PerCo (SD): Open Perceptual Compression

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce PerCo (SD), a perceptual image compression method based on Stable Diffusion v2.1, targeting the ultra-low bit range. PerCo (SD) serves as an open and competitive alternative to the state-of-the-art method PerCo, which relies on a proprietary variant of GLIDE and remains closed to the public. In this work, we review the theoretical foundations, discuss key engineering decisions in adapting PerCo to the Stable Diffusion ecosystem, and provide a comprehensive comparison, both quantitatively and qualitatively. On the MSCOCO-30k dataset, PerCo (SD) demonstrates improved perceptual characteristics at the cost of higher distortion. We partly attribute this gap to the different model capacities being used (866M vs. 1.4B). We hope our work contributes to a deeper understanding of the underlying mechanisms and paves the way for future advancements in the field. Code and trained models will be released at `https://github.com/Nikolai10/PerCo`.

1 Introduction

Perceptual compression, sometimes referred to as generative compression [1, 29] or distribution-preserving compression [42], refers to a class of neural image compression techniques that incorporate generative models (*e.g.*, generative adversarial networks [12], diffusion models [39, 15]) into their learning objective. Unlike traditional codecs such as JPEG, they additionally constrain the reconstructions to follow their underlying data distribution [5]. By leveraging powerful generative priors, missing details, such as textures, can be realistically synthesized, thus achieving higher perceptual quality at even lower bit rates. These characteristics make these methods particularly appealing for storage- and bandwidth-constrained applications.

Recently, foundation models [6], large-scale machine learning models trained on broad data at scale, have shown great potential in their adaption to a wide variety of downstream tasks, including ultra-low bit-rate perceptual image compression [32, 22, 8]. Notably, PerCo [8], the current state-of-the-art, is the first method to explore bit-rates from 0.1 down to 0.003bpp. For example, a bit-rate of 0.003bpp translates to approximately 115 bytes for an image of VGA resolution (480×640), which is less the size of a tweet. This is essentially achieved by extending the conditioning mechanism of a pre-trained text-conditional latent diffusion model (LDM) with vector-quantized hyper-latent features. In other words, only a short text description and a compressed image representation are required for decoding. Despite its great potential and fascinating results, PerCo has not been made publicly available. This is arguably due to the fact that PerCo relies on a proprietary LDM based on GLIDE [30].

To close this gap and to facilitate further research, we introduce PerCo (SD), an open and competitive alternative to PerCo based on the Stable Diffusion architecture [36], see fig. 1 for visual impressions. In the following, we review the theoretical foundations (section 2), discuss key engineering decisions in adapting PerCo to the Stable Diffusion ecosystem (section 3), and provide a comprehensive comparison, both quantitatively and qualitatively (section 4).

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

2 Background

Neural image compression. Neural image compression uses deep learning/ machine learning techniques to learn compact image representations. This is typically achieved by an auto-encoder-like structure consisting of an encoder E and a decoder D, as well as an optional entropy model P, which are trained jointly in a data-driven fashion. Specifically, E projects the input image x to a quantized latent representation $y = E(x)$, while D attempts to reverse this process $x' = G(y)$. The learning objective is to minimize the rate-distortion trade-off [9], with $\lambda > 0$:

$$\mathcal{L}_{RD} = \mathbb{E}_{x \sim p_X} [\lambda r(y) + d(x, x')]. \tag{1}$$

In eq. (1), the bit-rate is estimated using the cross entropy $r(y) = -\log P(y)$, where P represents a probability model of y. In practice, an entropy coding method based on P is used to obtain the final bit representation, *e.g.*, using adaptive arithmetic coding. The distortion is measured by a full-reference metric $d(x, x')$ that captures the distance of the reconstruction x' to the original input image x. Both terms are weighted by λ, which enables traversing the rate-distortion curve based on application needs. For a more general overview, we refer the reader to [47].

Diffusion models. Diffusion models [39, 15] are a type of generative models that approximate the underlying data distribution by learning the inverse of a diffusion process, which is defined as:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{2}$$

In eq. (2), $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ denotes the joint distribution of all samples generated across the trajectory of the forward diffusion process in T steps from $\mathbf{x}_1$ up to $\mathbf{x}_T$, given the input image $\mathbf{x}_0$. At each step, Gaussian noise is gradually added to the data following a noise schedule β_t, such that $q(\mathbf{x}_T|\mathbf{x}_{T-1}) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. In practice, the forward process can be simulated by $q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$, with $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, which enables the convenient parameterization $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon$, with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The reverse diffusion process is defined as:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}). \tag{3}$$

In eq. (3), $p_\theta(\mathbf{x}_{0:T})$ denotes the joint distribution of all samples generated across the trajectory of the reverse diffusion process in T steps from $\mathbf{x}_T$ up to $\mathbf{x}_0$, with $p(\mathbf{x}_T) = q(\mathbf{x}_T|\mathbf{x}_{T-1}) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$, where $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ approximates the true denoising distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ using a parametric Gaussian model (*e.g.*, time-conditional U-Net).

The learning objective is based on the variational lower bound, adapted to the diffusion setting:

$$\mathbb{E}_{q(\mathbf{x}_0)} [-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] := L. \tag{4}$$

Ho *et al*. [15] showed that this objective can be further simplified to a noise prediction task, neglecting multiplicative constants, which is widely used in practice and constitutes the foundation of the earlier variants of Stable Diffusion v1.1-v1.5 [36]:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right]. \tag{5}$$

An extension of eq. (5) to the conditional case can be achieved by adding side information z (*e.g.* text descriptions of $\mathbf{x}_0$) to the input of the noise prediction network $\epsilon_\theta(\mathbf{x}_t, z, t)$.

