Revisiting Associative Compression: I Can't Believe It's Not Better

Winnie Xu[1] **Matthew J. Muckley**[1] **Yann Dubois**[2] **Karen Ullrich**[1]

Abstract

Typically, unordered image datasets are individually and sequentially compressed in random order. Unfortunately, general set compression methods that improve over the default sequential treatment yield only small rate gains for high-dimensional objects such as images. We propose an approach for compressing image datasets by using an image-to-image conditional generative model on a reordered dataset. Our approach is inspired by Associative Compression Networks (Graves et al., 2018). Even though this variation of variational auto-encoders was primarily developed for representation learning, the authors of the paper show substantial gains in the lossless compression of latent variables. We apply the core idea of the aforementioned work; adapting the generative prior to a previously seen neighbor image, to a commonly used neural compression model; the mean-scale hyperprior model (MSHP) (Ballé et al., 2018; Minnen et al., 2018). However, the architecture changes we propose here are applicable to other methods such as ELIC (He et al., 2022) as well. We train our model on subsets of an ordered version of ImageNet, and report rate-distortion curves on the same dataset. Unfortunately, we only see gains in latent space. Hence we speculate as to the reason why the approach is not leading to more significant improvements.

1. Introduction

Our photo albums and memories have largely moved to the cloud through data centers owned by large tech corporations. In 2012 there existed half a million data centers compared to the 8 million today. Our hunger for archiving our lives at high definition comes at a high environmental cost: data centers are responsible for yearly CO2 emissions comparable to those produced by the global airline industry. This is due to the fact that millions of data centers worldwide consume electricity equivalent to that of entire countries such as South Africa, Egypt or Argentina and Columbia combined. Without intervention, models predict that data center energy usage could surpass 10% of the global electricity supply by 2030 (McNerney, 2019). Improving the consumption of data centers involves various strategies such as optimizing server usage, implementing effective cooling strategies and optimizing storage. While different types of data may require specific storage solutions (e.g., archival versus active use data), data compression is one of the most effective ways to enhance storage efficiency. (Moore, 202)

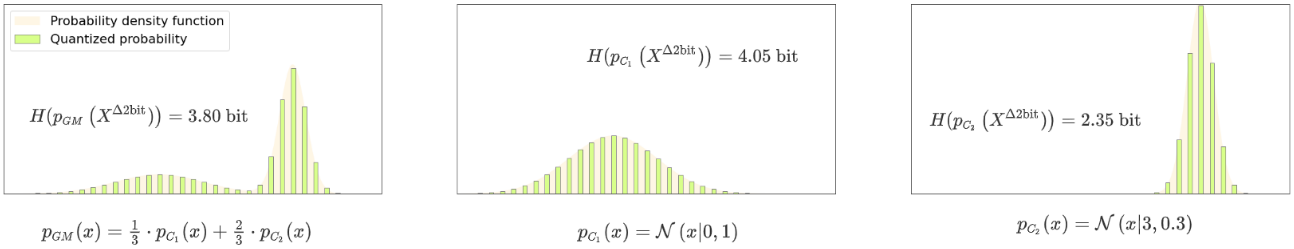
We can improve the compression rate of an image dataset if we were to compress it holistically as a multi-set. Under the assumption that images are drawn i.i.d. from an unknown source, the best possible rate improvement amounts to all information hidden in the ordering of the sequence (Varshney & Goyal, 2006b). For a multi-set of size N, the rate improvement is bounded by $-\log_2(N!)$ bits. In other words, a dataset of 10,000 images would save roughly only 15 MB or 11.8 bits per data point. Recent advances in bits-back coding have enabled the realization of optimal coders that efficiently achieve this precise rate with small margins of error (Severo et al., 2023).

However, for image data-sets these gains are negligible even for optimal implementations. We instead offer an alternative approach to image dataset compression. Since image datasets are rarely considered as i.i.d. samples from the natural image distribution, we assume that a dataset will often cluster in groups, e.g. around classes or perspectives. Hence, we may model them via a predictive coding approach, see e.g. (Barowsky et al., 2021; Graves et al., 2018) for similar ideas. We demonstrate the effectiveness of this principle next with a toy example.

