Bit Prioritization in Variational Autoencoders via Progressive Coding

Rui Shu [*1] **Stefano Ermon** [1]

Abstract

The hierarchical variational autoencoder (HVAE) is a popular generative model used for many representation learning tasks. However, its application to image synthesis often yields models with poor sample quality. In this work, we treat image synthesis itself as a hierarchical representation learning problem and regularize an HVAE toward representations that improve the model's image synthesis performance. We do so by leveraging the progressive coding hypothesis, which claims hierarchical latent variable models that are good at progressive lossy compression will generate high-quality samples. To test this hypothesis, we first show empirically that conventionally-trained HVAEs are not good progressive coders. We then propose a simple method that constrains the hierarchical representations to prioritize the encoding of information beneficial for lossy compression, and show that this modification leads to improved sample quality. Our work lends further support to the progressive coding hypothesis and demonstrates that this hypothesis should be exploited when designing variational autoencoders.

1. Introduction

The task of generating of high-dimensional image data has inspired the development of many successful deep generative models such as autoregressive models (Uria et al., 2016; Oord et al., 2016b), flow-based models (Dinh et al., 2014; 2016), generative adversarial networks (Goodfellow et al., 2014), variational autoencoders (Kingma & Welling, 2013; Rezende et al., 2014), and diffusion models (Ho et al., 2020; Song & Ermon, 2019). These models have since been applied to many other modeling tasks such as speech synthesis (Oord et al., 2016a; Donahue et al., 2018), reinforcement learning (Zhang et al., 2019; Levine et al., 2019), and scene-understanding (Eslami et al., 2018). In this work, we focus on the application of hierarchical variational autoencoders (VAEs) to image synthesis. Despite the success of VAEs in numerous fields (Akuzawa et al., 2018; Hsu et al., 2017; Gómez-Bombarelli et al., 2018; Sultan et al., 2018; Van Hoof et al., 2016), the application of VAEs to image synthesis has been challenging (Theis et al., 2015), and many existing works sought to improve VAEs' image synthesis capabilities by addressing the optimization challenges associated with VAE training (Rezende & Viola, 2018; Child, 2020; Vahdat & Kautz, 2020; Sønderby et al., 2016). Given the success of VAEs as a representation learning algorithm, we take the complementary position that VAEs should also be regularized toward representations bespoke to the task of interest (Levine et al., 2019). We apply this perspective to HVAE image synthesis and seek representations that will improve the model's image synthesis performance.

In order to find representations suitable for image synthesis, we critically examine diffusion models, a special class of hierarchical latent variable models that has demonstrated superior image synthesis performance in recent years (Ho et al., 2020; Song & Ermon, 2019). Ho et al. (2020) suggested that the diffusion model's strong performance on image synthesis may be attributable to such models having a *bit ordering* that is good for progressive lossy compression—which we shall refer to as the Progressive Coding Hypothesis. Our main contributions are thus as follows,

1. We formalize the principle of progressive coding and show that the diffusion model objective can be derived as a special application of this principle.
2. We show that conventionally-trained HVAEs are suboptimal progressive coders. We then develop an alternative training procedure that *prioritizes* the information-theoretic bits beneficial for progressive coding, and show that the resulting model has improved sample quality as measured by FID—thus lending support to the progressive coding hypothesis.
3. Finally, we take initial steps toward understanding why the progressive coding hypothesis works in practice.

Our work supports the use of progressive coding as an effective principle for designing hierarchical variational autoencoders and encourages further research into the theory and practice of progressive coding.