Latent diffusion models. Latent diffusion models (LDMs) [36] are a subset of diffusion models that formulate the learning objective eq. (4) in a latent space (*e.g.*, of a pre-trained auto-encoder), rather than in the pixel space. This change is primarily to reduce the high computational complexity during both training and sampling.

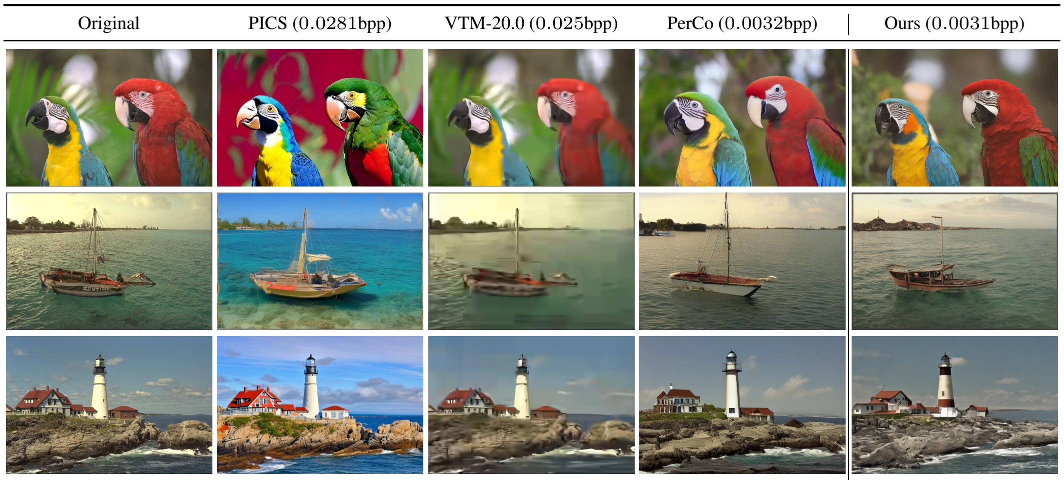


Figure 1: Visual comparison of PerCo (SD) to PICS [22], VTM-20.0, the state-of-the-art non-learned image codec, and PerCo [8]. Notably, PerCo and PerCo (SD) achieve an order of magnitude lower bits per pixel (bpp) compared to competing methods. **Best viewed electronically.**

3 Perceptual compression

Perceptual compression, sometimes referred to as generative compression [1, 29] or distribution-preserving compression [42], extends the traditional rate-distortion objective eq. (1) by an additional constraint that forces the reconstructions to follow the underlying data distribution, leading to the rate-distortion-perception trade-off [5].

The key idea of PerCo is to formulate the distortion term $d(x, x')$ in eq. (1) within a pre-trained text-conditional LDM, which serves as a powerful generative prior. This type of formulation has recently been also referred to as generative latent coding (as opposed to the regular transform coding paradigm in the pixel space) and is motivated by the fact that the latent space typically has greater sparsity, richer semantics, and better alignment with human perception [18].

3.1 Model overview

In this section, we provide a short model overview of PerCo (fig. 2). The core component is a conditional diffusion model (highlighted yellow) based on a proprietary variant of GLIDE [30], which we intend to replace with an open alternative (section 3.2).

Encoding. To better adapt the LDM to the compression setting, PerCo extracts side information at the encoder side of the form $z = (z_l, z_g)$, where z_l and z_g correspond to local and global features, respectively. In PerCo, z_l corresponds to vector-quantized (VQ) hyper-latent features, extracted by the hyper-encoder, and z_g corresponds to image captions extracted by a pre-trained large language model (BLIP-2 [24]). Both z_l and z_g are losslessly compressed using arithmetic coding and Lempel-Ziv coding. In PerCo, a uniform coding scheme is used to model z_l, *i.e.* the rate term $r(y)$ in eq. (1) can be ignored. Various bit-rates can be achieved by using different configurations for the spatial size, denoted by $(h \times w)$, and the codebook size V: $r(z_l) = \frac{hw \log_2 V}{HW}$ bpp, where $(H \times W)$ denotes the input size. The final bit-rate is obtained by $r(z) = r(z_l) + r(z_g)$, where $r(z_g)$ is controlled by the number of tokens (32 in the official configuration).

Decoding. At the decoder side, the compressed representations (z_l, z_g) are decoded and subsequently fed into the conditional diffusion model: z_l is upsampled using linear interpolation if required, and spatially concatenated with $\mathbf{x}_t$, the input of the first convolution of the denoising network. This is achieved by extending the pre-trained kernel with randomly initialized weights. z_g is passed to a pre-trained text encoder that computes textual embeddings, which are incorporated into the denoising network using cross-attention layers [43].

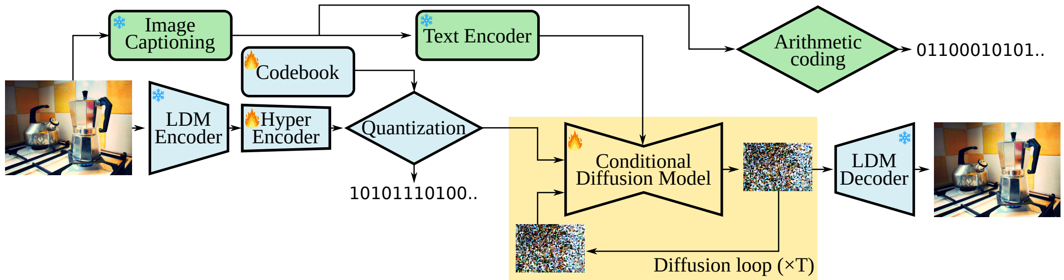


Figure 2: PerCo model overview, adopted from [8]. During training, the hyper-encoder, codebook, and diffusion model are trained, whereas all other components are fixed.