To compress a data instance from a discrete alphabet $\mathcal{X}$ *without error*, we need at least -$\sum_x p_X(x) \log_2 p_X(x)$ bits (Shannon, 1948; Cover & Thomas, 1991; MacKay, 2003). This quantity is also known as the Shannon entropy $H(P_X) = \mathbb{E}_{P_X}[-\log_2 P_X]$ of X. Next, assume a dataset sampled i.i.d. from a mixture $P_X = \sum_i \pi_i \cdot P_X^{(i)}$, $\sum_i \pi_i = 1$. We could sort the dataset and hope to assign each datapoint to the cluster component that its neighbors

[1] Meta AI [2] Stanford University. Correspondence to: Karen Ullrich <karenu@meta.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).



$p_{GM}(x) = \frac{1}{3} \cdot p_{C_1}(x) + \frac{2}{3} \cdot p_{C_2}(x)$

$p_{C_1}(x) = \mathcal{N}(x|0,1)$

$p_{C_2}(x) = \mathcal{N}(x|3,0.3)$

Figure 1. **Toy example:** Imagine we had a dataset of i.i.d random samples from a Gaussian mixture source distribution (**left**). Its associated discrete entropy is 3.8 bits per data point, discretized at 2 bit precision. The entropy indicates the optimal rate for symbols sampled form the mixture. Further imagine we sort all samples of the dataset and we compressed one datapoint. This compressed datapoint can now inform the source model for the next incoming datapoint to be compressed. For example, we could update the source model to be only one of the two modes of the mixture (**middle** or **right**). This adaptation of the source model allows us to compress at the component's source entropy i.e. 4.05 or 2.35 bit per symbol (assuming we did guess the component correctly). Considering, that in our example one third of samples will stem from the first component and two thirds from the second; this comes out to 2.92 bit per symbol in total or 0.88 bit in savings.

were assigned to as well. This is the key idea presented in (Graves et al., 2018), dubbed associative compression. The gains we expect are $H(P_X) - \sum_i \pi_i H(P_X^{(i)})$. In Figure 1, we show rate savings on a concrete example of a two-component Gaussian mixture model.

Our objective in this study is to explore the application of associative compression principles to lossy compression of image datasets. To achieve this, we build upon the existing body of research on neural image compression, which extensively utilizes deep variational auto-encoder (VAE) architectures. Specifically, we extend the work done on the MSHP architecture, a two-level latent VAE that uses a hyperprior model to generate the mean and scale parameters of the latent model. Instead of associating neighboring data points with specific mixture components, we condition the networks responsible for compressing parameters on the same data point. In principle, the latter approach generalizes beyond the former. To train our model, we learn to compress pairs of neighboring images and compare the performance of the MSHP architecture against our newly proposed conditional mean-scale hyperprior (cMSHP) architecture. Although we observe consistent improvements in compressing the second latent variable, the overall enhancements are still relatively small. This outcome prompts us to delve into the possible reasons behind these findings, which we discuss in the final section of this paper.

2. Background

Lossless compression, as demonstrated in our toy example, is not always necessary. In many cases, relaxing the hardness threshold on image distortion can result in significant rate savings. Historically, distortion is measured with a reference-based distortion metric $\rho(x, \tilde{x})$ such as peak signal-to-noise ratio (PSNR) or multi-scale structural similarity index (MS-SSIM) (Wang et al., 2003; 2004). For a given distortion metric and a defined tolerance level τ, Shannon (1948) established the limits of the optimal compression rate for lossy compression, known as the rate-distortion function:

$$R_{opt}(\tau) = \inf_{Q_{\hat{X}|X} : \mathbb{E}[\rho(X,\hat{X})] \leq \tau} I(X;\hat{X}) \tag{1}$$

The infimum is taken over possible distributions $Q_{\hat{X}|X}$ that satisfy the condition $\mathbb{E}[\rho(X,\hat{X})] \leq \tau$, ensuring the average distortion, i.e. mutual information I, remains within the defined tolerance. It is worth noting that while vector quantization can theoretically achieve the optimal rate (Cover & Thomas, 1991), it becomes intractable for high-dimensional data due to the inherent complexity of the associated algorithm.

