High-Fidelity Generative Image Compression

Fabian Mentzer[*] ETH Zürich

George Toderici Google Research

Michael Tschannen[†]

Eirikur Agustsson Google Research

Abstract

We extensively study how to combine Generative Adversarial Networks and learned compression to obtain a state-of-the-art generative lossy compression system. In particular, we investigate normalization layers, generator and discriminator architectures, training strategies, as well as perceptual losses. In contrast to previous work, i) we obtain visually pleasing reconstructions that are perceptually similar to the input, ii) we operate in a broad range of bitrates, and iii) our approach can be applied to high-resolution images. We bridge the gap between rate-distortion-perception theory and practice by evaluating our approach both quantitatively with various perceptual metrics, and with a user study. The study shows that our method is preferred to previous approaches even if they use more than $2\times$ the bitrate.

1 Introduction

The ever increasing availability of cameras produces an endless stream of images. To store them efficiently, lossy image compression algorithms are used in many applications. Instead of storing the raw RGB data, a lossy version of the image is stored, with—hopefully—minimal visual changes to the original. Various algorithms have been proposed over the years [50, 43, 54], including using state-of-the-art video compression algorithms for single image compression (BPG [7]). At the same time, deep learning-based lossy compression has seen great interest [45, 5, 31], where a neural network is directly optimized for the rate-distortion trade-off, which led to new state-of-the-art methods.

However, all of these approaches degrade images significantly as the compression factor increases. While classical algorithms start to exhibit algorithm-specific artifacts such as blocking or banding, learning-based approaches reveal issues with the distortion metric that was used to train the networks. Despite the development of a large variety of "perceptual metrics" (*e.g.*, (Multi-Scale) Structural Similarity Index ((MS-)SSIM) [53, 52], Learned Perceptual Image Patch Similarity (LPIPS) [57]), the weakness of each metric is exploited by the learning algorithm, *e.g.*, checkerboard artifacts may appear when targeting neural network derived metrics, relying on MS-SSIM can cause poor text reconstructions, and MSE yields blurry reconstructions.

In [3], Agustsson *et al.* demonstrated the potential of using GANs to prevent compression artifacts with a compression model that produces perceptually convincing reconstructions for extremely low bitrates (<0.08 bpp). However, their reconstructions tend to only preserve high-level semantics, deviating significantly from the input.

Recent work by Blau and Michaeli [9] characterized this phenomenon by showing the existence of a triple "rate-distortion-perception" trade-off, formalizing "distortion" as a similarity metric comparing pairs of images, and "perceptual quality" as the distance between the image distribution p_X and the distribution of the reconstructions $p_{\hat{X}}$ produced by the decoder, measured as a distribution divergence. They show that at a fixed rate, better perceptual quality always implies worse distortion. Conversely, only minimizing distortion will yield poor perceptual quality. To overcome this issue, distortion can

[*] Work done while interning at Google Research.

[†] Work done while at Google Research.

Project page and demo: hific.github.io

Correspondence: eirikur@google.com

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

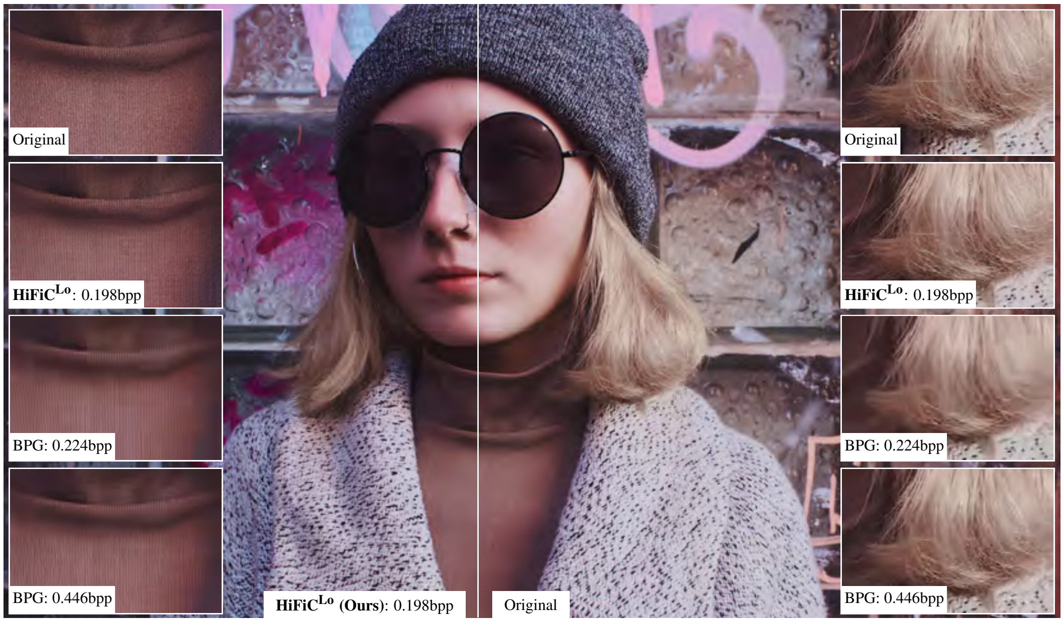

Original
$\textbf{HiFiC}^{\textbf{Lo}}$: 0.198bpp
BPG: 0.224bpp
BPG: 0.446bpp
$\textbf{HiFiC}^{\textbf{Lo}}$ **(Ours)**: 0.198bpp
Original
Original
$\textbf{HiFiC}^{\textbf{Lo}}$: 0.198bpp
BPG: 0.224bpp
BPG: 0.446bpp

Figure 1: Comparing our method, *HiFiC*, to the original, as well as BPG at a similar bitrate and at 2× the bitrate. We can see that our GAN model produces a high-fidelity reconstruction that is very close to the input, while BPG exhibits blocking artifacts, that are still present at double the bitrate. In the background, we show a split to the original to further indicate how close our reconstruction is. We show many more visual comparisons in Appendix B, including more methods, more bitrates, and various datasets. *Best viewed on screen.*

be traded for better perceptual quality by minimizing the mismatch between the distributions of the input and the reconstruction, *e.g.*, with Generative Adversarial Networks (GANs) [17]. While [9] presents comprehensive theoretical results, the rate-distortion-perception trade-off is only explored empirically on toy datasets.

In this paper, we address these issues with the following contributions:

1. We propose a generative compression method to achieve high quality reconstructions that are very close to the input, for high-resolution images (we test up to 2000×2000 px). In a user study, we show that our approach is visually preferred to previous approaches *even when these approaches use more than* $2\times$ *higher bitrates*, see Fig. 3.
2. We quantitatively evaluate our approach with FID [18], KID [8], NIQE [35], LPIPS [57], and the classical distortion metrics PSNR, MS-SSIM, and show how our results are consistent with the rate-distortion-perception theory. We also show that no metric would have predicted the full ranking of the user study, but that FID and KID are useful in guiding exploration. Considering this ensemble of diverse perceptual metrics including no-reference metrics, pair-wise similarities, and distributional similarities, as well as deep feature-based metrics derived from different network architectures, ensures a robust perceptual evaluation.
3. We extensively study the proposed architecture and its components, including normalization layers, generator and discriminator architectures, training strategies, as well as the loss, in terms of perceptual metrics and stability.

2 Related work

The most frequently used lossy compression algorithm is JPEG [50]. Various hand-crafted algorithms have been proposed to replace JPEG, including WebP [54] and JPEG2000 [43]. Relying on the video codec HEVC [42], BPG [7] achieves very high PSNR across varying bitrates. Neural compression approaches directly optimize Shannon's rate-distortion trade-off [14]. Initial works relied on RNNs [45, 47], while subsequent works were based on auto-encoders [5, 44, 1]. To decrease the required bitrate, various approaches have been used to more accurately model the probability density of the auto-encoder latents for more efficient arithmetic coding, using hierarchical priors, auto-regression with various context shapes, or a combination thereof [6, 31, 28, 39, 32, 26, 33]. State-of-the-art models now outperform BPG in PSNR, *e.g.*, the work by Minnen *et al.* [32].

Since their introduction by Goodfellow *et al.* [17], GANs have led to rapid progress in unconditional and conditional image generation. State-of-the-art GANs can now generate photo-realistic images at high resolution [10, 21, 38]. Important drivers for this progress were increased scale of training data and model size [10], innovation in network architectures [21], and new normalization techniques to stabilize training [36]. Beyond (un)conditional generation, adversarial losses led to advances in different image enhancement tasks, such as compression artifact removal [15], image de-noising [11], and image super-resolution [25]. Furthermore, adversarial losses were previously employed to improve the visual quality of neural compression systems [39, 40, 48, 3, 9]. [39] uses an adversarial loss as a component in their full-resolution compression system, but they do not systematically ablate and asses the benefits of this loss on the quality of their reconstructions. While [40] provides a proof-of-concept implementation of a low-resolution compression system with a GAN discriminator as decoder, [48, 9] focus on incorporating a GAN loss into the rate-distortion objective in a conceptually sound fashion. Specifically, [48] proposes to augment the rate-distortion objective with a distribution constraint to ensure that the distribution of the reconstructions match the input distribution at all rates, and [9] introduce and study a triple trade-off between rate, distortion, and distribution matching. Finally, [3] demonstrates that using GAN-based compression systems at very low bitrates can lead to bitrate savings of 2× over state-of-the-art engineered and learned compression algorithms.

