Correcting Diffusion-Based Perceptual Image Compression with Privileged End-to-End Decoder

Yiyang Ma [1] **Wenhan Yang** [2] **Jiaying Liu** [1]

Abstract

The images produced by diffusion models can attain excellent perceptual quality. However, it is challenging for diffusion models to guarantee distortion, hence the integration of diffusion models and image compression models still needs more comprehensive explorations. This paper presents a diffusion-based image compression method that employs a privileged end-to-end decoder model as correction, which achieves better perceptual quality while guaranteeing the distortion to an extent. We build a diffusion model and design a novel paradigm that combines the diffusion model and an end-to-end decoder, and the latter is responsible for transmitting the privileged information extracted at the encoder side. Specifically, we theoretically analyze the reconstruction process of the diffusion models at the encoder side with the original images being visible. Based on the analysis, we introduce an end-to-end convolutional decoder to provide a better approximation of the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ at the encoder side and effectively transmit the combination. Experiments demonstrate the superiority of our method in both distortion and perception compared with previous perceptual compression methods. The project is at https://realpasu.github.io/CorrDiff_Website.

1. Introduction

Image compression aims to minimize the amount of data required to represent the image while retaining as much relevant information as possible, to reconstruct the image with high fidelity. The persistent evolution of image compression technologies promotes the propensity of a series of emerging applications, *e.g.*, video streaming and augmented reality. With the rapid increase in existing image resolutions, *e.g.*, the emergence and polarity of High Definition, Full High Definition, and Ultra High Definition, image compression has drawn greater attention and interest.

In the past few decades, the conventional image compression pipeline consists of several fundamental modules: transformation, quantization, and entropy coding. The images are first decomposed into less correlated components, whose distributions are then estimated and modeled. This framework leads to the birth and success of a series of coding standards, *e.g.*, JPEG (Wallace, 1991) and BPG (BPG-Contributors, 2018). The compression method can also adaptively adjust by the corresponding input, achieving better-customized results (Fu et al., 2011; 2012). However, persistent manual optimization and pattern expansion result in an overly complex framework, gradually revealing performance bottlenecks as development progresses.

With the advancement of deep learning (DL), recent years have shown that DL-based image compression methods significantly surpass classical methods in balancing bit rate and reconstruction quality. In the beginning, deep neural networks are employed to capture the nonlinear mapping relationship, to augment the function of existing modules of image codecs (Theis et al., 2017; Toderici et al., 2017; Rippel & Bourdev, 2017). The later efforts (Ballé et al., 2017; 2018; Minnen et al., 2018; Toderici et al., 2016) make the entropy estimation learnable and lead compression techniques to enter the era of end-to-end training.

To evaluate the performance of image compression methods, there are two categories of metrics including distortion and perception (Blau & Michaeli, 2019). Distortion metrics (*e.g.*, PSNR, MS-SSIM (Wang et al., 2003)) which measure the fidelity of reconstructed images are leveraged by most image compression methods. Perception metrics refer to the subjective evaluation of human eyes. However, when image compression methods reach a certain level of fidelity, rich evidence is provided to theoretically and experimentally prove that optimizing distortion inevitably leads to the degradation of perceptual quality (Blau & Michaeli, 2019; Muckley et al., 2023). Such degradation usually includes

[1]Wangxuan Institute of Computer Technology, Peking University, Beijing, China [2]Pengcheng Laboratory, Shenzhen, China. Correspondence to: Jiaying Liu <liujiaying@pku.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Figure 1. Visual results compared to CDC (Yang & Mandt, 2023) and ILLM (Muckley et al., 2023). The patch is cropped from *daniel-robert-405.png* from CLIC *professional* dataset (Toderici et al., 2020). **[Zoom in for best view]**

over-smoothness or blurring, which has minor effects on the distortion metrics but incurs a significant decrease in visual perception. To address the issue, the generation models, which are good at generating human visually pleasing details, are incorporated into the image compression methods for achieving better subjective visual quality. Agustsson et al. (2019); Mentzer et al. (2020) propose to employ Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) as image decoders to generate the reconstruction results with rich details. He et al. (2022b) and Muckley et al. (2023) improve the GAN-based methods with advanced perception models. Due to the great recent success of diffusion-based models (Song & Ermon, 2019; Ho et al., 2020), Yang & Mandt (2023) leverage diffusion models to reconstruct images, getting impressive results in terms of perception. However, it has been validated that vanilla diffusion models tend to reconstruct images with richer visual details but less fidelity to the original images (Saharia et al., 2022b; Yang & Mandt, 2023). Intuitively, the powerful detail reconstruction capacity arises from the progressive process of adding/removing noise, which might not be friendly to distortion measures. Furthermore, the diffusion models rely on the efficacy of sampling, which further results in the difficulty of obtaining effective deterministic compression mapping.

To leverage both the generation capacity of diffusion models with less distortion loss, we propose to transmit a *correction item* compactly to correct the sampling process of the diffusion decoder. This correction is generated from the bitstream via an end-to-end convolutional decoder adaptively, which maintains the low bit rate while guaranteeing the distortion and improving the perceptual quality. In detail, we first theoretically analyze the approximation error of the score function which is leveraged in the reconstruction process of the diffusion model by a score network. Then, we introduce a privileged end-to-end convolutional decoder and linearly combine such decoder with the score network via a mathematically derived factor to build an approximation of the above-mentioned error. At last, we can simply send these linear factors that are used to combine the two components with a few bits as privileged information, assisting the decoder to correct the sampling process, which makes reconstruction results obtain improved visual quality. The proposed method is called "CorrDiff" (abbreviated from "Corrected Diffusion"). Noting that the target of reconstructing images with high fidelity of the original images, comparatively in our work, we refer the concept of "perceptual quality" to the general superiority of a set of image-level perception-oriented metrics (*e.g.*, LPIPS (Zhang et al., 2018)) to evaluate image pair-wise fidelity in this paper.

The contributions can be summarized as follows:

1. We propose a novel diffusion-based image compression framework, CorrDiff, with a privileged end-to-end decoder. This privileged decoder helps correct the sampling process with only a few bits to facilitate the decoder side to achieve better reconstruction.

2. We theoretically analyze the sampling process of diffusion models and further derive the design of the end-to-end correction paradigm.

3. We conduct extensive experiments including diverse metrics and give ablation studies to demonstrate the superiority of the proposed image compression method as well as the effectiveness of each component.

2. Related Works

2.1. Learned Image Compression Methods

In recent years, along with the development of deep learning, more and more DL-based image compression methods have been proposed. Ballé et al. (2016) design generalized divisive normalization which is widely used in the image compression task. Ballé et al. (2018) propose to use a hyper-bit rate to represent the mean of the main bit rate, reducing the cost of encoding the bit rates. Minnen et al. (2018) further employ a context model to predict the representation based on previous positions. Cheng et al. (2020) introduce attention mechanism (Vaswani et al., 2017) to handle the relation between different regions.

To achieve better perceptual quality which is closer to human perception, a series of methods leveraging generative models to build decoders are proposed (Agustsson et al., 2019; Mentzer et al., 2020; He et al., 2022b; Agustsson et al., 2023; Muckley et al., 2023; Yang & Mandt, 2023).

2.2. Generative Models

As the name implies, image generative models aim at creating novel images that contain visual details with high perceptual quality. Goodfellow et al. (2014) propose GANs which contain a generator and a discriminator to compete. Kingma & Welling (2014) design variational auto encoder to explicitly model the posterior probability distribution of images. Kingma & Dhariwal (2018) propose flow-based models to map the distribution of images to a Gaussian distribution in an invertible way and generate images through the reverse map.

In the past few years, Ho et al. (2020) introduce diffusion models in a simplified form. Diffusion models create images by gradually de-noising from the beginning of a Gaussian noise, which has been validated to have a great capacity to create high-quality contents (Saharia et al., 2022a; Rombach et al., 2022; Ma et al., 2023; Ruan et al., 2022). The corresponding theories grow fast (Song & Ermon, 2019; Song et al., 2021b; Kingma et al., 2021; Nichol & Dhariwal, 2021; Bao et al., 2022; Ho & Salimans, 2022), making it mathematically complete. In low-level vision tasks, diffusion models also achieve impressive performance (Xia et al., 2023; Ma et al., 2024). Thus, it is intuitive to employ diffusion models in the task of image compression to reconstruct images with high perceptual quality. However, it is challenging because the integration of diffusion models and image compression models is non-trivial as we have mentioned in the abstract.