In practice, lossy compression relies on a non-optimal approach. A typical framework for lossy compression involves three functions: an encoder, a decoder, and an entropy bottleneck. The encoder function maps input images to discrete latent representations $Y = f(X)$, while the decoder function performs the reverse process $\hat{X} = g(Y)$. The entropy bottleneck, often implemented as an entropy coder, performs lossless compression on the latent representations $\gamma(Y)$. In this context, the entropy bottleneck plays a crucial role. Given a model P_Y, it optimally compresses the data at a rate equal to cross-entropy $R = H(P_X P_{Y|X}, P_Y) = \mathbb{E}_{P_X P_{Y|X}}[-\log_2 P_Y]$. These three functions together form what we refer to as a neural compression codec, following (Yang et al., 2023). Our objective is to adopt a data-driven approach and parameterize these functions to learn them from the available data. To facilitate backpropagation for training, we relax Equation 1 using Lagrangian multipliers:

$$L(\lambda, \theta) = H(P_X P_{Y|X}, P_Y^\theta) + \lambda \cdot \mathbb{E}_{P_X}\left[\rho\left(X, \hat{X}\right)\right] \quad (2)$$

where λ is a Lagrange multiplier that relates to τ in the rate-distortion trade-off. The Lagrangian relaxation enables us to optimize the parameters θ of the neural compression codec using backpropagation. For instance, $Y = f^\theta(X)$ and $\hat{X} = g^\theta(Y)$ could be neural networks, and P_Y^θ a parameterized distribution model of Y, e.g. a discetized Gaussian $\mathcal{N}(Y|\mu, \sigma)$ where both distribution parameters are learned.

Upon examining Equation 2, we observe similarities to the field of generative modeling. In generative modeling, it is well-known that directly modeling a complex distribution P_Y can be challenging. To address this, a common approach is to introduce latent variable models, which alleviate the modeling burden. In this context, a joint model $P(Y, Z) = P_{Y|Z} P_Z$ is considered, where variational inference is often employed to provide an upper bound on the log-likelihood of such models, leading to improved overall performance in likelihood modelling. Motivated by these insights, this principle was adapted to the lossy compression framework by introducing a hyper-latent variable $Z = h^\theta(Y)$. Therefore, there are two entropy bottlenecks; the hyper-latent bottleneck and the latent bottleneck. The rates of the latent variable bottleneck evaluate

$$R_{\text{joint}} = \mathbb{E}_{P_X P_{Y|X}^\theta P_{Z|Y}^\theta}[-\log_2 P_Z^\theta - \log_2 P_{Y|Z}^\theta]. \quad (3)$$

It is important to note that this formulation is a loose variational bound, as it does not account for bits-back (for further details, refer to (Ballé et al., 2018)). The inclusion of latent variable models in the rate calculation improves compression rates. Additionally, it highlights the possibility of coding with multiple distributions. In section 5, we will delve deeper into this concept, expanding on the idea of using multiple distributions for coding.

3. Related Work

Neural lossy image compression Neural compression codecs as described in the previous section were developed and improved by (Ballé et al., 2018; Minnen et al., 2018; Cheng et al., 2020; He et al., 2022; El-Nouby et al., 2023). It is important to note that a large number of publications in lossy neural image compression focuses on optimizing perceptual metrics as opposed to reference based distortion metrics. One of the most common approaches in the field is to optimize weighted sum between the loss of a (conditional) GAN and a handcrafted metric such as MSE/MS-SSIM (Mentzer et al., 2020; Agustsson et al., 2019) but multiple other approaches exist (Tschannen et al., 2018; Ledig et al., 2017; Mentzer et al., 2020; Ding et al., 2021). Other methods in the field focus on improving the rate term Eq. 3 via bits-back coding (Theis et al., 2022; Yang & Mandt, 2022). For a more complete overview of the field see (Yang et al., 2023).

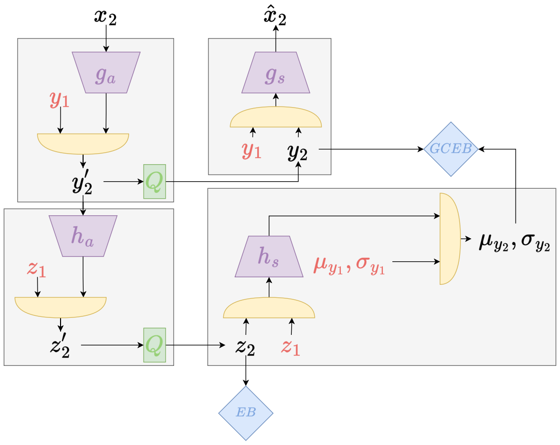


Figure 2. **Conditional MSHP (cMSHP) model**. We condition the entropy bottlenecks (blue) on previously compressed data x_1. Our alteration involves mixer modules (yellow) that map two variables to one where all variables have the same alphabet size $(W, S) \rightarrow T$, $W, S, T \in \mathcal{W}$. This allows to leave the other components of the architecture unmodified.

