Diffusion-based Compression Quality Tradeoffs without Retraining

Jonas Brenig
jonas.brenig@uni-wuerzburg.de

Radu Timofte
radu.timofte@uni-wuerzburg.de

Computer Vision Lab, CAIDAS & IFI, University of Würzburg, Germany

Abstract

Learned image compression methods using a generative decoder can reconstruct images at significantly higher perceptual quality than the new hand-crafted codecs or other learned methods. Recently, diffusion models have been integrated into the decoding process to further enhance image quality.

However, the diffusion process is sensitive to several hyperparameters, such as the number of steps, which are typically hard-coded and expected to perform well across various images. When applied to a single image, these parameters are often suboptimal.

In this work, we propose enhancing the reconstruction quality by optimizing the diffusion process's decoding parameters for each image individually during encoding. This approach improves the final quality with virtually no increase in bits-per-pixel. In addition, we compare methods to minimize the additional computational impact during encoding.

We validate our approach on the CDC (Yang et al., 2024) and PerCo (Careil et al., 2023) image compression models using datasets like Kodak and DIV2K. Our results show clear improvements in LPIPS and PSNR without negatively impacting bits-per-pixel. This concept of optimizing quality tradeoffs can be readily applied to other diffusion-based image compression methods without the necessity of additional network training.

1. Introduction

Image compression is a critical technology for managing the vast amounts of images and videos shared across the Internet. Although hand-crafted codecs such as JPEG [39] remain prevalent, newer codecs such as BPG [7] and VVC still image coding [11] offer substantial improvements, providing a robust baseline. However, recent learned image compression methods surpass these codecs in terms of compression rates [6, 21, 22, 32, 33].

Despite their efficiency, distortion-optimized methods often fail to align with human perception of quality. At low bitrates, optimizing solely for distortion can result in blurry or aesthetically displeasing images. Consequently, PSNR can be a misleading metric for image quality in many cases. Alternative metrics such as LPIPS [41] or FID [24] are often used. Several studies have aimed to enhance perceptual quality by employing generative modeling to better synthesize missing texture information. These methods often involve an adversarially trained generator [23, 31, 34], which can produce more realistic images even at low bitrates.

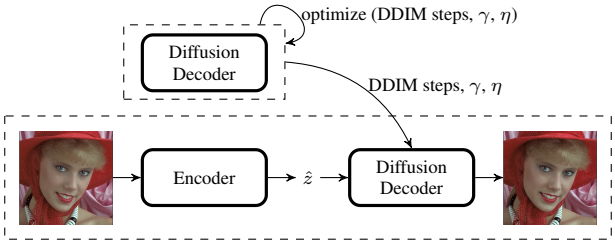

Figure 1. **Overview of the proposed method.**

With the increasing popularity of diffusion models for generative tasks [25, 36], they have also been applied to generative image compression. Theis *et al.* [38] proposed using an unconditional diffusion model and reverse channel coding to achieve impressive FID scores. Other approaches include using a conditional diffusion model as the generator [10, 40], enhancing a pre-trained (neural) codec with a diffusion model [16, 27], or leveraging the generative capabilities of pre-trained latent diffusion models for improved image reconstruction [12, 35].

Although diffusion models have shown promising results, the performance of the decoding process is heavily influenced by several hyperparameters. Typically, these parameters are handpicked to perform well in various cases, but they may not be optimal for each specific image. Consequently, existing methods do not fully exploit the potential of the diffusion process, as they rely on fixed hyperparameters that may not yield the best quality for each image.

In this paper, we propose enhancing existing diffusion-based image compression methods by optimizing their sam-