3 Method

3.1 Background

Conditional GANs Conditional GANs [17, 34] are a way to learn a generative model of a conditional distribution $p_{X|S}$, where each datapoint x is associated with additional information s (*e.g.*, class labels or a semantic map) and x, s are related through an unknown joint distribution $p_{X,S}$. Similar to standard GANs, in this setting we train two rivaling networks: a generator G that is conditioned on s to map samples y from a fixed known distribution p_Y to $p_{X|S}$, and a discriminator D that maps an input (x, s) to the probability that it is a sample from $p_{X|S}$ rather than from the output of G. The goal is for G to "fool" D into believing its samples are real, *i.e.*, from $p_{X|S}$. Fixing s, we can optimize the "non-saturating" loss [17]:

$$\begin{aligned}\mathcal{L}_G &= \mathbb{E}_{y\sim p_Y}[-\log(D(G(y,s),s)],\\ \mathcal{L}_D &= \mathbb{E}_{y\sim p_Y}[-\log(1-D(G(y,s),s)] + \mathbb{E}_{x\sim p_{X|s}}[-\log(D(x,s))].\end{aligned} \tag{1}$$

Neural Image Compression Learned lossy compression is based on Shannon's rate-distortion theory [14]. The problem is usually modeled with an auto-encoder consisting of an encoder E and a decoder G. To encode an image x, we obtain a quantized latent $y = E(x)$. To decode, we use G to obtain the lossy reconstruction $x' = G(y)$. The compression incurs a distortion $d(x, x')$, *e.g.*, $d =$ MSE (mean squared error). Storing y is achieved by introducing a probability model P of y. Using P and an entropy coding algorithm (*e.g.*, arithmetic coding [30]), we can store y losslessly using bitrate $r(y) = -\log(P(y))$ (in practice there is a negligible bit-overhead incurred by the entropy coder). If we parameterize E, G, and P as CNNs, we can train them jointly by minimizing the *rate-distortion trade-off*, where λ controls the trade-off:

$$\mathcal{L}_{EG} = \mathbb{E}_{x\sim p_X}[\lambda\, r(y) + d(x,x')]. \tag{2}$$

3.2 Formulation and Optimization

We augment the neural image compression formulation with a conditional GAN, *i.e.*, we merge Eqs. 1, 2 and learn networks E, G, P, D. We use $y = E(x)$, and $s = y$. Additionally, we use the "perceptual distortion" $d_P =$ LPIPS, inspired by [51], who showed that using a VGG [41]-based loss helps training. In the formalism of [9], d_P is a distortion (as it is applied point-wise), thus we group it together with MSE to form our distortion $d = k_M\text{MSE} + k_P d_P$, where k_M, k_P are hyper-parameters. Using hyper-parameters λ, β to control the trade-off between the terms, we obtain:

$$\mathcal{L}_{EGP} = \mathbb{E}_{x\sim p_X}[\lambda r(y) + d(x,x') - \beta\log(D(x',y))], \tag{3}$$

$$\mathcal{L}_D = \mathbb{E}_{x\sim p_X}[-\log(1-D(x',y))] + \mathbb{E}_{x\sim p_X}[-\log(D(x,y))]. \tag{4}$$

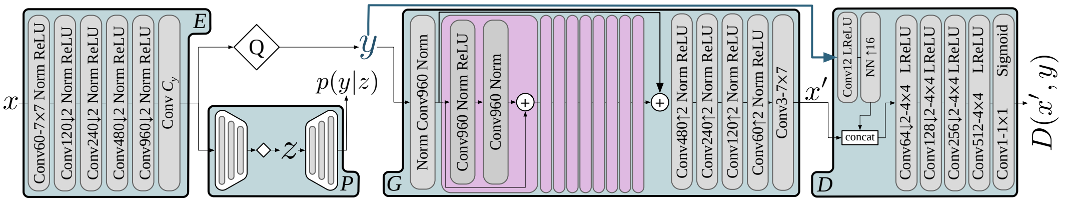


Figure 2: Our architecture. *ConvC* is a convolution with C channels, with 3×3 filters, except when denoted otherwise. $\downarrow 2$, $\uparrow 2$ indicate strided down or up convolutions. *Norm* is ChannelNorm (see text), *LReLU* the leaky ReLU [56] with α=0.2, *NN*↑16 nearest neighbor upsampling, Q quantization.

Constrained Rate When training a neural compression model w.r.t. the loss in Eq. 2, there is only a single term, $d(x, x')$, that is at odds with the rate term $r(y)$. The final (average) bitrate of the model can thus be controlled by varying only λ. In our setting however, MSE, d_P, and $-\log(D(x'))$ are at odds with the rate. For a fixed λ, different k_M, k_P, β would thus result in models with different bitrates, making comparison hard. To alleviate this, we introduce a "rate target" hyper-parameter r_t, and replace λ in Eq. 3 with an *adaptive* term λ' that uses the two hyper parameters $\lambda^{(a)}, \lambda^{(b)}$, where $\lambda' = \lambda^{(a)}$ if $r(y) > r_t$, and $\lambda' = \lambda^{(b)}$ otherwise. Setting $\lambda^{(a)} \gg \lambda^{(b)}$ allows us to learn a model with an average bitrate close to r_t.

3.3 Architecture

We show our architecture in Fig. 2, including the encoder E, generator G, discriminator D and probability model P. For P, we use the hyper-prior model proposed in [6], where we extract side information z to model the distribution of y and simulate quantization with uniform noise $\mathcal{U}(-1/2, 1/2)$ in the hyper-encoder and when estimating $p(y|z)$. However, when feeding y to G we use rounding instead of noise (*i.e.*, the straight-through estimator as in [44]), which ensures that G sees the same quantization noise during training and inference. E, G and D are based on [51, 3], with some key differences in the discriminator and in the normalization layers, described next.

Both [51, 3] use a multi-scale patch-discriminator D, while we only use a single scale, and we replace InstaceNorm [49] with SpectralNorm [36]. Importantly, and in contrast to [3], we condition D on y by concatenating an upscaled version to the image, as shown in Fig. 2. This is motivated from using a conditional GAN formulation, where D has access to the the conditioning information (which for us is y, see Section 3.2).

In [51] InstanceNorm is also used in E, G. In our experiments, we found that this introduces significant darkening artifacts when employing the model on images with a different resolution than the training crop size (see Section 5.3). We assume this is due to the spatial averaging of InstanceNorm, and note that [38] also saw issues with InstanceNorm. To alleviate these issues, we introduce **ChannelNorm**, which normalizes over channels. The input is a batch of $C \times H \times W$-dimensional feature maps f_{chw}, which are normalized to

$$f'_{chw} = \frac{f_{chw} - \mu_{hw}}{\sigma_{hw}} \alpha_c + \beta_c, \quad \text{where} \quad \begin{aligned} \mu_{hw} &= {}^1\!/_C \textstyle\sum_{c=1}^C f_{chw} \\ \sigma^2_{hw} &= {}^1\!/_C \textstyle\sum_{c=1}^C (f_{chw} - \mu_{hw})^2, \end{aligned} \tag{5}$$

using learned per-channel offsets α_c, β_c. We note that ChannelNorm has similarities to other normalizations. Most closely related, Positional Normalization [27] also normalizes over channels, but does not use learned α_c, β_c. Instead, they propose to use μ, σ from earlier layers as offsets for later layers, motivated by the idea that these μ, σ may contain useful information. Further, Kerras *et al.* [20] normalize each feature vector "pixelwise" to unit length, without centering ($\mu = 0$). We show a visualization and more details in Appendix A.4.

3.4 User Study

To visually validate our results, we set up a user study as two-alternative forced choice (2AFC) on N_I=20 random images of the CLIC2020 [46] dataset (datasets are described below), inpsired by the methodology used in [46]. Participants see crops of these images, which are determined as follows: when an image is first shown, a random 768×768 crop is selected. The participants are asked to request a new random crop until they find "an interesting crop" (*i.e.*, a region that is not completely flat or featureless), and then they use this crop for all comparisons on that image. We chose the crop size such that it is as large as possible while still allwoing two crops to fit side-by-side on all screens, downsampling would cause unwanted biases. The crops selected by users are linked in Appendix A.5.

At any time, the participants are comparing a pair of methods A and B. On screen, they always see two crops, the left is either method A or B, the right is always the original (Fig. A9 shows a screenshot). Participants are asked to toggle between methods to "select the method that looks closer to the original". This ensures the participants take into account the faithfulness of the reconstruction to the original image. We select the pairs of methods to compare by performing a binary search against the ranking of previously rated images, which reduces the number of comparisons per participant.

