

C. Training Objective

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

$$\begin{aligned} \mathcal{L}(x, e_z, e_h; \pi, \psi) &= -\log p(\hat{h}_2) - \log p(\hat{z}_2|\hat{h}_2) + \lambda_1 \|x_2 - 0\|^2 \\ &\quad - \log \left| \det \frac{\partial G_{\pi}(x, e_z, e_h)}{\partial (x, e_z, e_h)} \right|, \end{aligned} \quad (11)$$

where the Jacobian log-determinant generally prevents the collapse of the latent space. In our implementation, we replace it with a reconstruction loss $\lambda_2 d(x, \hat{x})$, with the distortion

metric $d(\cdot, \cdot)$ being the mean-squared error (MSE) or multi-scale structure similarity index (MS-SSIM):

$$\begin{aligned} \mathcal{L}(x, e_z, e_h; \pi, \psi) \\ = \underbrace{-\log p(\hat{h}_2) - \log p(\hat{z}_2|\hat{h}_2)}_R + \lambda_1 \|x_2 - 0\|^2 + \underbrace{\lambda_2 d(x, \hat{x})}_D, \end{aligned} \quad (12)$$

where π, ψ refer to the parameters of all the networks, including the quality enhancement network. Unlike the traditional weighted sum of rate R and distortion D , our training objective has the additional requirement that x_2 should approximate noughts. This drives $\hat{z}_2, \hat{h}_2$ to encode most of the information about the input x , provided that the reconstructed image $\hat{x}$ approximate x closely. In passing, we note that the reconstruction loss also prevents the latent space from collapsing. Apparently, it would be difficult to recover the input x if different x 's are all mapped to the same point in the latent space.

IV. EXPERIMENTAL RESULTS

This section evaluates the performance of ANFIC both objectively and subjectively. We first present the network architectures, training details, evaluation methodologies, and the baseline methods in Section IV-A. Next, we compare the rate-distortion performance of ANFIC with several state-of-the-art methods on commonly used datasets in Section IV-B. Lastly, we evaluate the subjective quality of the reconstructed images in Section IV-C.

A. Settings and Implementation Details

Network Architectures: Our autoencoding transforms for feature extraction (the left branch in Fig. 4) and hyperprior (the right branch in Fig. 4) share similar architectures to the VAE-based scheme in [2]. In addition, we use the same lightweight de-quantization network in [12] as the quality enhancement network. All the autoencoding transforms in our model have separate network weights. To keep the overall model size comparable to that of [2], we reduce the number of channels in every convolutional layer to 128. We adopt the autoregressive and Gaussian mixture model (Section III-B) for entropy coding in all the experiments, with the number K of mixture components set empirically to 3, which is found to be most effective in [5].

Training: For training, we use *vimeo-90k* dataset from [22]. It contains 91,701 training videos, each having 7 frames. In a training iteration, we randomly choose one frame from each video and crop it to 256×256 . We adopt the Adam [23] optimizer with a batch size of 32. The learning rate is fixed at $1e^{-4}$ during the first 3M iterations, and then we decay to $1e^{-5}$ for fine-tuning. The two hyper-parameters (see Eq. (12)) are chosen to have $\lambda_1 = 0.01 * \lambda_2$, where λ_2 is one of the values from $\{0.1, 0.05, 0.02, 0.01, 0.005, 0.002\}$ for MSE optimization and from $\{200, 100, 40, 20, 10, 4\}$ for optimizing MS-SSIM. 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 commonly used datasets, *Kodak* [24] and *Tecnick* [25], which include 24

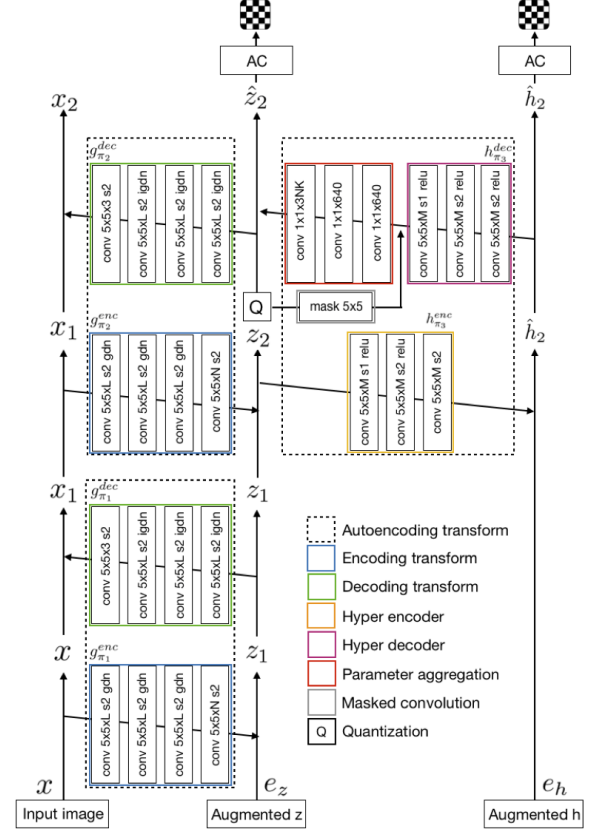


Fig. 4: The network architecture of our proposed ANFIC ($L = 128, N = 320, M = 192, K = 3$). We adopt the autoregressive and Gaussian mixture model for entropy coding. AC and mask denote arithmetic coding and masked convolution, respectively.

uncompressed images of size 768×512 and 40 images of size 1200×1200 , respectively. Additionally, we test our model on the CLIC validation datasets [26]. It contains two subdivided datasets: professional and mobile. The former has 41 higher resolution images and the latter 61 images. To evaluate the rate-distortion performance, we report rates in bits per pixel (bpp) and quality in PSNR-RGB and MS-SSIM. Moreover, we use BPG as an anchor in reporting the BD-rates. Note that rate inflation as compared to BPG is reflected by positive BD-rates while rate saving is shown as negative BD-rates.