Dataset compression The fundamental limits on lossless compression of multisets of size N were first determined by (Varshney & Goyal, 2006a). The first implementation of this rate for small alphabets was presented in (Gripon et al., 2012; Steinruecken, 2015; Reznik, 2011), a more practical method for general multisets was presented by (Severo et al., 2023). As in real world scenarios, we can improve upon the predicted bound when datasets can be assumed to be non-i.i.d. (Barowsky et al., 2021; Graves et al., 2018) leverage this insight to compress datasets by adapting the entropy bottleneck of one instance based on previously compressed instances.

4. Method

As indicated in the introduction, we will leverage neural codecs, as presented in the background section, to perform associative compression (Graves et al., 2018). In this approach, we pair neighboring images and compress one image conditionally on the other. The pairing is formulated based on a notion of proximity, defined using a distance metric in latent space. Instead of independently sampling $x \sim P_X$, we sample pairs (x_1, x_2) from the joint distribution $(x_1, x_2) \sim P_{X_1} P_{X_2|X_1}$. Consequently, our rate term is modified as follows:

$$\begin{aligned}&\tfrac{1}{2}R_{\text{joint}}(P_{X_1}, P^{\theta}_{Y_1|X_1}P^{\theta}_{Z_1|Y_1})+\\&\tfrac{1}{2}\mathbb{E}_{P_{X_1}}[R_{\text{joint}}(P_{X_2|X_1}, P^{\theta}_{Y_2|X_1;Y_1}P^{\theta}_{Z_2|Y_2;Y_1,Z_1})]\end{aligned} \tag{4}$$

Just as in the toy example, we can speculate about the rate savings by comparing Eq. 3 and 4. If the first term in Eq. 4 outweights the second we will see rate improvements.

4.1. Neural Codec Architecture

To build a conditional neural codec, the MSHP architecture (Ballé et al., 2018; Minnen et al., 2018) is altered as illustrated in Figure 4. The analysis and synthesis transforms g_a and g_s, respectively and the hyper-analysis and hyper-synthesis transforms h_a and h_s, are convolutional neural networks. In jointly training the cMSHP and MSHP, they do not share both levels of analysis and synthesis networks in the hierarchical compression model. Training the conditional model using shared modules g_a, g_s, h_a, and h_s was ablated but showed a slight decrease in performance. This was interpreted as placing a heavier burden on the rather shallow mixer modules to model the conditional probability and hence was not incorporated in the final architecture design.

The PMF of the hyper prior P^{θ}_{Z} is learned to encode Z with an entropy bottleneck (blue diamond), whereas the entropy bottleneck $P^{\theta}_{Y|Z}$ for the latents Y is a discrete conditional Gaussian. Further, we follow the notation presented in the original work. This differs from our previous presentation in that we referred to the decoder as $g(Y)$ instead of $g_s(Y)$ and we make quantization explicit (green box) to demonstrate that a variable is discrete.

We modify the architecture to be conditional by introducing a mixture module (yellow). This mixture module is a 1x1 convolution that has double the number of input channels as output channels. In other words, this is a simple construction to concatenate two variables of the same size along the channel dimension to obtain one variable of the same size as the previous two as output. This trick allows us to avoid modifying the entropy bottlenecks, or the analysis and synthesis transforms.

5. Experiments

To sample image pairs we first transform the entire dataset to latent space $Y = g_a(X)$ and project it onto one dimension (Tenenbaum et al., 2000). We than sort the latent representations via the isomap embedding algorithm (Tenenbaum et al., 2000) and store the indices. Note that sorting by latent space is one of many potential domains of comparison. We can now sample $(x_1, x_2) \sim P_{X_1}P_{X_2|X_1}$ by first sampling an arbitrary image and than getting its neighbor from the pre-computed index list. To compress x_1 we train a regular MSHP and to compress x_2 we train a cMSHP. We train and evaluate our model on blurred Imagenet (Russakovsky et al., 2015). We compute the rate distortion function for various tradeoffs $\lambda = 0.0004, 0.0008, 0.0016, 0.0032, 0.0075, 0.015, 0.03, 0.045$. We use a batchsie of 32, the Adam optimizer with learning rate $3 \cdot 10^{-4}$, about 100K gradient updates. While there are potentially performance gains from training for millions of steps, the training loss generally plateaus around 100k. We found this shorter training setup to be sufficient comparison across the two methods which should also be proportional to what would be gained from any baseline performance gains.