To obtain a ranking of methods, we use the Elo [16] rating system, widely used to rank chess players. We view each pair-wise comparison as a game between A and B, and all comparisons define a tournament. Running Elo in this tournament (using Elo parameters $k = 30$ and $N = 400$), yields a score per method. Since Elo is sensitive to game order, we run a Monte Carlo simulation, shuffling the order of comparisons N_E=10 000 times. We report the median score over the N_E simulations for each method, giving us an accurate estimate of the skill rating of each method and confidence intervals. We validated that the ordering is consistent with other methods for obtaining total orderings from pairwise comparisons that do not allow the computation of confidence intervals [37, 12].

4 Experiments

Datasets Our training set consists of a large set of high-resolution images collected from the Internet, which we downscale to a random size between 500 and 1000 pixels, and then randomly crop to 256×256. We evaluate on three diverse benchmark datasets collected independently of our training set to demonstrate that our method generalizes beyond the training distribution: the widely used *Kodak* [23] dataset (24 images), as well as the *CLIC2020* [46] testset (428 images), and *DIV2K* [2] validation set (100 images). We note that we ignore color profiles of CLIC images. The latter two mostly contain high-resolution images with shorter dimension greater than 1300px (see Appendix A.9 for more statistics). We emphasize that we do not adapt our models in any way to evaluate on these datasets, and that we evaluate on the full resolution images.

Metrics We evaluate our models and the baselines in PSNR as well as the perceptual distortions LPIPS [57] and MS-SSIM, and we use NIQE [35], FID [18], and KID [8] as perceptual quality indices. MS-SSIM is the most widely used perceptual metric to asses (and train) neural compression systems. LPIPS measures the distance in the feature space of a deep neural network originally trained for image classification, but adapted for predicting the similarity of distorted patches, which is validated to predict human scores for these distortions [57]. Three network architectures were adapted in [57], we use the variant based on AlexNet. NIQE is a no-reference metric, also based on assessing how strongly the statistics of a distorted image deviate from the statistics of unmodified natural images. In contrast to FID, non-learned features are used, and the metric entails focusing on patches selected with a "local sharpness measure".

FID is a widely used metrics to asses sample quality and diversity in the context of image generation, in particular for GANs. Unlike PSNR, MS-SSIM, and LPIPS, which measure the similarity between individual *pairs* of images, FID assesses the similarity between the *distribution* of the reference images and the distribution of the generated/distorted images. This similarity is measured in the feature space of an Inception network trained for image classification, by fitting Gaussians to the features and measuring a Wasserstein distance betweeen the Gaussians of the references and the Gaussians of the generated images. Heusel *et al.* [18] demonstrate that FID is consistent with increasing distortion and human perception. Furthermore, it was shown to detect common GAN failure modes such as mode dropping and mode collapse [29]. KID is similar, but unlike FID, is unbiased and does not make any assumptions about the distributions in the features space.

As the distortion loss ensures global consistency of the image, and due to the large variation of resolutions in our test sets, we calculate FID and KID on patches rather than on the full images, covering the image (details in Appendix A.7). We use a patch size of 256 pixels, which yields 28 650 patches for CLIC2020 and 6 573 patches for DIV2K. We do not report FID or KID for Kodak, as the 24 images only yield 192 patches.

Model Variants and Baselines We call our main model ***High Fidelity Compression (HiFiC)***. To see the effect of the GAN, we train a baseline with the same architecture and same distortion loss

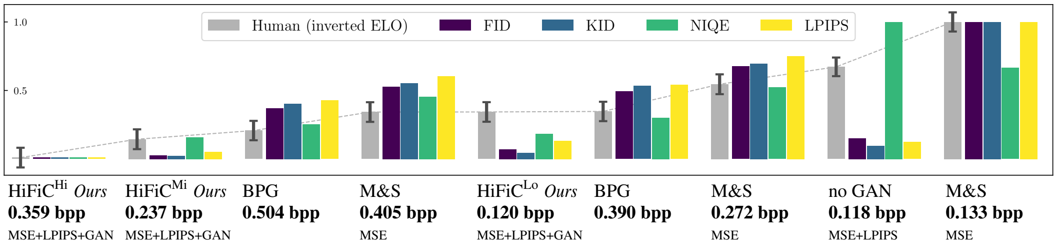


Figure 3: Normalized scores for the user study, compared to perceptual metrics. We invert human scores such that **lower is better** for all. Below each method, we show *average* bpp, and for learned methods we show the loss components. "no GAN" is our baseline, using the same architecture and distortion d as *HiFiC (Ours)*, but no GAN. "M&S" is the *Mean & Scale Hyperprior* MSE-optimized baseline. The study shows that training with a GAN yields reconstructions that outperform BPG at practical bitrates, for high-resolution images. Our model at 0.237bpp is preferred to BPG even if BPG uses $2.1\times$ the bitrate, and to MSE optimized models even if they use $1.7\times$ the bitrate.

$d = k_M \text{MSE} + k_P d_P$, but no GAN loss, called ***Baseline (no GAN)***. We train all models with Adam [22] for $2\,000\,000$ steps, and initialize our GAN models with weights trained for $\lambda' r + d$ only, which speeds up experiments (compared to training GAN models from scratch) and makes them more controllable. Exact values of all training hyper-parameters are tabulated in Appendix A.6.

We use the non-autoregressive ***Mean&Scale (M&S) Hyperprior*** model from Minnen *et al.* [32] as a strong baseline for an MSE-trained network. We emphasize that this model uses the same probability model P as we use, and that we train it with the same schedule as our models—the main architectural difference to ours is that Minnen *et al.* use a shallower auto-encoder specifically tuned for MSE. We train *M&S Hyperprior* for 15 rate points, allowing us to find a reconstruction with reasonably similar bitrate to our models for all images, and we obtain models outperforming BPG with similar performance to what is reported in [32] (Appendix A.1 shows a rate-distortion plot). When comparing against BPG, we use the highest PSNR setting, *i.e.*, no chroma subsampling and "slow" mode.

User Study We compare a total of N_M=9 methods: We use *HiFiC* models trained for $r_t \in \{0.14, 0.3, 0.45\}$, denoted ***HiFiC^Lo^***, ***HiFiC^Mi^***, ***HiFiC^Hi^***. For each such model, we go through all images and select the *M&S Hyperprior* model (out of our 15) that produces a reconstruction using *at least* as many bits as *HiFiC* for that image. Additionally, we use *Baseline (no GAN)* trained for r_t=0.14, and BPG at two operating points, namely at $1.5\times$ and $2\times$ the bitrate of *HiFiC^{Mi}*. The resulting bitrates do not exactly match r_t and are shown below models in Fig. 3. We asked N_P=14 participants to complete our study. Participants rated an average of 348 pairs of methods, taking them an average of one hour, yielding a total of 4876 comparisons.

5 Results

5.1 User Study

We visualize the outcome of the user study in Fig. 3. On the x-axis, we show the different methods sorted by the human ranking, with their average bits per pixel (bpp) on the N_I images, as well as the losses used for learned methods. We invert ELO and normalize all scores to fall between 0.01 and 1 for easier visualization. All metrics apart from the inverted ELO are calculated on the entire images instead of user-selected crops as we want to asses the amenability of the metrics for determining ratings and these crops would only be available through user studies.

We can observe the following: Our models (HiFiC) are always preferred to MSE-based models at equal bitrates. Comparing *HiFiC^{Lo}* to *Baseline (no GAN)*, we can see that adding a GAN clearly helps for human perception. Furthermore, *HiFiC^{Lo}* at 0.120bpp achieves similar ELO scores as BPG at 0.390bpp ($3.3\times$), and similar scores as *M&S Hyperprior* at 0.405bpp ($3.4\times$). *HiFiC^{Mi}* at 0.237bpp is preferred to BPG when BPG uses 0.504bpp, more than $2\times$ the bits, and preferred to *M&S Hyperprior* when it uses $1.7\times$ the bits. We note that BPG at around this bitrate is in a regime where the most severe artifacts start to disappear. The fact that *HiFiC^{Mi}* is preferred to BPG at half the bits shows how using a GAN for neural compression can yield high fidelity images with great bit savings compared to other approaches.

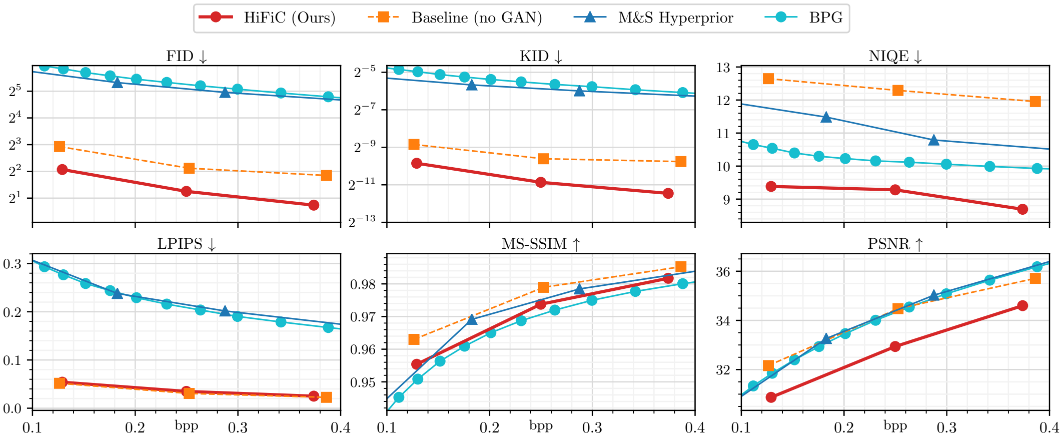


Figure 4: Rate-distortion and -perception curves on CLIC2020. Arrows in the title indicate whether lower is better (↓), or higher is better (↑). Methods are described in Section 4.