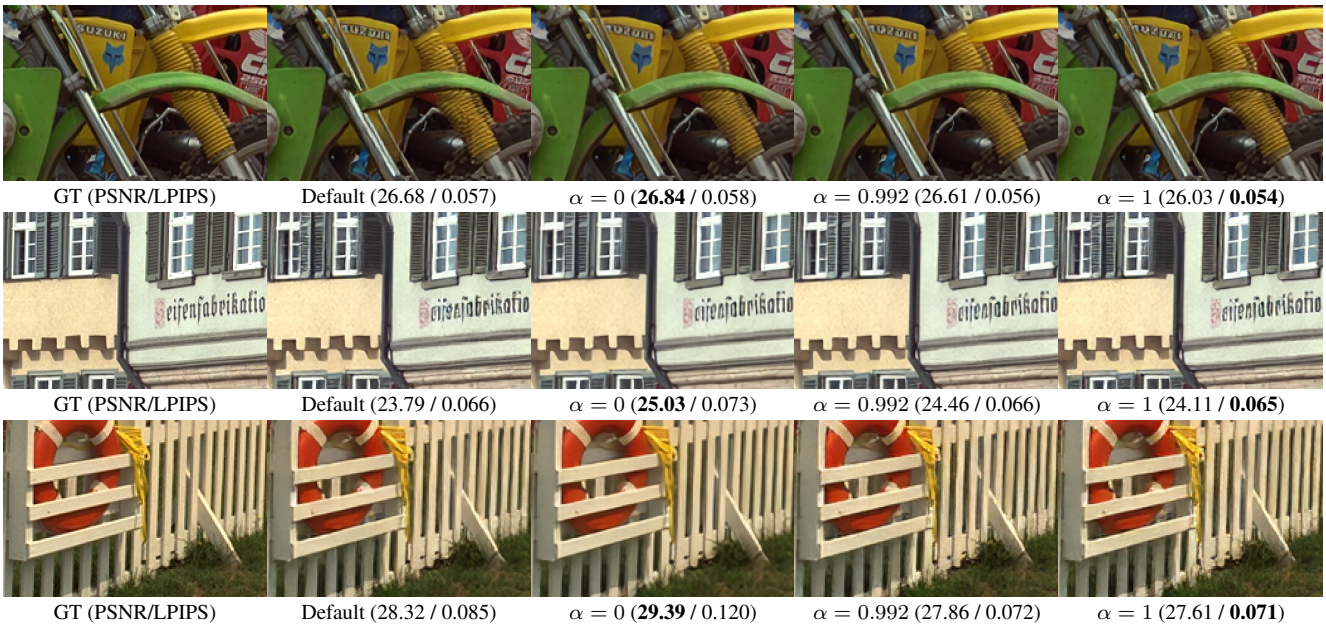

Figure 2. **Qualitative comparison of different tradeoffs between LPIPS and PSNR as generated by CDC x_0 [40]** for different crops of Kodak images. The **leftmost** image is the ground truth. The **second-left** image is the default configuration. The **remaining** images are optimized for different tradeoffs, with optimization for PSNR ($\alpha = 0$) up to optimization for LPIPS ($\alpha = 1$). Images optimized for PSNR tend to be more blurry while optimizing for LPIPS can result in more noisy images.

pling parameters specifically for the individual image being compressed. An overview is provided in Fig. 1. During the encoding process, we search for the parameter combination that results in the highest quality image. Depending on the chosen metric, different perception-distortion tradeoffs can be targeted [9]. Furthermore, the computational cost of this optimization is only incurred during compression, and the parameters can be transmitted with the compressed image at a negligible additional cost. We demonstrate the effectiveness of our approach on two different diffusion-based image compression methods: CDC [40] and PerCo [12]. Our results show significant improvements in perceptual quality without requiring additional training or significantly increasing bitrate. A project page is available online[1].

2. Related work

2.1. Learned image compression

Recent advances in the field of learned image compression have shown significant improvements, not only beating established handcrafted methods, such as JPEG [39], but also improving quality over newer ones, such as BPG [7] and the still image compression of VVC [11].

Most of the learned image compression methods are based on a nonlinear transform coding paradigm [18], in which an auto-encoder style model transforms the image into an easily compressible latent space [4, 5]. The latent space is quantized and then compressed by using arithmetic coding. To further improve this process, most works adopt an additional hyper-prior entropy model, which was first introduced by Ballé *et al*. [6], beating BPG.

Further improvements in entropy coding have been achieved using autoregressive context modeling [19, 33]. Recent works offer architectural improvements and allow for more parallelized decoding, resulting in faster decoding speeds while maintaining high performance [20–22, 30, 32, 42]. Recent works beat the state-of-the-art handcrafted still image compression codec of VVC [11].

When compressing natural images, PSNR is often not a very good indicator of image quality. Instead, perceptual metrics such as LPIPS [41] or FID [24] align much better with human perception.

Recently, several works focused on developing image compression techniques that prioritize perceptual quality. In most cases, these models use a generative adversarial network [17] (GAN) for the decoder. The first use of GANs in this context was introduced by Agustsson *et al*. [2]. Later works improved upon the initial concept [31], incorporating more advanced entropy models [23, 34] or allowing for control of the perception-distortion tradeoff [3, 28].

2.2. Diffusion-based learned image compression

More recently, diffusion models [25] offer promising potential to enhance the perceptual quality of image recon-

[1]Project: https://jbrenig.github.io/diff-bbopt-25/

struction, and several studies have begun to investigate their integration into the image compression pipeline.

Hoogeboom *et al*. [26] explored the use of Autoregressive Diffusion Models for lossless compression. In the realm of lossy compression, Theis *et al*. [38] proposed a scheme using an unconditional diffusion model combined with reverse channel coding. Although their experiments showed promising results, the approach is limited to small images due to computational complexity.

The first method to use diffusion models for image compression was introduced by Yang and Mandt [40]. They achieved impressive perceptual results across various metrics using a denoising diffusion model conditioned on the encoder latent. Alternatively, approaches such as DIRAC [16], HFD [27] and ResCDC [10] are trained to improve the perceptual quality of the reconstructions obtained using other image compression methods. These methods are trained on the degraded output of existing (neural) image compression codecs and can yield significant improvements in perceptual quality.

Relic *et al*. [35] recently proposed a lossy image compression method using Stable Diffusion [36]. Notably, their approach employs an integrated prediction network to estimate the number of decoding steps and quantization parameters. In contrast, our method does not require any training and estimates additional parameters as well. Another approach by Careil *et al*. [12] fine-tunes a pre-trained text-conditioned latent diffusion model. Conditioned on text provided by a pre-trained image captioning model, their proposed model (PerCo) restores details lost during the quantization of the latent, reconstructing high-realism images at very low bitrates.

2.3. Sampling of diffusion models

The original formulation of Denoising Diffusion Probabilistic Models (DDPM) [25] requires the same number of steps during sampling as was used during training. However, most works now employ alternative sampling schemes, such as Denoising Diffusion Implicit Models (DDIM) [37], which can generate images using significantly fewer sampling steps.