2.3. Distortion and Perception Metrics

To evaluate the performance of image compression methods, several metrics are employed. They can be divided into two categories, distortion and perception. In the category of distortion, PSNR is the most widely-used metric that measures the mean square error (MSE). Furthermore, Wang et al. (2003) propose multi-scale structural similarity (MS-SSIM) to compare patch-level similarity between two images. Xue et al. (2013) design GMSD, which compares the gradient of two images.

The perceptual metrics leverage features and their combinations of pre-trained neural networks. LPIPS (Zhang et al., 2018) calculates the summation of MSE between a pyramid of deep features, which can employ different networks as the backbone. DISTS (Ding et al., 2020) transforms images through Gaussian kernels and evaluates the transformed features. Unlike previous image-level metrics, FID (Heusel et al., 2017) is a widely-used metric to measure the divergence between two distributions. It is commonly leveraged to evaluate the performance of image generation models, as these models aim to create novel images within the data distribution. However, it is not suitable enough to measure the performance of image compression models because the target of such models is to reconstruct the original images with high fidelity, instead of generating new images.

3. Method

We first review general theories of diffusion models, including the score-matching perspective and its corresponding differential equations. Then we analyze the approximation error of the score function by the score network when the original images are visible, which can provide privileged information and facilitate correcting the error at the decoder side. Finally, we design a paradigm to approximate the error through an external end-to-end decoder and send the approximation to the decoder side with a few bits. With the proposed correction mechanism, we can achieve the goal of obtaining better reconstruction at the side of the decoder in terms of both distortion and perceptual quality. The proposed CorrDiff is illustrated in Fig. 2.

3.1. Preliminaries

Diffusion models (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021b) are a kind of generative model that can create impressively high-quality images. Diffusion models first perturb images $\mathbf{x}_0^*$ by adding Gaussian noise through a pre-specified distribution, then train a score function to estimate the noise injected into the images. By utilizing the score function iteratively, we can sample novel images from the distribution of pure Gaussian noise. The process of adding noise is called *forward process* and the de-noising process is called *reverse process*. The distribution of *forward process* is given below:

$$\begin{aligned} q(\mathbf{x}_t|\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \alpha(t)\mathbf{x}_0, \sigma^2(t)\mathbf{I}), \\ q(\mathbf{x}_T) &= \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}), \end{aligned} \tag{1}$$

where $\alpha(t), \sigma(t)$ are differentiable hyper-parameter functions of t. Furthermore, Kingma et al. (2021); Lu et al. (2022) prove that the following stochastic differential equation (SDE) has the same transition distribution with the conditional distribution $q(\mathbf{x}_t|\mathbf{x}_0)$ before at any $t \in [0, T]$:

$$\mathrm{d}\mathbf{x}_t = f(t)\mathbf{x}_t\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t, \tag{2}$$

where $\mathbf{w}_t$ is a standard Wiener process, and $f(t), g(t)$ are functions of $\alpha(t), \sigma(t)$, given by:

$$\begin{aligned} f(t) &= \frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}, \\ g^2(t) &= \frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}\sigma^2(t). \end{aligned} \tag{3}$$

Song et al. (2021b); Lu et al. (2022) prove that the *reverse process* can be done by solving the SDE below:

$$\begin{aligned} \mathrm{d}\mathbf{x}_t &= \left[f(t)\mathbf{x}_t - g^2(t)\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}_t, \\ \mathbf{x}_T &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{aligned} \tag{4}$$

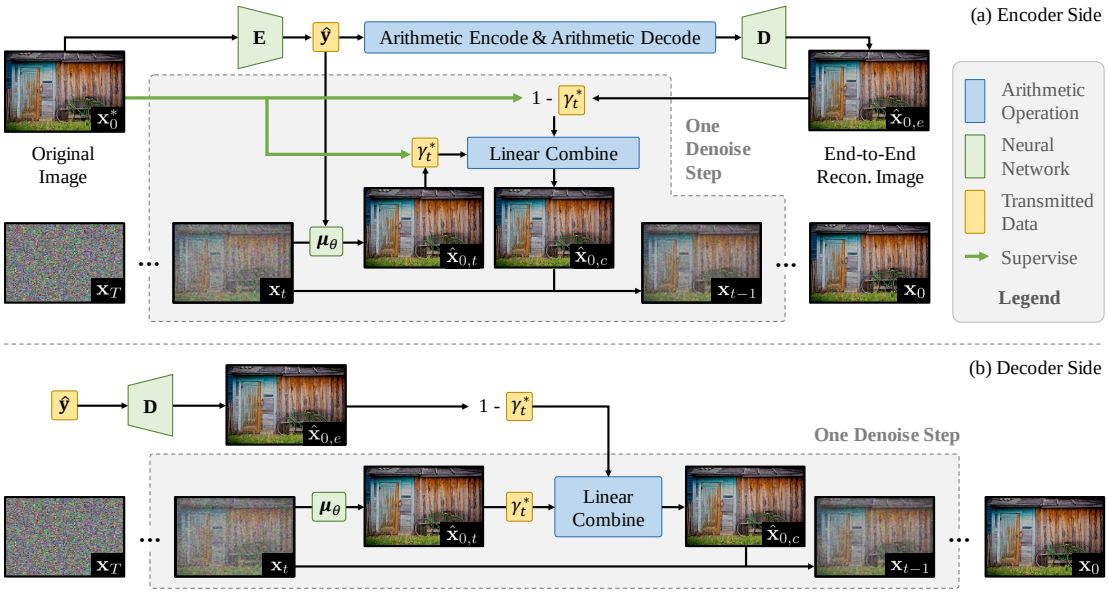


Figure 2. The framework of the proposed method. $\mathbf{E}$ denotes the encoder, $\mathbf{D}$ denotes the end-to-end decoder, and $\boldsymbol{\mu}_\theta$ denotes the score network. The yellow frame denotes the transmitted parts. The subimage (a) illustrates the pipeline at the encoder side, which obtains the representation $\hat{\mathbf{y}}$ and the factor set $\{\gamma_t^*\}_{t=1}^T$. The subimage (b) shows the reconstruction process on the decoder side.

where the score function $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ is estimated by a neural network. In practice, the network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ is trained to estimate the scaled score function $-\sigma(t)\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ by optimizing the loss function:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{t,\mathbf{x}_0}\left[\omega(t)\|\mathbf{s}_\theta(\mathbf{x}_t,t) + \sigma(t)\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\|^2\right] \\ &= \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\omega(t)\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \boldsymbol{\epsilon}\|^2\right], \end{aligned} \tag{5}$$

where $\omega(t)$ is the weighting function of loss terms of different t. Moreover, Song et al. (2021b) gives an ordinary differential equation which is the same as the marginal distribution of Eqn. (6):

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = f(t)\mathbf{x}_t + \frac{g^2(t)}{2}\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t), \mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6}$$

After the training process, we can utilize the trained score network $\mathbf{s}_\theta$ and solve the Eqn. (6) through numerical solvers like DPM-Solver (Lu et al., 2022) or DDIM (Song et al., 2021a) to sample from the diffusion model.

In this subsection, we discuss the general theories of diffusion models. In the following parts of this subsection, we will present our method of leveraging diffusion models for image compression via the proposed approach to correct the *reverse process* by taking the original images $\mathbf{x}_0^*$ as the available privileged information.