Furthermore, if we fix the architecture to ours and the distortion loss to $d = k_M\text{MSE} + k_P d_P$, the perceptual quality indices properly rank the resulting four models (*HiFiC*^{Lo}, *HiFiC*^{Mi}, *HiFiC*^{Hi}, *Baseline (no GAN)*). However, none of the metrics would have predicted the overall human ranking. Especially FID and KID overly punish MSE-based methods, and LPIPS improves when optimized for. On the positive side, this indicates these metrics can be used to order methods of similar architecture and distortion losses. However, we also see that currently there is no metric available to fully replace a user study.

In Appendix A.5, we show that running the Elo tournament per user (inter-participant agreement), and per image (participant consistency at image level) yields the same trends.

5.2 Visual Results

HiFiC In Fig. 1, we compare *HiFiC*^{Lo} to BPG at the same and at 2× the bitrate. The crops highlight how *HiFiC* reconstructs both the hair and sweater realistically, and very close to the input. BPG at the same bitrate exhibits significant blocking and blurring artifacts, which are lessened but still present at 2× the rate. In the background, we see that our full reconstruction is very similar to the original, including the skin and hat texture. In Appendix B, we show images from all of our datasets and compare to more methods, at various bitrates. There, we also provide download links to all reconstructions. For more visuals, see `hific.github.io`.

Failure Cases In Appendix B, we can also see examples of the two failure cases of *HiFiC*. The first is very small scale text, shown in CLIC2020/25bf4, which looks typeset in another script. The second is faces that are small relative to the image, as in Kodak/14, where our *HiFiC*^{Lo} model shows high-frequency noise.

Previous Work using GANs In Appendix A.3, we compare qualitatively to previous work by Agustsson *et al.* [3] and Rippel *et al.* [39].

5.3 Quantitative Results

Effect of GAN In Fig. 4, we show rate-distortion and rate-perception curves using our six metrics (see Section 4), on CLIC2020. We compare *HiFiC (Ours)*, *Baseline (no GAN)*, *M&S Hyperprior*, and BPG. We can see that, as expected, our GAN model dominates in all perceptual quality indices, but has comparatively poor PSNR and MS-SSIM. Comparing *Baseline (no GAN)* to *HiFiC*, which both have the same distortion $d = k_M\text{MSE} + k_P d_P$, reveals that the effect of adding a GAN loss is consistent with the theory, and with the user study: all perceptual metrics improve, and both components of d and thus the total distortion d get

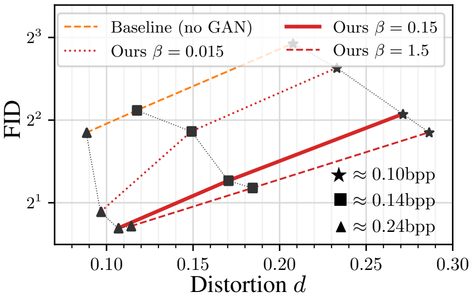


Figure 5: Distortion-perception trade-off.

		FID		KID		LPIPS		NIQE	
Norm in D	Conditional D	μ	Δ	μ	Δ	μ	Δ	μ	Δ
InstanceNorm		5.8	3.4	1.1E^{-3}	9.7E^{-4}	6.6E^{-2}	2.7E^{-3}	9.0	0.66
SpectralNorm		4.7	0.14	1.3E^{-3}	1.6E^{-4}	5.6E^{-2}	1.2E^{-3}	9.4	0.43
InstanceNorm	✓	4.3	0.68	7.1E^{-4}	2.4E^{-4}	6.3E^{-2}	1.6E^{-3}	8.8	0.64
SpectralNorm	✓	4.3	0.36	1.2E^{-3}	2.2E^{-4}	5.5E^{-2}	7.4E^{-4}	9.4	0.32

Table 1: Exploring across-run variation. Each model (row) is run four times in exactly the same configuration, and we show mean μ and difference between maximal and minimal value Δ.

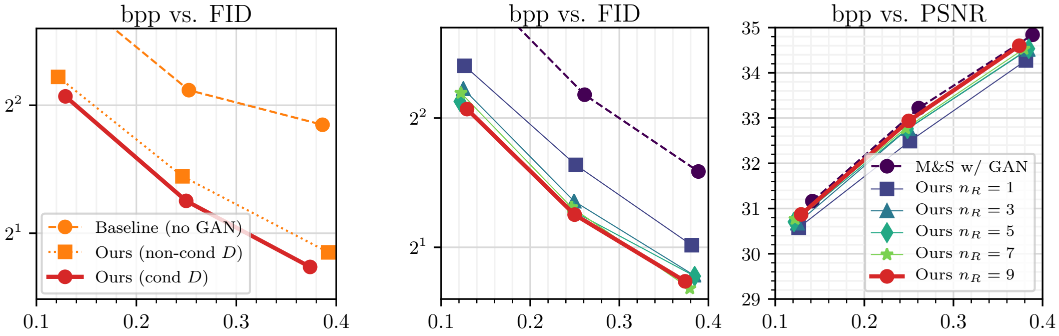
worse (see also next paragraph). We observe that MSE (PSNR) is more at odds with the GAN loss than LPIPS, which gets only marginally worse when we add it. These observations motivate us to use FID in ablation studies, as long as d and the overall training setup is fixed. We show similar figures for the other datasets in Appendix A.8.

Distortion-Perception Trade-off In Fig. 5, we show how varying the GAN loss weight β navigates the distortion-perception trade-off. We plot FID on the y axis, and the full $d = k_M \text{MSE} + k_P d_P$ on the x axis. We show *HiFiC* on a exponential grid $\beta \in \{0.015, 0.15, 1.5\}$, and *Baseline (no GAN)* as a reference. Each model is shown at three different bitrates, and models with similar bitrate are connected with black dotted lines. We see that across rates, as we increase β, the perceptual quality index improves, while d suffers. This effect is lessened at higher rates (towards left of figure), possibly due to different relative loss magnitudes at this rate, *i.e.*, d is much smaller, r is larger.

5.4 Studies

Discriminator: On Conditioning and Normalization Recall that we use a conditional D, in contrast to previous work. While our formulation can be adapted to include a non-conditional D, this changes training dynamics significantly: Theoretically, ignoring the distortion d for a moment, G can then learn to produce a natural image that is completely unrelated do with the input, and still achieve a good discriminator loss. While we guide G with d, we nevertheless found that training a *non-conditional* discriminator leads to images that are less sharp. The resulting models also obtain a worse FID, see Fig. 6a.

In [51], InstanceNorm is used in D, which causes artifacts in our setup, prompting us to replace it with SpectralNorm [36]. In pure generation, SpectralNorm was shown to reduce the variance of FID across multiple runs [24]. To explore this in our setup, we run each of the models shown in Table 1 four times, and measure the mean and largest difference between runs, on our metrics. We find that in the *non-conditional* setting, SpectralNorm reduces the variability of FID and KID significantly, and, to a lesser extent, also that of LPIPS and NIQE. This effect is weakened when using the *conditional* D, where SpectralNorm only marginally reduces variability in all metrics, which we view as a further benefit of conditioning. Overall, keeping the normalization dimension fixed, we also see that conditioning D improves all metrics.



(a) Non-conditional D

(b) Varying generator capacity

Figure 6: a) Shows the effect of a non-conditional D, b) shows models with different number of residual blocks n_R in G.

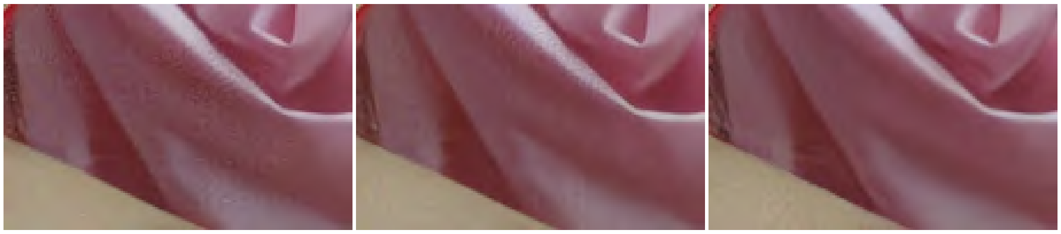
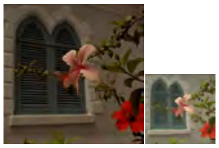

(a) no d_P (b) d_P = VGG (c) d_P = LPIPS

Figure 8: Effect of varying the perceptual distortion d_P. All models were also trained with an MSE loss and a GAN loss.