The DDIM sampling process is governed by several hyperparameters that influence the final result. The number of DDIM steps controls the number of timesteps for which the diffusion model is sampled. Furthermore, η controls the probabilistic component of the sampling, with $\eta = 0$ representing the fully deterministic DDIM and $\eta = 1$ representing the probabilistic DDPM. For more details, refer to the original DDIM paper [37]. Another parameter to consider is γ, which represents the standard deviation of the initial noise distribution $\mathcal{N}(0, \gamma^2 I)$ used at the beginning of the diffusion sampling process.

CDC [40] and PerCo [12] diffusion-based compression methods are evaluated in this paper and both use DDIM for sampling by default. In this work, we only consider the numeric DDIM parameters γ, η, and the number of DDIM steps. For the Stable Diffusion [36]-based PerCo [12], we also consider the guidance-scale parameter, which controls the model's adherence to the text conditioning.

3. Proposed method

Our approach is based on existing diffusion-based image compression methods and does not require any additional training. Although most diffusion-based methods use sensible default parameters, such as the number of DDIM steps, these values may not be optimal for any specific image.

We propose improving the quality of the reconstruction by selecting the optimal decoding parameters for each image individually. This requires additional computational effort during encoding; however, the decoding process of the underlying learned image compression method remains mostly unchanged. Since most images are only compressed once, we argue that it is worthwhile to invest the additional effort during encoding to enhance the quality of the decoded image. A conceptual overview of the proposed method is provided in Fig. 1.

3.1. Optimizing DDIM parameters during encoding

Since the final result significantly depends on the parameters chosen for the sampling procedure, we propose investing additional computational effort during the encoding process to store the optimal parameters for each image. Although well-chosen default diffusion parameters provide adequate results, optimizing these parameters for specific images can further improve reconstruction quality and, in some cases, expedite the decoding process by using fewer sampling steps.

In this work, we consider the parameters of two different diffusion-based image compression methods: CDC [40] and PerCo [12]. For CDC, we focus on the parameters γ, η, and DDIM steps. As PerCo uses a text-conditioned latent diffusion model, we use DDIM steps and the guidance scale. We limit the possible options explored to a reasonable range around the default parameters.

Future approaches might also consider optimizing other parameters, such as the timestep schedule.

3.2. Optimization criterion

Selecting the *best* parameters for decoding requires choosing an appropriate optimization metric. There is an inherent tradeoff between perceptual quality and distortion, as highlighted by previous studies [9]. Therefore, the optimal parameters for the diffusion process are subjective and depend on the targeted tradeoff between perceptual quality and distortion. Solely optimizing for PSNR can significantly degrade perceptual quality, and vice versa.

Following the existing literature, we evaluate our results using LPIPS [41] as a perceptual measure and PSNR as a distortion measure. Although diffusion models are optimized to produce images of good perceptual quality, there are diminishing returns when optimizing solely for LPIPS, with an increasing loss in quality in terms of PSNR.

To balance perceptual quality and distortion, we propose an optimization criterion that combines both metrics. The optimization objective is given by the following tradeoff term, which we seek to maximize:

$$\mathcal{T} = (1-\alpha) \cdot PSNR^{\uparrow} - \alpha \cdot LPIPS^{\downarrow} \qquad (1)$$

Here, $\alpha \in [0, 1]$ controls the relative importance of PSNR versus LPIPS. When $\alpha = 1$, this is equivalent to minimizing LPIPS, while $\alpha = 0$ is equivalent to maximizing PSNR. Since LPIPS typically operates on a much smaller scale than PSNR, we expect α to be close to 1 in practice.

For our experiments, we selected $\alpha = 0.992$, which provides a good balance between the two metrics, as shown in Fig. 3. This means that a 1 dB increase in PSNR is considered equivalent to a reduction in LPIPS of about -0.008.

3.3. Note on additional bit-cost

The added bits-per-pixel (BPP) overhead of transmitting the parameters for the diffusion process is negligible. Even assuming full-precision 32-bit values for each parameter, the additional cost is minimal—about 12 bytes for all three parameters. Given the overall size of a high-resolution image, the increase in BPP is insignificant. In practice, even allocating 8 bits for every parameter is more than enough (limiting the parameters to 256 different values). Thus, the overall impact on the bits-per-pixel is negligible (i.e. < 0.0003 bpp for Kodak images).

4. Experiments

4.1. Experimental setup

Datasets For most of our experiments and ablations, we use the **Kodak** [15] dataset, which contains 24 images with a resolution of 512 by 768 pixels. We also provide additional results for the **DIV2K** [1] validation set, which contains 100 images with a width of 2040 pixels and varying height, as well as the validation set of the **CLIC2022** [13] dataset, which contains 30 images with the larger side being 2048 pixels.

Measures To assess the effectiveness of our proposed approach, we use two standard image quality metrics: **PSNR** and **LPIPS** [41] (using AlexNet).

Evaluated methods We apply our approach to two different diffusion-based image compression methods, with publicly available codes and pre-trained checkpoints. The primary method for most of our experiments is the **CDC**[2] model [40]. Additionally, we provide results for the Stable-Diffusion-based **PerCo**[3] [12].