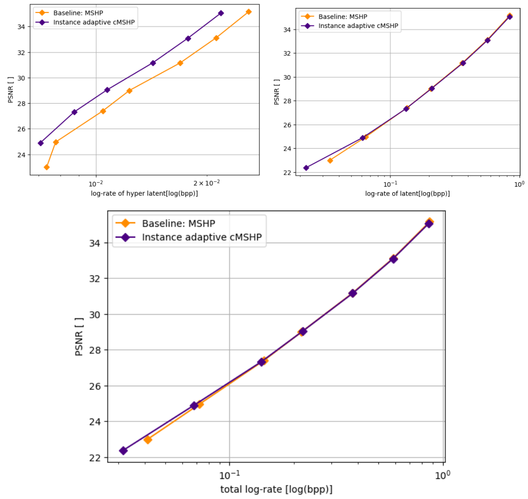


Figure 3. LogRate-Distortion curves for cMSHP (purple). The overall compression rate (bottom) is not improved significantly, however, we see consistent gains in the hyper-latent compression (top left).

We present the results of our methods in Figure 3. The purple line is the rate computed as in Eq. 4, the orange line refers to the rate as in Eq. 3. We can observe gains only for the hyper-latent variable.

6. Discussion

In this work, we have revisited the idea of associative compression using neighboring data instances to achieve better compression rates. We have integrated the idea in a commonly used model for neural image compression codec, the MSHP. We see only negligable rate improvments mainly pertaining to the latent $P_{Z_2|Z_1}$.

Following, we will discuss the failure of the proposal. Intuitively, it should be the case that the best conditioning information would be obtained from the hyperprior. At the stage of evaluating the Gaussian conditional, the perfect conditioning information should hence be met. This could imply that the maximum performance gains from conditioning on the previous image would solely be to remove the bits allocated to the hyperprior. Note that the gains we see in Figure 3 (top left plot) are on the order of 10%, which aligns with this intuition.

Moreover, there could always be architectural limitations where the mixer architecture might be too shallow. Perhaps the entropy bottleneck of the latent should also depend explicitly on Z_1 as did the Gaussian Conditional entropy bottleneck used for the latents $P_{Y_1|Z_1}$.

Besides architecture, the issues could run deeper in the importance of ordering assumed when using our method. The sorting algorithm used may be not suitable for our needs. Specifically, if there is no clear way to assign which image is x_1 and x_2, we might model the joint distribution as $P_{Y_1|Y_2}P_{Y_2}$ or as $P_{Y_2|Y_1}P_{Y_1}$, implying $P_{Y_2,Y_1} = P_{Y_1}P_{Y_2}$.

On the other hand, while it cannot generally be assumed that there exists two images in a dataset whose contents overlap to a high degree, we can model the conditional distribution $P_{X'|X}$ as opposed to the joint $P_{X'X}$ so that samples may still be interpreted as i.i.d. For two different images, they may still be similar under the conditional distribution and hence the differences between two images of the same set becomes reduced compared to their pure similarity score under the marginal distribution P_X.

In future explorations, we will investigate these issues and extend the model to also work for chaining long sequences of images. By doing this we might be able to only train one model for compression that can be iteratively applied to a sequence of pre-sorted data. We would need to change training on subsets of N neighbors, but still evaluate on the entire dataset in one chain at once. We leave the idiosyncrasies of these ideas to future work.

References

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

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

Barowsky, M., Mariona, A., and Calmon, F. P. Predictive coding for lossless dataset compression. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1545–1549. IEEE, 2021.

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

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley Series in Telecommunications. 1991.

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

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

Graves, A., Menick, J., and Oord, A. v. d. Associative compression networks for representation learning. *arXiv preprint arXiv:1804.02476*, 2018.

Gripon, V., Rabbat, M., Skachek, V., and Gross, W. J. Compressing multisets using tries. In *2012 IEEE Information Theory Workshop*, pp. 642–646, 2012.

He, D., Yang, Z., Peng, W., Ma, R., Qin, H., and Wang, Y. ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *CVPR*, pp. 5718–5727, June 2022.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

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

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

McNerney, M. The data center dilemma: Is our data destroying the environment? *Data Center Knowledge*, 2019. URL https://www.datacenterknowledge.com/industry-perspectives/data-center-dilemma-our-data-destroying-environme

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

Minnen, D., Ballé, J., and Toderici, G. D. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pp. 10771–10780, 2018.