Instance Norm in Auto-Encoder We visualize the darkening caused by InstanceNorm in E, G (see Section 3.3) in the inset figure, where a model is evaluated on an image at a resolution of 512×512px as well as at the training resolution (256×256px). We also explored using BatchNorm [19] but found that this resulted in unstable training in our setup.

Generator Capacity By default, we use n_R=9 residual blocks in G (see Fig. 2). We show the result of varying n_R in Fig. 6b. While n_R=1 gives significantly worse FID and PSNR, both metrics start to saturate around n_R=5. We also show the *M&S Hyperprior* baseline trained with our loss ("M&S w/ GAN"), *i.e.*, with LPIPS and a conditional D, and exactly the same training schedule. While this yields slightly better PSNR as HiFiC, FID drops by a factor 2, further indicating the importance of capacity, but also how suited the hyperprior architecture is for MSE training.

Training Stability and Losses Training GANs for unconditional or class-conditional image generation is notoriously hard [29, 36, 10]. However, our setup is inherently different through the distortion loss d, which guides the optimization at the pixel-level. While our initialization scheme of using weights trained to minimize $\lambda' r + d$ (see Section 4) can help, MSE seems to be the driving force for stability in our setup: For Fig. 9, we initialize the network with $\lambda' r + d$-optimized weights, then train with a GAN loss but without a distortion loss d. We compare fixing E to using the $\lambda' r + d$ weights, to jointly learning E. We see that the GAN collapses if we learn E. Additionally, we explore varying d_P: In Fig. 8a, we see that removing d_P introduces significant gridding artifacts. Using d_P=VGG, *i.e.*, L_1 in the feature space of VGG, some gridding is still visible (Fig. 8b). Using d_P=LPIPS alleviates these artifacts. We note that only using LPIPS *without a GAN* also produces minor repetitive patterns as well as less sharp reconstructions, which is validated by the fact that *Baseline (no GAN)* ranks worse than our GAN models in the user study. We show a visual example in Appendix A.2.

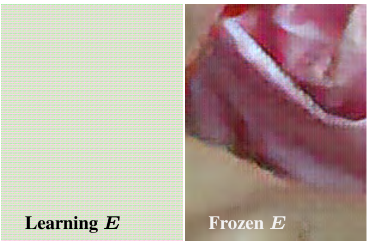


Figure 9: Training with a GAN loss, but no MSE or d_P.

6 Conclusion

In this paper, we showed how optimizing a neural compression scheme with a GAN yields reconstructions with high perceptual fidelity that are visually close to the input, and that are preferred to previous methods even when these approaches use more than double the bits. We evaluated our approach with a diverse set of metrics and interpreted the results with rate-distortion-perception theory. Guided by the metrics, we studied various components of our architecture and loss. By comparing our metrics with the outcome of the user study, we showed that no existing metric is perfect for ordering arbitrary models right now, but that using FID and KID can be a valuable tool in exploring architectures and other design choices. Future work could focus on further studying perceptual score indices and metrics (to better predict human preferences), and further investigate failure cases such as small faces. Additionally, generative video compression is a very interesting direction for future work. To ensure temporal consistency, mechanisms to enforce smoothness could be borrowed from, e.g., GAN-based video super resolution models.

Broader Impact

Users of our compression method can benefit from better reconstructions at lower bitrates, reducing the amount of storage needed to save pictures and the bandwidth required to transmit pictures. The latter is important as wireless technology typically lags behind user trends which have been continuously requiring higher bandwidths over the past decades, and there is no end in sight with emerging applications such as virtual reality. Furthermore, better compression technology improves accessibility in developing areas of the world where the wireless infrastructure is less performant and robust than in developed countries. It is important to keep in mind that we employ a generator G that in theory can produce images that are very different from the input. While this is the case for *any* lossy image compression algorithm, this has a bigger impact here as we specifically train G for realistic reconstructions. Therefore, we emphasize that our method is not suitable for sensitive image contents, such as, *e.g.*, storing medical images, or important documents.

Funding

This work was done at Google Research.

Acknowledgments

The authors would like to thank Johannes Balle, Sergi Caelles, Sander Dielmann, and David Minnen for the insightful discussions and feedback.

References

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017.

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[3] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

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

[7] Fabrice Bellard. BPG Image format. https://bellard.org/bpg/.

[8] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

[9] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. *arXiv preprint arXiv:1901.07821*, 2019.

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[11] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[12] Rémi Coulom. Whole-history rating: A bayesian rating system for players of time-varying strength. In *International Conference on Computers and Games*, pages 113–124. Springer, 2008.

[13] Stéphane Coulombe and Steven Pigeon. Low-complexity transcoding of jpeg images with near-optimal quality using a predictive quality factor and scaling parameters. *IEEE Transactions on Image Processing*, 19(3):712–721, 2009.

[14] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[15] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4826–4835, 2017.

[16] Mark E Glickman. A comprehensive guide to chess ratings. *American Chess Journal*, 3(1):59–102, 1995.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2017.

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[23] Kodak PhotoCD dataset. `http://r0k.us/graphics/kodak/`.

[24] Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *International Conference on Machine Learning*, pages 3581–3590, 2019.

[25] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.

[26] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[27] Boyi Li, Felix Wu, Kilian Q Weinberger, and Serge Belongie. Positional normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1622–1634. Curran Associates, Inc., 2019.

[28] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. *arXiv preprint arXiv:1703.10553*, 2017.

[29] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, pages 700–709, 2018.

[30] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand. Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):620–636, 2003.

[31] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.

[33] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. *arXiv preprint arXiv:2007.08739*, 2020.

[34] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[35] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012.

[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[37] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Advances in neural information processing systems*, pages 2474–2482, 2012.

[38] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[39] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2922–2930, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[40] Shibani Santurkar, David Budden, and Nir Shavit. Generative compression. *arXiv preprint arXiv:1703.01467*, 2017.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[43] David S. Taubman and Michael W. Marcellin. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[44] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszar. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017.

[45] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.

[46] George Toderici, Lucas Theis, Nick Johnston, Eirikur Agustsson, Fabian Mentzer, Johannes Ballé, Wenzhe Shi, and Radu Timofte. *CLIC 2020: Challenge on Learned Image Compression*, 2020. http://compression.cc.

[47] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. *arXiv preprint arXiv:1608.05148*, 2016.

[48] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. In *Advances in Neural Information Processing Systems*, pages 5929–5940, 2018.

[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[50] Gregory K Wallace. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[51] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[52] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, Nov 2003.

[53] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[54] WebP Image format. https://developers.google.com/speed/webp/.

[55] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[56] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

A Supplementary Material – High-Fidelity Generative Image Compression

We show further details of our method in A.1–A.9, and more visual results in Appendix B.

A.1 Comparing MSE models based on Minnen *et al.* [32]

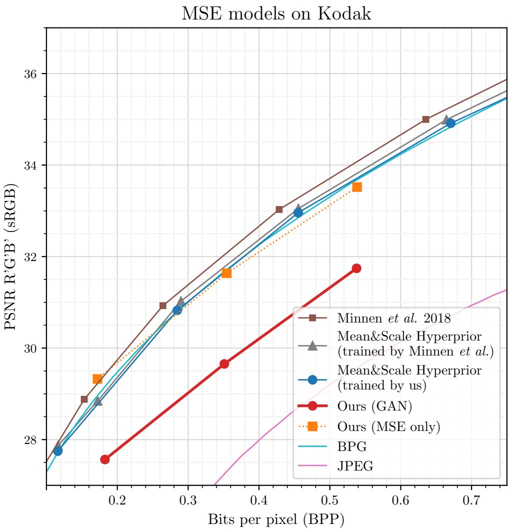
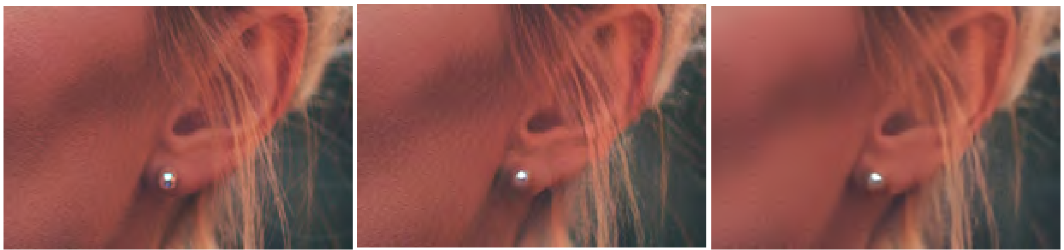


Figure A1: Comparing the *Mean & Scale Hyperprior* model trained by us to the one reported by Minnen *et al.* [32], and to their main model, denoted "Minnen *et al.* 2018".