3.2. Correcting the Score Function with Original Images

As we have discussed in the previous subsection, we train a score network to estimate the score function of the diffusion model. When we manage to leverage diffusion models in the task of image compression, we first obtain the discretized image representation $\hat{\mathbf{y}}$ by an encoder $\mathbf{E}$ and further quantization following previous DL-based image compression methods (Ballé et al., 2018; Minnen et al., 2018):

$$\hat{\mathbf{y}} = Q\big(\mathbf{E}(\mathbf{x}_0^*)\big), \tag{7}$$

where we use the subscript 0 to indicate noise-free images following the setting of diffusion models and the superscript $*$ to indicate original images. We extend the score network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ with $\hat{\mathbf{y}}$ as conditions $\mathbf{s}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t)$. However, it is notable that the original images $\mathbf{x}_0^*$ are visible at the encoder side. It is intuitive to analyze how to correct the estimation of the conditioned score function $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}})$ at the encoder side with $\mathbf{x}_0^*$ as an additional condition. We have:

$$q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*) = \frac{q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t) q_t(\mathbf{x}_t|\hat{\mathbf{y}})}{p(\mathbf{x}_0^*|\hat{\mathbf{y}})}, \tag{8}$$

through Bayes rule. Thus, we have:

$$\begin{aligned}\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*) =& \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) + \\ & \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t),\end{aligned} \tag{9}$$

because $p(\mathbf{x}_0^*|\hat{\mathbf{y}})$ is not related to $\mathbf{x}_t$. In Eqn. (9), it is noticed that the first item $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}})$ is estimated by the score function $\mathbf{s}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t)$, and the second term $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t)$ is a correction item resulted from the visible original images $\mathbf{x}_0^*$. Therefore, if we can transmit such an item, we can assist the decoder side in reconstructing images with better quality. However, such correction items have the same dimensions as the original images. So, it is not effective to transmit them to the decoder side directly. Hence, we design a protocol to approximate the correction items through an external end-to-end decoder which only

Algorithm 1 Encoder Side with DDIM.

Input: original image $\mathbf{x}_0^*$
Require: encoder $\mathbf{E}$, score network $\boldsymbol{\mu}_\theta$, end-to-end decoder $\mathbf{D}$, hyper-parameter functions $\alpha(t), \sigma(t)$
Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\hat{\mathbf{y}} \leftarrow Q(\mathbf{E}(\mathbf{x}_0^*))$
for $t = T$ **to** 1 **do**
  $\hat{\mathbf{x}}_{0,t} \leftarrow \boldsymbol{\mu}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t)$
  $\hat{\mathbf{x}}_{0,e} \leftarrow \mathbf{D}(\hat{\mathbf{y}})$
  $\gamma_t^* \leftarrow \arg\min_\gamma \left[\mathrm{M}(\mathbf{x}_0^*, \gamma\hat{\mathbf{x}}_{0,t} + (1-\gamma)\hat{\mathbf{x}}_{0,e})\right]$
  $\hat{\mathbf{x}}_{0,c} \leftarrow \gamma_t^*\hat{\mathbf{x}}_{0,t} + (1-\gamma_t^*)\hat{\mathbf{x}}_{0,e}$
  $\hat{\boldsymbol{\epsilon}}_t \leftarrow \frac{1}{\sigma(t)}(\mathbf{x}_t - \alpha(t)\hat{\mathbf{x}}_{0,c})$
  $\mathbf{x}_{t-1} \leftarrow \alpha(t-1)\hat{\mathbf{x}}_{0,c} + \sigma(t-1)\hat{\boldsymbol{\epsilon}}_t$
end for
Send: representation $\hat{\mathbf{y}}$, linear factors $\{\gamma_t^*\}_{t=1}^T$

Algorithm 2 Decoder Side with DDIM.

Receive: representation $\hat{\mathbf{y}}$, linear factors $\{\gamma_t^*\}_{t=1}^T$
Require: score network $\boldsymbol{\mu}_\theta$, end-to-end decoder $\mathbf{D}$, hyper-parameter functions $\alpha(t), \sigma(t)$
Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T$ **to** 1 **do**
  $\hat{\mathbf{x}}_{0,t} \leftarrow \boldsymbol{\mu}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t)$
  $\hat{\mathbf{x}}_{0,e} \leftarrow \mathbf{D}(\hat{\mathbf{y}})$
  $\hat{\mathbf{x}}_{0,c} \leftarrow \gamma_t^*\hat{\mathbf{x}}_{0,t} + (1-\gamma_t^*)\hat{\mathbf{x}}_{0,e}$
  $\hat{\boldsymbol{\epsilon}}_t \leftarrow \frac{1}{\sigma(t)}(\mathbf{x}_t - \alpha(t)\hat{\mathbf{x}}_{0,c})$
  $\mathbf{x}_{t-1} \leftarrow \alpha(t-1)\hat{\mathbf{x}}_{0,c} + \sigma(t-1)\hat{\boldsymbol{\epsilon}}_t$
end for
Return: reconstructed image $\mathbf{x}_0$

needs a few bits to send. We will state our protocol and its corresponding theories in the following subsection.

3.3. Approximation of the Correction via an External End-to-End Decoder

First, as the score network actually estimates the noise injected into the original images, we can directly build pseudo noise-free images $\hat{\mathbf{x}}_{0,t}$ at any time-step t following (Ho et al., 2020; Chung et al., 2023):

$$\begin{aligned}\hat{\mathbf{x}}_{0,t} &:= \mathbb{E}_{\mathbf{x}_0 \sim q_t(\mathbf{x}_0|\mathbf{x}_t, \hat{\mathbf{y}})}[\mathbf{x}_0|\mathbf{x}_t, \hat{\mathbf{y}}] \\ &= \frac{1}{\alpha(t)}\big(\mathbf{x}_t + \sigma^2(t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\hat{\mathbf{y}})\big). \end{aligned} \tag{10}$$

Then, we can prove that (refer to Appendix Sec. A):

Theorem 3.1. *The conditional distribution $q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t)$ can be approximated by $q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \hat{\mathbf{x}}_{0,t})$.*

When given the pseudo noise-free image $\hat{\mathbf{x}}_{0,t}$, the $\mathbf{x}^*$ will distribute around $\hat{\mathbf{x}}_{0,t}$ which is less relative to the representation $\hat{\mathbf{y}}$. Thus, we have:

$$q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \hat{\mathbf{x}}_{0,t}) \approx q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}). \tag{11}$$

In the following derivation, we agree that images $\mathbf{x}_0^*$ are vectors (which are actually matrices) for formal simplicity. The exact shapes do not affect the correctness of the theories. With the approximation of Theorem 3.1 and Eqn. (11), we transform the correction item by:

$$\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t) \approx \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}). \tag{12}$$

We have the chain rule of vector differentiation:

$$\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) = \left(\frac{\partial\hat{\mathbf{x}}_{0,t}}{\partial\mathbf{x}_t}\right)^\top \nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}). \tag{13}$$

It is noticed that:

$$\begin{aligned}&(\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}))^\top \nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) \\ =&\left[\left(\frac{\partial\hat{\mathbf{x}}_{0,t}}{\partial\mathbf{x}_t}\right)^\top \nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})\right]^\top \nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) \\ =&\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})^\top \frac{\partial\hat{\mathbf{x}}_{0,t}}{\partial\mathbf{x}_t}\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}).\end{aligned} \tag{14}$$

It is noticed that the pseudo noise-free images $\hat{\mathbf{x}}_{0,t}$ are obtained by the score network from $\mathbf{x}_t$ to estimate the original images $\mathbf{x}_0^*$. Considering that the original images $\mathbf{x}_0^*$ have positive correlation with $\mathbf{x}_t$ on average, we have:

$$\mathbb{E}_{\mathbf{x}_t}\left[\frac{\partial\hat{\mathbf{x}}_{0,t}}{\partial\mathbf{x}_t}\right] \approx \mathbb{E}_{\mathbf{x}_t}\left[\frac{\partial\mathbf{x}_0^*}{\partial\mathbf{x}_t}\right] \succ \mathbf{0}. \tag{15}$$

Thus we approximately have:

$$\left[\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})\right]^\top \nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) > 0, \tag{16}$$

which indicates $-\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})$ is a descent direction of the function $\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}))$ of $\mathbf{x}_t$. Hence, we can leverage $\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})$ as an approximation of the direction of $\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})$. Considering the relation between $\mathbf{x}_t$ and $\hat{\mathbf{x}}_{0,t}$ defined in Eqn. 10, we multiply $\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})$ with the reciprocal of the factor $\frac{\sigma^2(t)}{\alpha(t)}$ to ensure similar numerical scale:

$$\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) \approx \frac{\alpha(t)}{\sigma^2(t)}\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}). \tag{17}$$