[*]Equal contribution [1]Stanford Computer Science.. Correspondence to: Rui Shu <ruishu@stanford.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

2. Background

2.1. Hierarchical Latent Variable Models

A hierarchical latent variable model (HLVM) $p(x, z_{1:T})$ models an observed variable x as well as T levels of hierarchical latent variables $\{z_i\}_{i=1}^T$, where each z_i is a multivariate random variable. The hierarchical latent variables are sampled in sequential order during generation, and the full sampling process can be represented by the factorization

$$p(x, z_{1:T}) = p(x \mid z_{1:T}) \prod_{i=1}^{T} p(z_i \mid z_{<i}).$$

Given a data distribution $p_{\text{data}}(x)$, the goal of distribution modeling is to optimize the HLVM p such that the marginal distribution $p(x)$ is close to the data distribution.

2.1.1. Hierarchical Variational Autoencoders

A hierarchical variational autoencoder (HVAE) is an application of the variational autoencoding framework (Kingma & Welling, 2013; Rezende et al., 2014) to the hierarchical generator p. This framework introduces a variational inference model $q(z_{1:T} \mid x)$ and then optimizes (p, q) jointly via the Evidence Lower Bound (ELBO),

$$\max_{p,q} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:T})} \left[\ln \frac{p(x, z_{1:T})}{q(z_{1:T} \mid x)} \right] \equiv \mathbb{E}_{p_{\text{data}}(x)} \Big[\mathbb{E}_{q(z_{1:T}|x)} \ln p(x \mid z_{1:T}) + D_{\text{KL}}(q(z_{1:T} \mid x) \parallel p(z_{1:T})) \Big],$$

which admits a decomposition (shown RHS) where the first term minimizes reconstruction error and the second term regularizes q toward the prior distribution $p(z_{1:T})$.

An important design choice when building an HVAE is the conditioning structure for the inference model q. Although earlier works on HVAEs (Burda et al., 2015) chose to factorize the q in reverse generation order (also known as *bottom-up* inference), Sønderby et al. (2016) proposed the *top-down* inference process,

$$q(z_{1:T} \mid x) = \prod_{i=1}^{T} q(z_i \mid x, z_{<i}),$$

which has become the predominant inference method of choice in the current literature (Vahdat & Kautz, 2020; Child, 2020). In addition to its empirical efficacy, the top-down factorization is also beneficial in that it decomposes the regularization term into the following,

$$\mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:T} \mid x) \parallel p(z_{1:T})) = \sum_{i=1}^{T} \mathbb{E}_{p_{\text{data}}(x)} \Big[\mathbb{E}_{q(z_{<i}|x)} D_{\text{KL}}(q(z_i \mid x, z_{<i}) \parallel p(z_i \mid z_{<i})) \Big],$$

which can be seen as a summation of conditional mutual information bounds (Alemi et al., 2018; Poole et al., 2019) since

$$I(Z_i ; X \mid Z_{<i}) \leq \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{<i}|x)} D_{\text{KL}}(q(z_i \mid x, z_{<i}) \parallel p(z_i \mid z_{<i})),$$

where the mutual information $I(Z_i; X \mid Z_{<i})$ is with respect to the distribution $p_{\text{data}}(x) q(z_{1:T} \mid x)$. This relation motivates the interpretation of the regularizer as an information-theoretic *rate* which we shall discuss and take advantage of in our examination of progressive coding in Section 3.

2.1.2. Denoising Diffusion Probabilistic Models

A denoising diffusion probabilistic model (DDPM), proposed by Ho et al. (2020), is a special case of the HLVM where each z_i has the same dimensionality as x and p is Markovian,

$$p(x, z_{1:T}) = p(x \mid z_T) \prod_{i=1}^{T} p(z_i \mid z_{i-1}).$$

Similar to the HVAE, the DDPM also introduces an inference model $q(z_{1:T} \mid x)$. Crucially, however, $q(z_{1:T} \mid x)$ is fixed to a prescribed diffusion process,

$$q(z_{1:T} \mid x) = q(z_T \mid x) \prod_{i=1}^{T-1} q(z_i \mid z_{i+1}),$$

where each z_i is simply a Gaussian perturbation with prescribed variance σ_i^2 of its conditioning parent variable. Because of the special restrictions imposed by the DDPM, the generator p can be trained to maximize the likelihood