To validate that the *M&S Hyperprior* model trained by us is a strong MSE model, we compare it to models reported by Minnen *et al.* [32] in Fig. A1. First, we compare the model trained by us to the "Mean & Scale Hyperprior" baseline reported in [32], which is equivalent to the *M&S Hyperprior* model in terms of architecture. We observe a very minor drop in PSNR, which is likely attributable to our training schedule of 2 000 000 steps, vs. the 6 000 000 steps used in [32]. Then, we compare to the fully autoregressive main model of [32], denoted "Minnen *et al.* 2018". We can see that this model is on average ≈ 0.4dB better than our *M&S Hyperprior* model. It is important to note that a) an auto-regressive probability model would also increase the performance of all of our models, and b) that in the user study, our GAN models were preferred to MSE-trained Hyperprior models even when the Hyperprior models used $4\times$ the bitrate, which amounts roughly to a $\approx$4dB PSNR difference.

A.2 Using LPIPS without a GAN loss

We saw in the user study that the model trained for MSE and LPIPS without a GAN loss (*Baseline (no GAN)*) ranks worse than the one using a GAN loss (*HiFiC^{Lo}*). In Fig. A2, we visualize a reconstruction of *Baseline (no GAN)* to show that this training setup can also cause gridding artifacts on some images. Furthermore, we see that adding a GAN loss leads to sharper reconstructions and to a faithful preservation of the image noise.



Orignal

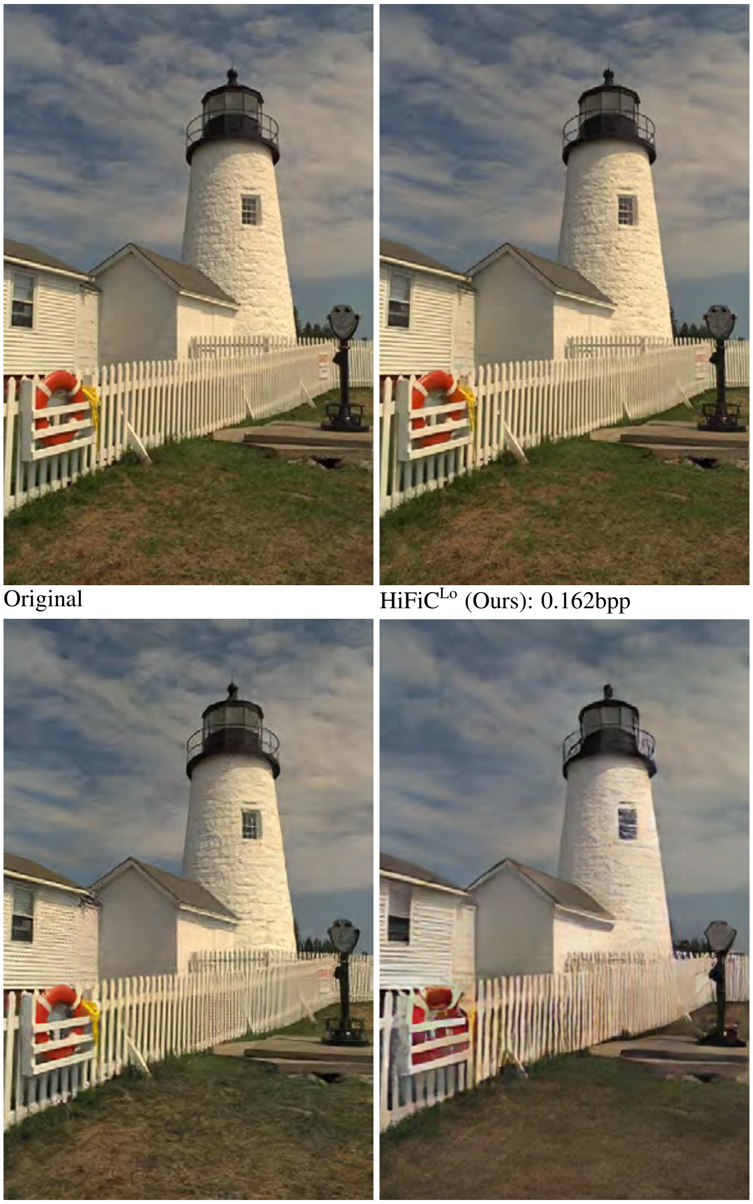
r, MSE, LPIPS, GAN (*HiFiC^{Lo}*)

r, MSE, LPIPS (*Baseline (no GAN)*)

Figure A2: LPIPS without a GAN (right) leads to gridding and less sharpness than using a GAN (middle). We show the loss components below the figure, where r is the rate loss. *Best viewed on screen.*

A.3 Qualitative Comparison to Previous Generative Approaches

We compare against Agustsson *et al.* [3] as well as Rippel *et al.* [39] in Fig. A3 on an image from the Kodak dataset, as both incorporated adverserial losses into their training. We do no further comparisons, as Agustsson *et al.* targeted "extremely low" bitrates and thus operate in a very different regime than our models, and Rippel *et al.* only released a handful of images.



Original

HiFiC$^{\text{Lo}}$ (Ours): 0.162bpp

Rippel *et al.* [39]: 0.194bpp

Agustsson *et al.* [3]: 0.0668bpp

Figure A3: Comparison between neural compression approaches using an adverserial loss on Kodak/19. Note the high-frequency artifacts in the tower and fence for [39], as well as the different grass texture. Note that the comparison to [3] is not ideal, as their highest bpp reconstructions use less than half the rate of *HiFiC*, but we see that the reconstruction deviates significantly from the input. Furthermore, both [39] and [3] exhibit a small color shift.

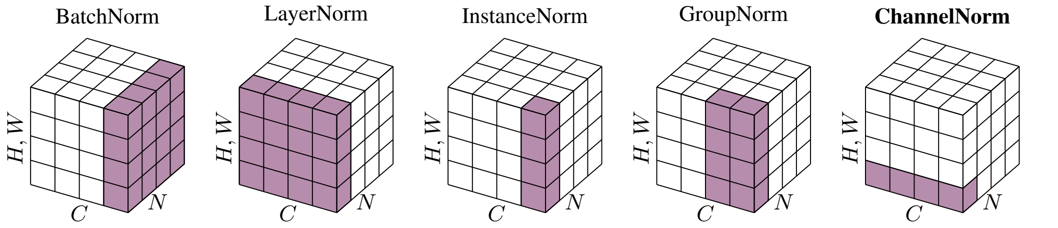


Figure A4: Visualizing which axes different normalization layers normalize over. N is the batch dimension, C the channel dimension, and H, W is the spatial dimensions. If the shaded area spans over an axis, this axis is normalized over. For example, BatchNorm normalizes over space and batches, LayerNorm over space and channels. Our normalization layer, ChannelNorm, normalizes over channels only. *Figure adapted from [55].*

A.4 ChannelNorm

Fig. A4 shows a visual comparison of different normalization layers, which is based on the visualization provided in [55]. We note that BatchNorm [19], LayerNorm [4], InstanceNorm [49], and GroupNorm [55], all average over space.

We compare ChannelNorm to InstanceNorm via the following equations. To simplify notation, we assume that normal broadcasting rules apply, *e.g.*, $f_{chw} - \mu_c$ means that when calculating channel c, we subtract μ_c from each spatial location h, w in the c-th channel of f. As introduced in Section 3.3, ChannelNorm uses learned per-channel offsets α_c, β_c and normalizes over channels:

$$f'_{chw} = \frac{f_{chw} - \mu_{hw}}{\sigma_{hw}} \alpha_c + \beta_c, \qquad \begin{aligned} \mu_{hw} &= {}^1\!/_C \textstyle\sum_{c=1}^{C} f_{chw} \\ \sigma^2_{hw} &= {}^1\!/_C \textstyle\sum_{c=1}^{C} (f_{chw} - \mu_{hw})^2. \end{aligned}$$

InstanceNorm also uses α_c, β_c, but normalizes spatially:

$$f'_{chw} = \frac{f_{chw} - \mu_c}{\sigma_c} \alpha_c + \beta_c, \qquad \begin{aligned} \mu_c &= {}^1\!/_{HW} \textstyle\sum_{h=1}^{H} \sum_{w=1}^{W} f_{chw} \\ \sigma^2_c &= {}^1\!/_{HW} \textstyle\sum_{h=1}^{H} \sum_{w=1}^{W} (f_{chw} - \mu_c)^2. \end{aligned}$$

We hypothesize that the dependency on H, W causes the generalization problems we see.

A.5 User Study: More Results

For the user study plot in the main text (Fig. 3), we averaged Elo scores across all raters, across all images, *i.e.*, the Elo tournament is over all comparisons (see Section 3.4). This data is visualized as a box plot in Fig. A5.

In this section, we also show the result of averaging over participants in Fig. A6 (running an Elo tournament per participant), and over images in Fig. A7 (running an Elo tournament per image). As we can see, the overall order remains unchanged, except for the three methods that were very close in median Elo score in Fig. 3 (HiFiC at 0.120bpp, BPG at 0.390bpp, and M&S Hyperprior at 0.405bpp), which change their order. Also, different images obtain a wide range of Elo scores.

In Table 2, we show the first 5 characters of the names of the $N_I = 20$ images used for the user study, and the rank each method scored on this image, where lower is better. We note that for all images, one of our GAN models earns first or second place – except for the last image. This image contains a lot of small scale text, and is shown as part of our visualizations in Appendix B. There, we also provide a link to all images used in the user study.