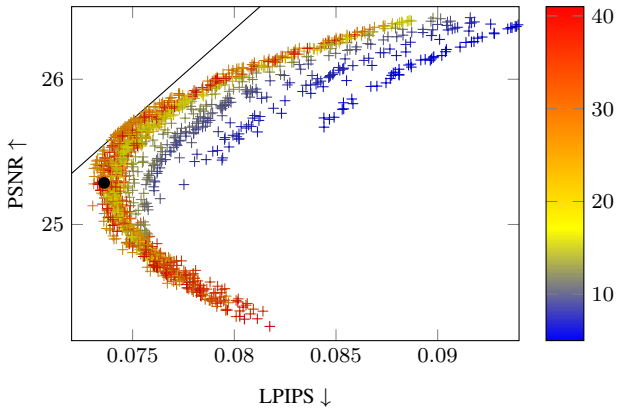

Figure 3. **Different configurations of CDC** x_0 in terms of PSNR and LPIPS for the first Kodak image. Color represents the amount of DDIM steps. The CDC x_0 default is marked as **a black dot**. The tradeoff $\mathcal{T}$ between PSNR and LPIPS for $\alpha = 0.992$ is shown as **a black line**.

4.2. Effect of DDIM parameters for CDC

In order to determine the benefits of per-image parameter optimization, we use the mid-quality checkpoint of the CDC x_0 model (optimized for perceptual quality) and perform a grid search over the DDIM parameters. We limit our search grid, as detailed in Tab. 1 and evaluate the resulting 3600 parameter combinations on the Kodak images. For every configuration, we sample the model twice and average the result.

In Fig. 3 we show the effect of different sampling parameters on the reconstruction quality of the CDC [40] methods for the first image of the Kodak image suite. The default configuration of CDC performs very well across the different images, but it is rarely the optimal setting for any particular image, regardless of the choice for α. Although there are some general trends, such as the use of more DDIM steps for better perceptual quality, the optimal parameters vary significantly between images. For many images, it is beneficial for performance in LPIPS to use more than the default 17 DDIM sampling steps, while fewer steps are sufficient when optimizing for PSNR. By finding and transmitting/encoding these parameters for the individual image, the reconstruction quality can be increased at virtually no cost in terms of bits-per-pixel. We provide a more detailed analysis in the supplementary material.

[2]CDC: https://github.com/buggyyang/CDC_compression

[3]PerCo: https://github.com/Nikolai10/PerCo

On the Kodak images, the parameter optimization allows for an average improvement of -0.002 in terms of LPIPS (for $\alpha = 0.992$) at roughly the same PSNR, compared to the default configuration of CDC. When optimizing only for LPIPS ($\alpha = 1.0$), the improvement increases to -0.004, with a reduction in PSNR of -0.295dB on average. Similarly, when optimizing for PSNR ($\alpha = 0$), we observe an average increase in PSNR of 1dB, while increasing the average LPIPS by 0.015. This effect can also be seen visually, as shown in Fig. 2, where images optimized for PSNR are slightly more blurry, whereas images optimized for LPIPS can be slightly more noisy.

Parameter	Min	Max	Step	Default (CDC x_0)
DDIM steps	5	41	2	17
γ	0.0	1.0	0.05	0.8
η	0.0	0.5	0.05	0.0

Table 1. **Parameter-bounds for the CDC x_0 models.**

4.3. Blackbox optimization

Performing a full grid search over all DDIM parameters is very resource intensive. To address this, we compare a variety of blackbox optimization techniques that reduce the number of required iterations while still yielding high-quality results.

A simple random search, which randomly samples a fixed number of configurations of the parameter space, serves as a simple baseline. Additionally, we compare three different surrogate models for Bayesian optimization.

- **Gaussian Processes (GP)**, implemented in Skopt[4].
- **Hyperbands (HB)** [14], implemented in blackboxopt[5].
- **Probabilistic Random Forests (PRF)**, implemented in OpenBox [29].

All of these methods are evaluated with 10, 20, 30, 50, 100 and 200 iterations. We select our optimization criterion $\mathcal{T}$ with $\alpha = 0.992$. This choice of α was primarily designed to optimize both LPIPS and PSNR compared to the default configuration of CDC [40]. However, other values for α are also reasonable and depend on the specific use case.

Results We present our results on the Kodak dataset in Fig. 4, where the grid-search result is labeled as *Grid*. The dotted line indicates the targeted tradeoff with $\alpha = 0.992$. Any point to the top-left of this line is considered a better result according to the criterion $\mathcal{T}$. For a direct comparison of the different methods w.r.t. $\mathcal{T}$ and the number of iterations, see Fig. 6.

[4]Skopt: https://scikit-optimize.github.io

[5]Blackboxopt: https://github.com/boschresearch/blackboxopt

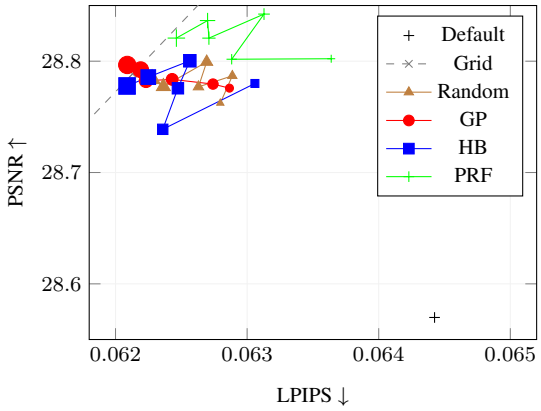

Figure 4. **Comparing CDC x_0 default with CDC optimized by different methods** for up to 200 iterations, in terms of average PSNR and average LPIPS on Kodak [15].