It is noticed that $\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})$ describes the similarity between the original image $\mathbf{x}_0^*$ and the pseudo noise-free image $\hat{\mathbf{x}}_{0,t}$. Thus, in consideration of the perceptual characteristic of images, we utilize a perception-oriented metric $\mathrm{M}(\cdot,\cdot)$ which measures the distance of two images to estimate $\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t})$:

$$\nabla_{\hat{\mathbf{x}}_{0,t}}\log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) \approx -\nabla_{\hat{\mathbf{x}}_{0,t}}\mathrm{M}(\mathbf{x}_0^*, \hat{\mathbf{x}}_{0,t}). \tag{18}$$

However, transmitting such an item is still not efficient. To give a further approximation with a few bits, we introduce an end-to-end decoder $\mathbf{D}$ which directly decodes the representation $\hat{\mathbf{y}}$ to images $\hat{\mathbf{x}}_{0,e} = \mathbf{D}(\hat{\mathbf{y}})$ (we use the subscript 0 to indicate noise-free images and e to indicate end-to-end results). We approximate $-\nabla_{\hat{\mathbf{x}}_{0,t}} \mathrm{M}(\mathbf{x}_0^*, \hat{\mathbf{x}}_{0,t})$ on the direction of $\hat{\mathbf{x}}_{0,e} - \hat{\mathbf{x}}_{0,t}$. The accuracy of such an approximation depends on the accuracy of $\hat{\mathbf{x}}_{0,e}$, which is the result of end-to-end decoder $\mathbf{D}$. If the model $\mathbf{D}$ is well trained, the quality of $\hat{\mathbf{x}}_{0,e}$ can be guaranteed. The approximation is:

$$-\nabla_{\hat{\mathbf{x}}_{0,t}} \mathrm{M}(\mathbf{x}_0^*, \hat{\mathbf{x}}_{0,t}) \approx \left[\gamma_t^* \hat{\mathbf{x}}_{0,t} + (1-\gamma_t^*)\hat{\mathbf{x}}_{0,e}\right] - \hat{\mathbf{x}}_{0,t}, \quad (19)$$

where the linear factor γ_t^* is defined as:

$$\gamma_t^* := \arg\min_{\gamma} \left[\mathrm{M}(\mathbf{x}_0^*, \gamma\hat{\mathbf{x}}_{0,t} + (1-\gamma)\hat{\mathbf{x}}_{0,e})\right], \quad (20)$$

to ensure that the combination is closer to the original image than $\hat{\mathbf{x}}_{0,t}$. As long as $\gamma_t^* \neq 1$, the difference $[\gamma_t^* \hat{\mathbf{x}}_{0,t} + (1-\gamma_t^*)\hat{\mathbf{x}}_{0,e}] - \hat{\mathbf{x}}_{0,t}$ will be a descent direction of $\mathrm{M}(\mathbf{x}_0^*, \hat{\mathbf{x}}_{0,t})$. The linear factor γ_t^* is just a float number, which is effortless to transmit. Such a γ_t^* is quite easy to obtain by gradient descent through $\mathrm{M}(\cdot,\cdot)$ at every time-step t. The implementations of $\mathrm{M}(\cdot,\cdot)$ can be variable (*e.g.*, LPIPS (Zhang et al., 2018) and DISTS (Ding et al., 2020)). In summary, taking Eqn. (10), (12), (17), (18), (19) into (9), the corrected score function is given below (the corresponding proof refers to Appendix Sec. A):

Theorem 3.2. *$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*)$ can be approximated by the following combination:*

$$\begin{aligned}&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*) \\ \approx &\frac{\alpha(t)}{\sigma^2(t)}\left[\gamma_t^* \hat{\mathbf{x}}_{0,t} + (1-\gamma_t^*)\hat{\mathbf{x}}_{0,e}\right] - \frac{\mathbf{x}_t}{\sigma^2(t)}. \end{aligned} \quad (21)$$

Furthermore, considering the characteristic of the image compression task, we train the score network $\boldsymbol{\mu}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t)$ to predict $\hat{\mathbf{x}}_{0,t}$ directly instead of training $\mathbf{s}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t)$ to predict $\boldsymbol{\epsilon}$. Thus, the actual used score function is:

$$\begin{aligned}&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*) \\ \approx &\frac{\alpha(t)}{\sigma^2(t)}\left[\gamma_t^* \boldsymbol{\mu}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t) + (1-\gamma_t^*)\mathbf{D}(\hat{\mathbf{y}})\right] - \frac{\mathbf{x}_t}{\sigma^2(t)}. \end{aligned} \quad (22)$$

In conclusion, we depict the protocol below:

1. **At the encoder side**, we solve the reverse process Eqn. (6) with corrected score function Eqn. (22) at the encoder side and calculate the linear factors γ_t^* at each time-step t with the original images $\mathbf{x}_0^*$ being visible. The corresponding algorithm is shown in Algorithm 1.
2. **Sending** the representation $\hat{\mathbf{y}}$ and the set of linear factors $\{\gamma_t^*\}_{t=1}^T$ to the decoder.
3. **At the decoder side**, we leverage the received representation $\hat{\mathbf{y}}$ and the set $\{\gamma_t^*\}_{t=1}^T$, and use the corrected score function Eqn. (22) to reconstruct the images. The corresponding algorithm is shown in Algorithm 2.

3.4. Model Training

As we have depicted in previous subsections, our framework contains three main parts: the encoder $\mathbf{E}$, the end-to-end decoder $\mathbf{D}$ and the score network $\boldsymbol{\mu}_\theta$. Besides, being similar to previous image compression methods, our framework also contains an entropy model to predict the mean and variance of the representation $\hat{\mathbf{y}}$ to arithmetically encode and decode $\hat{\mathbf{y}}$ and estimate the bit rate during training. We train the model in two phases. First, we load the parameters of $\mathbf{E}$ and the entropy model from a pre-trained end-to-end image compression model and only train the score network $\boldsymbol{\mu}_\theta$ with the loss below:

$$\begin{aligned}\mathcal{L}_{\text{phase}_1} = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\big[&\|\boldsymbol{\mu}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t) - \mathbf{x}_0^*\|^2 + \\ &\lambda_\mu \mathrm{M}_\mu(\boldsymbol{\mu}_\theta(\mathbf{x}_t, \hat{\mathbf{y}}, t), \mathbf{x}_0^*)\big], \end{aligned} \quad (23)$$

where M_μ is the perceptual loss leveraged in the training process following previous perceptual image compression methods (Mentzer et al., 2020; Yang & Mandt, 2023; Muckley et al., 2023) and λ_μ is the corresponding loss weight. After the training process of only the score network $\boldsymbol{\mu}_\theta$, we train the entire framework including $\mathbf{E}$, $\mathbf{D}$, the entropy model and $\boldsymbol{\mu}_\theta$ with the loss below:

$$\begin{aligned}\mathcal{L}_{\text{phase}_2} = \mathcal{L}_{\text{phase}_1} + &\|\mathbf{D}(\hat{\mathbf{y}}) - \mathbf{x}_0^*\|^2 + \\ &\lambda_e \mathrm{M}_e(\mathbf{D}(\hat{\mathbf{y}}), \mathbf{x}_0^*) + \lambda_r R(\hat{\mathbf{y}}), \end{aligned} \quad (24)$$

where M_e is the perceptual loss of end-to-end decoder, $R(\cdot)$ is the estimated bit rate, and λ_e, λ_r are the corresponding loss weights. The implementations will be given in Sec. 4.1.

4. Experiments

4.1. Implementations

Model, Training and Inferring Settings. We implement our score network based on the architecture of ADM (Dhariwal & Nichol, 2021) with fewer parameters. We leverage ELIC (He et al., 2022a) including its encoder as $\mathbf{E}$, its entropy model and its decoder as our end-to-end decoder $\mathbf{D}$. Please refer to the supplementary materials for details.

