Improving Statistical Fidelity for Neural Image Compression with Implicit Local Likelihood Models

Matthew Muckley [1] **Alaaeldin El-Nouby** [1 2] **Karen Ullrich** [1] **Hervé Jégou** [1] **Jakob Verbeek** [1]

Abstract

Lossy image compression aims to represent images in as few bits as possible while maintaining fidelity to the original. Theoretical results indicate that optimizing distortion metrics such as PSNR or MS-SSIM necessarily leads to a discrepancy in the statistics of original images from those of reconstructions, in particular at low bitrates, often manifested by the blurring of the compressed images. Previous work has leveraged adversarial discriminators to improve statistical fidelity. Yet these binary discriminators adopted from generative modeling tasks may not be ideal for image compression. In this paper, we introduce a non-binary discriminator that is conditioned on quantized local image representations obtained via VQ-VAE autoencoders. Our evaluations on the CLIC2020, DIV2K and Kodak datasets show that our discriminator is more effective for jointly optimizing distortion (e.g., PSNR) and statistical fidelity (e.g., FID) than the PatchGAN of the state-of-the-art HiFiC model. On CLIC2020, we obtain the same FID as HiFiC with 30-40% fewer bits.

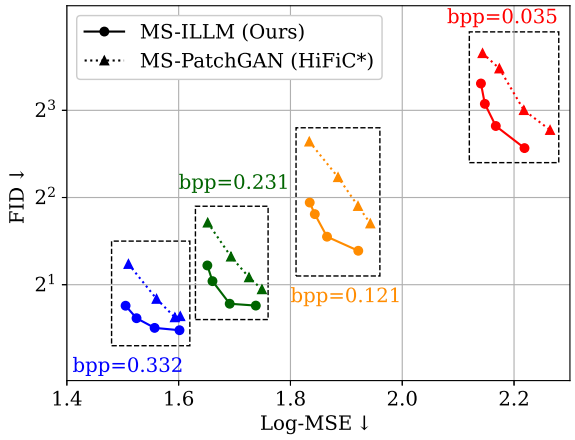


Figure 1. Comparison of distortion vs. statistical fidelity tradeoff across different bitrates. As the bitrate decreases, both distortion (as measured by MSE) and statistical fidelity (as measured by FID) degrade. Throughout all compression regimes, our discriminator achieves better trade-offs between distortion and statistical fidelity than the competing PatchGAN discriminator, as used in HiFiC.

1. Introduction

The principal task for designing digital image compression systems is to build functions that transform images into the fewest amount of bits while maintaining a fixed, predefined distortion level. For much of digital compression history, efforts involved designing three components: (1) an autoencoding transformation function, such as the discrete cosine transform (DCT) (Ahmed et al., 1974), (2) a scheme for lossy quantization of the transform coefficients, and (3) a lossless entropy coder for the quantized coefficients. Arithmetic coding (Rissanen, 1976; Pasco, 1976) solves (3) by achieving theoretical bounds (Shannon, 1948) for lossless compression. For the autoencoding transform in (1), handcrafted transforms such as DCT or wavelets (Antonini et al., 1992; Le Gall & Tabatabai, 1988) can be used. Combinations of these with carefully-designed quantization tables for step (2) form the backbone of compression standards such as JPEG and JPEG2000.

Much of the benefit of JPEG2000 over JPEG was acquired by improving the quality of the autoencoding transform. In this respect, there is great potential for neural networks to obtain further improvements as end-to-end image compression systems, as has been actively explored, see e.g. (Ballé et al., 2018; Minnen et al., 2018; Cheng et al., 2020; Mentzer et al., 2020). Deep learning methods have now surpassed both old standards (e.g., JPEG and JPEG2000) and new standards (e.g., BPG (Bellard)) in rate-distortion performance.

A common thread in the design of both traditional image compressors as well as deep learning ones is the rate-distortion optimization objective. In most cases, methods utilize a handcrafted definition of distortion such as peak-

[1]Meta AI [2]Inria. Correspondence to: Matthew Muckley <mmuckley@meta.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

signal-to-noise ratio (PSNR) or multi-scale structural similarity index (MS-SSIM) (Wang et al., 2003; 2004). However, it can be shown theoretically that especially for very low rates, optimizing for distortion necessarily pulls the statistics of the output samples away from the true distribution (Blau & Michaeli, 2019), typically via blurring or smoothing. Deep learning compressors that incorporate GANs (Agustsson et al., 2019; Mentzer et al., 2020) mitigate this by balancing distortion loss with an adversarial discriminator that attempts to align the compressed image distribution with the true distribution. More recent work has shown that the balancing of distortion and statistical fidelity can be controlled at decode time to determine how many details are synthesized (Agustsson et al., 2023). HiFiC (Mentzer et al., 2020) demonstrated that the use of GANs can lead to great benefits from the perspective of human observers, with HiFiC being preferred to BPG even when using half the bits. Similar results were observed for video compression (Mentzer et al., 2022).

While effective, previous approaches for improving statistical fidelity in image compression have primarily relied on discriminator designs from the image generation literature. Image generation models were designed to model global image distributions. For compression, the task is different in that the purpose of an adversarial discriminator is a much smaller projection from one image to another, typically from a blurry image to a sharpened image on the statistical manifold of the original natural images. This process is primarily governed by detail synthesis. For this reason, we opt to adapt the design of the adversarial training to emphasize this locality. At a high level, our proposal quantizes all possible images to local neighborhoods, thus aligning the discriminator modeling with its task in compression.

Our contributions are as follows:

1. We introduce a new adversarial discriminator based on VQ-VAE autoencoders (van den Oord et al., 2017; Razavi et al., 2019). Our new discriminator optimizes likelihood functions in the neighborhood of local images, which we call an "implicit local likelihood model" (ILLM). We combine our discriminator with the Mean-Scale Hyperprior (Minnen et al., 2018) neural compression architecture to create a new compressor that we call a Mean-Scale-ILLM (MS-ILLM).

2. We perform experiments with MS-ILLM over the CLIC2020, DIV2K, and Kodak datasets, where we demonstrate that we can surpass the statistical fidelity scores of HiFiC (as measured by FID) without sacrificing PSNR, see Figure 1.

3. We ablate our designs over latent dimensions for the VQ-VAE labeler and the U-Net discriminator to validate our architectural design choices.

2. Related work

Distortion and divergence metrics. From early on, the standard mean-squared error as a measure of distortion has been criticized for misaligning with human perception of distortion (Snyder, 1985). The (multiscale) structural similarity index measure (MS-SSIM) was proposed to fix this misalignment (Wang et al., 2003; 2004) by comparing the statistics of local image patches at multiple resolutions. Neural alternatives include metrics derived from comparing feature maps of deep networks. For example, the learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) is based on features from a VGG (Simonyan & Zisserman, 2014) or AlexNet (Krizhevsky et al., 2012) classifier. The perceptual information metric (PIM) is based on features from unsupervised training (Bhardwaj et al., 2020). Another class of measures includes no-reference measures such as NIQE (Mittal et al., 2012), FID (Heusel et al., 2017), and KID (Bińkowski et al., 2018). These consider the distributional alignment of the reconstructions and do not measure the distance to the reference data. More recently, approaches have been developed to learn distortion metrics indirectly via contrastive learning (Dubois et al., 2021). This work relates to other task-centric distortion metrics such as (Tishby et al., 1999; Alemi et al., 2017; Torfason et al., 2018; Singh et al., 2020; Matsubara et al., 2022).

Neural image compression methods. Optimizing for the standard handcrafted distortion metrics with neural networks can give promising rate-distortion performance (Ballé et al., 2018; Minnen et al., 2018; Cheng et al., 2020; El-Nouby et al., 2023; He et al., 2022). Optimizing for perceptual metrics is more difficult, as it results in significant compression artifacts (Ballé et al., 2018; Ledig et al., 2017; Mentzer et al., 2020; Ding et al., 2021). Thus, in practice often a weighted sum between the loss of a (conditional) GAN and a handcrafted metric such as MSE/MS-SSIM can provide perceptual benefits with stability (Mentzer et al., 2020; Agustsson et al., 2019). One can also use other divergences such as the Wasserstein distance (Tschannen et al., 2018) or KL-divergence of a deep latent variable model (Theis et al., 2022; Yang & Mandt, 2022; Ghouse et al., 2023).

Rate-distortion-perception tradeoff. The theoretical limits of the rate-perception-distortion trade-off were investigated by Blau & Michaeli (2018). A key finding is that at a given rate, improving the distortion comes at a cost of decreasing the perceptual quality of an image. Later studies have investigated the rate-distortion-perception trade-off, finding that realism generally comes at the expense of rate-distortion (Blau & Michaeli, 2019; Qian et al., 2022), which has also been demonstrated empirically (Theis et al., 2022; Yang & Mandt, 2022). Specifically, perfect realism can be achieved with at most two-fold increase in MSE (Yan et al., 2021; Blau & Michaeli, 2019).