3.2 PerCo (SD)

Our general goal is to provide an open alternative to PerCo, with ideally highly competitive performance, while following the official design decisions as closely as possible.

Challenges. Among the many available Stable Diffusion options, we choose version 2.1[1], which is similar to the proprietary GLIDE-based LDM, a native v-prediction model [37]. Prior to the adoption, we had the following concerns: i) the LDM of Stable Diffusion v2.1 is much smaller than the one used in PerCo. In SD v2.1, we have 866M, 84M, and 340M parameters for the denoising network, auto-encoder, and text-encoder, respectively. PerCo uses a 1.4B-parameter denoising network ($1.62\times$), a 4.7B-parameter text encoder ($13.82\times$), and an 83M-parameter auto-encoder. ii) SD v2.1 by default uses a larger input resolution (768×768) compared to the target resolution (512×512). iii) Finally, it remains unclear how the proprietary LDM performs in comparison to existing off-the-shelf models, given the current analysis of the consistency-diversity-realism fronts [2].

Core design decisions/ deviations. In this section, we discuss the core changes over the official configuration. A full detailed comparison is provided in table 1.

- Training steps. We limit the number of training steps to 150k iterations due to resource considerations, which roughly corresponds to 50% of the computation budget of PerCo.
- Peak learning rate. We generally find it beneficial to use small learning rates ($1e-5$).
- U-Net finetuning. We finetune the whole U-Net, which we find to provide slightly better results. We attribute this observation to an initial resolution/ distribution mismatch.
- Extended kernel. We initialize the extended kernel with zeros [49], which encourages the model to gradually incorporate the additional conditional information (z_l).
- VQ-module. We additionally ℓ_2-normalize the codes [48], which we find to be crucial to ensure stable training.

Further considerations. We explored finite-scalar quantization (FSQ) [28] as a simpler alternative to the sensitive codebook learning paradigm. While FSQ does indeed streamline the training process, it falls short of matching the performance of its VQ counterparts. We further investigated the use of LoRa [17] as an alternative to solely fine-tuning the linear layers to somewhat better quantify the issue of catastrophic forgetting [25]. However, this approach did not yield improved results. Lastly, we explored various l_z-conditioning formulations of the diffusion model. We experimented with an additional hyper-decoder, as an alternative to the simple upsample with linear interpolation operation, either by directly using the hyper-decoded local features as input to the diffusion model, or to support auxiliary loss formulations to regularize the hyper-encoder (*e.g.* by enforcing good reconstruction quality of the latent features). In both scenarios, we did not observe additional improvements. This can be partly attributed to the observation that downstream learning tasks yield comparable results in both the latent and pixel spaces [41].

[1] `https://github.com/Stability-AI/stablediffusion`

Table 1: Comparison of the design decisions: PerCo (official) vs. PerCo (SD). Key deviations are highlighted in gray and discussed in the main text.

	PerCo (official)	PerCo (SD)
Training		
Training dataset	OpenImagesV6 [21] (9M)	OpenImagesV6 [21] (9M)
Optimizer	AdamW [27]	AdamW [27]
Training steps	5 epochs/ $\approx$ 300k	150k (50%)
Peak learning rate	$1e-4$	$1e-5$
Weight decay	0.01	0.01
Linear warm-up	10k	10k
Batch size	160 (w/o LPIPS), 40 w/ LPIPS	80 w/ LPIPS
U-Net finetuning	linear layers (15%)	all layers
LPIPS auxilliary loss	bit-rates > 0.05bpp	all bit-rates
Text conditioning	drop in 10%	drop in 10%
Finetuning grid	50 steps	1000 steps (unchanged)
Extended kernel	random initialization	zero initialization
VQ-module	improved VQ [48]	improved VQ [48] + cosine similarity
Inference		
Scheduler	DDIM [40]	DDIM [40]
Denoising steps	5 for > 0.05bpp, else 20	20
CFG [16]	3.0	3.0

4 Experimental results

Implementational details. PerCo (SD) is written in PyTorch [33] and built around the diffusers library [44]. As such, PerCo (SD), in general, allows for testing various Stable Diffusion versions (v1, v2) out-of-the-box, with minor adjustments. We use a single DGX H100 system to train all models in a distributed, multi-GPU ($8\times$ H100) setup using full precision. To further accelerate training, all captions are pre-computed and loaded into memory during runtime. PerCo (SD) is also accompanied by a simplified Google Colab demo, which enables training on a single A100-GPU.

Evaluation setup. We adopt the same evaluation protocol as in PerCo [8]. We consider the Kodak [20] and the MSCOCO-30k [7] datasets, which contain 24 and 30k images at resolution 512×768 and 512×512, respectively. We report the FID [14] and KID [4] as a measure of perception, the MS-SSIM [45] and LPIPS [50] as a measure of distortion, the CLIP-score [13] as a measure of global alignment of reconstructed images and ground truth captions (in PerCo: BLIP 2 generated captions) and finally, the mean intersection over union (mIoU) as a measure of semantic preservation [38]. For more details, we refer the reader to [8, Section 4.1 and A Experiment details].

4.1 Main results

In this section, we quantitatively compare the performance of PerCo (SD v2.1) to the officially reported numbers (fig. 3). All models were trained using a reduced set of optimization steps (150k, 50% of the official configuration). Note that the performance is bounded by the LDM auto-encoder, denoted as SD v2.1 auto-encoder.