During training process, we utilize DISTS (Ding et al., 2020) as M_μ and Alex-based (Krizhevsky et al., 2012) LPIPS (Zhang et al., 2018) as M_e. We use different implementations of M_μ and M_e to avoid overfitting on one single metric. The perceptual weights are $\lambda_\mu = 0.16$ and $\lambda_e = 0.64$. We use 5 different bit rate weights λ_r including $[0.5, 0.2, 0.1, 0.05, 0.02]$ to train 5 models with different bit rates. We first train only the score network for $400,000$

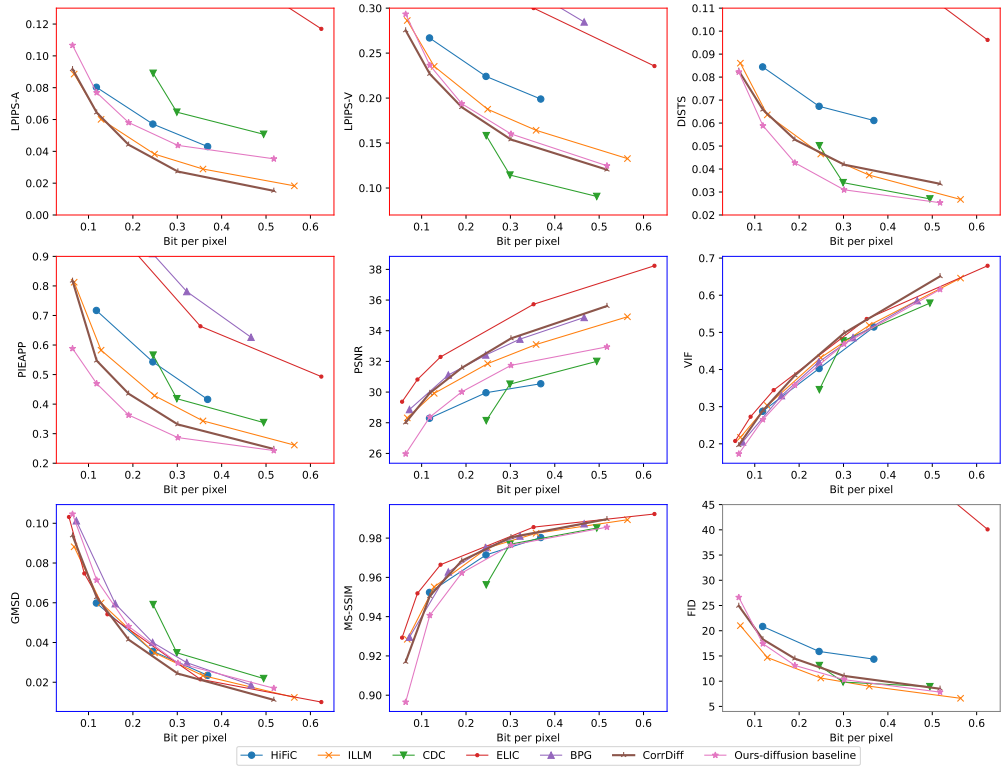


Figure 3. performance of diverse metrics on CLIC *professional* dataset. **[Zoom in for best view]**

iterations and then train the entire framework for another $400,000$ iterations with a batch size of 8, learning rate of $5e-5$ and optimizer of Adam (Kingma & Ba, 2014). We train all the models on the dataset of DIV2K (Agustsson & Timofte, 2017) which includes 800 high-resolution images. We randomly crop them into 256×256 patches in the training process. We also employ EMA with the rate of 0.9999 to stabilize the training process following previous diffusion-based methods.

During inference, we use DDIM (Song et al., 2021a) as the diffusion sampler with 8 steps and transmit γ_t^* in the form of `float16` with 16 bits, which means only 128 bits in total. We leverage VGG-based (Simonyan & Zisserman, 2014) LPIPS as M and use the vanilla gradient descent method provided by PyTorch (PyTorch-Contributors, 2024) to obtain all the γ_t^* at every time-step.

Datasets and Metrics. We evaluate our method on 3 datasets: Kodak (Kodak, 2024), CLIC *professional* (Toderici et al., 2020) and DIV2K-test (Agustsson & Timofte, 2017). Kodak which includes 24 images is one of the most widely-used datasets in the image compression task. CLIC *professional* and DIV2K-test which contain 41 and 100 high-resolution images respectively are utilized to demonstrate the performance on large images.

To demonstrate the superiority of our method in both distortion and perceptual quality, we leverage a set of diverse metrics. For distortion, we utilize PSNR, VIF (Sheikh & Bovik, 2006) and GMSD (Xue et al., 2013). For perception, we utilize LPIPS (Zhang et al., 2018), DISTS (Ding et al., 2020), and PIEAPP (Prashnani et al., 2018). It is noticed that LPIPS can be employed with different backbones including AlexNet (Krizhevsky et al., 2012) and VGG (Simonyan & Zisserman, 2014). Furthermore, we also leverage FID (Heusel et al., 2017), which measures the distance between two image distributions, to evaluate the general quality of our reconstructed images. In Fig. 3 which shows quantitative results, the charts with red frames are perceptual metrics, and curves with blue frames are distortion metrics. FID is shown in a gray frame due to its particularity.

Baseline Methods. We evaluate several image compression methods as baselines. We leverage BPG (BPG-Contributors, 2018) as a representative of classical methods. For DL-based methods, we employ an MSE-oriented method, ELIC (CVPR 2022) (He et al., 2022a), and a series of perceptual image compression methods including HiFiC (NIPS 2020) (Mentzer et al., 2020), ILLM (ICML 2023) (Muckley et al., 2023) and CDC (NIPS 2023) (Yang & Mandt, 2023).

4.2. Quantitative Results

We show the R-D curves of the 9 metrics mentioned before on the dataset of CLIC *professional* in Fig. 3. We further give results on the dataset of DIV2K and Kodak in Appendix Sec. C. We also show the results of our method without the correction as an ablation study, which refer to Sec. 4.4. It

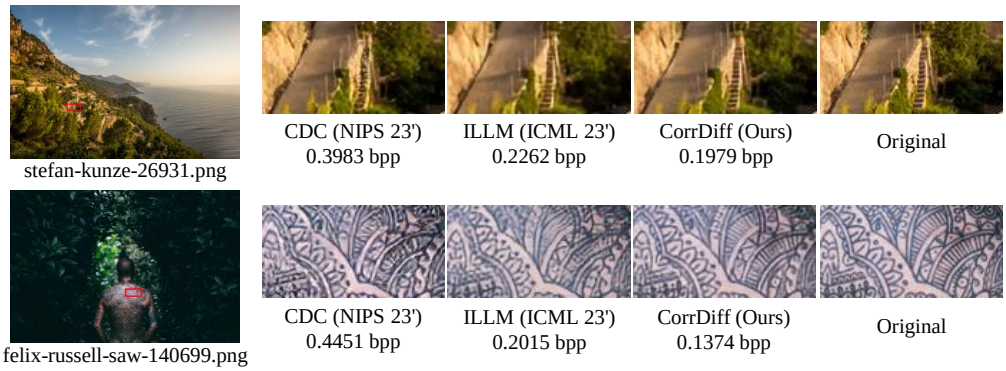


Figure 4. Visual results compared to CDC (Yang & Mandt, 2023) and ILLM (Muckley et al., 2023). **[Zoom in for best view]**

can be seen that the proposed method achieves general superiority of diverse perceptual metrics along with better distortion compared with other perceptual image compression methods, demonstrating the efficiency of the proposed method. The only diffusion-based method, CDC, outperforms on LPIPS-VGG because LPIPS-VGG is part of its training target, while it fails to achieve satisfactory results in other metrics especially the distortion metrics (*e.g.*, PSNR, GMSD), revealing the limitation of vanilla diffusion-based image compression models. ILLM, as the state-of-the-art perceptual image compression model, achieves competitive FID which is explicitly modeled during its training process, but performs inferiorly in all other metrics. Furthermore, FID is not a suitable enough metric to evaluate the performance of image compression methods because it measures the distance between two image distributions but the task of image compression focuses more on the fidelity of reconstructing the original images themselves instead of the distribution of all the reconstructed images. It is notable that, at the same time performing well on diverse perceptual metrics, our method also performs better on distortion metrics than previous perceptual image compression methods (*e.g.*, CDC, ILLM, and HiFiC), indicating that our method achieves a better distortion-perception trade-off.

4.3. Qualitative Results

To further demonstrate the perceptual quality of our results, we give several cases of different perceptual image compression methods in Fig. 1 and Fig. 4. It is obvious that the reconstructed results of our method have more visual details with higher fidelity costing fewer bits. The full versions of the visual results refer to Appendix Sec. D.