Baselines: For comparison, the baseline methods include VTM-444, BPG-444, ICLR'18 [8], NIPS'18 [2], ICLR'19 [9], TPAMI'20 [12], CVPR'20 [5], TPAMI'21 [10], and TIP'21 [4]. It is worth noting that TPAMI'20 [12] is a flow-based model, while the other learned codecs are VAE-based.

B. Rate-Distortion Performance

Fig. 5 compares the rate-distortion performance of the competing methods on Kodak, Tecnick, and CLIC (professional and mobile combined) datasets, with the BD-rate numbers summarized in Table I. Following some prior works, the BD-rate figures for CLIC professional dataset are reported separately in Table I.

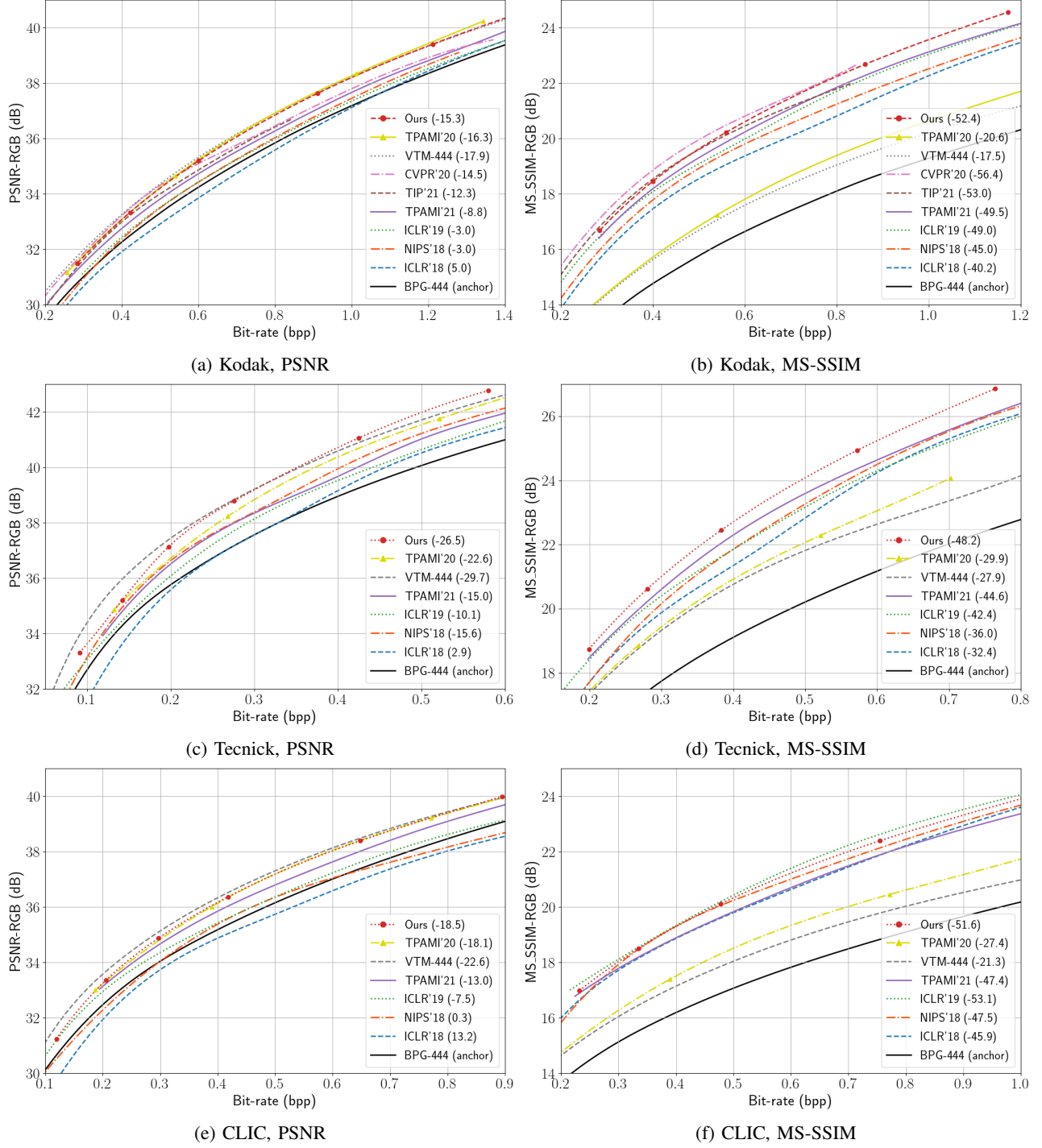


Fig. 5: Rate-distortion performance evaluation on Kodak, Tecnick, and CLIC datasets for both PSNR and MS-SSIM. The numbers in the parentheses are the BD-rates with BPG-444 as anchor.

In terms of PSNR-RGB, one can see that our method shows comparable performance to the state-of-the-art learned codecs, CVPR'20 [5] and TPAMI'20 [12], on Kodak and CLIC datasets. Remarkably, it achieves the best performance among all the learned codecs on Tecnick and CLIC datasets. It however falls short of the VTM model slightly on Kodak, Tecnick and CLIC datasets. In particular, ANFIC displays a tendency to perform worse at low rates. This may be attributed to the fact that additive coupling layers are susceptible to the

accumulation and propagation of quantization errors (Fig. 3b). It is important to note in Table I that ANFIC is inferior to VTM in BD-rate saving by a significant margin (7%) on CLIC Professional dataset. Careful examination of the dataset reveals that some images are extremely challenging and not typical of the images found in our training data. All the competing methods are faced with the same issue. It is expected that increasing the diversity of training data will help. Nevertheless, the superiority of ANFIC over BPG is apparent on all the

TABLE I: Comparison of the BD-rate savings and model sizes of the competing methods (optimized by MSE). The BD-rate savings are reported with BPG-444 serving as the anchor. The best performer is marked with “†”, and the second best with “*”.