We generally obtain highly competitive results in terms of perception (FID, KID), especially for the ultra-low bit rates. For our lowest bit rate configuration, 0.0036bpp, we obtain considerably better FID and KID scores compared to PerCo at 0.0041bpp (4.49 vs. 5.49 and 0.0009 vs. 0.0011). This benefit comes, however, at the cost of consistently lower image fidelity (MS-SSIM, LPIPS). Besides the notorious rate-distortion-perception trade-off [5], we attribute this gap to the different model capacities being used (LDM 866M vs. 1.4B, Text encoder 340M vs. 4.7B). Intuitively, PerCo attempts to recover the latent image code from only a short text description and vector-quantized hyper-latent features, which arguably requires a sophisticated generative prior. We further obtain superior CLIP and mIoU scores. PerCo (SD) tends, however, to use slightly shorter, perhaps more generic text descriptions (0.00165bpp vs. 0.0022bpp) due to presumably different BLIP 2 configurations. As

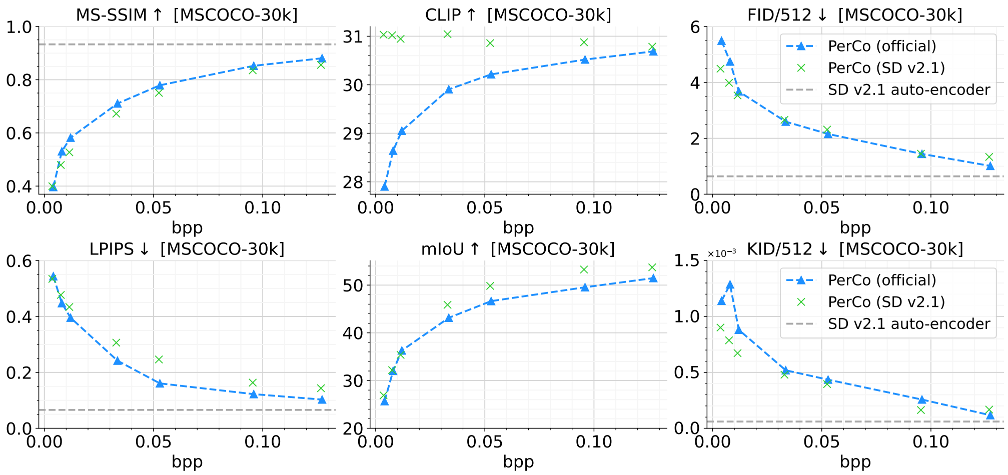


Figure 3: Quantitative comparison: PerCo (official) vs. PerCo (SD)

such, the CLIP scores might not be directly comparable. In our case, the CLIP scores also seem less dependent on the bit rate.

Finally, it is worth mentioning that we did not apply post-hoc filtering methods [19] to further boost performance. Like all probabilistic methods, PerCo (SD) is sensitive to the initial random seed. Therefore, future work should report the mean and standard deviation across multiple test runs.

4.2 Ablations/ further results

Both PerCo and PerCo (SD) rely on the DDIM scheduler. We find that the default configuration remains a good choice (classifier-free guidance scale of 3 and 20 sampling steps). For additional details and further results, see appendix A.

5 Related work

We limit this section to concurrent approaches for ultra-low bit-rate image compression that leverage powerful pre-trained foundation models and refer the reader for a broader overview to [8, Related work]. Conditioning modalities explored in these methods include prompt inversion and compressed sketches [46, 22], text descriptions obtained by a commercial large language model (GPT-4 Vision [31]), semantic label maps and compressed image features [23], CLIP image features and color palettes [34, 3], and textual inversion combined with a variant of classifier guidance, dubbed compression guidance [11, 10, 32]. Relic *et al.* [35] takes a slightly different approach by treating the removal of quantization error as the denoising task, aiming to recover lost information in the transmitted image latent. In all cases, some form of Stable Diffusion [36] is used (ControlNet [49], DiffBIR [26] and Stable unCLIP [36]), with no changes to the official weights.

6 Conclusion

In this paper, we introduced PerCo (SD), an open and competitive alternative to PerCo, the current state-of-the-art for ultra-low bit-rate image compression. We revisited the theoretical foundations, described our engineering efforts in translating PerCo to the Stable Diffusion ecosystem, and provided an in-depth analysis of both approaches. We hope our work contributes to a deeper understanding of the underlying mechanisms and paves the way for future advancements in the field. Code and models will be released at `https://github.com/Nikolai10/PerCo`.

Limitations. PerCo (SD) inherits the limitations described in the original work. In its current state, PerCo (SD) can only handle medium-sized images (*e.g.*, 512×512). Possible solutions have been discussed in [41, section 5]. Finally, PerCo (SD) is based on a much smaller LDM (866M vs. 1.4B) - we leave the exploration of more powerful foundation models for future work.

References

[1] 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 International Conference on Computer Vision (ICCV)*, 2019.

[2] P. Astolfi, M. Careil, M. Hall, O. Mañas, M. Muckley, J. Verbeek, A. R. Soriano, and M. Drozdzal. Consistency-diversity-realism pareto fronts of conditional image generative models. *arXiv: 2406.10429*, 2024.

[3] Tom Bachard, Tom Bordin, and Thomas Maugey. CoCliCo: Extremely low bitrate image compression based on CLIP semantic and tiny color map. In *PCS 2024 - Picture Coding Symposium*, 2024.

[4] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

[5] Yochai Blau and Tomer Michaeli. Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[6] Rishi Bommasani et al. On the opportunities and risks of foundation models. *arXiv: 2108.07258*, 2021.