4.4. Ablation Studies

We conduct the ablation study on the proposed design of introducing an external end-to-end decoder. We provide performance of using only the diffusion part in Fig. 3. We further give average BD rates (Bjontegaard, 2001) compared with the final setting on perception and distortion metrics

Table 1. Average BD rates on perception and distortion metrics of different settings of our CorrDiff on CLIC *professional* dataset.

Setting	BD Rate (%)		
	Perception	Distortion	Total
CorrDiff (final)	-	-	-
Only End-to-End	+22.177	−3.923	+9.127
Only Diffusion	+4.833	+28.756	+16.794
Direct $\boldsymbol{\mu}_\theta$	+12.113	+10.358	+11.235

of leveraging only the end-to-end part and using the score network $\boldsymbol{\mu}_\theta$ for one step to directly reconstruct the images in Tab. 1. When calculating the BD rates, we include FID as a perceptual metric and exclude the metrics that were leveraged as the targets during training (LPIPS-A for "only end-to-end" and DISTS for "only diffusion" and "direct $\boldsymbol{\mu}_\theta$"). We further calculate the average BD rate of all the metrics. As illustrated, leveraging only the diffusion can achieve fair perceptual results because it is a powerful generative model, but leads to poor distortion being similar to CDC (which is a vanilla diffusion-based method). Utilizing only the end-to-end decoder performs well in distortion as expected, but does not have the general superiority in diverse perceptual metrics, indicating the loss of perceptual quality. Employing only the score function has poor performance, revealing the significance of the diffusion model itself. The ablation studies prove that the proposed method can achieve a better trade-off between distortion and perceptual quality.

5. Conclusion

In this paper, we propose a diffusion-based perceptual image compression method that leverages an priviliaged end-to-end decoder to correct the score function. We leverage the fact that the original images are visible at the encoder side, propose the correction item and theoretically analyze the approximation of the correction which can be transmitted effectively. Experiments demonstrate the proposed method's superiority in terms of both distortion and perception. Further ablation studies validate the efficiency of the designed method of introducing correction of diffusion models.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62332010, and in part by the Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agustsson, E. and Timofte, R. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshops*, 2017.

Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., and Gool, L. V. Generative adversarial networks for extreme learned image compression. In *Proc. IEEE Int. Conf. Comput. Vision*, 2019.

Agustsson, E., Minnen, D., Toderici, G., and Mentzer, F. Multi-realism image compression with a conditional generator. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2023.

Ballé, J., Laparra, V., and Simoncelli, E. P. Density modeling of images using a generalized normalization transformation. In *Proc. Int'l Conf. Learning Representations*, 2016.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. In *Proc. Int'l Conf. Learning Representations*, 2017.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. In *Proc. Int'l Conf. Learning Representations*, 2018.

Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022.

Bjontegaard, G. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001.

Blau, Y. and Michaeli, T. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *Proc. Int'l Conf. Machine Learning*, 2019.

BPG-Contributors. BPG image format. https://bellard.org/bpg/, 2018.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2020.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *Proc. Int'l Conf. Learning Representations*, 2023.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2021.

Ding, K., Ma, K., Wang, S., and Simoncelli, E. P. Image quality assessment: Unifying structure and texture similarity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020.

Fu, C.-M., Chen, C.-Y., Huang, Y.-W., and Lei, S. Sample adaptive offset for HEVC. In *2011 IEEE 13th Int'l Workshop on Multimedia Signal Processing*. IEEE, 2011.

Fu, C.-M., Alshina, E., Alshin, A., Huang, Y.-W., Chen, C.-Y., Tsai, C.-Y., Hsu, C.-W., Lei, S.-M., Park, J.-H., and Han, W.-J. Sample adaptive offset in the hevc standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 2012.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2014.

He, D., Yang, Z., Peng, W., Ma, R., Qin, H., and Wang, Y. ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2022a.

He, D., Yang, Z., Yu, H., Xu, T., Luo, J., Chen, Y., Gao, C., Shi, X., Qin, H., and Wang, Y. PO-ELIC: Perception-oriented efficient learned image coding. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshops*, 2022b.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2020.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *Proc. Int'l Conf. Learning Representations*, 2014.

Kodak, E. Kodak lossless true color image suite. https://r0k.us/graphics/kodak/, 2024.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2012.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2022.

Ma, Y., Yang, H., Wang, W., Fu, J., and Liu, J. Unified multi-modal latent diffusion for joint subject and text conditional image generation. *arXiv preprint arXiv:2303.09319*, 2023.

Ma, Y., Yang, H., Yang, W., Fu, J., and Liu, J. Solving diffusion odes with optimal boundary conditions for better image super-resolution. In *Proc. Int'l Conf. Learning Representations*, 2024.

Mentzer, F., Toderici, G. D., Tschannen, M., and Agustsson, E. High-fidelity generative image compression. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2020.

Minnen, D., Ballé, J., and Toderici, G. D. Joint autoregressive and hierarchical priors for learned image compression. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2018.

Muckley, M. J., El-Nouby, A., Ullrich, K., Jégou, H., and Verbeek, J. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *Proc. Int'l Conf. Machine Learning*, 2023.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *Proc. Int'l Conf. Machine Learning*, 2021.

Prashnani, E., Cai, H., Mostofi, Y., and Sen, P. Pieapp: Perceptual image-error assessment through pairwise preference. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018.

PyTorch-Contributors. Pytorch. https://pytorch.org/, 2024.

Rippel, O. and Bourdev, L. Real-time adaptive image compression. In *Proc. Int'l Conf. Machine Learning*, 2017.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2022.

Ruan, L., Ma, Y., Yang, H., He, H., Liu, B., Fu, J., Yuan, N. J., Jin, Q., and Guo, B. MM-Diffusion: Learning multi-modal diffusion models for joint audio and video generation. *arXiv preprint arXiv:2212.09478*, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2022a.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2022b.

Sheikh, H. R. and Bovik, A. C. Image information and visual quality. *IEEE Trans. on Image Processing*, 2006.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *Proc. Int'l Conf. Learning Representations*, 2021a.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *Proc. Int'l Conf. Learning Representations*, 2021b.

Theis, L., Shi, W., Cunningham, A., and Huszár, F. Lossy image compression with compressive autoencoders. In *Proc. Int'l Conf. Learning Representations*, 2017.

Toderici, G., O'Malley, S. M., Hwang, S. J., Vincent, D., Minnen, D. C., Baluja, S., Covell, M., and Sukthankar, R. Variable rate image compression with recurrent neural

networks. In *Proc. Int'l Conf. Learning Representations*, 2016.

Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., and Covell, M. Full resolution image compression with recurrent neural networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2017.

Toderici, G., Shi, W., Timofte, R., Theis, L., Balle, J., Agustsson, E., Johnston, N., and Mentzer, F. Workshop and challenge on learned image compression (CLIC2020). In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshops*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2017.

Wallace, G. K. The JPEG still picture compression standard. *Communications of the ACM*, 1991.

Wang, Z., Simoncelli, E. P., and Bovik, A. C. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems & Computers*, 2003.

Xia, B., Zhang, Y., Wang, S., Wang, Y., Wu, X., Tian, Y., Yang, W., and Van Gool, L. DiffIR: Efficient diffusion model for image restoration. In *Proc. IEEE Int. Conf. Comput. Vision*, 2023.

Xue, W., Zhang, L., Mou, X., and Bovik, A. C. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Trans. on Image Processing*, 2013.

Yang, R. and Mandt, S. Lossy image compression with conditional diffusion models. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018.