$$\max_{p} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:T}|x)} \ln p(x, z_{1:T}),$$

which thus gives p the interpretation of reversing the diffusion process defined by q. In practice, the DDPM introduces a series of denoising functions $d_i : \mathcal{X} \to \mathcal{X}$ and parameterizes p as

$$p(z_i \mid z_{i-1}) = q(z_i \mid z_{i-1}, x = d_{i-1}(z_{i-1})) \; \forall i > 1$$
$$p(x \mid z_T) = \mathcal{N}(x \mid d_T(z_T), \sigma_{T+1}^2 I).$$

Recall that each $q(z_i \mid \cdot)$ is also a Gaussian distribution with a prescribed σ_i. By substituting this parameterization into the likelihood objective and omitting all of the weighting terms introduced by $\sigma_{2:T+1}$, the likelihood objective thus reduces to the DDPM objective,

$$\min_{d_{1:T}} \sum_{i=1}^{T} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_i|x)} \|x - d_i(z_i)\|^2.$$

Despite its simplicity, the DDPM has strong image synthesis performance, inspiring many subsequent works (Song et al., 2020; Kingma et al., 2021; Dockhorn et al., 2021; Jalal et al., 2021).

2.2. Lossy Compression

We consider lossy compressors consisting of a decoder $d : \mathcal{Z} \to \mathcal{X}$, a prior $p(z)$, and a stochastic encoder $q(z \mid x)$. Such a model can perform lossy compression of a data point x by lossily mapping x to a point z in latent space via the encoder, performing compression/decompression of z in the latent space using the prior[1], and then decoding back into the space of $\mathcal{X}$ using the decoder.

The model incurs a communication cost (i.e., rate) of

$$R = \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z \mid x) \parallel p(z))$$

to lossily reconstruct the sample $\hat{x} = d(z)$ for $z \sim q(z \mid x)$. The reconstructed sample $\hat{x}$'s fidelity to the original sample x can then be measured via a distortion function f. On average, the model will incur a distortion of

$$D = \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z|x)} f(x, d(z)).$$

We adopt a rate-distortion trade-off perspective to describe the optimal rate for a given distortion and vice versa (Alemi et al., 2018; Blau et al., 2018). We say that a lossy compressor is γ-optimal if the compressor has rate $R \leq \gamma$ and achieves the optimal distortion $D = \lambda^*$ under f, where λ^* is the optimal value to the optimization problem

$$\begin{aligned} \lambda^* = \min_{d,p,q} & \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z|x)} f(x, d(z)) \\ \text{s. t. } & \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z \mid x) \parallel p(z)) \leq \gamma. \end{aligned} \tag{1}$$

Conversely, the lossy compressor is λ-optimal if the compressor has distortion $D \leq \lambda$ and achieves the optimal rate $R = \gamma^*$, where

$$\begin{aligned} \gamma^* = \min_{d,p,q} & \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z \mid x) \parallel p(z)) \\ \text{s. t. } & \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z|x)} f(x, d(z)) \leq \lambda. \end{aligned} \tag{2}$$

Both (γ, λ^*) and (λ, γ^*) optimally trade off rate for a given distortion (and vice versa) and are thus considered pareto-optimal points.

3. The Progressive Coding Hypothesis

In this section, we formalize the progressive coding hypothesis presented in Ho et al. (2020). We shall begin by showing how any HLVM paired with a top-down inference model can be naturally converted into a progressive coder. Next, we discuss why conventional ELBO-based training of HLVMs may not necessarily lead to *good* progressive coders. We then demonstrate how the special restrictions imposed by DDPMs explain its progressive coding capabilities. Finally, we present the progressive coding hypothesis, based on the empirical observation that DDPMs are both good progressive coders and strong image synthesizers.

[1]Continuous z's must be discretized for source coding.