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] 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.

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

[10] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

[11] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023.

[12] 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*, 2014.

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[14] 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*, 2017.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[17] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[18] Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[19] Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. If at first you don't succeed, try, try again: Faithful diffusion-based text-to-image generation by selection. *arXiv: 2305.13308*, 2023.

[20] Eastman Kodak. Kodak lossless true color image suite (PhotoCD PCD0992).

[21] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[22] Eric Lei, Yiğit Berkay Uslu, Hamed Hassani, and Shirin Saeedi Bidokhti. Text+ sketch: Image compression at ultra low rates. In *ICML 2023 Workshop on Neural Compression: From Information Theory to Applications*, 2023.

[23] Chunyi Li, Guo Lu, Donghui Feng, Haoning Wu, Zicheng Zhang, Xiaohong Liu, Guangtao Zhai, Weisi Lin, and Wenjun Zhang. Misc: Ultra-low bitrate image semantic compression driven by large multimodal model. *arXiv: 2402.16749*, 2024.

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[26] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv: 2308.15070*, 2024.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[28] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*, 2024.

[29] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 2020.

[30] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[31] OpenAI, Josh Achiam, et al. Gpt-4 technical report. *arXiv: 2303.08774*, 2024.

[32] Zhihong Pan, Xin Zhou, and Hao Tian. Extreme generative image compression by learning text embedding from diffusion models. *arXiv: 2211.07793*, 2022.

[33] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[35] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. *arXiv: 2303.08774*, 2024.

[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 (CVPR)*, 2022.

[37] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[38] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning Representations*, 2021.

[39] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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

[41] Róbert Torfason, Fabian Mentzer, Eiríkur Ágústsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Towards image understanding from deep compression without decoding. In *International Conference on Learning Representations*, 2018.

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

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[44] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[45] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2003.

[46] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[47] Yibo Yang, Stephan Mandt, and Lucas Theis. An introduction to neural data compression. *Foundations and Trends® in Computer Graphics and Vision*, 2023.

[48] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022.

[49] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[50] 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 (CVPR)*, 2018.

A Appendix / supplemental material

A.1 Additional quantitative results

More results. We provide additional quantitative results on the MSCOC0-30k and Kodak datasets in fig. 4 and fig. 5, respectively. We observe characteristics similar to those of the main results.

Inference speed. We refer the reader to [8, A Experimental details, Inference Speed]. The encoder speed of PerCo (SD) is supposed to be identical to PerCo, as the only difference lies within the LDM, which runs on the decoder side. As the LDM in PerCo (SD) is considerably smaller (866M vs. 1.4B) and based on a similar architecture, we assume that the decoder speed of PerCo (SD) is at least comparable to PerCo.

A.2 Additional visual results

Additional visual comparisons. We provide additional visual results in fig. 6 and fig. 7. We find that PerCo (SD) produces pleasing reconstructions that are comparable to PerCo.

Global conditioning. In fig. 8, we analyze the impact of the global conditioning and show that PerCo (SD) offers similar internal characteristics.

Reconstructions across various bit rates. In fig. 9, we visualize reconstructions with increasing access to local conditioning information.

Semantic preservation. Finally, in fig. 10, we visualize the semantic preservation capabilities of PerCo (SD) across all tested bit-rates.

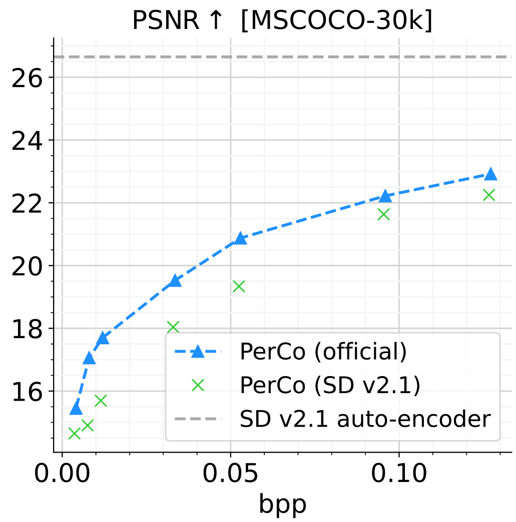


Figure 4: Quantitative comparison on the MSCOCO-30k dataset: PerCo (official) vs. PerCo (SD). We have not tried to tune our model towards better PSNR scores, as these low-level distortion metrics are known to be less meaningful for low rates [8].