Even Random search (*Random*) already provides clear improvements over the default configuration using significantly fewer iterations. More sophisticated methods—such as Bayesian Optimization using Gaussian Processes (GP)—achieve similar or better results with even fewer iterations.

Generally, increasing the number of iterations has diminishing returns. Bayesian Optimization using GP already almost matches the performance of the grid search with only 100 iterations and provides a decent improvement over the default configuration with 30-50 iterations. With 200 iterations it even outperforms the grid search, which is limited by the predefined grid of parameters.

BD-rate As seen in Fig. 5, the performance gains remain constant across different bitrates. Using 30 iterations of Bayesian Optimization using GP, we achieve consistent improvements in both PSNR and LPIPS on the Kodak dataset with BD-rate [8] improvement of -4.6% for PSNR and -4.8% for LPIPS.

We also compare with the results for HiFiC [31] which serves as a baseline for generative compression methods and BPG [7] for non-learned compression. HiFiC performs significantly better than CDC in terms of PSNR and LPIPS. While our method improves the performance of CDC in these metrics, it does not reach the performance of HiFiC.

Dataset optimized parameters For completeness, we also evaluate the configuration that performs best according to $\mathcal{T}$ across all images in the Kodak dataset. This configuration achieves better LPIPS compared to the default configuration (-0.02), with similar PSNR ($+0.03$dB). However, it still falls short of the performance achieved by our proposed per-image optimization approach.

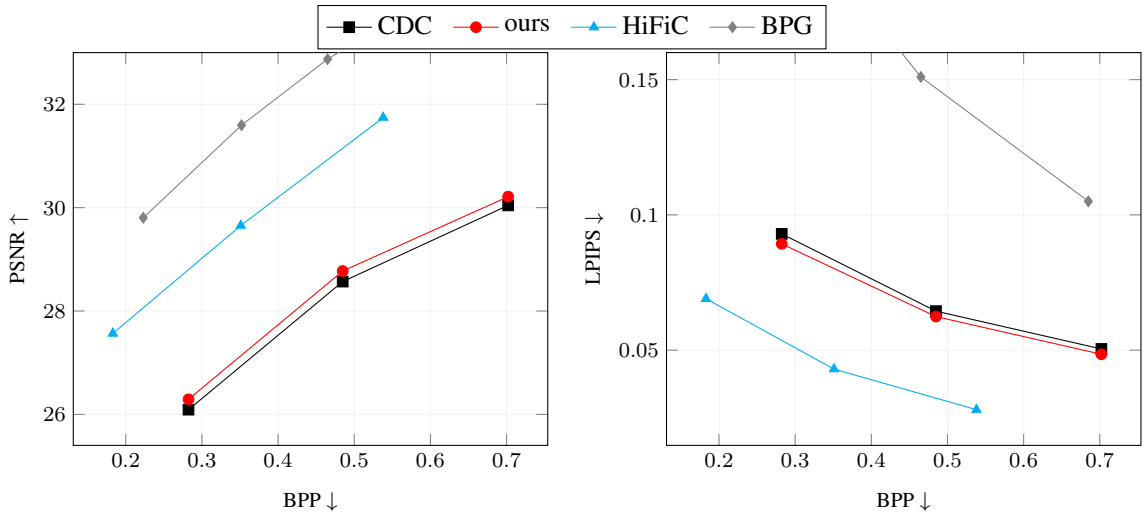

Figure 5. **CDC x_0 default [40] vs. our optimized CDC model** in terms of average PSNR and average LPIPS on Kodak [15] dataset. For *ours* we employed Bayesian Optimization using Gaussian Processes and 30 iterations.

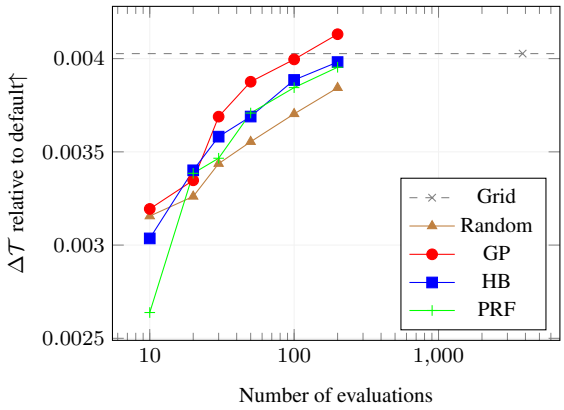

Figure 6. **Differences for various optimization methods to the CDC x_0 default [40]** configuration with regard to the optimization criterion $\mathcal{T}$ vs. number of iterations on Kodak [15].