3.1. HLVM as Progressively Lossier Compressors

Given a hierarchical latent variable model (HLVM)

$$p(x, z_{1:T}) = p(x \mid z_{1:T}) \prod_{i=1}^{T} p(z_i \mid z_{<i})$$

equipped with inference model

$$q(z_{1:T} \mid x) = \prod_{i=1}^{T} q(z_i \mid z_{<i}, x),$$

we make the observation that an HLVM defines, not just one, but a sequence of T lossy compressors. Using the first k latent variables in the hierarchy, we can define a lossy compressor by considering the prior $p(z_{1:k})$, the encoder $q(z_{1:k} \mid x)$, and then learning a decoder via the optimization problem

$$\min_{d} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:k}|x)} f(x, d(z_{1:k})).$$

The tuple (d, p, q) thus defines a compressor that employs the first k latent variables of the HLVM and has an associated rate and distortion of (R_k, D_k). By decreasing k from T to 1, we can build a series of progressively lossier compressors that trades off rate against distortion according to the points $\{(R_k, D_k)\}_{k=1}^{T}$. Note that $R_{k+1} \geq R_k$ because of the data processing inequality. We shall refer to this series of progressively lossier compressors collectively as a progressive coder.

3.2. HLVMs are Not Always Good Progressive Coders

A key observation of the progressive coding hypothesis is that the points $\{(R_k, D_k)\}_{k=1}^{T}$ are not necessarily pareto-optimal. It is worth noting that for $k = T$, the point (R_T, D_T) will be pareto-optimal if the HLVM was trained via an ELBO objective that employs f as the reconstruction term. Even then, however, there is no guarantee that (R_k, D_k) is pareto-optimal for $k < T$, since strict subsets of the HLVM's latent variables were never directly optimized to lower distortion.

3.3. DDPMs are Good Progressive Coders

If our goal were to encourage the HLVM to be a good progressive coder for all values $k \in \{1, \ldots, T\}$, one approach then is to define a sequence of rate targets $\{\gamma_1, \ldots, \gamma_T\}$ and simultaneously optimize all T constrained optimization objectives described by Equation (1).[2] Using scalarization,

[2]We treat the rate target sequence as a hyperparameter.

this reduces to the following *progressive coding objective*

$$\min_{d_{1:T},p,q} \sum_{k=1}^{T} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z|x)} f(x, d_k(z_{1:k})) \quad (3)$$
$$\text{s. t. } \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:k} \mid x) \parallel p(z_{1:k})) \leq \gamma_k \quad \forall k \in \{1, \dots, T\}.$$

Suppose, for the moment, that we restrict our HLVM to the Gaussian variational family where each latent code z_i has the same dimensionality as x (i.e., q is a diffusion process). The optimal q^* to the objective Equation (3) will then simply be a Gaussian Markov chain adhering to a specific *noise schedule*, where each z_i is simply a linear transformation of x followed by the addition of some amount of noise. Conversely, this also means that fixing q to a prescribed noise schedule and then optimizing

$$d_k^* = \arg\min_{d_k} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:k}|x)} f(x, d_k(z_{1:k})) \quad (4)$$
$$p^* = \arg\min_{p} \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:T} \mid x) \parallel p(z_{1:T})) \quad (5)$$

will result in a compressor (d_k^*, p^*, q) that is γ-optimal for some choice of rate target γ_k implied by q's prescribed noise schedule. In particular, the implicit rate target will be

$$\gamma_k = \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:k} \mid x) \parallel p^*(z_{1:k})).$$