Methods	BD-rate (%)				Model Size
	Kodak	Tecnick	CLIC	CLIC Pro	
ICLR’18 [8]	5.0	2.9	13.2	-	12M
NIPS’18 [2]	-3.0	-15.6	0.3	-	20M
ICLR’19 [9]	-3.0	-10.1	-7.5	-13.8	73M
TPAMI’20 [12]	-16.3*	-22.6	-18.1	-20.3	18M
CVPR’20 [5]	-14.5	-	-	-25.3*	27M
TPAMI’21 [10]	-8.8	-15.0	-13.0	-	73M
ANFIC (Ours)	-15.3	-26.5*	-18.5*	-24.5	23M
VTM-444	-17.9†	-29.7†	-22.6†	-31.3†	-

datasets.

In terms of MS-SSIM, our method performs among top two. It is slightly worse than the top performer, CVPR’20 [5], on Kodak dataset, especially at low rates (See Fig. 5b), but is comparable to ICLR’19 [9], which achieves the best MS-SSIM performance on the CLIC dataset. It is worth mentioning that TPAMI’20 [12], a strong baseline when evaluated with PSNR-RGB, exhibits poor MS-SSIM results because the released model is optimized for MSE only. Also, as noted previously in other studies, the learned codecs outperform VTM and BPG considerably when trained and tested by MS-SSIM.

The model size comparison in Table I suggests that the rate-distortion benefits of ANFIC do not come at the expense of unreasonably huge models. Its model size is between that of TPAMI’20 [12] and CVPR’20 [5], both show competitive rate-distortion performance.

C. Subjective Quality Comparison

Figs. 6 and 7 show the subjective quality comparison between ANFIC (ours), VVC, BPG, and TPAMI’20 [12] on images *kodim01* and *kodim16* from Kodak dataset. It is seen that our MSE model achieves comparable subjective quality to VVC and TPAMI’20 [12]. As expected, ANFIC optimized for MSE tends to smooth the highly-textured areas, while VVC and HEVC generates clear blocking artifacts in Fig. 7. In particular, TPAMI’20 [12] suffers from geometric distortion especially in the “door” area in Fig. 6 and produces some artificial noisy dots on the “water surface” in Fig. 7. In contrast, our MS-SSIM model shows much better subjective quality, preserving most high-frequency details.

V. ABLATION STUDIES

In this section, we conduct ablation studies to understand ANFIC’s properties. Firstly, we show how the ANF framework improves the VAE-based scheme by stacking its autoencoding transform (Section V-A). Secondly, we investigate the effect of the quality enhancement network on ANFIC and its VAE-based counterpart (Section V-B). Thirdly, we discuss the effect of imposing different regularization strategies on x_2 (Section V-C). Fourthly, we analyze the inner workings of ANFIC by visualizing the output of each autoencoding transform in both spatial and frequency domains (Section V-D). Fifthly, we study the compression performance of ANFIC across low and high rates (Section V-E). Finally, we extend ANFIC to support

variable rate compression and compare its performance with the other baselines (Section V-F). Unless otherwise specified, Kodak dataset is used for ablation experiments.

A. Number of Autoencoding Transforms

To see the rate-distortion benefits of stacking autoencoding transforms, we compare between the VAE-based scheme [2] and ANFIC with a varied number of autoencoding transforms. It is important to note that the VAE-based scheme can be interpreted as one-step ANFIC (see Section III-A). For a fair comparison, the VAE-based scheme (which is termed “NIPS’18+GMM” and is modified from [2] by additionally including Gaussian mixture-based entropy coding and the quality enhancement network [12]) and ANFIC share the same autoencoding architecture, entropy coding scheme, and quality enhancement network. To keep the model size comparable, the channel number of every autoencoding transform in ANFIC is set to 128 (See Fig. 4), while that of the VAE-based counterpart is 192. This ensures that ANFIC with two autoencoding transforms (the main setting used throughout this paper) has a similar model size to the VAE-based one. Nevertheless, when the number of autoencoding transforms increases beyond two, the model size of ANFIC increases linearly.

From Fig. 8, it is seen that increasing the number of autoencoding transforms from one layer (VAE-based) to two layers (Ours 2-step) improves the rate-distortion performance significantly. However, the gain diminishes sharply when the number goes beyond two. We thus choose two autoencoding transforms as our default setting.

A side experiment shows that increasing the channel number (i.e. the L value in Fig. 4) of the autoencoding transform from 128 to 192 improves the BD-rate saving only marginally by 1.1%. The channel number is defaulted to 128 for lower complexity and fair comparison.

B. Effect of Quality Enhancement Networks

Fig. 9 shows the effect of the quality enhancement network (as a post-processing network) on the rate-distortion performance of ANFIC and the VAE-based scheme [2]. In addition to the default quality enhancement network from [12], we experiment with another popular one, known as MCNet [1], which is often used in the end-to-end learned video codecs to enhance the quality of the motion-compensated frame [1]. The two quality enhancement networks have similar model sizes. The major difference between them is that the default one [12] does not have striding and pooling operations, whereas MCNet [1] has a U-net structure, where the resolution of the feature maps shrinks first and stretches later.

We observe that ANFIC benefits more from the use of the default quality enhancement network [12], which boosts the BD-rate saving of ANFIC by 6.6% as compared to 3.5% with the NIPS’18+GMM (VAE-based with default quality enhancement network [12]) scheme [2]. This suggests that ANFIC literally separates the image transformation and the (quantization) error compensation into two orthogonal parts. The former is addressed by invertible autoencoding transforms while the latter relies on the quality enhancement network. The