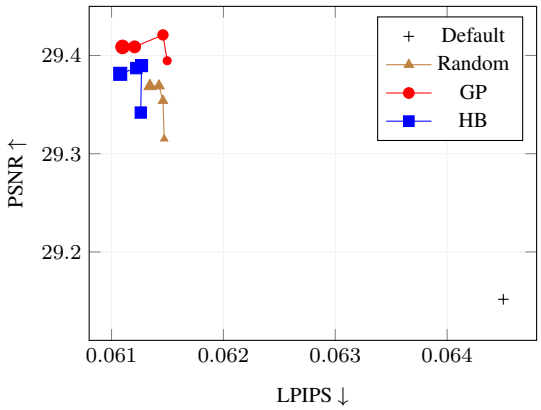

Figure 7. **CDC x_0 default [40] vs. CDC optimized** with different methods with 10, 20, 30 or 50 iterations in terms of average PSNR and average LPIPS on DIV2K [1].

4.4. Evaluation on high-resolution images

To test the scalability of our approach, we evaluate it on the validation splits of the high-resolution **DIV2K** [1] and **CLIC2022** [13] datasets. We apply three blackbox optimization methods: Bayesian Optimization with Gaussian Processes (GP), Hyperbands (HB), and Random Search, with a maximum of 50 iterations.

As shown in Fig. 7 and Fig. 8, we observe consistent improvements in both PSNR and LPIPS compared to the default configuration of CDC. Performance trends mirror those seen on the Kodak dataset: Bayesian Optimization with GP delivers the largest gains, followed by Hyperbands and Random Search.

4.5. Extreme tradeoffs

For our main experiments, we select $\alpha = 0.992$, which improves over the default configuration of CDC in both PSNR and LPIPS. However, depending on the application, other values of α may be preferable to emphasize either perceptual quality or distortion.

In Fig. 9, we compare results for extreme tradeoff settings with $\alpha = 0$ and $\alpha = 1$ against the default of CDC x_0 [40] on the Kodak dataset across varying bitrates. As expected [9], optimizing for either PSNR ($\alpha = 0$) or LPIPS ($\alpha = 1$) results in a drop in the other metric compared to the default configuration.

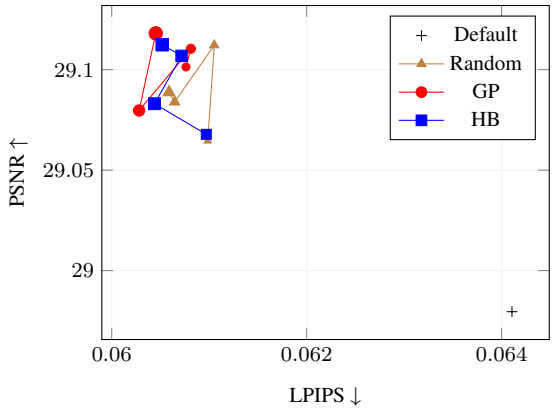

Figure 8. **CDC x_0 default [40] vs. CDC optimized** with different methods with 10, 20, 30, or 50 iterations in terms of average PSNR and average LPIPS on CLIC2022 [13].

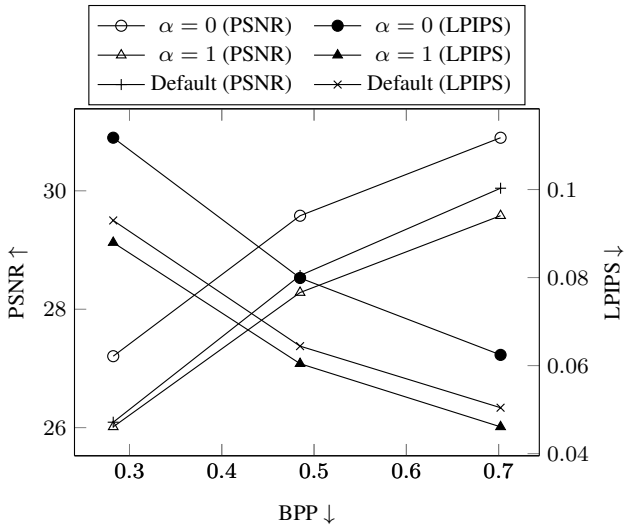

Figure 9. **Extreme tradeoffs for CDC x_0 [40]** with $\alpha = 0$ and $\alpha = 1$ and the default configuration. For the optimized configurations we employed Bayesian Optimization using Gaussian Processes and 30 iterations. The proposed method allows to select a tradeoff between PSNR and LPIPS performance without any retraining. However, when optimizing only for a single metric (PSNR or LPIPS), performance in the other metric drops.

4.6. Computational effort

Since sampling a diffusion model is many times slower than the normal encoding process (which normally takes less than one second), our method requires significantly more time during encoding. However, decoding remains unaffected apart from potential adjustments to the number of DDIM sampling steps.

On an RTX 4090, sampling a DIV2K [1] image with the CDC x_0 model [40] using 17 steps takes approximately 5.6 seconds. The time required for the optimization depends on the number of iterations and the average step count. For example, 10 iterations of Bayesian Optimization with GP (at $\alpha = 0.992$) use an average of 23.7 steps, resulting in an average optimization time of 82 seconds per image. This scales linearly with the number of optimization iterations, but is further influenced by the selection of α. As discussed before, higher values of α often lead to more DDIM steps, which in turn increases the optimization time.

Given that images are typically only encoded once and then stored, this time investment can be justified by the improved reconstruction quality that our method delivers.