From here, we can further simplify the objectives in Equations (4) and (5) as follows. First, since $z_{<k}$ is independent of x when conditioned on z_k due to the Markov chain structure of q, it suffices to let z_k be the only input to the decoder d_k (i.e., $\hat{x} = d_k(z_k)$). Next, since q is a Markov chain, it suffices for p to also be a Markov chain where $p(z_{1:T}) = \prod_{k=1}^{T} p(z_k \mid z_{k-1})$. We can then perform parameter-sharing between p and d_k by implicitly defining the prior as

$$p(z_k \mid z_{k-1}) = q(z_k \mid x = d_{k-1}(z_{k-1})). \quad (6)$$

These simplifications thus reduce the progressive coding objective to simply being

$$\min_{d_{1:T}} \sum_{k=1}^{T} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_k|x)} f(x, d_k(z_k)) \quad (7)$$

for a fixed choice of diffusion process q. When the distortion function is taken to be mean-squared error $f = \| \cdot \|^2$, Equation (7) recovers the DDPM objective. This shows that the DDPM objective is a special case of the progressing coding objective, and explains why DDPMs have an inductive bias to be good progressive coders (Ho et al., 2020).

3.4. Progressive Coding and Sampling Quality

In the previous section, we observed that the DDPM objective is a special case of the progressive coding objective. This observation is significant because Ho et al. (2020) suggested that DDPM's superior image synthesis capabilities may be attributable to it being a good progressive coder with respect to mean-squared error (MSE) distortion. The progressive coding hypothesis, thusly stated, is that training an HLVM to be a good progressive coder with respect to mean-squared error will improve its sampling quality.[3] For notational simplicity, all subsequent mentions of progressive coding will be with respect to mean-squared error distortion unless otherwise specified.

For the remainder of this paper, we shall test this hypothesis in the context of another class of HLVMs—namely, the hierarchical variational autoencoder (HVAE). Since we can analyze and train HVAEs via the more generalized concept of progressive coding presented in Equations (1) to (3), we shall examine the extent to which conventionally-trained HVAEs are good at progressive coding, and test whether training HVAEs explicitly via a progressive coding objective will improve its sampling quality.

4. HVAEs are Not Good Progressive Coders

Our paper will consider the simplest case of progressive coding, where we partition an HVAE's hierarchical latent variables $z_{1:T}$ into two groups $(z_{1:K}, z_{K+1:T})$, where the choice of K is prescribed. Our decision to tackle the most basic case of progressive coding stems from the simplicity of the experimental design as well as the observation that diffusion-based models appear to work well even with very few diffusion steps (Anonymous, 2022; Xiao et al., 2021). Of particular interest to us is the pareto-optimality of the information encoded by $z_{1:K}$, since conventionally-trained HVAE do not directly subject the strict subset $z_{1:K}$ to direct decoding and distortion minimization. Our goal in this section is to analyze the rate-distortion (R_K, D_K) associated with the representation $z_{1:K}$ learned by a conventionally-trained HVAE and assess its pareto-optimality.

4.1. Method

Our experimental procedure begins by training an HVAE conventionally via ELBO maximization. We then measure the rate,

$$R_K = \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:K} \mid x) \parallel p(z_{1:K})),$$

[3] It may come as a surprise to see mean-squared error as the distortion function of choice for improving image sample quality. We shall examine the connection between pixel-space MSE and the perceptual similarity metric LPIPS in Section 6.

associated with the first K latent variables learned by the HVAE for some choice of K. To decode directly from $z_{1:K}$, we fix our learned q and measure the associated distortion D_K via the following optimization problem

$$D_K = \min_d \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:K}|x)} \|x - d(z_{1:K})\|^2.$$

To determine the pareto-optimality of (R_K, D_K), we set the target rate to $\gamma = R_K$ and determine the γ-optimal distortion via

$$\lambda^* = \min_{d,p,q} \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:K}|x)} \|x - d(z_{1:K})\|^2$$
$$\text{s.t. } \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:K} \mid x) \parallel p(z_{1:K})) \leq R_K.$$

We optimize this objective using a Lagrange multiplier (Zhao et al. (2018); Appendix A.5). Our solution to the optimization problem gives us an (upper bound) estimate of λ^*. This then allows us to compare D_K against λ^* to assess the pareto-optimality of (R_K, D_K).