Minimizing distributional divergence. One class of divergences that can be used for two-sample hypothesis testing are f-divergences, also known as Ali-Silvey divergences (Ali & Silvey, 1966) or Csiszar's ϕ-divergences (Csiszár, 1967). These divergences are connected to the problem of two-sample hypothesis testing because they represent an integrated Bayes risk through their relationship to the density ratio (Liese & Vajda, 2008). Nowozin et al. (2016) developed a class of generative models, so called f-GANs based on that insight, including the Jensen-Shannon Divergence. Other GAN architectures have been designed, such as c-GANs (Mirza & Osindero, 2014) that condition image generation on class labels. Corresponding models exist for when the labels are expanded to spatial semantic maps, such as OASIS (Sushko et al., 2022). Our work is related to OASIS, the primary differences are 1) they apply their model to the task of semantic image generation, whereas we are doing compression and 2) they use a pixel-space semantic map, whereas we use a latent-space semantic map based on a VQ-VAE (van den Oord et al., 2017; Razavi et al., 2019).

3. Background

In this section we review rate-distortion theory (Shannon, 1948), as well as more recent work on the rate-distortion-perception tradeoff (Blau & Michaeli, 2019). At the end of the section, we review how these theories can be applied to the design of neural codecs.

3.1. Notation

Throughout this work, we assume $(\Omega, \mathcal{F}, \mathbb{P})$ to be a probability space where Ω is the sample space, $\mathcal{F}$ is the event space, and $\mathbb{P}$ denotes the probability function such that $X : \Omega \to \mathcal{X}$ is a random variable (r.v.) defined on the space. Equivalently, $Y : \Omega \to \mathcal{Y}$. We will use capital letters for random variables, e.g. X; lower case letters for their realizations, e.g. $x \in \mathcal{X}$; P_X is a distribution of X; and p_X is the probability mass function of P_X. We will denote conditional distributions as $P_{X|Y}$, which we think of as a collection of probability measures on $\mathcal{X}$, for each value y there exists $P_{X|Y=y}$. Expectations will be denoted as $\mathbb{E}_{x\sim P_X}[q(x)]$, or abbreviated as $\mathbb{E}[q(x)]$.

3.2. Rate-distortion-perception theory

The goal of lossy compression is to store the outcomes $x \sim P_X$ of a discrete random variable X, e.g. natural images with as few bits (bit-rate) as possible while simultaneously ensuring that a reconstruction $\hat{x} \sim P_{\hat{X}|x}$ is of a certain quality level no lesser than τ. The problem has been formulated more precisely as rate-distortion theory (Shannon, 1948). Shannon concludes that the best bit-rate R is characterized by the rate-distortion function;

$$R(\tau) = \min_{P_{\hat{X}|X}} I(\hat{X}; X) \tag{1}$$
$$\text{s.t. } \mathbb{E}_{x,\hat{x}\sim P_X P_{\hat{X}|x}}[\rho(\hat{x}, x)] \leq \tau,$$

where $I(\cdot;\cdot)$ denotes the mutual information, and $\rho(\cdot,\cdot)$ is a distortion measure. Blau & Michaeli (2019) recently extended the aforementioned rate-distortion function to include an additional constraint that characterizes how well the statistics of the reconstructions resemble statistics of the real data distribution;

$$d(P_{\hat{X}}, P_X) \leq \sigma, \tag{2}$$

where $d(\cdot,\cdot)$ is some divergence between distributions, e.g. the Kulback-Leibler divergence (KLD). While the authors refer to this as "perception", for sake of clarity we refer to this metric as "statistical fidelity" to differentiate it from human perception, as well as metrics such as LPIPS (Zhang et al., 2018) that are considered "perceptual" in the literature. A key finding of Blau & Michaeli (2018) is that in many cases statistical fidelity comes at the expense of distortion, at constant rate.

Below, we show how to build a differentiable training objective by approximating and relaxing the constrained rate-distortion-perception function $R(\tau, \sigma)$.

3.3. Lossy codec optimization

For the purpose of this paper, we will constrain ourselves to lossy compression algorithms (codecs) of the following kind: We assume source symbols x will be encoded into a (quantized) latent representation $y = f(x)$, $f : \mathcal{X} \to \mathcal{Y}$. Subsequently, we employ an entropy coder[1] $g_\omega(y)$ with parameters ω to losslessly compress y into its shortest possible bit string. We write $r(x) := |g_\omega(y)|$ to denote the bit rate or length of the binary string generated by g_ω. Since sender and receiver have common knowledge of the entropy coder, the receiver can apply a decoder $h : \mathcal{Y} \to \mathcal{X}$ to recover the source signal $\hat{x} = h(y) = f \circ h(x)$. We will refer to the tuple of encoder, decoder and entropy coder as a lossy codec (f, g_ω, h). See Figure 2 for an overall system diagram.

The goal of optimization is to learn a parameterized lossy codec, $(f_\varphi, g_\omega, h_\upsilon)$, where φ, ω, and υ denote the parameters of each component. In our case, the encoder f_φ and the decoder h_υ are neural networks and the entropy coder g_ω will be defined by a parameterized approximation of the marginal over representations, in other words we need to learn $P_{Y|\omega}$. See Ballé et al. (2017) for details of the overall structure; we describe ours in Section 4.

[1] An entropy coder is a map that solves the lossless compression problem optimally. Please see Cover & Thomas (1991); MacKay (2003) for more details.

Using Lagrange multipliers to relax (1) and (2), the training objective comes out to be

$$\begin{aligned}\mathcal{L}(\varphi,\omega,\upsilon) =&\lambda_r\, \mathbb{E}_{x\sim P_X}\left[r_{\varphi,\omega}\left(x\right)\right] \\ &+\lambda_\rho\, \mathbb{E}_{x\sim P_X}\left[\rho\left(f_\varphi \circ h_\upsilon(x), x\right)\right] \\ &+\lambda_d\, d\left(P_{\hat{X}}, P_X\right).\end{aligned} \tag{3}$$

Before we give detailed descriptions of the functional forms of our lossy codec, we need to specify how we can approximate the distributional divergence in practice.

3.4. Approximating the distributional divergence

Mechanistically, it is typically straightforward to specify distortion functions for $\rho\left(f_\varphi \circ h_\upsilon(x), x\right)$, but minimizing the divergence term, $d\left(P_{\hat{X}}, P_X\right)$ can be move involved. A standard technique is to use GANs to optimize the symmetric Jensen-Shannon divergence (JSD) (Goodfellow et al., 2014; Nowozin et al., 2016). The JSD is a proper divergence measure between distributions, meaning that if there are enough training samples and the model class $P_{\hat{X}}$ is sufficiently rich, P_X can be accurately approximated. Goodfellow et al. (2014) show the JSD is minimized by the well-known GAN minimax optimization problem

$$\begin{aligned}\min_{\phi}\max_{\varphi,\omega,\upsilon}\ &\mathbb{E}_{x\sim P_X}\left[-\log D_\phi(x)\right] \\ &+\mathbb{E}_{\hat{x}\sim P_{\hat{X}}}\left[-\log\left(1-D_\phi(\hat{x})\right)\right],\end{aligned} \tag{4}$$

where D_ϕ is a parameterized discriminator function (with parameters ϕ) that estimates if a sample was drawn from the real data distribution and we have used the shorthand $\hat{x} = f_\varphi \circ h_\upsilon(x)$. In neural compression, we assume $P_{\hat{X}}$ to be the marginal over the joint $P_X P_{\hat{X}|x}$. For computing the empirical risk in (4), we draw different samples for x and $\hat{x}$. The sign-flipped generator-loss function from (4) is

$$\mathcal{L}_G(\varphi,\omega,\upsilon) = \mathbb{E}_{\hat{x}\sim P_{\hat{X}}}\left[\log\left(1-D_\phi(\hat{x})\right)\right], \tag{5}$$

which we can use as a drop-in replacement for $d(P_{\hat{X}}, P_X)$ in (3) (alternating minimization for the discriminator). This approach has been applied to several neural compression systems (Agustsson et al., 2019; Mentzer et al., 2020). Compared to GANs for image generation, the task for $P_{\hat{X}}$ is greatly simplified by the properties of the neural compression task. For this reason, we redesign the discriminator to reflect the locality of projection needed for compression.

4. Method

Our approach follows that of HiFiC (Mentzer et al., 2020), with the primary distinction being in a novel proposal for modeling local likelihoods in the neighborhood of the compressed image. We describe our vector-valued OASIS-type discriminator D_ϕ (Sushko et al., 2022), the most crucial adaptation as compared to other neural compression schemes, and point to relevant training specifics.

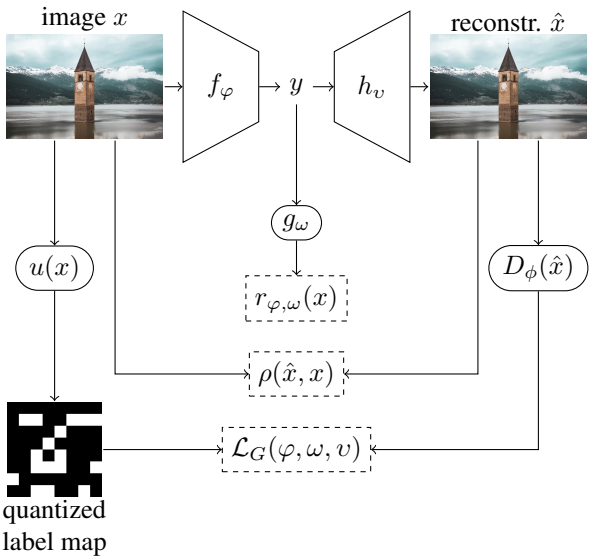


Figure 2. Overview of our learned lossy compression system with discriminator-based divergence minimization. An image x is encoded and quantized to latent y. A likelihood model g_ω enables entropy coding of y and an estimate of the rate $r_{\varphi,\omega}$. A decoder h_υ converts quantized y back into a compressed image $\hat{x}$. To improve statistical fidelity, we also train with a discriminator, D_ϕ, that attempts to match labels from a pretrained labeler u.