Decoding steps The number of DDIM steps used during decompression is a parameter that will influence the decoding speed of the learned image compression method. In many cases, it might be beneficial to perceptual quality to increase the number of DDIM steps. As a consequence, optimizing the parameter configuration for LPIPS will often result in a higher number of DDIM steps than the default (given the parameter bounds selected in Tab. 1). For example, using significantly fewer DDIM steps than the default (17) harms performance in terms of LPIPS, as shown for the first Kodak image in Fig. 3. Furthermore, most of the best configurations use more than the default 17 steps. However, increasing the number of steps also slows down the decoding process, which can be undesirable when decoding speed is a critical factor.

To evaluate the tradeoff between quality and decoding efficiency, we repeat the optimization while constraining the maximum number of DDIM steps to 17. As shown in Fig. 10, limiting the maximum number of decoding steps reduces performance in both PSNR and LPIPS relative to the unconstrained search, but still improves over the CDC default.

If the number of DDIM steps is a concern, it might also be worthwhile to include them in the optimization criterion, balancing possible performance improvements against the increase in decoding time.

Optimizing on crops Preliminary experiments on a smaller image crop show some further potential to reduce the computational impact. Optimizing using Gaussian Processes for 30 iterations on a 256x256 center-crop of DIV2K reduces the time needed significantly, to less than 10s per image. However, this comes at the cost of reduced performance. Compared to the default configuration of CDC, this approach achieves a $+0.45$dB increase in PSNR at the cost of a small increase in LPIPS ($+0.0016$) when evaluated on the full-size DIV2K images.

4.7. Generalizability to other methods

In addition to our primary experiments with the CDC [40] model, we also evaluate the effectiveness of our approach

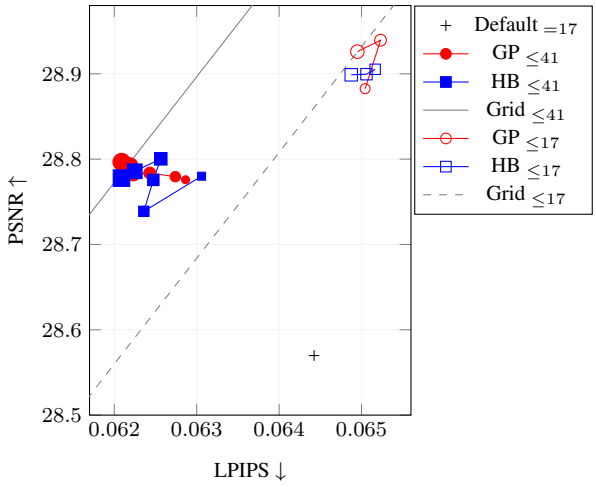

Figure 10. **CDC x_0 default [40] vs. our CDC configurations** optimized using Gaussian Processes or Hyperbands in terms of average PSNR and average LPIPS on Kodak with $\alpha = 0.992$. Comparing the normal optimization settings to limiting the number of DDIM sampling steps to 17.

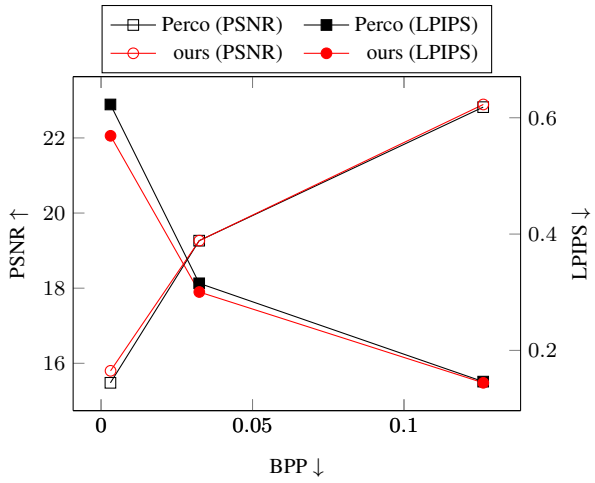

Figure 11. **PerCo [12] vs. our optimized PerCo model** in terms of average PSNR and average LPIPS on Kodak. For *ours* we employed Bayesian Optimization using Gaussian Processes and 30 iterations. Significant improvements are achieved for low (< 0.1) bpp.

on the PerCo [12] method. For this model, we optimize two easily accessible parameters: the number of DDIM steps ($[2, 60]$) and the guidance scale ($[0.0, 5.0]$), which controls how strongly the diffusion model adheres to the text-conditioning information. As before, we optimize for the tradeoff objective $\mathcal{T}$ with $\alpha = 0.992$ using 30 iterations of Bayesian Optimization with GP.

Results As shown in Fig. 11, our approach yields improvements in both LPIPS and PSNR. Specifically, we achieve an improvement in BD-Rate [8] of -3.8% for PSNR and -16.3% for LPIPS compared to the default configuration.

Interestingly, at the highest bitrate, the gains in PSNR and LPIPS are modest ($+0.07$ and -0.002). However, the average number of decoding steps required decreases significantly from 20 to 13 steps, resulting in faster decoding. At the lowest bitrate, the improvements are much more substantial ($+0.3$ for PSNR and -0.06 for LPIPS), although this comes at the cost of using a higher number of DDIM steps (increasing to 40 on average).