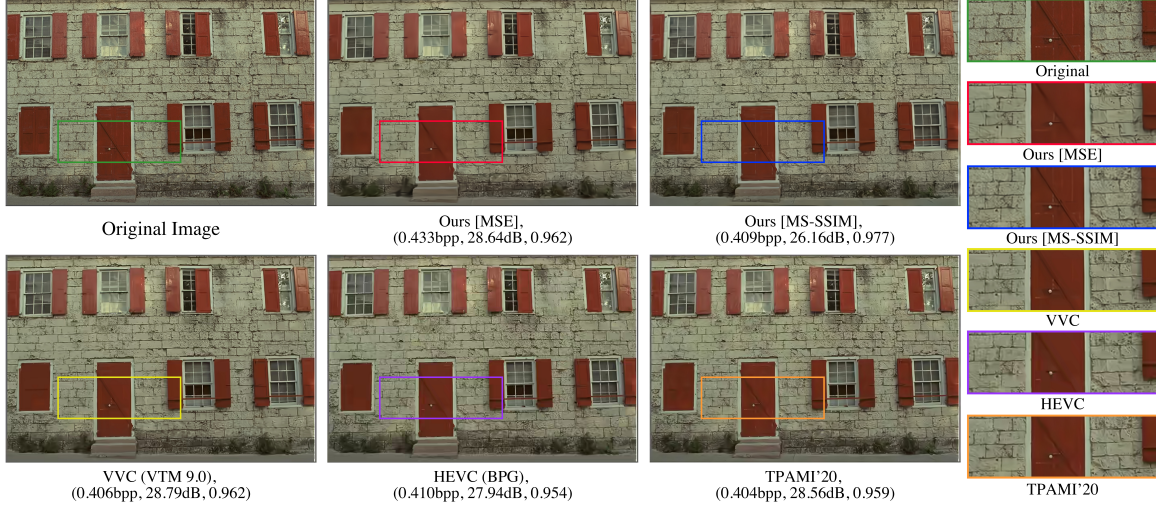


Fig. 6: Subjective quality comparison of image *kodim01* from Kodak dataset.

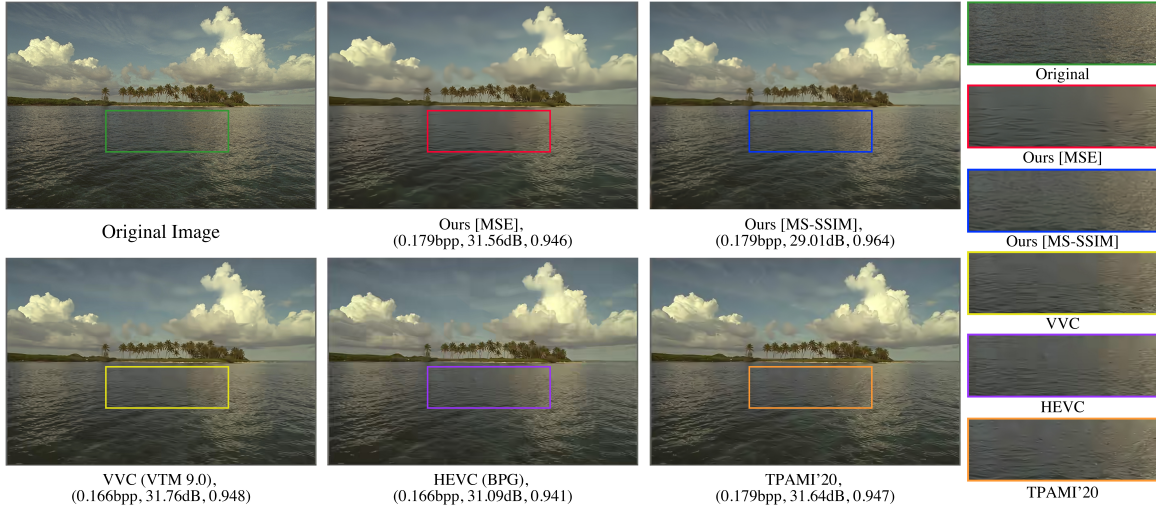


Fig. 7: Subjective quality comparison of image *kodim16* from Kodak dataset.

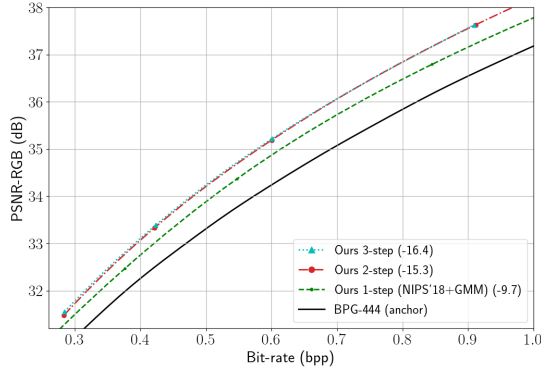


Fig. 8: Rate-distortion curves for different number of autoencoding transforms.

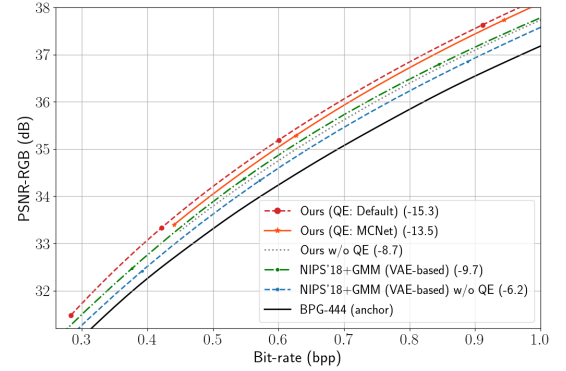


Fig. 9: Rate-distortion performance with and without the quality enhancement network.

fact that the feature extraction and the image reconstruction in ANFIC have to go through the same invertible coupling layers make it difficult to learn autoencoding transforms that can handle well both image representation and error compensation. This however is not the case with the NIPS'18+GMM (VAE-

based) scheme, where the analysis and the synthesis transforms do not share the same network. Usually, the synthesis transform can learn to compensate partially for quantization errors. As such, the gain from the quality enhancement network becomes limited when the synthesis network is already capable

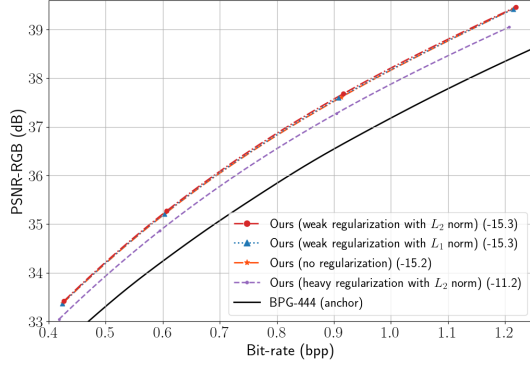


Fig. 10: ANFIC with different regularization strategies imposed on x_2 .

enough.