4.1. Autoencoder architecture

Here, we describe the encoder f_φ, decoder h_υ and the latent marginal $P_{Y|\omega}$ as used by the entropy coder, g_ω. Due to the relationship between the rate-distortion function and variational inference, a neural codec can be viewed as a type of variational autoencoder (Ballé et al., 2017). In line with previous work in neural compression, we model the data distribution using a two-level hierarchical autoencoder, also known as a hyperprior model (Ballé et al., 2018). This model class is named after the governing prior $P_{Y|\omega}$ that itself is modeled as a latent variable model $P_{Y|\omega} = \sum P_{Y|z;\omega} P_z$.

Previous work has used different choices for the prior distribution, such as a conditional Gaussian (Ballé et al., 2018; Minnen et al., 2018; Mentzer et al., 2020). We adopt the approach of Minnen et al. (2018) that conditions the means and scales of the Gaussian, and refer to it as the mean-scale hyperprior model. For the architecture of the encoder and decoder we follow Mentzer et al. (2020), who used larger and deeper models than Minnen et al. (2018).

4.2. Implicit local likelihood models

To develop our likelihood models, we assume that we have access to a labeling vector function, $u : \mathcal{X} \to \{0,1\}^{(C+1)\times W\times H}$. For any x in our dataset, u outputs a 3D spatially-distributed one-hot target vector map with

dimensions $(C+1) \times W \times H$, where C is the number of labels, W is the latent space width, and H is the latent space height. We reserve the zero-th label as a "fake" class to designate reconstructed images, and use the remaining C classes to label original images. We define b_0 as a one-hot target $(C+1) \times W \times H$ tensor where values are 1 for the zero-th element in the C dimension, effectively the "fake" class in standard GAN terminology.

Following Sushko et al. (2022), we can now define the two c-GAN-style adversarial loss functions:

$$\begin{aligned} \mathcal{L}_D(\phi) = \mathbb{E}_{x \sim P_X} \left[-\langle u(x), \log D_\phi(x) \rangle \right] \\ + \mathbb{E}_{\hat{x} \sim P_{\hat{X}}} \left[-\langle b_0, \log D_\phi(\hat{x}) \rangle \right], \end{aligned} \tag{6}$$

$$\mathcal{L}_G(\varphi, \omega, \upsilon) = \mathbb{E}_{\hat{x} \sim P_{\hat{X}}} \left[-\langle u(x), \log D_\phi(\hat{x}) \rangle \right], \tag{7}$$

where $D_\phi(x)$ is now a vector-valued function, and $\langle \cdot, \cdot \rangle$ denotes the inner product. Note that rather than just distinguishing original from reconstructed images, here the goal of the discriminator is to distinguish among the C image labels of original images, as well as detecting reconstructed images. The generator loss corresponds to the non-saturating GAN loss proposed by Goodfellow et al. (2014), extended to the multi-label case. It aims, for reconstructed images, to maximize the discriminator likelihood of the labels of the corresponding real image.

Effectively, this allows the discriminator to use more local information in the label. We note that HiFiC also includes an alternative form of locality in that it uses the latent as an extra input to the PatchGAN discriminator (Mentzer et al., 2020). However, in this case the discriminator still has the opportunity to ignore local information, because locality is not enforced in the loss function. Conversely, our approach enforces locality by including a latent code in the loss function and inputing this information via backpropagation. The distinction between forward-conditioning and loss-based conditioning is discussed in the OASIS paper (Sushko et al., 2022), and we refer readers there for further details.

4.3. Choice of labeling function

The choice of labeling function influences the success of the method. Equations (6) and (7) allow broad classes of labels, including global image labels when setting $W = H = 1$ and spatially-distributed labels for $H, W > 1$. Since our goal is to enforce locality in the implicit likelihood model, we opt to apply VQ-VAEs (van den Oord et al., 2017; Razavi et al., 2019). Using vector quantization, we partition the latent space of a VQ-VAE autoencoder into C clusters with cluster means $\{m_c\}_{c=1}^{C}$. For original images, we set

$$u^{(c,i,j)}(x) = \begin{cases} 1 & \text{if } c = \underset{q=1,\dots,C}{\arg\min} \left\| e^{(i,j)} - m_q \right\|_2^2, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $e^{(i,j)}$ is the channel vector at location (i, j) from the VQ encoder. The VQ-VAE encoder partitions the space of the latent vectors into C Voronoi cells. Since Voronoi cells are convex, any two points are path connected. Further, since our decoder architecture is a continuous mapping, the path connectedness remains even in reconstruction space $\mathcal{X}$. In other words, with the VQ-VAE approach, locality in the label space implies locality in image space. This contrasts with other labeling approaches, such as with ImageNet classes. Without a model for generating latent codes, our approach does not consider unconditional generation, although this could be achieved using autoregressive techniques (van den Oord et al., 2017; Razavi et al., 2019).

Our VQ-VAE architecture is based on the VQ-GAN variant (Esser et al., 2021) with a couple of modifications:

- We use ChannelNorm (Mentzer et al., 2020) instead of GroupNorm to improve normalization statistical stability across different image regions.
- We use XCiT (El-Nouby et al., 2021) for the attention layers to improve compute efficiency.

Given this architecture, the number of likelihood functions is governed by the spatial size of the latent space $(W \times H)$ and codebook size C of the VQ-VAE. A larger latent space and codebook will lead to smaller likelihood neighborhoods. Unless specified otherwise, we utilize a 32×32 latent space for images of size 256×256 and a codebook size of 1024. We did not observe a huge variation in results depending on these parameters (see ablations in Section 5.4).

4.4. Discriminator architecture

For the discriminator we use the U-Net architecture (Ronneberger et al., 2015) previously proposed by Sushko et al. (2022) in the context of semantic image synthesis. In their case, the discriminator aims to label pixels of real images with the corresponding label in the conditioning semantic segmentation map, while labeling pixels in generated images as "fake". In our case, rather than predicting manually annotated semantic classes, the discriminator predicts among the labels provided by the labeling function u.

Our U-Net variant uses LeakyReLU (Xu et al., 2015) for the activations and is built on top of residual blocks rather than the feed-forward blocks of the original (Ronneberger et al., 2015). Since our latent resolution is different from the image resolution, we cut the output path of the U-Net at the level of the 32×32 latent provided by $u(x)$. Contrary to the OASIS discriminator architecture, we do not use normalization for the convolutional layers. We considered several types of normalization, including spectral (Miyato et al., 2018) and instance normalization (Ulyanov et al., 2017), but we found that no normalization at all was most effective. We show ablations for the normalization layers in Section 5.4.

4.5. Training

We utilize the two-stage training process of HiFiC (Mentzer et al., 2020). In the first stage, we train the autoencoder (i.e., encoder f_φ, entropy coder g_ω, and decoder h_υ) without the discriminator for 1M steps. In pretraining the autoencoder we observed two phenomena. First, there were very large gradients at the start of training, leading to high variance. Second, in late training, the model requires very small gradient steps in order to optimize for the last few dB of PSNR. For these reasons, we depart from the learning rate schedule of HiFiC: we begin with linear warmup (Liu et al., 2020) for 10,000 steps and adopt a cosine learning rate decay for the rest of training. We set a peak learning rate of 3×10^{-4} and train using the AdamW optimizer (Loshchilov & Hutter, 2019) with a weight decay of 5×10^{-5}. We adopt the rate targeting strategy of HiFiC for six bitrates.

In addition, we pass the quantized latents to the decoder with backpropagation via the straight-through estimator (Theis et al., 2017) to ensure the decoder sees the same values at training and test-time. The models from this first training stage are used in all subsequent fine-tuning experiments. Our procedure for training the quantizing model $u(x)$ is almost identical, except for setting a smaller value of the MSE parameter in ρ (see Appendix), aligning features more with LPIPS (Zhang et al., 2018).

In the second stage we finetune the decoder, h_υ, part of the autoencoder using the full rate-distortion-perception loss of (2). We use a learning rate of 4×10^{-4} for the U-Net discriminator and 1×10^{-4} for the generator, which are the same values used in OASIS (Sushko et al., 2022). We tried tuning the discriminator and generator learning rates around these points, but we found that for higher bitrates in particular it was necessary for the discriminator to train faster than the generator, and the $(4 \times 10^{-4}, 1 \times 10^{-4})$ tuple represented a sweet spot for achieving this. For fine-tuning, we used the AdamW (Loshchilov & Hutter, 2019) optimizer with the same parametes of pretraining, except we lower the betas to (0.5, 0.9).

We refer to the overall model from this training process "MS-ILLM", for Mean & Scale Hyperprior fine-tuned with the Implicit Local Likelihood Model, to emphasize the locality of the discriminator vs. previous methods.

5. Experiments

In this section we validate our approach with numerical experiments. We consider metrics that act as surrogates for both reference distortion (i.e., $\rho\left(\hat{x}, x\right)$) and statistical fidelity (i.e., $d\left(P_{\hat{X}}, P_X\right)$) and demonstrate that our approach more efficiently optimizes for distortion and statistical fidelity simultaneously than previous methods.