In Fig. A9, we show a screenshot of the GUI shown to raters. The crops selected by the participants are available at `https://hific.github.io/raw/userstudy.json`.

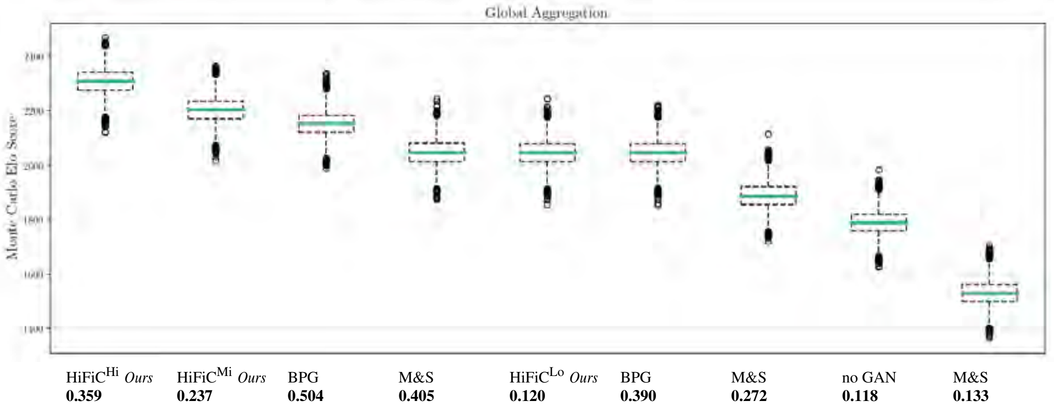


Figure A5: Global Monte Carlo Elo Scores. We use the standard box plot visualization: The horizontal green thick line indicates the median, the box extends from Q1 to Q3 quartiles, the whiskers extend to $1.5 \cdot (Q3 - Q1)$, and outlier points are points past the whiskers.

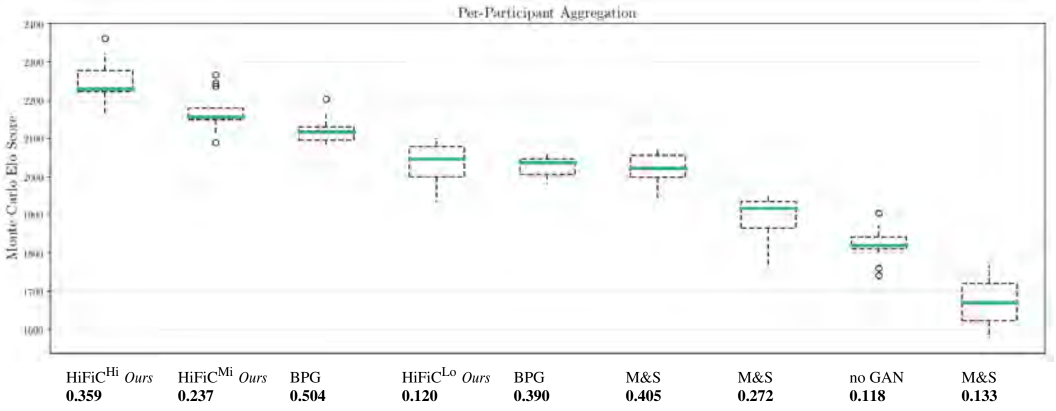


Figure A6: Per-participant Monte Carlo Elo Scores. We use the standard box plot visualization: The horizontal green thick line indicates the median, the box extends from Q1 to Q3 quartiles, the whiskers extend to $1.5 \cdot (Q3 - Q1)$, and outlier points are points past the whiskers.

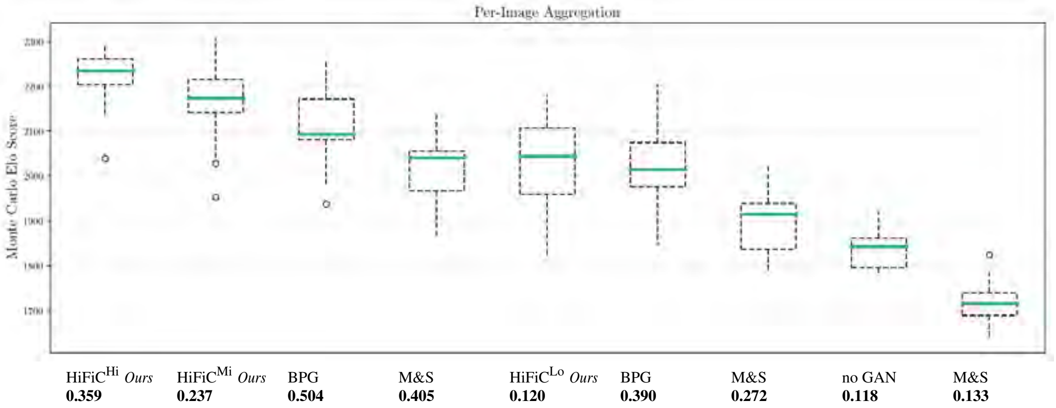


Figure A7: Per-image Monte Carlo Elo Scores. We use the standard box plot visualization: The horizontal green thick line indicates the median, the box extends from Q1 to Q3 quartiles, the whiskers extend to $1.5 \cdot (Q3 - Q1)$, and outlier points are points past the whiskers.

Image name	HiFiC$^{\text{Hi}}$ 0.359 bpp	HiFiC$^{\text{Mi}}$ 0.237 bpp	BPG 0.504 bpp	M&S 0.405 bpp	HiFiC$^{\text{Lo}}$ 0.120 bpp	BPG 0.390 bpp	M&S 0.272 bpp	no GAN 0.118 bpp	M&S 0.133 bpp
e0256	1	2	3	5	4	6	8	7	9
a251f	1	2	4	8	3	5	7	6	9
0ae78	1	2	3	6	5	4	8	7	9
95e7d	1	2	3	6	4	5	8	7	9
2145f	1	2	4	3	5	6	7	8	9
58c13	2	1	3	5	6	4	7	9	8
f063e	2	4	1	6	5	3	7	8	9
dcb53	1	2	4	3	5	6	7	8	9
d5424	1	2	3	5	6	4	8	7	9
72e19	1	2	5	4	3	8	6	7	9
1c55a	1	5	2	4	6	3	7	8	9
ad249	2	1	5	4	3	7	8	6	9
d9692	2	5	1	4	6	3	7	8	9
18089	1	3	2	5	6	4	7	8	9
f7a9e	1	2	4	3	5	6	7	8	9
a09ce	1	3	2	5	7	4	6	8	9
6e8e3	1	2	4	5	3	6	8	7	9
afa0a	1	2	4	6	3	5	7	8	9
8ba19	2	1	3	5	4	6	7	8	9
25bf4	4	6	1	3	8	2	5	7	9

Table 2: Per image rankings of the user study. We show the average bpp of each method in the header

A.6 Training Details

We follow standard practice of alternating between training E, G, P (jointly) for one step and training D for one step. We use the same learning rate of 1E^{-4} for all networks, and train using the Adam optimizer [22]. At the beginning of training, the rate loss can dominate. To alleviate this, we always first train with a higher λ, as in [32]. As mentioned in Section 4, we initialize our GAN models (HiFiC$^{\text{Hi}}$, HiFiC$^{\text{Mi}}$, HiFiC$^{\text{Lo}}$) from a model trained for MSE and d_P =LPIPS. Table 3 shows our different models and the LR and λ schedules we use. The GAN initialization is using the *Warmup* model. Together, this yields 2M steps for all models.

We fix hyper-parameters shown in Fig. A8a for all experiments (unless noted), and vary $\lambda^{(a)}$ depending on r_t as shown in Fig. A8b.

The training code and configs for *HiFiC^{Lo}* and *Baseline (no GAN)* is available at `hific.github.io`.

	Losses	Initialize with	Training	LR decay	Higher λ
Baseline (no GAN)	MSE+LPIPS	-	2M steps	1.6M steps	1M steps
M&S Hyperprior	MSE	-	2M steps	1.6M steps	1M steps
Warmup	MSE+LPIPS	-	1M steps	0.5M steps	50k steps
HiFiC $^{\text{Hi,Mi,Lo}}$	MSE+LPIPS+GAN	*Warmup*	1M steps	0.5M steps	50k steps

Table 3: Training schedules, using “M” for million, “k” for thousand.

Parameter	Value	Note
$\lambda^{(b)}$	2^{-4}	
k_M	$0.075 \cdot 2^{-5}$	
k_P	1	
C_y	220	
β	0.15	Except for Fig. 5

(a) Fixed hyper-parameters.

Target Rate r_t	$\lambda^{(a)}$
0.14	2^1
0.30	2^0
0.45	2^{-1}

(b) Varying hyper-parameters.

Figure A8: Hyper-parameters.

A.7 Patch-based FID and KID

As mentioned in Section 4, we extract patches to calculate FID and KID. From each $H \times W$ image, we first extract $\lfloor H/f \rfloor \cdot \lfloor W/f \rfloor$ non-overlapping $f \times f$ crops, and then shift the extraction origin by $f/2$ in both dimensions to extract another $(\lfloor H/f \rfloor - 1) \cdot (\lfloor W/f \rfloor - 1)$ patches. We use $f = 256$ in all evaluations.