From Fig. 9, it is also seen that the default quality enhancement network [12] shows better rate-distortion performance than MCNet [1], especially at lower rates. This may be attributed to the fact that the striding and pooling of MCNet [1] could cause the loss of some spatial information. In any case, ANFIC with either quality enhancement network outperforms NIPS'18+GMM.

C. Effect of x_2 Regularization

Fig. 10 compares the rate-distortion curves for different regularization strategies imposed on x_2 , including (1) weak regularization ($\lambda_1 = 0.01 * \lambda_2$) with the L_2 norm (the proposed method), (2) weak regularization ($\lambda_1 = 0.01 * \lambda_2$) with the L_1 norm, (3) heavy regularization ($\lambda_1 = 1 * \lambda_2$) with the L_2 norm, and (4) no regularization ($\lambda_1 = 0$). It can be observed that weak regularization with either the L_2 norm or L_1 norm achieves the best rate-distortion performance, presenting 15.3% BD-rate reductions. Heavy regularization with the L_2 norm, however, degrades the rate-distortion performance, because the regularization loss is weighted equally as the reconstruction loss. No regularization, interestingly, shows marginally worse rate-distortion performance (15.2% BD-rate reduction) than the weak regularization with the L_2 norm (the proposed method).

The fact that no regularization shows marginal impact on the final rate-distortion performance has partially to do with our setting x_2 to 0 for reconstruction during training. Recall that the mapping between the input (x, e_z, e_h) and the latent representation (x_2, z_2, h_2) is invertible (See Fig. 3a). In the absence of quantization, using $(0, z_2, h_2)$ in place of (x_2, z_2, h_2) for decoding while ensuring the invertibility by minimizing the reconstruction loss $d(x, \hat{x})$ would compel x_2 to approximate noughts during encoding without any additional regularization. The same trend carries roughly over to the case when x_2, z_2, h_2 are quantized. We however notice that imposing weak regularization on x_2 during encoding will make the training more stable.

D. Visualization of Autoencoding Transforms

Fig. 11 visualizes how our ANFIC model (see Fig. 4) transforms the input image x step-by-step into a residual image x_2

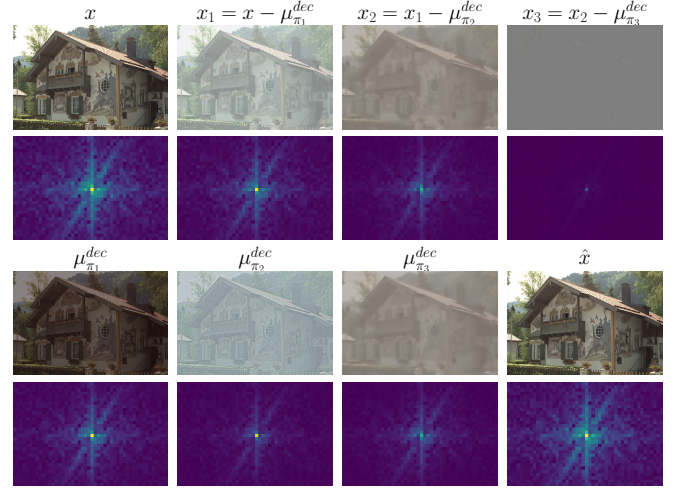


Fig. 11: Visualization of the autoencoding transform outputs $\{x_i\}_{i=1}^3$ and the decoder outputs $\{\mu_{\pi_i}^{dec}\}_{i=1}^3$ in the autoencoding transforms in three-step ANFIC, where the average image intensity has been shifted to 128 for better viewing. The signal spectra in frequency domain are plotted as heatmaps.

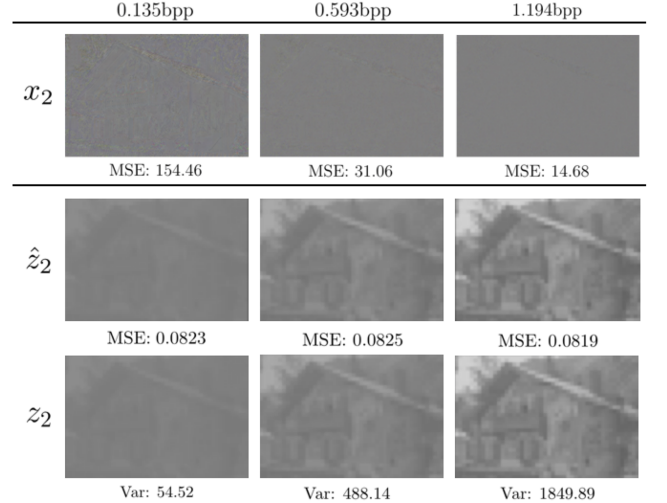


Fig. 12: Visualization of x_2 , $\hat{z}_2$, and z_2 , where only few channels with the largest variances are shown for $\hat{z}_2$ and z_2 . The Mean Square Error (MSE) of x_2 is measured against a zero image while that of $\hat{z}_2$ is against z_2 .

and what information is captured by the corresponding latent code $z_i, i = 1, 2$ in each step. Additionally, the corresponding signal spectra in frequency domain are presented to understand the system response of every autoencoding transform. For better visualizing the evolution of signals, we extend the architecture in Fig. 4 to three-step ANFIC, with the final outputs being x_3 and z_3 (instead of x_2 and z_2 as depicted in Fig. 4). Also presented in this figure are the decoder outputs $\{\mu_{\pi_i}^{dec}\}_{i=1}^3$ of the autoencoding transforms (see Eq. (3) and Fig. 4), which reveal the information captured by the latent code $\{z_i\}_{i=1}^3$. As an example, the first autoencoding transform converts the image x into the latent code z_1 , which is then decoded as $\mu_{\pi_1}^{dec}$ to be subtracted from x . Hence, $\mu_{\pi_1}^{dec}$ stands for an estimate of x that is derived from the latent z_1 .