5.1. Datasets and metrics

For training we utilize the train split of OpenImages V6 (Kuznetsova et al., 2020) for all models. We used the full-resolution versions of the images. The first augmentation transform is randomly selected to be either a random resized crop or a simple crop to a standard 256×256 training resolution. Our intuition is that this would expose the model to both interpolation statistics as well as raw quantization statistics from standard image codecs. The second augmentation consists of random horizontal flipping.

For evaluation we adopt the test split of CLIC2020 (Toderici et al., 2020), the validation split of DIV2K (Agustsson & Timofte, 2017), and Kodak.[2] We focus most of our results in the main body on CLIC2020 because it is commonly used by the neural image compression community. We present results for DIV2K and Kodak in the Appendix. Results for DIV2K and Kodak exhibited similar trends to CLIC2020.

Our evaluation metrics are of two general classes: reference-based and no-reference metrics. The reference metrics are computed in the form $\rho(\hat{x}, x)$, where x is a ground-truth image and $\hat{x}$ is a compressed version of x. The handcrafted reference metrics of MS-SSIM (Wang et al., 2003) and PSNR are standards for evaluating image compression methods. A drawback of optimization for the handcrafted metrics is that it can lead to blurring of the reconstructed images. For this reason, other reference metrics such as LPIPS (Zhang et al., 2018) and DISTS (Ding et al., 2020) have been developed that more heavily favor preservation of texture and are more correlated with human judgment (Ding et al., 2020), but it is important to note that as reference metrics they can still trade off some statistical fidelity (Blau & Michaeli, 2019).

Alternatively, no-reference metrics measure statistical fidelity via distributional alignment. No-reference metrics include FID (Heusel et al., 2017) and KID (Bińkowski et al., 2018). Both of these metrics use features from a pretrained Inception V3 model (Szegedy et al., 2016) to parametrize distributional alignment and are used for image generation.

5.2. Baseline models

We compare MS-ILLM to a suite of baseline models. We show details of how we calculated each baseline in the appendix. **No-GAN (Ours):** This method is our pretrained model at 1 million steps with no GAN fine-tuning. **MS-PatchGAN (HiFiC*):** This the No-GAN model with PatchGAN discriminator fine-tuning, essentially our own training recipe for HiFiC (Mentzer et al., 2020). Since we do not have access to the HiFiC training data, this model can be considered as similar to the original, but not an exact replica.

[2] Kodak PhotoCD dataset, http://r0k.us/graphics/kodak.

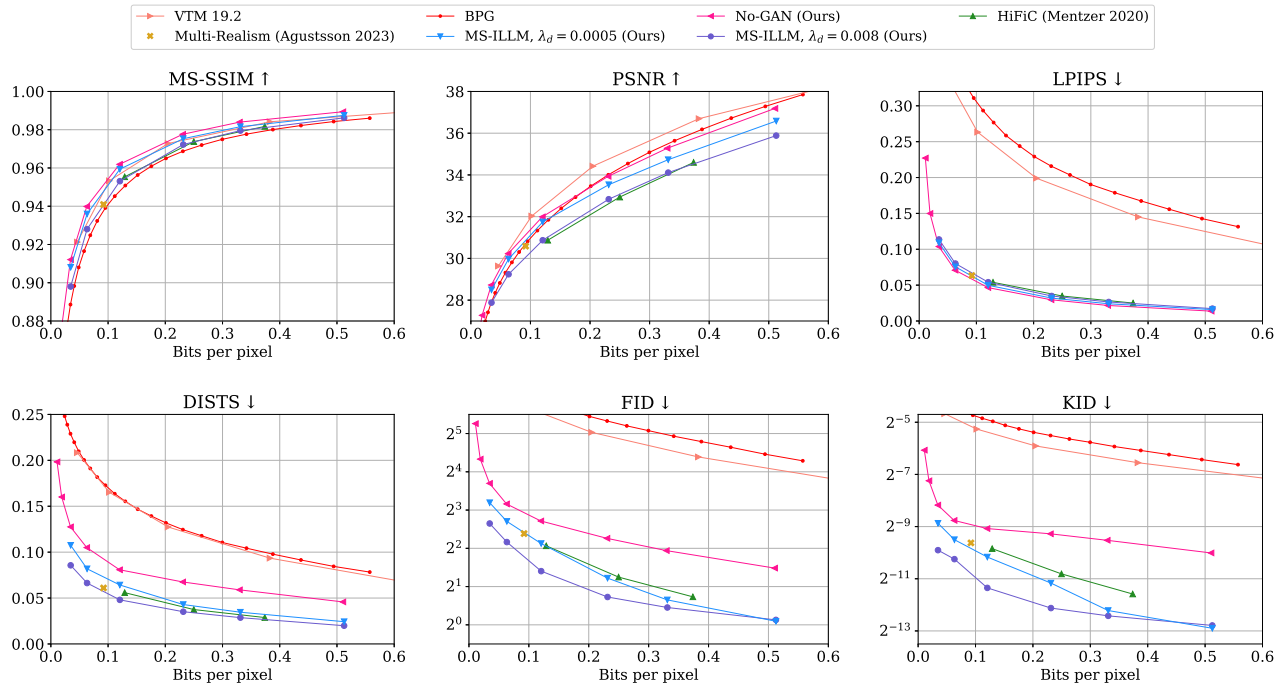


Figure 3. Comparisons of methods across various distortion and statistical fidelity metrics for the CLIC 2020 test set. Reference models (VTM 19.2, BPG, and No-GAN (Ours)) achieve the best MS-SSIM and PSNR scores, but display poor statistical fidelity as measured by FID and KID. GAN methods (MS-ILLM, HiFiC, and Multi-Realism) are able to trade some distortion for statistical fidelity. MS-ILLM (Ours) is able to achieve better statistical fidelity as measured by FID and KID vs. HiFiC at equivalent distortion levels.

VTM 19.2:[3] The image compression component of VVC, essentially the state-of-the-art handcrafted image compressor. **BPG:**[4] The image compression component of HEVC, adapted to work on images and give minimally-sized headers. The Mean-Scale Hyperprior autoencoder architecture that we use in our model has similar rate-distortion performance to BPG (Minnen et al., 2018). For this reason, we consider BPG to also be a stand-in for Mean-Scale Hyperprior performance. **HiFiC:** The (previous) state-of-the-art generative image compression method of Mentzer et al. (2020). **Multi-Realism:** A concurrent generative compression method utilizng the recent ELIC autoencoder (He et al., 2022) combined with PatchGAN, showing improved performance vs. HiFiC, recently published by Agustsson et al. (2023). For the multi-realism paper, the authors released a single operating point for CLIC2020, which we include.

To differentiate between in our own training pipeline and that of the original paper, in our plots we use the name "HiFiC" only when pulling data from the original paper, and for all other plots we use the name "MS-PatchGAN (HiFiC*)" for our own HiFiC training recipe.

[3] https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

[4] https://bellard.org/bpg/

5.3. Main results

Figure 3 shows performance of the models across bitrates. For this figure, we plot using the discriminator weighting λ_d in equation (3) such that MS-ILLM, $\lambda_d = 0.008$ matches the PSNR performance of HiFiC from the original paper. Figure 3 shows that while matching PSNR, MS-ILLM has uniformly better statistical fidelity than HiFiC as measured by FID and KID over all bitrates. Alternatively, with $\lambda_d = 0.0005$ we find that we can match HiFiC at a specified FID/KID rate while achieving a higher PSNR.

Figure 1 shows the same information from an alternative perspective, where in this case we utilize our own HiFiC training recipe (MS-PatchGAN (HIFiC*)) to evaluate at many rate-distortion-perception tradeoff points. Figure 1 shows that for all of the distortion points investigated (distortion in this case being measured by mean-squared error), our ILLM discriminator is able to achieve better statistical fidelity as measured by FID than PatchGAN.

In Figure 4 we compare the methods qualitatively in the setting of a cityscape image from CLIC2020 test and a custom pet image. For the cityscape image, at a bitrate of 0.177 bpp the BPG codec has substantial blurring, most easily observable in the trees. HiFiC is able to sharpen the image substantially, but this comes at the cost of introduc-