A.8 Further Quantitative Results

In this section, we provide plots similar to Fig. 4 on the other two datasets: In Fig. A10, we show rate-distortion and rate-perception curves for DIV2K [2], and in Fig. A11, we show curves for Kodak [23]. As noted in Section 4, the 24 Kodak images only yield 192 patches to calculate FID and KID, and we thus omit the two metrics.

A.9 Image Dimensions of the Datasets

We use the three datasets mentioned in Section 4 for our evaluation. Kodak contains images of 768×512 pixels, the other two datasets contain images of varying dimensions. To visualize the distribution of these dimensions, we show histograms for the shorter dimensions, for the product of both dimensions, and for the aspect ratio in Fig. A12. We see that most images cluster around shorter sides of 1400px, and go up to 2000px. We note that the three biggest images for CLIC are of dimensions 2048×2048, 2048×2048, 2000×2000, for DIV2K they are 2040×2040, 1872×2040, 1740×2040.

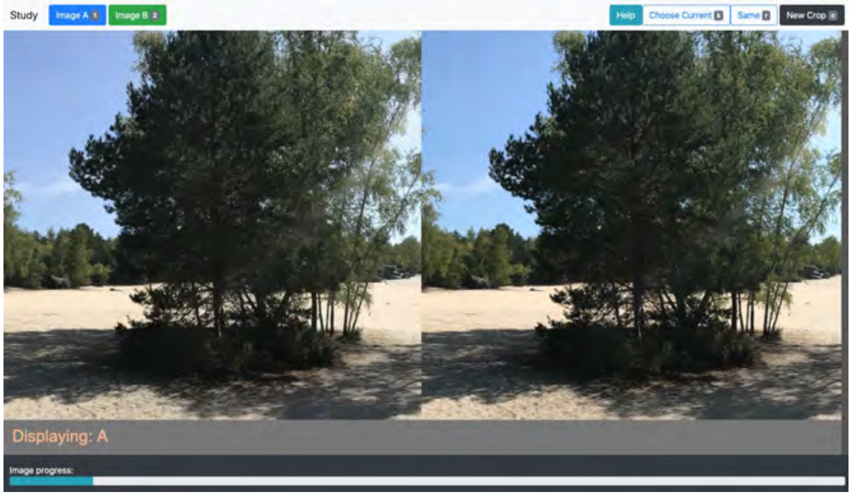


Figure A9: Screenshot of the user study GUI.

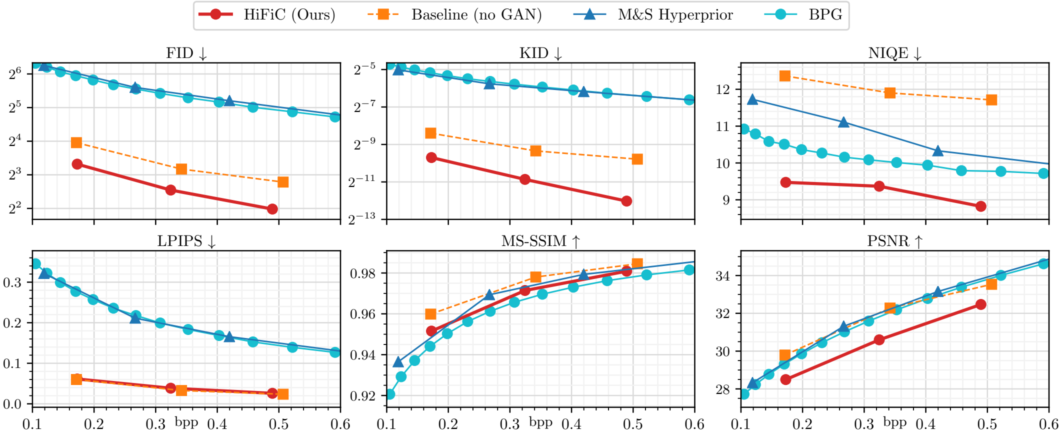


Figure A10: Rate-distortion and -perception curves on DIV2K. Arrows in the title indicate whether lower is better (↓), or higher is better (↑).

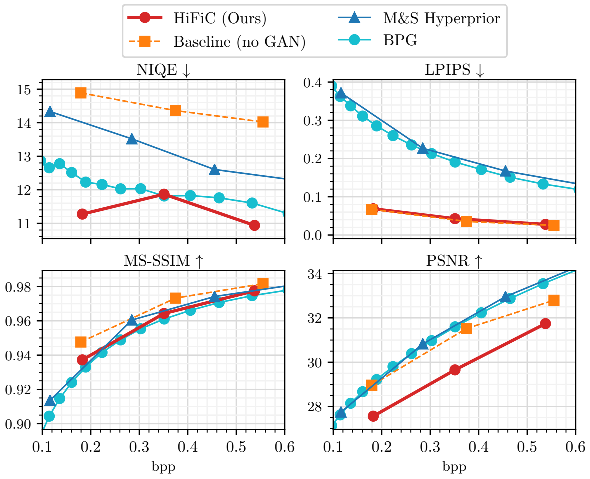


Figure A11: Rate-distortion and -perception curves on Kodak. Arrows in the title indicate whether lower is better (↓), or higher is better (↑).

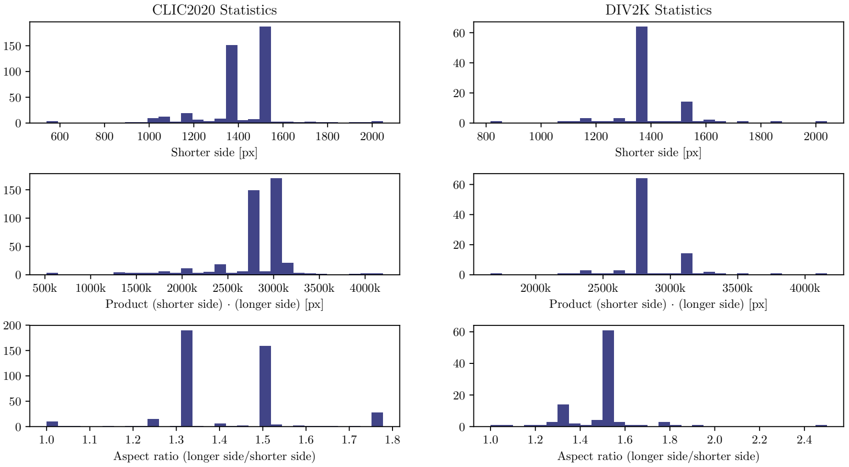


Figure A12: Histograms to show image dimensions for CLIC2020 and DIV2K.

B Further Visual Results

B.1 PDF Visualization

Due to size constraints, the PDF is hosted at `https://hific.github.io/appendixb`.

In the PDF, we show images from all datasets. For each image, we show the full reconstruction by one of our models, next to the original. On the top left of this image, we show the dataset and image ID. Additionally, we pick one or two crops for each image, where we compare the original to the following methods:

1. HiFiC$^{\text{Mi}}$ (Ours)
2. HiFiC$^{\text{Lo}}$ (Ours)
3. *M&S Hyperprior* at a bitrate greater than *HiFiC^{Lo}*
4. BPG at the 2× the bitrate of *HiFiC^{Mi}*
5. BPG at the same bitrate as *HiFiC^{Mi}*
6. BPG at the same bitrate as *HiFiC^{Lo}*
7. JPEG at $Q = 80$.

We chose to add JPEG at $Q = 80$ as a further reference, as this is a quality factor in common use [13].

Results

Throughout the examples, we can see that our GAN models shine at reconstructing plausible textures, yielding reconstructions that are very close to the original. We see that BPG at the same bitrate as *HiFiC^{Lo} (Ours)* tends to exibit block artifacts, while *HiFiC^{Lo}* looks very realistic. For most images, our *HiFiC^{Mi}* model also looks significantly better than BPG at the same bitrate. When BPG uses 2× the rate as *HiFiC^{Mi}*, it starts to look similar to our reconstructions.

B.2 More Comparisons and Raw Images

We note that the crops are embedded as PNGs in the PDF, but the large background images are embedded as JPEGs to prevent a huge file. However, we package full-sized PNGs for various methods into ZIPs. The ZIP files also contain a HTML file that can be used for easy side-by-side comparisons, as this is the best way to visualize differences.

The images are also available directly in the browser at:

`https://hific.github.io/raw/index.html`.

The ZIP files:

1. The following ZIP contains the 20 images from CLIC2020 used for the user study, compressed with each of the 9 methods used in the study.
   `https://hific.github.io/raw/userstudy`
   Size: 610MB

2. The following ZIP files contain all the images of the respective datasets, compressed with *HiFiC^{Hi}*, *HiFiC^{Mi}*, *HiFiC^{Lo}*.
   `https://hific.github.io/raw/kodak`
   Size: 61MB
   `https://hific.github.io/raw/clic2020`
   Size: 6.3GB
   `https://hific.github.io/raw/div2k`
   Size: 1.7GB