From left to right in the top two rows, one can see that the high-frequency details of the input image x are filtered out in successive autoencoding transforms, arriving at a residual image x_3 with little high-frequency information (see the sub-figure in the top-right corner). As such, the autoencoding transforms in ANFIC act as low-pass filters, where their cut-off frequency decreases with the increasing transform step in the feature extraction process. Because x_3 will be discarded during the reconstruction process, the remaining high-frequency details in x_3 will be lost completely. Thus, ANFIC is lossy.

The decoder outputs of the autoencoding transforms further shed light on how the latent code is transformed from e_z into a form suitable for compression (i.e. $e_z \rightarrow z_1 \rightarrow z_2 \rightarrow z_3$). From left to right in the bottom two rows, we see that $\mu_{\pi_1}^{dec}$ (decoded from z_1) presents a rough estimate of the input x . Its spectrum looks similar to that of x , but is not exactly the same. We conjecture that $\mu_{\pi_1}^{dec}$ focuses more on the approximation of the high-frequency part of the input x . The corroborating fact is that when it is subtracted from x , the resulting output $x_1 = x - \mu_{\pi_1}^{dec}$ has relatively less high-frequency information. This becomes even more obvious in the following autoencoding transform, where $\mu_{\pi_2}^{dec}$ (decoded from z_2) addresses primarily the remaining mid-frequency part in x_1 ; as a result, the output $x_2 = x_1 - \mu_{\pi_2}^{dec}$ of the second transform becomes an even lower-frequency signal. In the end, the latent code z_3 , which will be compressed into the bitstream, only needs to represent a low-pass filtered version of the original input, which is relatively easy to compress. The reconstruction process updates a zero image in x_3 by those decoder outputs in reverse order (i.e. $\mu_{\pi_3}^{dec} \rightarrow \mu_{\pi_2}^{dec} \rightarrow \mu_{\pi_1}^{dec} \rightarrow x$), to recover the low-frequency, mid-frequency, and high-frequency details of the input x step-by-step.

Fig. 12 further visualizes x_2 , z_2 , and $\hat{z}_2$ at different bit rates ranging from 0.135bpp to 1.194bpp. It is seen that more residuals appear in x_2 at low rates than at high rates, suggesting that setting x_2 to a zero image at low rates would introduce more distortion than at high rates. As for z_2 and $\hat{z}_2$, because a fixed, uniform quantization step size, i.e. 1, is used for all the rate points, the MSE between z_2 and $\hat{z}_2$ does not change significantly. However, the network learns to adjust the variance of z_2 in order to control the signal-to-noise ratio in the latent space. We see that the higher the bit rate is, the more information is captured by z_2 ; namely, z_2 tends to have larger variances at high rates. All in all, the information captured by x_2 decreases with the increasing bit rate, whereas that by z_2 increases accordingly.

E. Compression Performance across Low and High Rates

This study investigates the compression performance of ANFIC over a wide range of bit rates. It is reported in [11], [12] that most VAE-based compression schemes suffer from the autoencoder limitation; that is, the reconstruction by the autoencoder is generally lossy, even without quantization. As a result, it is difficult for a VAE-based model to support efficient compression over a wide range of bit rates without changing the network architecture, for example, by adjusting the number of channels. ANFIC, although being a flow-based model, is

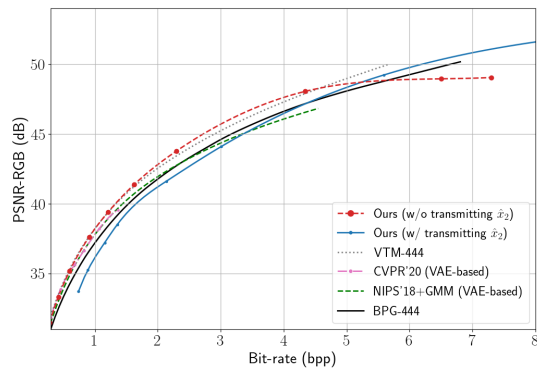


Fig. 13: Rate-distortion comparison between our ANFIC and VAE-based schemes across low and high rates.

lossy due to discarding the high-frequency information in the residual image x_2 (see Fig. 4) for reconstruction.

Fig. 13 compares ANFIC with two state-of-the-art VAE-based schemes over a wide range of bit rates. In particular, ANFIC has the same number of channels (i.e. 320 channels) in latent space as NIPS'18 [2], whereas CVPR'20 [5] has only 192 channels yet with a larger model size. We see that our ANFIC (w/o transmitting $\hat{x}_2$) matches the performance of VTM closely from extremely low-rate compression up to perceptually lossless compression, while the two VAE-based schemes tend to fall short of VTM and even BPG at high rates. The reason why ANFIC is able to work well across low and high rates are two-fold: (1) the ANF-based backbone is fully invertible, and (2) our training strategies, which require x_2 to approximate noughts in the feature extraction process and use noughts exactly for x_2 during reconstruction, force the image latent $\hat{z}_2$ and its hyperprior $\hat{h}_2$ to capture as much information about the input x as possible (see Fig. 4).