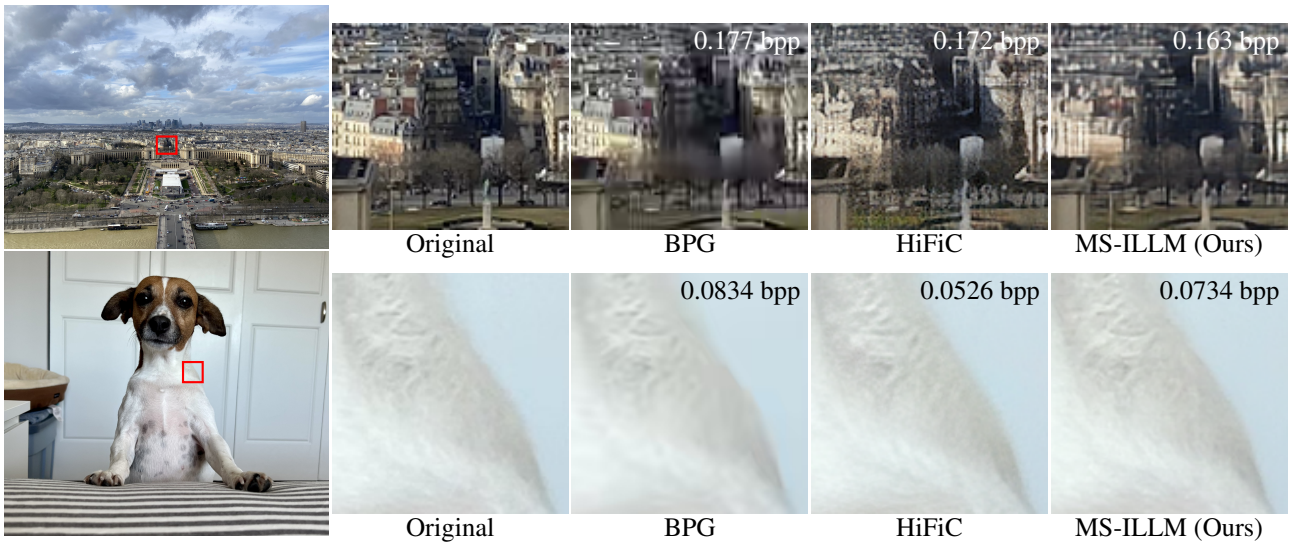


Figure 4. Qualitative examples of compressed images. (*top*) A cityscape image from the CLIC2020 test set. For the zoomed insets, BPG shows good quality for the white building in the foreground with quality degrading in the background. HiFiC provides more definition for the background at the cost of compression artifacts. Our method is able to provide the increased definition of HiFiC with fewer compression artifacts. (*bottom*) A custom example with a dog. In this case, BPG leads to some blurring of subtle fur features, as well as blocking compression artifacts. Both blurring and artifacts are removed with both the HiFiC method and ours.

Figure 5. An example with text. Incorporating adversarial models for both HiFiC and MS-ILLM leads to a degradation in textual quality vs. the No-GAN method that only uses reference-based loss functions. Compared to HiFiC, our discriminator is able to slightly improve legibility with the word, "centrum".

ing distortion artifacts. Meanwhile, MS-ILLM is able to achieve sharpening with less artifact introduction compared to HiFiC. For the pet image, again BPG leads to substantial blurring of subtle textures like fur, while both HiFiC and MS-ILLM are able to restore much of the missing texture. We show further qualitative examples with various sharpening levels on the Kodak dataset in the Appendix.

5.4. Ablations

In Figure 5 we consider a qualitative text example from CLIC2020, a difficult setting for generative compression models. In this case we can observe that the No-GAN method is the best at reconstructing text, while both HiFiC and MS-ILLM lead to text degradation. As was the case in Figure 4, MS-ILLM introduces fewer artifacts than HiFiC.

Table 1. Metrics for different normalization layers.

	None		Instance	
Bitrate	FID↓	PSNR↑	FID↓	PSNR↑
≈ 0.035	6.27	27.9	8.28	28.0
≈ 0.121	2.65	30.9	3.04	30.7
≈ 0.231	1.77	33.0	2.21	32.8

We performed ablations over the normalization layers in our U-Net discriminator. Normalizations tested included spectral reparametrization (Miyato et al., 2018), instance normalization (Ulyanov et al., 2017), and no normalization. Although spectral normalization was previously demonstrated to work best for PatchGAN in HiFiC (Mentzer et al., 2020) and was also used for OASIS (Sushko et al., 2022), for our setting we found that spectral norm trained extremely slow, taking as much as 25 times the number of gradient steps before it began matching the performance of the other methods. Surprisingly, we found that no normalization at all was ideal for our setting as demonstrated in Table 1. Despite not having normalizations, we did not observe major issues with stability in training.

In Table 2 we examine the effect of latent dimension for the labeling function u. We find that varying the spatial dimension of the latent space has a mild effect on the FID-PSNR tradeoff. The effect of codebook size is almost negligible.

Table 2. Metrics for different spatial latent size ($H \times W$) and codebook size C, for models with bpp ≈ 0.121.

SPATIAL SIZE	CODEBOOK SIZE	FID↓	PSNR↑
16×16	256	2.66	31.0
16×16	1024	2.65	31.1
32×32	256	2.57	30.9
32×32	1024	2.55	30.8

6. Limitations

The use of adversarial models and neural networks for compression could lead to bias of the output images based on demographic factors such as race or gender. As such, our results should be considered for research purposes only and not for production systems without further assessment over these factors. Second, our use of the HiFiC autoencoder requires substantial compute. Deployment in real-world settings will require miniaturization and heavy quantization of the model weights. Finally, although we showed improved performance with our method for statistical fidelity as measured by FID/KID, this does not guarantee that our approach would do better in terms of human preference over competing methods.

7. Conclusion

We developed a neural image compression model, called MS-ILLM, that improves statistical fidelity by using local adversarial discriminators. Unlike past discriminators employed in compression, our method emphasizes the locality necessary for the compression task. The benefits of this translates into better FID and KID metrics. As such, we empirically concur with the theory behind the rate-distortion-perception tradeoff for the task of image compression and move the current state of the art closer to the theoretical optimum of perfect statistical fidelity.

References

Agustsson, E. and Timofte, R. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In CVPRW, 2017.

Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., and Gool, L. V. Generative adversarial networks for extreme learned image compression. In ICCV, 2019.

Agustsson, E., Minnen, D., Toderici, G., and Mentzer, F. Multi-realism image compression with a conditional generator. In CVPR, 2023.

Ahmed, N., Natarajan, T., and Rao, K. Discrete cosine transform. IEEE Trans. Comput., C-23(1):90–93, 1974.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In ICLR, 2017.

Ali, S. M. and Silvey, S. D. A general class of coefficients of divergence of one distribution from another. J R Stat Soc Series B Stat Methodol, 28(1):131–142, 1966.

Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. Image coding using wavelet transform. IEEE TIP, 1(2): 205–220, 1992.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. In ICLR, 2017.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. In ICLR, 2018.

Ballé, J., Hwang, S. J., and Agustsson, E. TensorFlow Compression: Learned data compression, 2022. URL http://github.com/tensorflow/compression.

Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. arXiv preprint, arXiv:2011.03029, 2020.

Bellard, F. BPG image format. URL https://bellard.org/bpg/.

Bhardwaj, S., Fischer, I., Ballé, J., and Chinen, T. An unsupervised information-theoretic perceptual quality metric. In NeurIPS, 2020.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying MMD GANs. ICLR, 2018.

Blau, Y. and Michaeli, T. The perception-distortion tradeoff. In CVPR, 2018.

Blau, Y. and Michaeli, T. Rethinking lossy compression: The rate-distortion-perception tradeoff. In ICML, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. NeurIPS, 2020.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In CVPR, 2020.

Cover, T. and Thomas, J. Elements of information theory. John Wiley & Sons, 1991.

Csiszár, I. Information-type measures of difference of probability distributions and indirect observation. studia scientiarum Mathematicarum Hungarica, 2:229–318, 1967.

Ding, K., Ma, K., Wang, S., and Simoncelli, E. P. Image quality assessment: Unifying structure and texture similarity. IEEE TPAMI, 44(5):2567–2581, 2020.

Ding, K., Ma, K., Wang, S., and Simoncelli, E. P. Comparison of full-reference image quality models for optimization of image processing systems. IJCV, 129(4): 1258–1281, 2021.

Dubois, Y., Bloem-Reddy, B., Ullrich, K., and Maddison, C. J. Lossy compression for lossless prediction. In NeurIPS, 2021.

El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., and Jegou, H. XCiT: Cross-covariance image transformers. In NeurIPS, 2021.

El-Nouby, A., Muckley, M. J., Ullrich, K., Laptev, I., Verbeek, J., and Jégou, H. Image compression with product quantized masked image modeling. Trans. Mach. Learn. Res., 2023.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In CVPR, 2021.

Ghouse, N. F., Petersen, J., Wiggers, A., Xu, T., and Sautière, G. A residual diffusion model for high perceptual quality codec augmentation. arXiv preprint, arXiv:2301.05489, 2023.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. In NeurIPS, 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 CVPR, pp. 5718–5727, June 2022.

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. NeurIPS, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NeurIPS, 2012.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. IJCV, 128(7):1956–1981, 2020.

Kynkäänniemi, T., Karras, T., Aittala, M., Aila, T., and Lehtinen, J. The role of ImageNet classes in Fréchet inception distance. In ICLR, 2023.

Le Gall, D. and Tabatabai, A. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In ICASSP, 1988.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017.

Liese, F. and Vajda, I. f-divergences: Sufficiency, deficiency and testing of hypotheses. In Barnett, N. S. (ed.), Advances in Inequalities from Probability Theory & Statistics, pp. 113–173. Nova Publishers, 2008.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. In ICLR, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. ICLR, 2019.

MacKay, D. J. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.

Matsubara, Y., Yang, R., Levorato, M., and Mandt, S. Supervised compression for resource-constrained edge computing systems. In CVPR, 2022.

Mentzer, F., Toderici, G. D., Tschannen, M., and Agustsson, E. High-fidelity generative image compression. In NeurIPS, 2020.

Mentzer, F., Agustsson, E., Ballé, J., Minnen, D., Johnston, N., and Toderici, G. Neural video compression using GANs for detail synthesis and propagation. In ECCV, 2022.

Minnen, D., Ballé, J., and Toderici, G. D. Joint autoregressive and hierarchical priors for learned image compression. In NeurIPS, 2018.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. arXiv preprint, arXiv:1411.1784, 2014.