Appendix

A. Proofs

Lemma A.1. *(Chung et al., 2023) Let $\phi(\cdot)$ be a multivariate Gaussian distribution with covariance matrix $\sigma^2\mathbf{I}$ being diagonal and mean $\boldsymbol{\mu}$. There exists a constant L, s.t., $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,*

$$|\phi(\mathbf{x}) - \phi(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|, \tag{A.1}$$

where:

$$L = \frac{d}{\sqrt{2\pi}\sigma} e^{-1/2\sigma^2}. \tag{A.2}$$

Theorem A.2. *The conditional distribution $q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t)$ can be approximated by $q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \hat{\mathbf{x}}_{0,t})$.*

Proof. We have:

$$\begin{aligned} q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t) &= \int q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_0) q_t(\mathbf{x}_0|\hat{\mathbf{y}}, \mathbf{x}_t) \mathrm{d}\mathbf{x}_0 \\ &= \mathbb{E}_{\mathbf{x}_0 \sim q_t(\mathbf{x}_0|\hat{\mathbf{y}}, \mathbf{x}_t)}[q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_0)]. \end{aligned} \tag{A.3}$$

The distribution of $q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_0)$ can be approximated by a Gaussian distribution due to the characteristic of images:

$$q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_0) \approx \phi(\mathbf{x}_0). \tag{A.4}$$

Thus we have the difference between the two items:

$$\begin{aligned} &|q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t) - q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \hat{\mathbf{x}}_{0,t})| \\ =&|\mathbb{E}_{\mathbf{x}_0 \sim q_t(\mathbf{x}_0|\hat{\mathbf{y}}, \mathbf{x}_t)}[\phi(\mathbf{x}_0)] - \phi(\hat{\mathbf{x}}_{0,t})| \\ \leq& \int |\phi(\mathbf{x}_0) - \phi(\hat{\mathbf{x}}_{0,t})| \mathrm{d}Q(\mathbf{x}_0|\hat{\mathbf{y}}, \mathbf{x}_t) \\ \leq& \frac{d}{\sqrt{2\pi}\sigma} e^{-1/2\sigma^2} \int \|\mathbf{x}_0 - \hat{\mathbf{x}}_{0,t}\| \mathrm{d}Q(\mathbf{x}_0|\hat{\mathbf{y}}, \mathbf{x}_t), \end{aligned} \tag{A.5}$$

and the item $\int \|\mathbf{x}_0 - \hat{\mathbf{x}}_{0,t}\| \mathrm{d}Q(\mathbf{x}_0|\hat{\mathbf{y}}, \mathbf{x}_t)$ is limited if the model is well-trained. □

The proof is inspired by Chung et al. (2023).

Theorem A.3. *$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*)$ can be approximated by the following combination:*

$$\begin{aligned} &\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*) \\ \approx& \frac{\alpha(t)}{\sigma^2(t)} \left[\gamma_t^* \hat{\mathbf{x}}_{0,t} + (1 - \gamma_t^*) \hat{\mathbf{x}}_{0,e}\right] - \frac{\mathbf{x}_t}{\sigma^2(t)}. \end{aligned} \tag{21}$$

Proof. Taking Eqn. (10), (12), (17), (18), (19) into (9), we have:

$$
\begin{aligned}
&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}, \mathbf{x}_0^*) \\
=&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) + \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \mathbf{x}_t) \\
\approx&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) + \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_0^*|\hat{\mathbf{y}}, \hat{\mathbf{x}}_{0,t}) \\
=&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) + \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) \\
\approx&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) + \frac{\alpha(t)}{\sigma^2(t)} \nabla_{\hat{\mathbf{x}}_{0,t}} \log q_t(\mathbf{x}_0^*|\hat{\mathbf{x}}_{0,t}) \\
\approx&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) - \frac{\alpha(t)}{\sigma^2(t)} \nabla_{\hat{\mathbf{x}}_{0,t}} \mathrm{M}(\mathbf{x}_0^*, \hat{\mathbf{x}}_{0,t}) \\
\approx&\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\hat{\mathbf{y}}) + \frac{\alpha(t)}{\sigma^2(t)} \left[(\gamma_t^* - 1)\hat{\mathbf{x}}_{0,t} + (1 - \gamma_t^*)\hat{\mathbf{x}}_{0,e}\right] \\
=&\frac{\alpha(t)}{\sigma^2(t)} \hat{\mathbf{x}}_{0,t} - \frac{\mathbf{x}_t}{\sigma^2(t)} + \frac{\alpha(t)}{\sigma^2(t)} \left[(\gamma_t^* - 1)\hat{\mathbf{x}}_{0,t} + (1 - \gamma_t^*)\hat{\mathbf{x}}_{0,e}\right] \\
=&\frac{\alpha(t)}{\sigma^2(t)} \left[\gamma_t^* \hat{\mathbf{x}}_{0,t} + (1 - \gamma_t^*)\hat{\mathbf{x}}_{0,e}\right] - \frac{\mathbf{x}_t}{\sigma^2(t)}.
\end{aligned} \tag{A.6}
$$

□

B. Further Implementation Details

Model Details. We leverage the code-base of ADM (Dhariwal & Nichol, 2021) to implement the score network $\boldsymbol{\mu}_\theta$. The detailed architecture is shown in Tab. 3. The architectures of the Encoder $\mathbf{E}$ and the end-to-end decoder $\mathbf{D}$ are the same with ELIC (He et al., 2022a). For the models trained with $\lambda_r \in [0.5, 0.2]$ and $\lambda_r \in [0.1, 0.05, 0.02]$, we load the pre-trained ELIC with $\lambda = 0.004$ and $\lambda = 0.008$ respectively. The sources of the compared methods are given in Tab. 2. We thank their owners for their contributions to the community. Following previous works, we pad the images to integral multiple to the patch of 64×64 during inference.

Table 2. The sources of the compared methods.

Classification	Method	URL
MSE-Oriented	ELIC	https://github.com/VincentChandelier/ELiC-ReImplemetation
Perceptual	HiFiC	https://github.com/Justin-Tan/high-fidelity-generative-compression
	CDC	https://github.com/buggyyang/CDC_compression
	ILLM	https://github.com/facebookresearch/NeuralCompression/tree/main/projects/illm

Implementations of other Methods and Metrics. We implement LPIPS by https://github.com/S-aiueo32/lpips-pytorch/tree/master and all other metrics by https://github.com/chaofengc/IQA-PyTorch. When calculating FID, we crop images to 256×256 patches. We crop all the images two times from the start position $(0, 0)$ and $(128, 128)$ without overlap. For BPG, we employ BPG v0.9.8 through the following script:

```
# Encode
bpgenc -o $binary_file -q $qp $input_image
# Decode
bpgdec -o $recon_image $binary_file
```

We leverage quantizer parameters in the set of $[32, 35, 37, 40, 45]$.

Table 3. Detailed architecture of our Model. The model size is the summation of the score network $\boldsymbol{\mu}_\theta$, the encoder $\mathbf{E}$, the entropy model and the end-to-end decoder $\mathbf{D}$.

	Entire Model
Model size	73.79M
Channels	96
Depth	2
Channels multiple	1,1,2,2,3
Heads	4
Attention resolution	None
BigGAN up/downsample	✓
Dropout	0.0