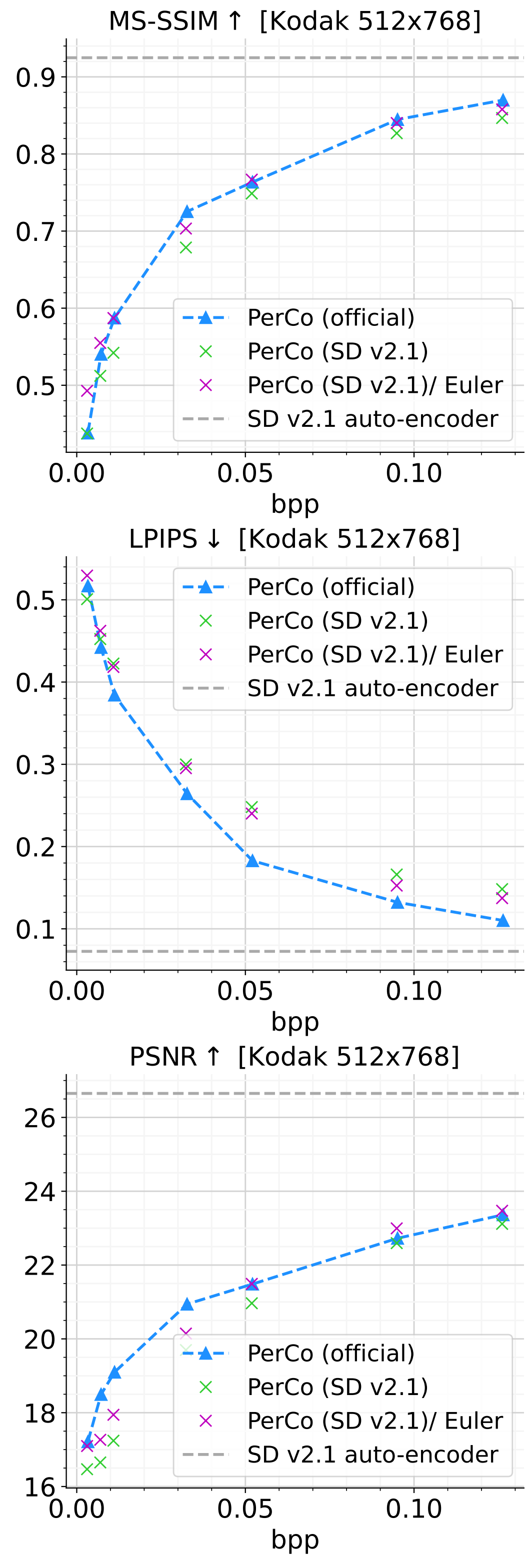


Figure 5: Quantitative comparison on the Kodak dataset: PerCo (official) vs. PerCo (SD). We further show another model configuration based on the `EulerAncestralDiscreteScheduler`, which we found to produce consistently lower distortion at the cost of, however, slightly decreased perceptual characteristics. Note that the PerCo (SD) performance is bounded by the auto-encoder.