Mittal, A., Soundararajan, R., and Bovik, A. C. Making a "completely blind" image quality analyzer. IEEE Sign. Process. Letters, 20(3):209–212, 2012.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In ICLR, 2018.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. NeurIPS, 2016.

Pasco, R. C. Source coding algorithms for fast data compression. PhD thesis, Stanford University CA, 1976.

Qian, J., Zhang, G., Chen, J., and Khisti, A. A rate-distortion-perception theory for binary sources. In International Zurich Seminar on Information and Communication, pp. 34–38, 2022.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. In NeurIPS, 2019.

Rissanen, J. J. Generalized Kraft inequality and arithmetic coding. IBM J. Res. Dev., 20(3):198–203, 1976.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015.

Shannon, C. E. A mathematical theory of communication. Bell Syst. Tech. J., 27(3):379–423, 1948.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In ICLR, 2014.

Singh, S., Abu-El-Haija, S., Johnston, N., Ballé, J., Shrivastava, A., and Toderici, G. End-to-end learning of compressible features. In ICIP, 2020.

Snyder, H. L. Image quality: Measures and visual performance. In Flat-panel displays and CRTs, pp. 70–90. Springer, 1985.

Sushko, V., Schönfeld, E., Zhang, D., Gall, J., Schiele, B., and Khoreva, A. OASIS: Only adversarial supervision for semantic image synthesis. IJCV, 130(12):2903–2923, 2022.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the Inception architecture for computer vision. In CVPR, 2016.

Theis, L., Shi, W., Cunningham, A., and Huszár, F. Lossy image compression with compressive autoencoders. In ICLR, 2017.

Theis, L., Salimans, T., Hoffman, M. D., and Mentzer, F. Lossy compression with Gaussian diffusion. arXiv preprint, arXiv:2206.08889, 2022.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. In Annu. Allerton Conf. Commun. Control Comput., pp. 368–377, 1999.

Toderici, G., Theis, L., Johnston, N., Agustsson, E., Mentzer, F., Ballé, J., Shi, W., and Timofte, R. CLIC 2020: Challenge on learned image compression, 2020.

Torfason, R., Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., and Van Gool, L. Towards image understanding from deep compression without decoding. In ICLR, 2018.

Tschannen, M., Agustsson, E., and Lucic, M. Deep generative models for distribution-preserving lossy compression. In NeurIPS, 2018.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In CVPR, 2017.

van den Oord, A., Vinyals, O., and Kavukcuoglu, k. Neural discrete representation learning. In NeurIPS, 2017.

Wang, Z., Simoncelli, E., and Bovik, A. Multiscale structural similarity for image quality assessment. In ACSSC, volume 2, pp. 1398–1402, 2003.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. IEEE TIP, 13(4):600–612, 2004.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. arXiv preprint, arXiv:1505.00853, 2015.

Yan, Z., Wen, F., Ying, R., Ma, C., and Liu, P. On perceptual lossy compression: The cost of perceptual reconstruction and an optimal training framework. In ICML, 2021.

Yang, R. and Mandt, S. Lossy image compression with conditional diffusion models. arXiv preprint, arXiv:2209.06950, 2022.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR, 2018.

Improving Statistical Fidelity for Neural Image Compression with Implicit Local Likelihood Models — Supplementary Material

We will upload code for reproducing our results to the NeuralCompression repository at https://github.com/facebookresearch/NeuralCompression.

A. Further training details and hyperparameters

A.1. Autoencoder pretraining

For pretraining the autoencoder (i.e., f_φ, g_ω, and h_υ), we used the same overall approach as HiFiC (Mentzer et al., 2020). The original paper used adaptive rate targeting by oscillating the value of λ_ρ depending on the current empirical rate. We applied the same approach, where we used a looser λ_ρ in early training and increased it after 50,000 training steps. We trained models with eight different rate targets. The targets and corresponding values for $\lambda_{\rho,a}$ and $\lambda_{\rho,b}$ are listed in Table 3.

Table 3. Hyperparameters for autoencoder pretraining.

TARGET RATE	$\lambda_{\rho,a}$	$\lambda_{\rho,b}$
0.00875	2^5	2^{-4}
0.0175	2^4	2^{-4}
0.035	2^3	2^{-4}
0.07	2^2	2^{-4}
0.14	2^1	2^{-4}
0.30	2^0	2^{-4}
0.45	2^{-1}	2^{-4}
0.9	2^{-2}	2^{-4}

All models used 1 million steps for pretraining, with the target rate being 1.429 higher for the first 50,000 steps. Also, the value of $\lambda_{\rho,a}$ was held 50% lower for the first 50,000 steps.

Complete specification of the cost function also requires specifying $\rho(\hat{x}, x)$. We utilized a combination of MSE and (AlexNet-based) LPIPS. All of our images were normalized to the range [0.0, 1.0]. To bring this in line with the original HiFiC implementation, we used the following:

$$\rho(\hat{x}, x) = \lambda_{\text{MSE}} \left\|\hat{x} - x\right\|_2^2 + \text{LPIPS}_{\text{Alex}}\left(\hat{x}, x\right), \tag{9}$$

where $\lambda_{\text{MSE}} = 150$. We applied the same loss at the fine-tuning stage.

Our last modification for pretraining was to include variable learning rates. As mentioned in the main body, we observed large gradients at the beginning of training. At the end of training, very small steps were necessary to acquire the last few dB of PSNR. The standard approach is to step-decay the LR by a factor of 10 after a significant amount of training (e.g., 500,000 steps). Our training used a slightly different strategy. To increase the time the model spent at lower learning rates, we used a linear ramp for the first 10,000 steps followed by cosine rate decay. We observed this gave a small boost to PSNR on the CLIC 2020 test set as shown in Figure A1.

A.2. Labeler pretraining

For pretraining the label function, $u(x)$, we applied the same general approach as van den Oord et al. (2017). For this case the loss does not include a rate loss, but rather a VQ loss. To specify the loss, we can introduce the VQ-VAE parameters as γ for the encoder, ζ for the entropy coder (i.e., codebook), and ψ for the decoder. Then, the loss is

$$\mathcal{L}(\gamma, \zeta, \psi) = \lambda_\rho \, \mathbb{E}_{x \sim P_X}\left[\rho\left(f_\gamma \circ h_\psi(x), x\right)\right] + l_{\text{embedding},\gamma,\zeta}(x) + \beta l_{\text{commitment},\gamma,\zeta}(x), \tag{10}$$

where $l_{\text{embedding},\gamma,\zeta}(x)$ and $l_{\text{commitment},\gamma,\zeta}(x)$ are the commitment and embedding losses of the original paper. For $\rho(\hat{x}, x)$ we used the same equation as in (9), but with $\lambda_{\text{MSE}} = 1.0$ and a VGG (Simonyan & Zisserman, 2014) backbone instead of AlexNet.

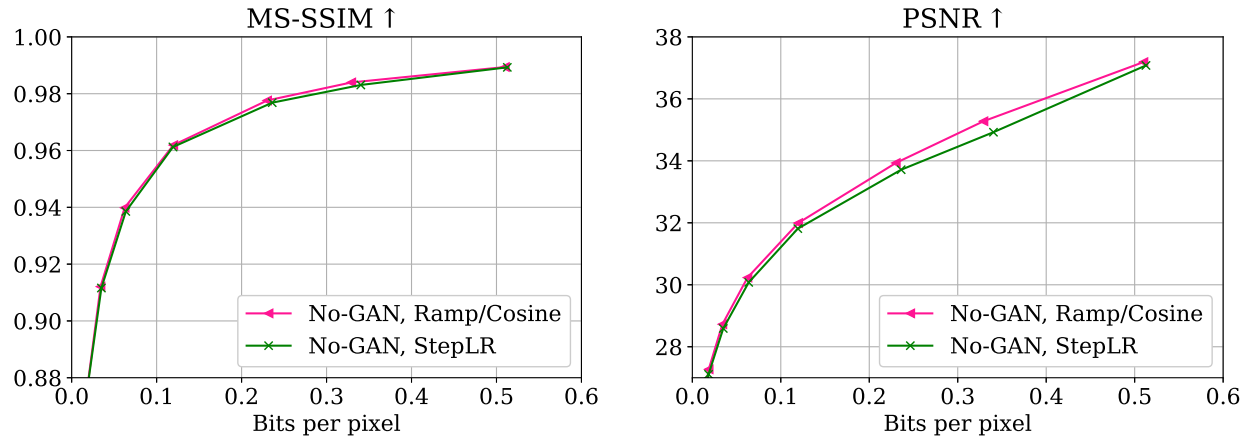


Figure A1. Comparison of learning rate strategies for training hyperprior models. Ramp/cosine applies a linear warmup for 10,000 steps, followed by cosine decay. The StepLR applies a learning rate decay of 0.1 after 500,000 steps.

We applied the same learning rate schedule as for the autoencoder pretraining over 1 million training steps. As the VQ model has a fixed rate, we did not utilize any rate targeting mechanisms.

Despite its role being restricted to labeling for the discriminator, we found that incorporation of perceptual losses in the training of the label function to be critical for its success. Empirically, we did not observe any good results without the inclusion of LPIPS in $\rho(\hat{x}, x)$ for training the VQ-VAE.