Moore, F. Data center energy consumption - enormous data centers creating a hyperscale heat wave. *horison*, 202. URL https://asset.fujifilm.com/master/americas/files/2020-03/

8ef23fd892733dd9d1df8de46f64cfc9/Hyperscale_Heat_Wave.pdf.

Reznik, Y. A. Coding of sets of words. In *2011 Data Compression Conference (DCC)*, pp. 43–52. IEEE, 2011.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Severo, D., Townsend, J., Khisti, A., Makhzani, A., and Ullrich, K. Compressing multisets with large alphabets. *IEEE Journal on Selected Areas in Information Theory*, 2023.

Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

Steinruecken, C. Compressing Sets and Multisets of Sequences. *IEEE Transactions on Information Theory*, 61 (3):1485–1490, 2015.

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

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

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

Varshney, L. and Goyal, V. Toward a source coding theory for sets. In *2006 Data Compression Conference (DCC)*, pp. 13–22. IEEE, 2006a.

Varshney, L. R. and Goyal, V. K. Toward a source coding theory for sets. In *Data Compression Conference (DCC'06)*, pp. 13–22. IEEE, 2006b.

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

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

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

Yang, Y., Mandt, S., Theis, L., et al. An introduction to neural data compression. *Foundations and Trends® in Computer Graphics and Vision*, 15(2):113–200, 2023.

A. Notation

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

B. Sequence Ordering

To sort our features in latent space, we simply run an out of box isomap algorithm on the latent space features of the training set, which were obtained using a pre-trained MSHP model's hyper analysis network. We then use the learned model transform function to transform the evaluation set as well to account for the distributional shift between the training and test sets. During training, we train the $cMSHP$ model using this re-ordered dataset.

C. Training Procedure

The baseline model was implemented as in the original MSHP paper (Ballé et al., 2018; Minnen et al., 2018) and trained on the blurred ImageNet dataset (Russakovsky et al., 2015) for 100K steps. The cMSHP model was implemented with the baseline MSHP modules and additional trainable parameters from separate mixer and hyper prior layers. We had also ablated training a cMSHP that was always conditioned on zeros to mimic the basic case. We confirmed that we may successfully recover the baseline MSHP module, which allowed us to experiment solely with the proposed conditional architecture without needing to pre-train a separate MSHP module that can be used as a warm-start for the coinciding modules in the cMSHP experiments.

D. Additional Ablations

We experimented with various other ablations on architecture and conditioning schemes. Neither significant improvements on the rates nor differences between different levels of represnetation in the hierarchy were observed.

Below we report results on PSNR after chained compression of images using the $cMSHP$ model. They were trained either one chains or 3 or 4 images at a time where the conditioning layer was either a 1x1 convolution or a FiLM layer.

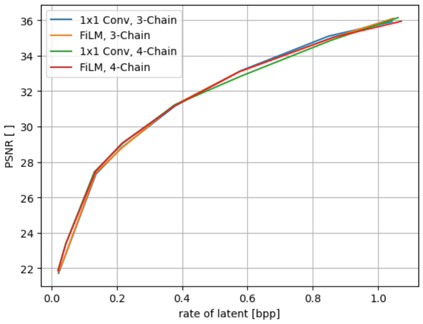


Figure 4. Hyperlatent rate.

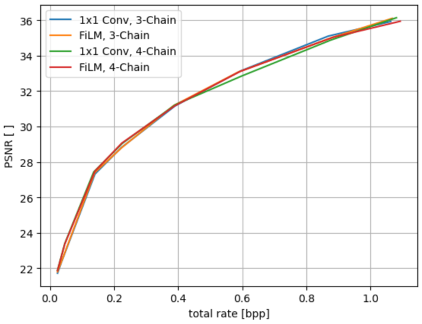


Figure 5. Latent rate.

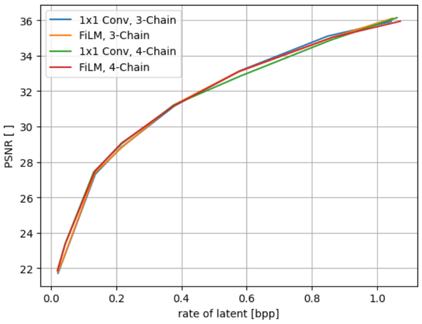


Figure 6. Total rate.