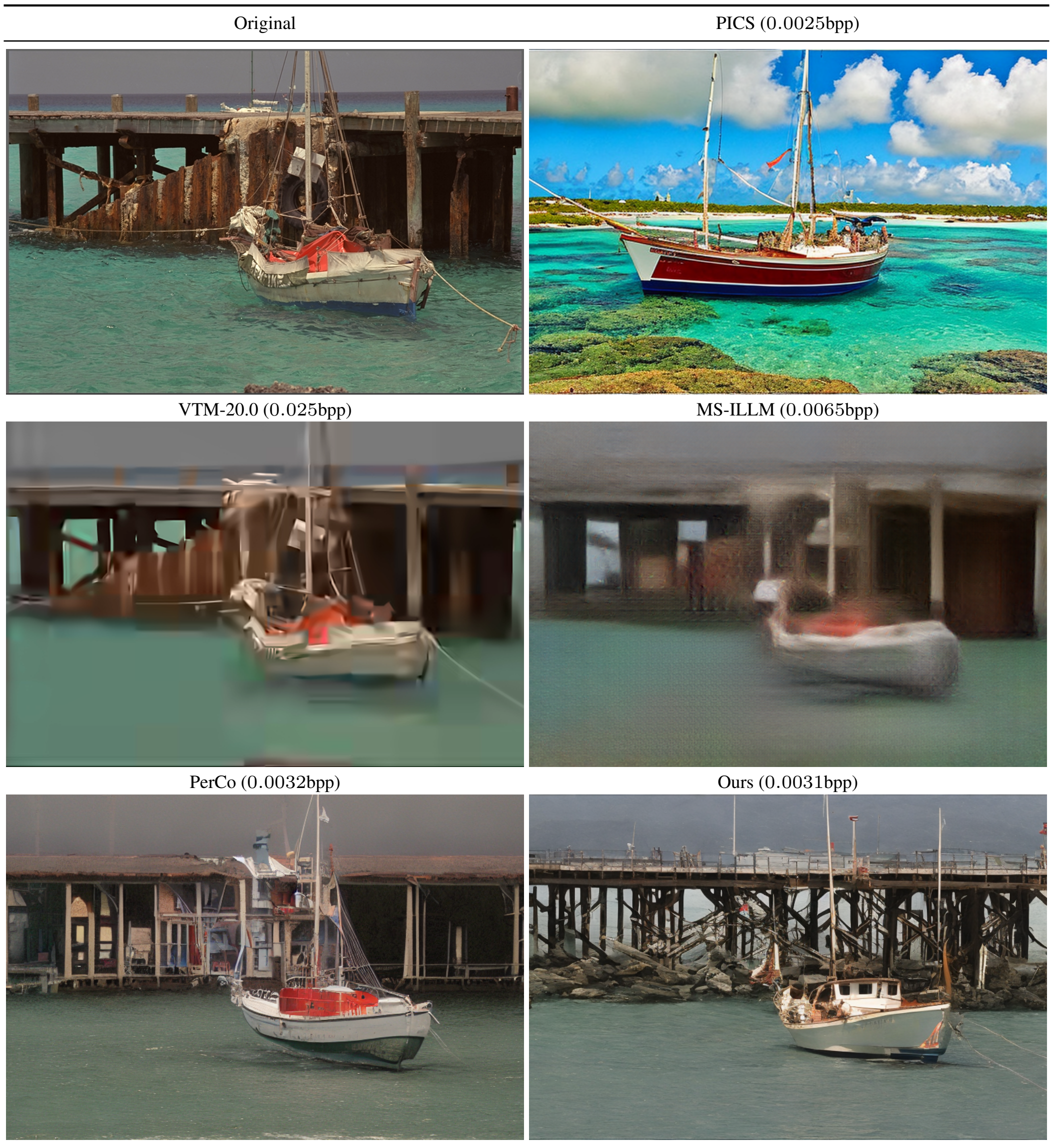


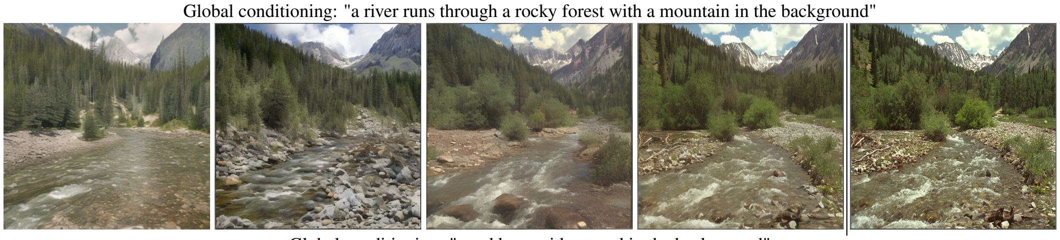
Figure 6: Visual comparison of PerCo (SD) to PICS [22], VTM-20.0, the state-of-the-art non-learned image codec, MS-ILLM (Muckley *et al*. ICML 2023), and PerCo [8].

Figure 7: Visual comparison of PerCo (SD) to PICS [22], VTM-20.0, the state-of-the-art non-learned image codec, MS-ILLM (Muckley *et al*. ICML 2023), and PerCo [8].

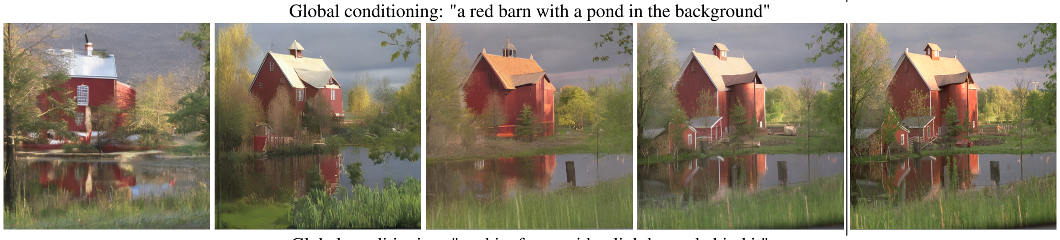
Figure 8: Visual illustration of the impact of the global conditioning on the Kodak dataset (`kodim19`), with a spatial bit-rate of 0.0019bpp. Samples are generated from the same initial Gaussian noise. Inspiration taken from [8, fig. 13].

0.0036bpp | 0.011bpp | 0.0329bpp | 0.1267bpp | Original

Global conditioning: "a river runs through a rocky forest with a mountain in the background"



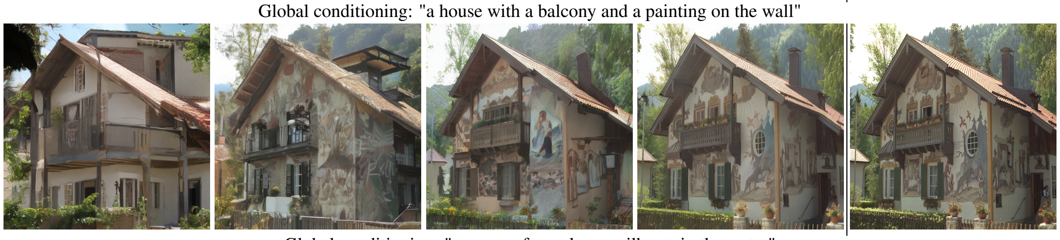
Global conditioning: "a red barn with a pond in the background"



Global conditioning: "a white fence with a lighthouse behind it"

Global conditioning: "a pink flower is in front of a window with blue shutters"

Global conditioning: "a house with a balcony and a painting on the wall"



Global conditioning: "a group of people on sailboats in the water"

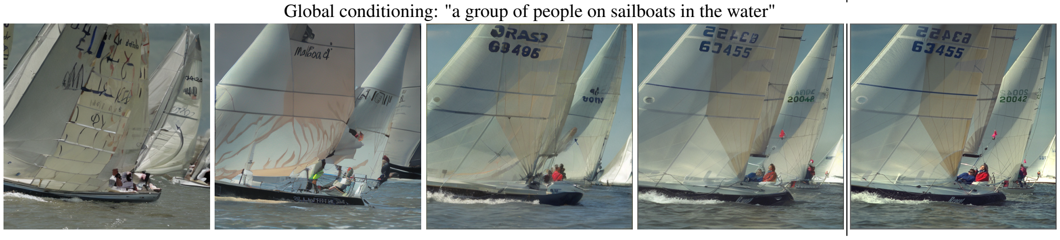


Figure 9: Visual comparison of PerCo (SD) across various bit-rates on the Kodak dataset