To further study the invertibility of ANFIC by additionally encoding x_2 , we model the distribution of the quantized x_2 , denoted by $\hat{x}_2 = \lfloor x_1 - \mu_{\pi_2}^{dec}(\hat{z}_2) \rfloor$, by the convolution of a Gaussian and a uniform distribution. For better coding efficiency, the distribution is conditional on $\hat{z}_2$:

$$p(\hat{x}_2|\hat{z}_2) = \mathcal{N}(0, \sigma_{\pi_2}^{dec}(\hat{z}_2)^2) * \mathcal{U}(-0.5, 0.5) \quad (13)$$

A closer look at the rate-distortion performance w/o and w/ transmitting $\hat{x}_2$ in Fig. 13 reveals that (1) at lower rates, transmitting $\hat{x}_2$ shows worse rate-distortion performance than not transmitting $\hat{x}_2$, and that (2) at higher rates, transmitting $\hat{x}_2$ helps mitigate the quality gap between lossy and (mathematically) lossless compression. In particular, not transmitting $\hat{x}_2$ puts a limit on the highest achievable reconstruction quality (i.e. the rate-distortion curve plateaus after 6bpp). The second observation is in line with the invertibility property of ANF. Focusing on lossy image compression, we opt for not transmitting $\hat{x}_2$ in this paper. However, how to adapt ANFIC to support mathematically lossless coding is an interesting open issue that is among our future work.

F. Variable Rate Compression

Recognizing that ANFIC can work well over a wide range of bit rates, we take one step further to adapt ANFIC to variable

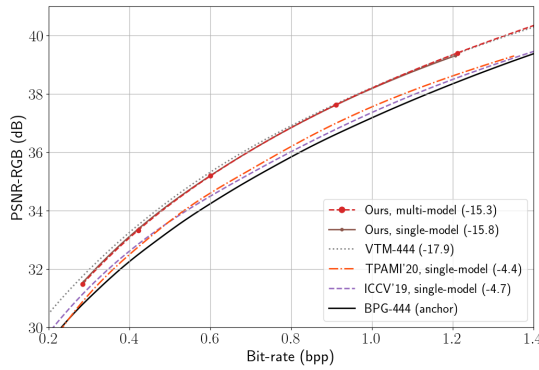


Fig. 14: Rate-distortion comparison between variable rate models. Multi-model: separate models for distinct rate points; Single-model: a single model for multiple rate points.

rate compression with a single model. To this end, we implement the notion of the conditional convolution in [17], replacing every convolutional layer with one that is conditional on the λ_2 (see Eq. (12)). The conditional convolution layer applies an affine transformation to every feature map, with the affine parameters derived from a network conditional on the rate parameter λ_2 . For the experiment, we train a single ANFIC model using 5 distinct λ_2 values $\{0.1, 0.05, 0.02, 0.01, 0.005\}$. The training objective is an extension of Eq. (12) by substituting different λ_2 's into Eq. (12) and averaging over these variants.

Fig. 14 shows the rate-distortion comparison of the state-of-the-art variable rate models, including VVC, BPG, ICCV'19 [17], TPAF'20 [12], and ANFIC (ours). Compared with our multi-model setting, our single-model setting performs comparably well, with slightly increased rate saving due to training variance. It also shows comparable performance to VTM across the 5 rate points, but outperforms significantly the other learning-based methods in single-model mode.

VI. CONCLUSION

In this paper, we propose an ANF-based image compression system (ANFIC). It is motivated by the fact that VAE, which forms the basis of most end-to-end learned image compression, is a special case of ANF and can be extended by ANF to offer greater expressiveness. ANFIC is the first work that introduces VAE-based compression in a flow-based framework, enjoying the benefits of both approaches. Experimental results show that ANFIC performs comparably to or better than the state-of-the-art learned image compression and is able to offer a wide range of quality levels without changing the network architecture. Furthermore, its variable rate version shows little performance degradation. Flow-based models are relatively new to learned image compression. We believe there remains widely open space for further research; for example, how to achieve mathematically lossless coding with ANFIC is yet to be addressed.

REFERENCES

[1] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An End-To-End Deep Video Compression Framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[2] D. Minnen, J. Ballé, and G. Toderici, "Joint Autoregressive and Hierarchical Priors for Learned Image Compression," in *Neural Information Processing Systems*, 2018.

[3] Y. Hu, W. Yang, and J. Liu, "Coarse-to-Fine Hyper-Prior Modeling for Learned Image Compression," in *AAAI Conference on Artificial Intelligence*, 2020.

[4] 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*, vol. 30, pp. 3179–3191, 2021.

[5] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

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

[7] C. Lin, J. Yao, F. Chen, and L. Wang, "A Spatial RNN Codec for End-to-End Image Compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[8] J. Ballé, D. Minnen, S. Singh, S. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[9] J. Lee, S. Cho, and S. Beack, "Context-adaptive Entropy Model for End-to-end Optimized Image Compression," in *International Conference on Learning Representations*, 2019.

[10] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning End-to-End Lossy Image Compression: A Benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[11] L. Helming, A. Djelouah, M. Gross, and C. Schroers, "Lossy Image Compression with Normalizing Flows," in *International Conference on Learning Representations Workshop*, 2021.

[12] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, "End-to-End Optimized Versatile Image Compression With Wavelet-Like Transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[13] 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.

[14] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *International Conference on Learning Representations*, OpenReview.net, 2017.

[15] E. Hoogeboom, R. Peters, J. van den Berg, and M. Welling, "Integer Discrete Flows and Lossless Compression," in *Advances in Neural Information Processing Systems*, 2019.

[16] C. Huang, L. Dinh, and A. C. Courville, "Augmented Normalizing Flows: Bridging the Gap Between Generative Flows and Latent Variable Models," *CoRR*, 2020.

[17] Y. Choi, M. El-Khamy, and J. Lee, "Variable Rate Deep Image Compression With a Conditional Autoencoder," in *2019 IEEE/CVF International Conference on Computer Vision*, 2019.

[18] Y.-H. Ho, C.-C. Chan, W.-H. Peng, and H.-M. Hang, "End-to-end learned image compression with augmented normalizing flows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

[19] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2014.

[20] J. Ballé, V. Laparra, and E. Simoncelli, "End-to-end Optimized Image Compression," in *International Conference on Learning Representations*, 2017.

[21] A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," in *Proceedings of International Conference on Machine Learning*, 2016.

[22] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video Enhancement with Task-Oriented Flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015.