These results confirm the general applicability of our per-image optimization strategy across different generative compression frameworks.

5. Conclusion

In this work, we propose using blackbox optimization techniques during image compression for diffusion-based methods to optimize the sampling parameters used during decoding. By applying additional compute once during the compression process, we show that it is possible to improve the final reconstruction of existing diffusion-based image compression methods at virtually no cost in terms of bits-per-pixel (less than 3 additional bytes per image). This approach also allows users to decide on the tradeoff between perceptual quality and distortion when encoding the image and does not require any sort of further model training.

We demonstrate that our approach can be applied to different pre-trained diffusion-based learned image compression methods (*e.g.*, CDC [40], PerCo [12]), showing consistent improvements on Kodak [15], as well as DIV2K [1] and CLIC2022 [13].

5.1. Future work

Although our approach shows promising results, it currently requires multiple samplings of the diffusion model during encoding, which increases the computational cost during that phase. Future work could explore methods to enable the heuristic prediction of optimal decoding parameters, potentially reducing the time and resources needed during the compression process.

Acknowledgments

This work was partly supported by the Alexander von Humboldt Foundation.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pat-*

tern recognition workshops, pages 126–135, 2017. 4, 6, 7, 8

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF ICCV*, pages 221–231, 2019. 2

[3] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22324–22333, 2023. 2

[4] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017. 2

[5] Johannes Ballé, Philip A Chou, David Minnen, Saurabh Singh, Nick Johnston, Eirikur Agustsson, Sung Jin Hwang, and George Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020. 2

[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*, 2018. 1, 2

[7] Fabrice Bellard. Bpg image format. https://bellard.org/bpg/, 2015. 1, 2, 5, 13

[8] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. itu-t q. 6/sg16, doc. vceg-m33. In *15th Meeting. Austin, Texas*, 2001. 5, 8

[9] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018. 2, 3, 6

[10] Jonas Brenig and Radu Timofte. Higher fidelity perceptual image and video compression with a latent conditioned residual denoising diffusion model. In *European Conference on Computer Vision*, pages 194–210. Springer, 2024. 1, 3

[11] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1, 2

[12] Marlene Careil, Matthew J. Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 4, 8

[13] CLIC. Workshop and challenge on learned image compression (clic). http://clic.compression.cc/2022/, 2022. 4, 6, 7, 8

[14] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *ICML*, pages 1436–1445, 2018. 5

[15] Rich Franzen. Kodak lossless true color image suite. https://r0k.us/graphics/kodak/, 1999. 4, 5, 6, 8, 1

[16] Noor Fathima Ghouse, Jens Petersen, Auke Wiggers, Tianlin Xu, and Guillaume Sautiere. A residual diffusion model for high perceptual quality codec augmentation. *arXiv preprint arXiv:2301.05489*, 2023. 1, 3

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[18] Vivek K Goyal. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5):9–21, 2001. 2

[19] Zongyu Guo, Zhizheng Zhang, Runsen Feng, and Zhibo Chen. Causal contextual prediction for learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(4):2329–2341, 2021. 2

[20] Wang Guo-Hua, Jiahao Li, Bin Li, and Yan Lu. EVC: Towards real-time neural image compression with mask decay. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[21] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14771–14780, 2021. 1

[22] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5718–5727, 2022. 1, 2

[23] Dailan He, Ziming Yang, Hongjiu Yu, Tongda Xu, Jixiang Luo, Yuan Chen, Chenjian Gao, Xinjie Shi, Hongwei Qin, and Yan Wang. Po-elic: Perception-oriented efficient learned image coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1764–1769, 2022. 1, 2

[24] 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. *Advances in neural information processing systems*, 30, 2017. 1, 2

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, pages 6840–6851, 2020. 1, 2, 3

[26] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. 3

[27] Emiel Hoogeboom, Eirikur Agustsson, Fabian Mentzer, Luca Versari, George Toderici, and Lucas Theis. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023. 1, 3

[28] Shoma Iwai, Tomo Miyazaki, Yoshihiro Sugaya, and Shinichiro Omachi. Fidelity-controllable extreme image compression with generative adversarial networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8235–8242. IEEE, 2021. 2

[29] Huaijun Jiang, Yu Shen, Yang Li, Beicheng Xu, Sixian Du, Wentao Zhang, Ce Zhang, and Bin Cui. Openbox: A python toolkit for generalized black-box optimization. *Journal of Machine Learning Research*, 25(120):1–11, 2024. 5

[30] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14388–14397, 2023. 2

[31] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020. 1, 2, 5, 14

[32] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020. 1, 2

[33] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 1, 2

[34] Peirong Ning, Wei Jiang, and Ronggang Wang. Hflic: Human friendly perceptual learned image compression with reinforced transform. In *2023 International Conference on Communications, Computing and Artificial Intelligence (CCCAI)*, pages 188–194. IEEE, 2023. 1, 2

[35] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision*, pages 303–319. Springer, 2024. 1, 3

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 3

[37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3

[38] Lucas Theis, Tim Salimans, Matthew D Hoffman, and Fabian Mentzer. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022. 1, 3

[39] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 1, 2

[40] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. In *Advances in Neural Information Processing Systems*, pages 64971–64995, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

[41] 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. 1, 2, 4

[42] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International Conference on Learning Representations*, 2022. 2