Here, we note that training HVAEs with a rate-constraint bears similarity to the Generalized ELBO with Constrained Optimization (GECO) described by Rezende & Viola (2018). However, GECO gradually anneals the constraint as a tool for improving optimization stability; in contrast, we adopt a fixed constraint and apply rate-constrained optimization simply as a means of investigating the progressive coding hypothesis.

We conduct the experiment on SVHN, CelebA, and LSUN-Bedrooms by first training a VDVAE (Child, 2020). Since VDVAE has a convolutional architecture with spatial latent variables, we set K to be such that z_K's spatial resolution is eight times smaller than the original height and width dimensions of the image dataset. In particular, for SVHN, we chose K so that $z_{1:K}$ is the set of all latent variables with spatial resolution 4×4 or smaller; for CelebA-64 and LSUN-Bedrooms-64, we chose K so that $z_{1:K}$ has resolution 8×8 or smaller.

4.2. Results

Table 1 compares the rate-distortion trade-off in the first K latent variables achieved by conventional training versus γ-optimal training. Since we set the rate-target to $\gamma = R_K$, both conventional training and γ-optimal training achieve roughing similar rates. However, γ-optimal training achieves significantly better mean-squared error—roughly halving the mean-squared error D_K achieved by conventional training across all three datasets. This indicates that the point (R_K, D_K) achieved by conventional training is not pareto-optimal.

We compare the reconstructed images achieved by conventional versus γ-optimal training in Figure 1. We see that γ-optimal training results in sharper and higher-fidelity reconstructions, demonstrating that the improved mean-squared error has a perceptually-noticeable impact on the reconstructions. Overall, we conclude that the representations $z_{1:K}$ learned by conventional training are not good for lossy compression, and thus that conventionally-trained HVAEs are not good progressive coders.

Table 1. Comparison of D_K associated with a conventionally-trained HVAE against our estimate of the γ-optimal distortion λ^* achieved by rate-targeted training where $\gamma = R_K$. We report our estimate as an upper bound on λ^*. We also report the realized rate R'_K that is actually achieved by the rate-targeted model. All values are per-dimension (i.e., normalized by the data dimensionality).

	Conventional Training		γ-Optimal Training	
Dataset	R_K	D_K	R'_K	$\geq \lambda^*$
SVHN	0.0168	0.0200	0.0162	**0.0063**
CelebA	0.0142	0.0274	0.0135	**0.0128**
LSUN	0.0187	0.0611	0.0181	**0.0339**

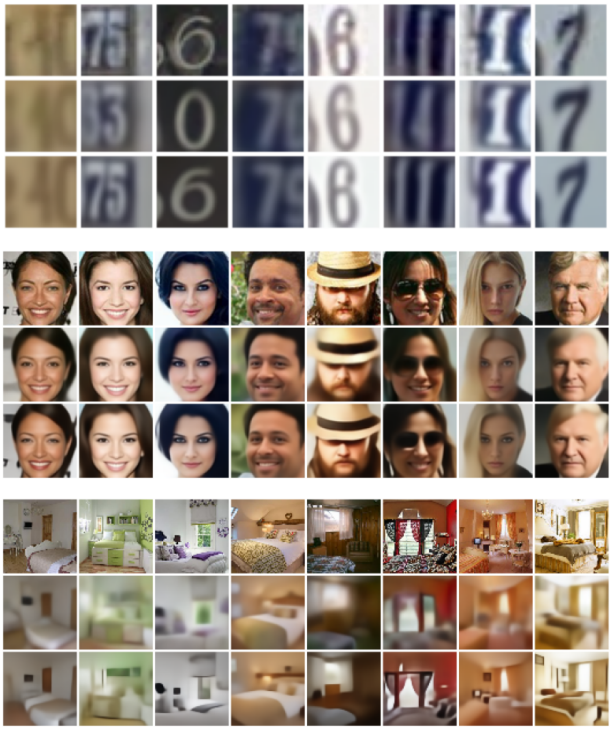

Figure 1. For each dataset (in order: SVHN, CelebA, LSUN), we compare the original image (top row), the reconstructions from conventional training (middle row), and the reconstructions from γ-optimal training (bottom row). Using γ-optimal training results in sharper and higher-fidelity reconstructions.