A.3. Fine-tuning with GAN loss

For discriminator fine-tuning, we froze both the encoder and the bottleneck of the autoencoder and only fine-tuned the decoder. For all methods, we utilized a learning rate of 0.0001 for the autoencoder. For the discriminator, we used a learning rate of 0.0001 for PatchGAN and 0.0004 for ILLM. For both discriminators, we set the Adam betas to 0.5 and 0.9, following OASIS (Sushko et al., 2022). We used the non-saturating crossentropy loss for all discriminator training.

B. Calculation of baseline metrics

For all methods, we first compressed images with the respective method on each dataset. Then we ran all methods through our own evaluation pipeline to ensure consistency of comparisons.

For VTM we installed VTM 19.2 from the reference at `https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM`. First, we converted the images to YCbCr colorspace using the `rgb2ycbcr` command from `compressai` (Bégaint et al., 2020). Then, we ran the following:

```
# Encode
$VTM_DIR/bin/EncoderAppStatic \
    -i $INPUT_YUV \
    -c $VTM_DIR/cfg/encoder_intra_vtm.cfg \
    -q $QUALITY \
    -o /dev/null \
    -b $COMPRESSED_FILE \
    -wdt $WIDTH \
    -hgt $HEIGHT \
    -fr 1 \
    -f 1 \
    --InputChromaFromat=444 \
    --InputBitDepth=8 \
```

```
    --ConformanceWindowMode=1

# Decode
$VTM_DIR/bin/DecoderAppStatic \
    -b $COMPRESSED_FILE \
    -o $OUTPUT_YUV \
    -d 8
```

For BPG, we used

```
# encode
bpgenc -q $QUALITY $INPUT_IMAGE -o $COMPRESSED_FILE -m 9 -f 444

# decode
bpgdec -o $OUTPUT_IMAGE $COMPRESSED_FILE
```

For HiFiC (Mentzer et al., 2020), we used the `tfci` command from Tensorflow Compression (Ballé et al., 2022).

For the multirealism results (Agustsson et al., 2023), we used the images uploaded at `https://storage.googleapis.com/multi-realism-paper/multi_realism_paper_supplement.zip`.

C. Note on metrics computation: bitrate, FID, and KID

In order to decode, our entropy coder g_ω and image decoder h_υ require both the bitstreams and metadata that specify the size of the image. To consider this for the bitrate, we wrote the bistreams and metadata to `pickle` files and measured the size of the file to estimate bitrates for all methods. For the hyperprior methods and HiFiC, we measured the size of the `tfci` file output by `tensorflow-compression` to estimate their rates. For the codec methods (BPG and VTM), we measured the size of the file to measure its rate. BPG and VTM are fully-featured and contain some extra metadata vs. the other methods, so this leads to a slight overestimate of the rate. However, this is a small cost overhead per image and does not majorly impact the results we present.

For the calculation of FID and KID, we used the `torch-fidelity` package (available from `https://github.com/toshas/torch-fidelity`) to calculate all metrics on our paper. Previous methods have used `tensorflow` (e.g., `tensorflow-gan`), but `torch-fidelity` has a few differences from `tensorflow-gan`:

1. `torch-fidelity` includes a standardized Inception V3 module (Szegedy et al., 2016) for feature extraction that includes Tensorflow 1.0-compatible resizing of the input images to a 299×299 resolution, whereas `tensorflow-gan` leaves the image resizing up to the user.
2. For the calculation of KID, `torch-fidelity` bootstraps the estimates by sampling subsets with replacement, whereas `tensorflow-gan` evenly divides the features into equal subsets with a maximum number of elements per subset.

After adjusting for these changes, we were able to match the two package implementations up to machine precision, but despite this we were unable to match the exact numbers from previous papers. For example, HiFiC (Mentzer et al., 2020) reports a 4.19736528 FID for HiFiC-Lo, whereas our implementation finds an FID of 4.18498420715332 for the released images. The differences for KID are even larger.

To provide a fair comparison, we opt instead to recalculate FID and KID for all baseline models in our work with the `torch-fidelity` package, meaning that in our plots we use the more optimistic FID for HiFiC of ≈ 4.185 rather than the pessimistic FID of ≈ 4.197. We also use the KID implementation of `torch-fidelity` to match those implemented by other groups in the PyTorch community.

D. Further analysis of the rate-distortion-perception tradeoff

Figure A2 shows rate-distortion curves over several weight settings λ_d for eq. (2) when training the generator. As the weight increases, statistical alignment (as measured by FID) between compressed and real images improves while distortion suffers,

confirming the theoretical results of Blau & Michaeli (2019).

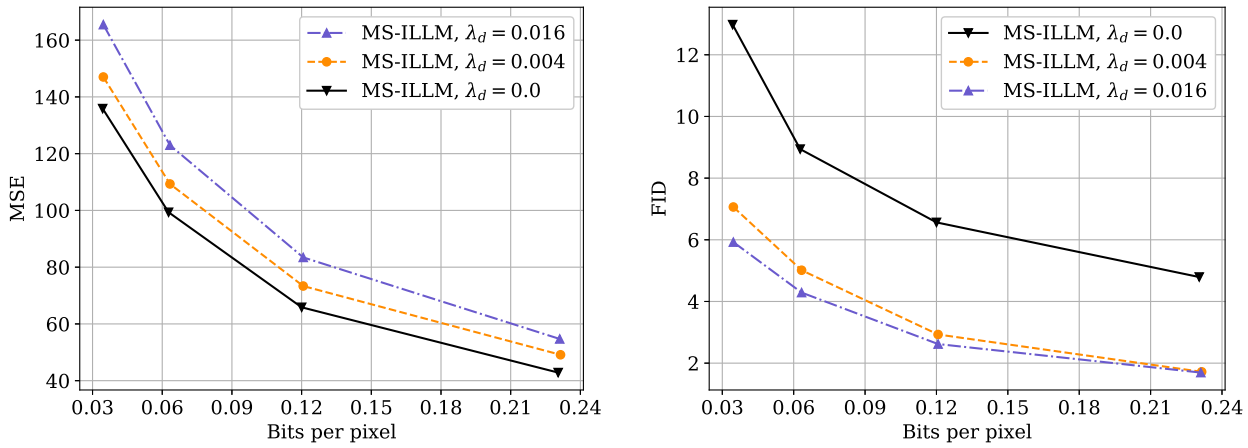


Figure A2. (*left*) Empirical validation of Figure 1 from (Blau & Michaeli, 2019) for the task of image compression. As the weight for statistical fidelity, λ_d is increased, we find that distortion (as measured by MSE) deteriorates, while statistical fidelity as measured by FID improves (*right*).

E. Additional qualitative examples

In Figure A3 we show the effect of more aggressive discriminator weighting using the ILLM discriminator vs. PatchGAN (as used in HiFiC). When the two methods are matched for PSNR, ILLM has a better FID (see Figure 3), and shows fewer artifacts. As we increase the discriminator weight, ILLM allows even more sharpening and detail addition without artifact reduction.

F. Experimental results on DIV2K and Kodak

We provide further experimental results on the DIV2K validation set (Agustsson & Timofte, 2017), shown in Figure A4. As with Figure 3 in the main body, MS-ILLM is able to match HiFiC on reference-based metrics (MS-SSIM, PSNR, LPIPS, DISTS) while outperforming HiFiC on no-reference distributional metrics (FID and KID). This further supports the claim that our discriminator is more efficient in trading distortion for statistical fidelity.

We note that for DIV2K we observed some instabilities in calculating KID. These were small enough that it was not an issue for CLIC2020 test or DIV2K at lower bitrates. However, at higher bitrates we observed that it was possible for our method to yield negative KID scores. For this reason, in Figure A4 we do not plot the last point of KID for our method.

In Figure A5 we show results on the Kodak dataset. For this case, there are too few patches to calculate distributional metrics like FID and KID. Nonetheless, the distortion metrics corroborate the performance of MS-ILLM compared other competing methods with MS-ILLM achieving lower distortion values than HiFiC at equivalent bitrates.

G. Investigation of ImageNet feature alignment

It has recently been demonstrated that there is a potential perceptual null space in FID due to the use of Inception V3 features in the calculation of the metric (Kynkäänniemi et al., 2023). Essentially, Kynkäänniemi et al. (2023) demonstrates that great improvements in FID scores can be gained by aligning images with ImageNet features without improving their perceptual quality. Kynkäänniemi et al. (2023) also demonstrates that FID calculation with other feature extracting backbones (such as those from SSL models like SwAV (Caron et al., 2020) are not as susceptible to this effect.