[24] E. Kodak, "Kodak lossless true color image suite (photocd pcd0992). <http://r0k.us/graphics/kodak/>,"

[25] A. Nicola and G. Andrea, "TESTIMAGES: a Large-scale Archive for Testing Visual Devices and Basic Image Processing Algorithms," in *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*, 2014.

[26] G. Toderici, W. Shi, R. Timofte, L. Theis, J. Balle, E. Agustsson, N. Johnston, and F. Mentzer, "Workshop and Challenge on Learned Image Compression (CLIC2020)," 2020.

Yung-Han Ho received his B.S. and M.S. degrees in electrophysics and electronics engineering from National Chiao Tung University (NCTU) in 2008 and 2010, respectively.

He is currently pursuing his Ph.D. degree in the department of computer science, National Yang Ming Chiao Tung University (NYCU). His research interests are learning-based image/video coding, computer vision, and machine learning.

Chih-Chun Chan received his B.S. degree in computer science and information engineering from National Taiwan Normal University in 2019.

He is currently pursuing his M.S. degree in the department of computer science, National Yang Ming Chiao Tung University (NYCU). His research interests are learning-based image/video coding, computer vision, and machine learning.

Wen-Hsiao Peng (M'09-SM'13) received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1997, 1999, and 2005, respectively, all in electronics engineering.

He was with the Intel Microprocessor Research Laboratory, Santa Clara, CA, USA, from 2000 to 2001, where he was involved in the development of International Organization for Standardization (ISO) Moving Picture Experts Group (MPEG)-4 fine granularity scalability and demonstrated its application in 3-D peer-to-peer video conferencing. Since 2003, he has actively participated in the ISO/IEC MPEG digital video coding standardization process and contributed to the development of the High Efficiency Video Coding (HEVC) standard and MPEG-4 Part 10 Advanced Video Coding Amd.3 Scalable Video Coding standard. His research group at NCTU is one of the few university teams around the world that participated in the Call-for-Proposals on HEVC and its Screen Content Coding extensions. He is currently a Professor with the Computer Science Department, National Yang Ming Chiao Tung University (NYCU). He was a Visiting Scholar with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, from 2015 to 2016. He has authored over 70 technical papers in the field of video/image processing and communications and over 60 standards contributions. His research interests include learning-based video/image coding, multimedia analytics, and computer vision.

Dr. Peng is a Technical Committee Member of the Visual Signal Processing and Communications and Multimedia Systems and Application tracks of the IEEE Circuits and Systems Society (CASS). He was Technical Program Co-chair for 2021 IEEE VCIP, 2011 IEEE VCIP, 2017 IEEE ISAPCS, and 2018 APSIPA ASC; Publication Chair for 2019 IEEE ICIP; Area Chair/Session Chair/Tutorial Speaker for IEEE ICME and VCIP; and Track Chair/Session Chair/Review Committee Member for IEEE ISCAS. He served as AEIC for Digital Communications/Lead Guest Editor/Guest Editor/SEB Member for IEEE JETCAS, Associate Editor/Special Session Organizer for IEEE TCSVT, and Guest Editor for IEEE TCAS-II. He was Distinguished Lecturer of APSIPA and is Chair of IEEE CASS VSPPC Technical Committee.

Hsueh-Ming Hang (M'78-SM'91-F'02) received the B.S. and M.S. degrees in control engineering and electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and Ph.D. in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984.

From 1984 to 1991, he was with AT&T Bell Laboratories, Holmdel, NJ, and then he joined the Electronics Engineering Department of National Chiao Tung University (NCTU), Hsinchu, Taiwan, in December 1991. From 2006 to 2009, he took a leave from NCTU and was appointed as Dean of the EECS College at National Taipei University of Technology (NTUT). From 2014 to 2017, he served as the Dean of the ECE College at NCTU. He was appointed as the Dean of Faculty of System Engineering, NCTU in 2019. He has been actively involved in the international MPEG standards since 1984 and his current research interests include multimedia compression, multiview image/video processing, and deep-learning based image/video processing.

Dr. Hang holds 14 patents (Taiwan, US and Japan) and has published over 200 technical papers related to image compression, signal processing, and video codec architecture. He was an associate editor (AE) of the IEEE Transactions on Image Processing (1992-1994, 2008-2012) and the IEEE Transactions on Circuits and Systems for Video Technology (1997-1999). He is a co-editor and contributor of the Handbook of Visual Communications published by Academic Press in 1995. He was an IEEE Circuits and Systems Society Distinguished Lecturer (2014-2015) and is currently a Board Member of the Asia-Pacific Signal and Information Processing Association (APSIPA). He received the Distinguished Engineering Professor Award from Chinese Institute of Engineers (2005) and Chinese Institute of Electrical Engineering (2012). He is a recipient of the IEEE Third Millennium Medal and is a Fellow of IEEE and IET and a member of Sigma Xi.

Marek Domański received the M.Sc., Ph.D., and Habilitation degrees from the Poznań University of Technology, Poland, in 1978, 1983, and 1990, respectively.

Since 1993, he has been a Professor with the Poznań University of Technology, where he leads the Institute of Multimedia Telecommunications. He coauthored one of the very first AVC decoders for tv set-top boxes, in 2004, highly ranked technology proposals to MPEG for scalable video compression, in 2004, 3-D video coding, in 2011, and immersive video coding, in 2019. He authored three books and more than 300 articles in journals and conference proceedings. His contributions were mostly on image, video and audio compression, virtual navigation, free-viewpoint television, image processing, multimedia systems, 3-D video and color image technology, digital filters, and multidimensional signal processing.

Dr. Domański served as a member of various steering, program, and editorial committees of international journals and international conferences. He was the General Chairman/the Co-Chairman and the Host of several international conferences, including Picture Coding Symposium, PCS 2012; IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2013; European Signal Processing Conference, EUSIPCO 2007; 73rd and 112nd Meetings of MPEG; International Workshop on Signals, Systems and Image Processing, IWSSIP 1997 and 2004; and International Conference Signals and Electronic Systems, ICSES 2004.