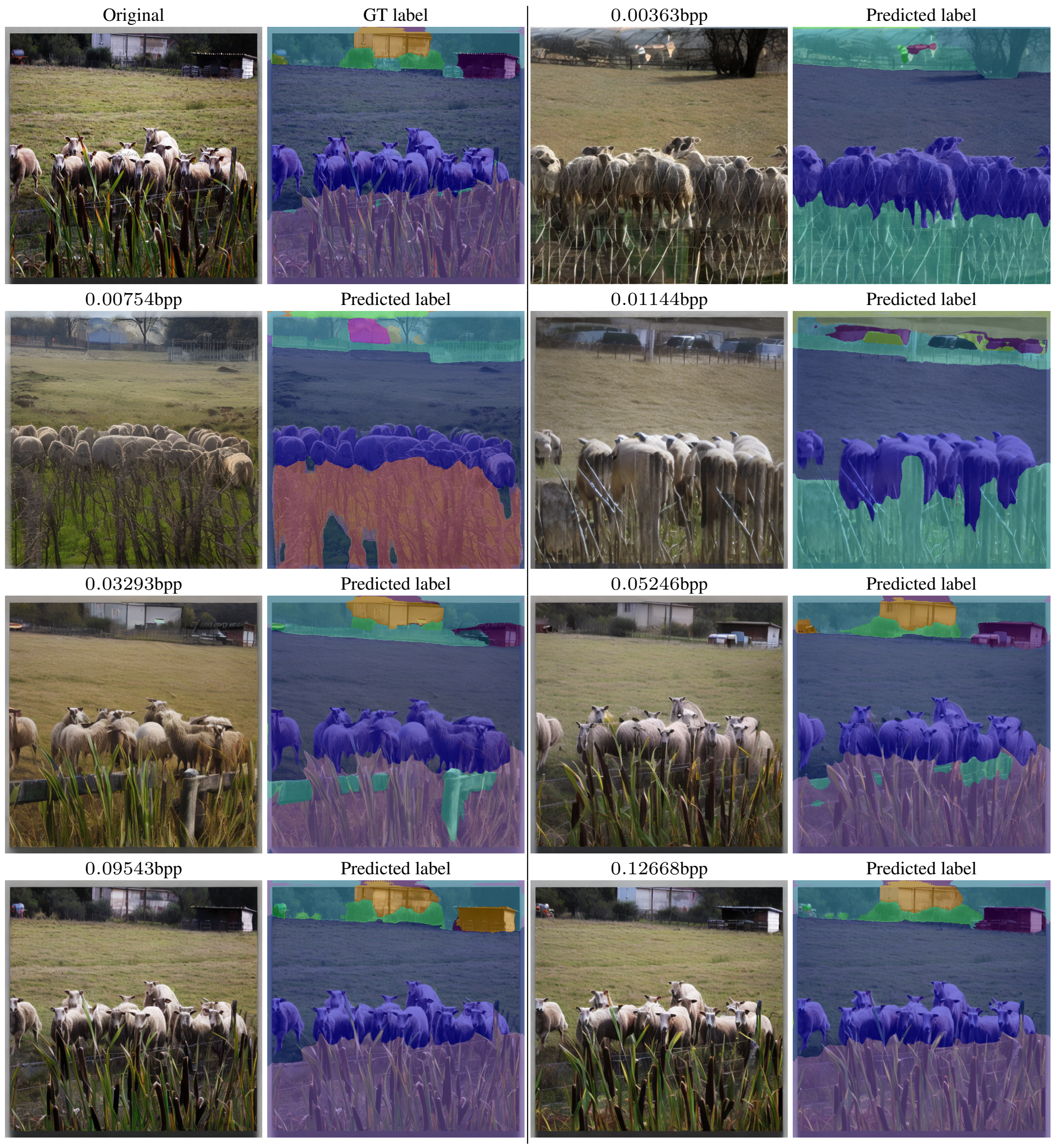


Figure 10: Visual comparison of the semantic preservation of PerCo (SD) across various bit-rates on the MSCOCO-30k dataset (000000442539), using the ViT-Adapter segmentation network (Chen *et al*. ICLR 2023). Global conditioning: "a herd of sheep standing in a field next to a fence".

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract, including the introduction of PerCo (SD) as an alternative to PerCo and the focus on adapting it to the Stable Diffusion ecosystem, accurately reflect the paper's contributions. The abstract outlines both the theoretical and practical aspects of the work, aligning well with the paper's scope and goals.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results. The theoretical background is, however, properly reviewed in section 2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper offers all the information needed to reproduce the main experimental results (see table 1 and section 4). Scripts, code, and trained models of PerCo (SD) will be released at `https://github.com/Nikolai10/PerCo`.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper offers open access to code and data. The code and trained models of PerCo (SD) will be released at `https://github.com/Nikolai10/PerCo` (an early version has already been released). The training data is available online `https://storage.googleapis.com/openimages/web/index.html`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper follows the exact evaluation protocol defined in PerCo [8], which is widely adopted in the field of image compression. The paper does not report error bars or other appropriate information about the statistical significance of the experiments due to the high computational cost.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See section 4, Implementational details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper does not contain research involving human subjects or participants or data-related concerns. PerCo (SD) inherits the risks of Stable Diffusion, which are summarized in `https://huggingface.co/stabilityai/stable-diffusion-2-1`, section Misuse, Malicious Use, and Out-of-Scope Use. The model is intended for research purposes only.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper inherits the risks of Stable Diffusion, which are summarized in `https://huggingface.co/stabilityai/stable-diffusion-2-1`, section Misuse, Malicious Use, and Out-of-Scope Use. The model is intended for research purposes only. PerCo (SD) does not add any new potential positive/ negative societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: The paper inherits the risks of Stable Diffusion, which are summarized in `https://huggingface.co/stabilityai/stable-diffusion-2-1`, section Misuse, Malicious Use, and Out-of-Scope Use. The model is intended for research purposes only.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper is accompanied by an anonymized and well documented GitHub project `https://github.com/Nikolai10/PerCo`. The GitHub page contains a separate section (Acknowledgment) to acknowledge all third party libraries. Each source file contains proper licensing information.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper is accompanied by an anonymized and well documented GitHub project `https://github.com/Nikolai10/PerCo`. A license is included.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.