This could be a problem for our method, as we use VGG for training the labeling function $u(x)$. This could mean that the improvement mechanism for our results might be based on ImageNet class alignment rather than improvements in image quality. For this reason, we recalculated FID scores on CLIC2020 and DIV2K using a SwAV ResNet50 background, with

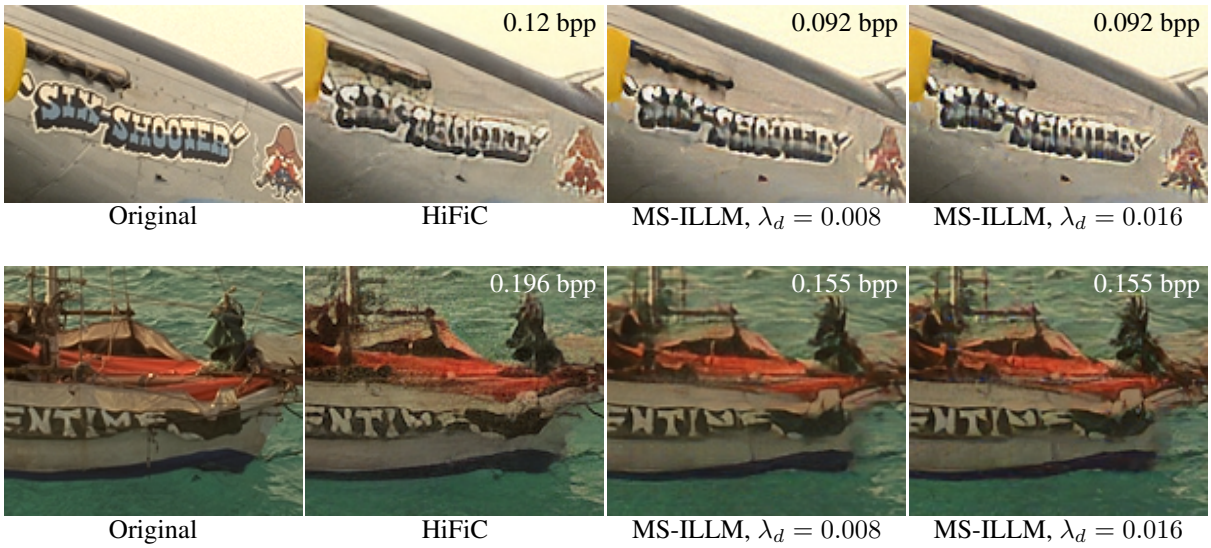


Figure A3. Qualitative examples with MS-ILLM compared to HiFiC on images from the Kodak dataset. The $\lambda_d = 0.008$ result shows MS-ILLM with approximately the same PSNR as HiFiC as computed on the CLIC 2020 test dataset. This model is operating at a lower FID than HiFiC, and displays fewer artifacts in the examples. By increasing λ_d MS-ILLM can more gracefully add details and textures to the image without increasing artifacts compared to the PatchGAN method of HiFiC.

results in Figure A6. Figure A6 demonstrates our results are still upheld when using the SwAV FID extractor, indicating that our metrics improvements arise from effects beyond simple ImageNet class alignment.

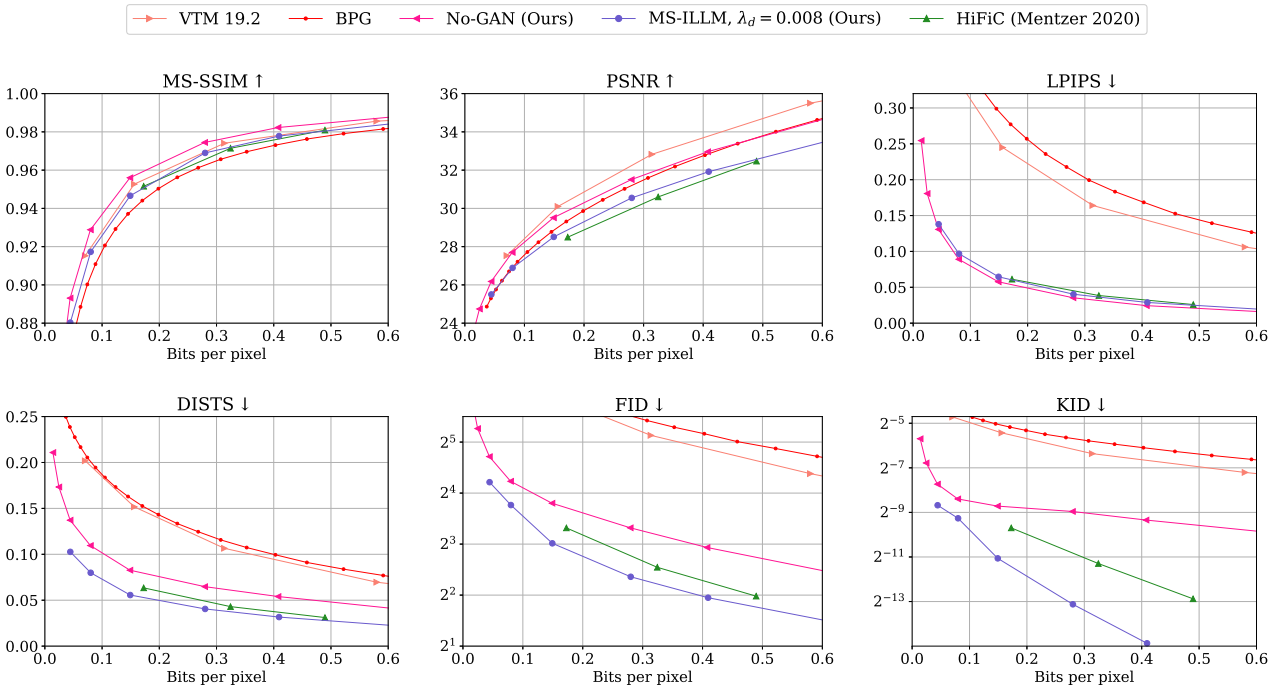


Figure A4. Comparisons of methods across various distortion and statistical fidelity metrics for the DIV2K validation set. As with Figure 3 in the main body, MS-ILLM is able to match HiFiC in reference metrics (MS-SSIM, PSNR, LPIPS, and DISTS) while outperforming HIFiC in no-reference metrics (FID and KID) that indicate statistical fidelity.

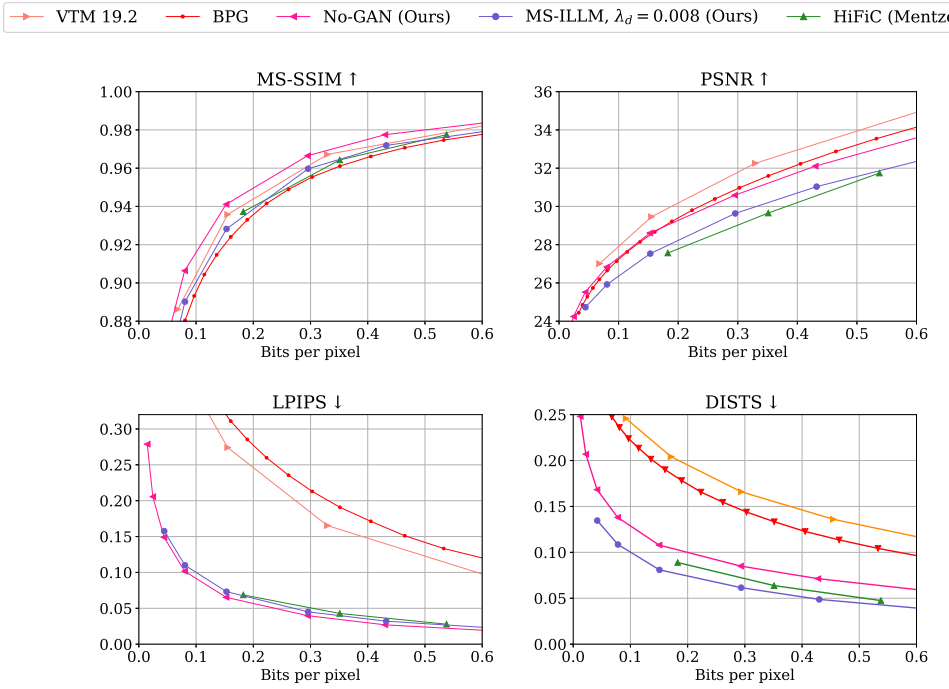
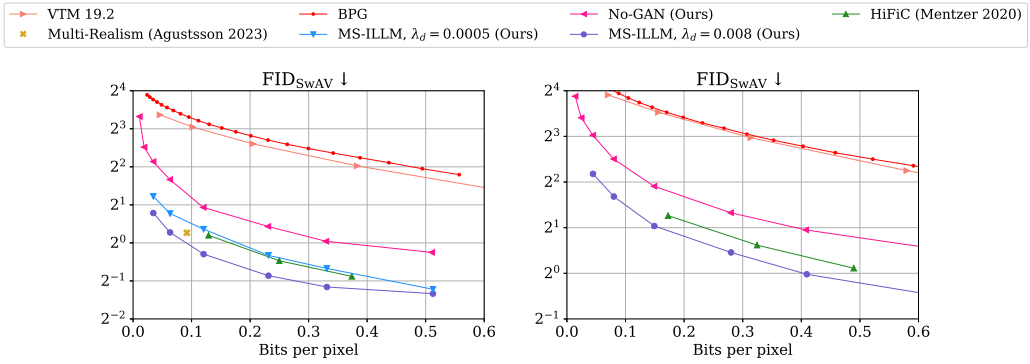


Figure A5. Plots of metrics for different bitrates on the Kodak dataset. In this case, we are unable to calculate FID or KID as the Kodak dataset has too few images (24) to yield useful metrics. Nonetheless, we can observe that MS-ILLM achieves similar distortion values to HiFiC across the different bitrates.



(a) CLIC2020 test (b) DIV2K validation

Figure A6. Rate-fidelity plots with claculation of FID using a ResNet50 trained via the SwAV self-supervised method (Caron et al., 2020). Similarly to Figures 3 and A4, MS-ILLM acquires higher statistical fidelity at all bitrates than the competing methods. This indicates that the improvements of MS-ILLM arise from effects beyond ImageNet class alignment.