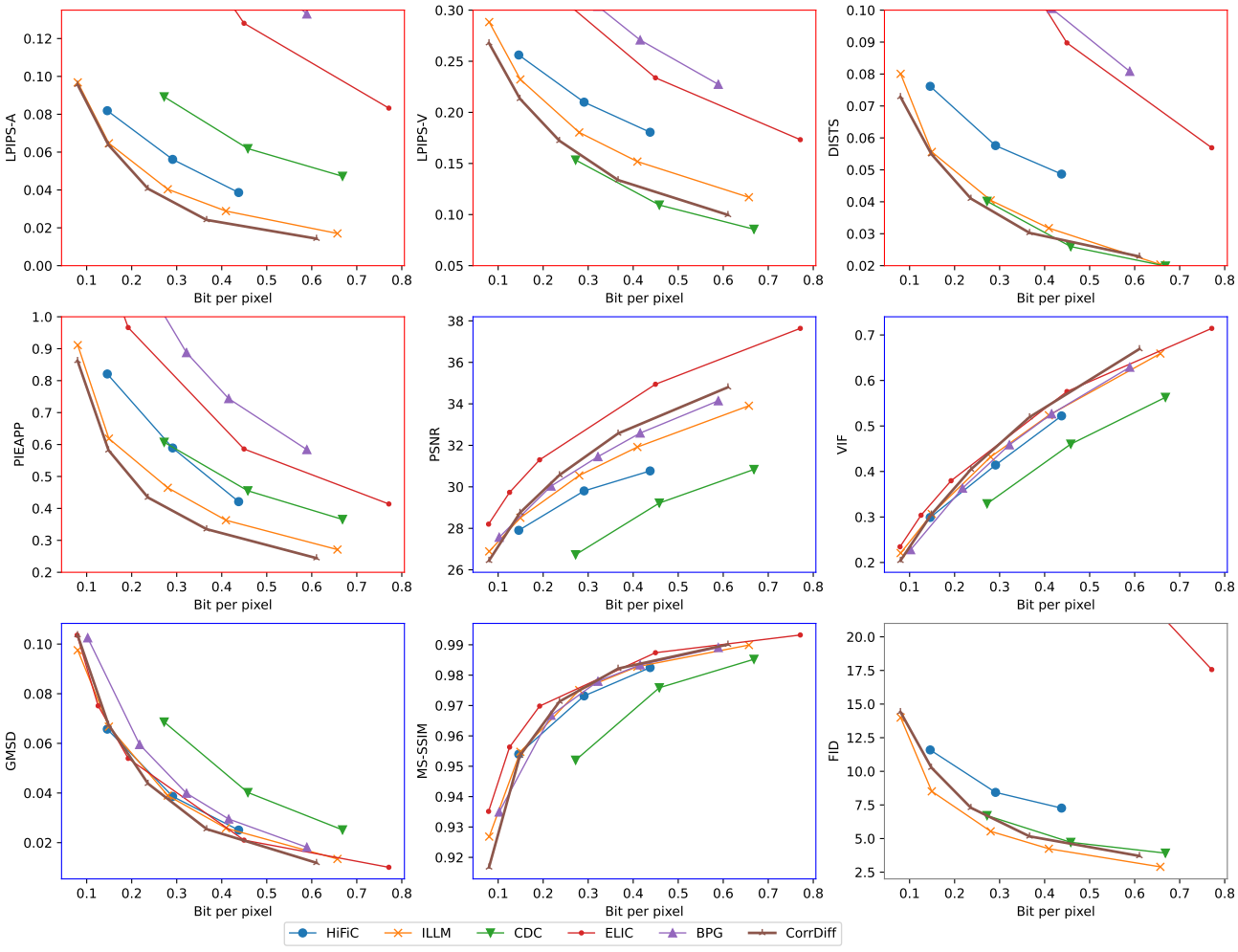


Figure 5. performance of diverse metrics on DIV2K test dataset. **[Zoom in for best view]**

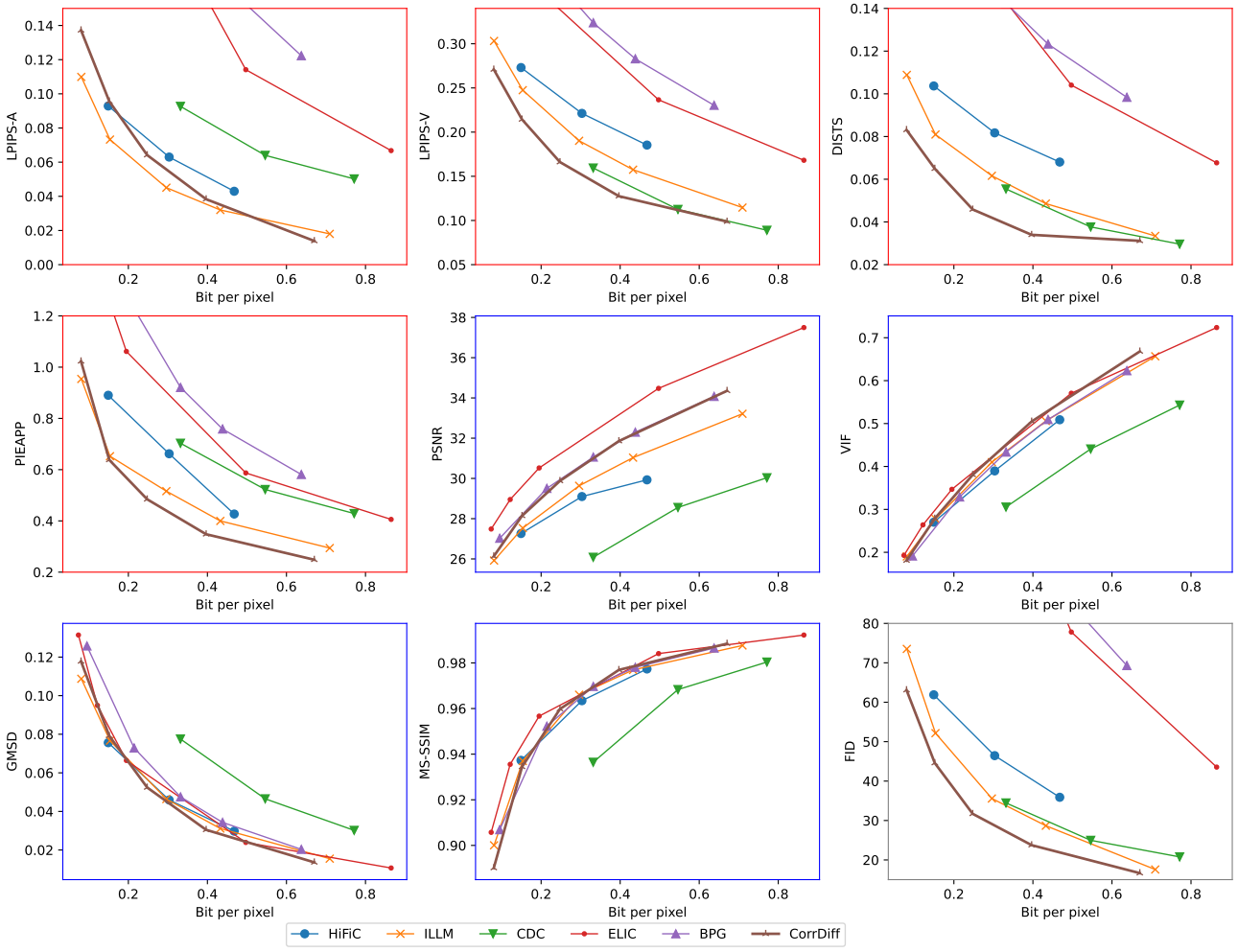


Figure 6. performance of diverse metrics on Kodak dataset. **[Zoom in for best view]**

C. performance on other datasets

We further show the performance on the dataset of Kodak (Kodak, 2024) and DIV2K-test (Agustsson & Timofte, 2017) in Fig. 6 and Fig. 5. The performance on these datasets are similar to the performance on CLIC *professional* which has been shown in the main paper.

D. Full Versions of Visual Results

We show the full versions of the images we have given in the main paper in this section in Fig. 7, 8, 9.

Original daniel-robert-405.png

CorrDiff (Ours) 0.2640 bpp

CDC (NIPS 23') 0.5125 bpp

Figure 7. Full version of daniel-robert-405.png from CLIC *professional* dataset. **[Zoom in for best view]**

Original stefan-kunze-26931.png

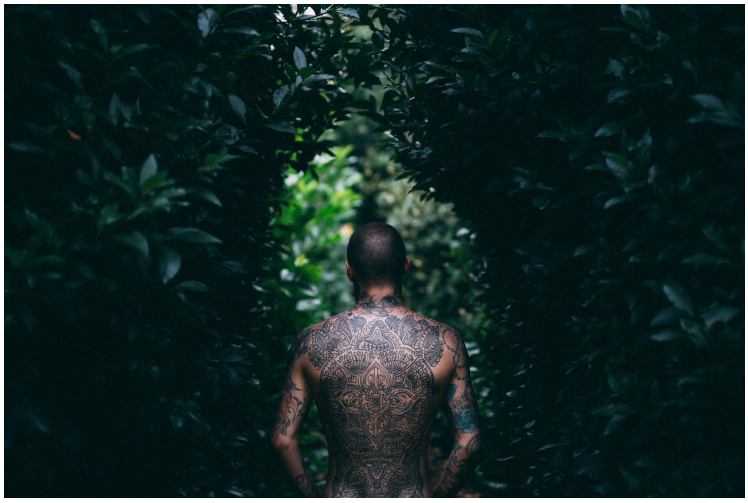
CorrDiff (Ours) 0.1979 bpp

ILLM (ICML 23') 0.2262 bpp

Figure 8. Full version of stefan-kunze-26931.png from CLIC *professional* dataset. **[Zoom in for best view]**

Original felix-russel-saw-140699.png

CorrDiff (Ours) 0.1374 bpp



ILLM (ICML 23') 0.2015 bpp

Figure 9. Full version of felix-russel-saw-140699.png from CLIC *professional* dataset. **[Zoom in for best view]**