5. Progressive Coding Improves Sample Quality in HVAEs

Previously, we partitioned the hierarchical latent variables of a conventionally-trained HVAE into two groups $(z_{1:K}, z_{K+1:T})$ and showed that the associated rate-distortion point (R_K, D_K) is not pareto-optimal, indicating that conventionally-trained HVAEs are not good progressive coders. In this section, we shall encourage the HVAE to be better at progressive coding by proposing an alternative training procedure—one which forces (R_K, D_K) to be pareto-optimal. We shall then test whether doing so improves the HVAE's sampling quality.

Here, we note that the exact choice of pareto-optimal point may be critical; choosing the pareto-optimal point where $R_K = 0$ will in fact harm the model's performance since it will prevent the HVAE from making statistical use of its first K latent variables. This would be analogous to injecting an arbitrarily large amount of noise in the middle of a DDPM's diffusion process. Similar to how the choice of noise schedule is important for diffusion models with a discrete number of diffusion steps (Kingma et al., 2021), we treat the selection of the pareto-optimal point as a hyperparameter.

Our procedure for enforcing (R_K, D_K) to be pareto-optimal is simple; Equations (1) and (2) offer two such methods for selecting a pareto-optimal point: either fix a rate target γ and directly optimize $z_{1:K}$ to minimize distortion, or fix a distortion target λ and directly optimize $z_{1:K}$ to minimize rate. These procedures, respectively, will require us to prescribe a choice of rate target or distortion target as a hyperparameter. We choose to work with distortion-targeting because we find the Euclidean distance more intuitive than an information-theoretic quantity; in particular, it is useful that a mean-squared error of zero corresponds to perfect reconstruction, whereas it is unclear how large the rate should be to achieve near-perfect reconstructions. We shall thus proceed by describing the our distortion-targeting procedure, and how we trained the remaining $z_{K+1:T}$ latent variables in the full HVAE.

5.1. Method

To enforce pareto-optimality of (R_K, D_K), we optimized the following objective function,

$$\min_{d,p,q} \mathbb{E}_{p_{\text{data}}(x)} D_{\text{KL}}(q(z_{1:K} \mid x) \parallel p(z_{1:K}))$$
$$\text{s.t. } \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q(z_{1:K}|x)} f(x, d(z_{1:K})) \leq \lambda,$$

for various choices of distortion target λ. Optimizing this objective allows us to learn the subcomponents $p(z_{1:K})$ and $q(z_{1:K} \mid x)$ of the HVAE, and ensures that the latent variables $z_{1:K}$ are trained so that (R_K, D_K) is pareto-optimal. To complete the full HVAE, we must still learn $p(x, z_{K+1:T} \mid z_{1:K})$ and $q(z_{K+1:T} \mid x, z_{1:K})$. To do so, we simply freeze our learned choices for $p(z_{1:K})$ and $q(z_{1:K} \mid x)$ and optimize the remaining subcomponents of the HVAE via the ELBO objective

$$\min_{p',q'} \mathbb{E}_{p_{\text{data}}(x)} \Big[\mathbb{E}_{q(z_{1:T})} \ln p(x \mid z_{1:T}) + D_{\text{KL}}(q(z_{1:T} \mid x) \parallel p(z_{1:T})) \Big],$$

where p' and q' denote optimization over the specific subcomponents $p(x, z_{K+1:T} \mid z_{1:K})$ and $q(z_{K+1:T} \mid x, z_{1:K})$.

Stated as such, our method for encouraging progressive coding is simply a two-stage training procedure that first learns the representations $z_{1:K}$ in the first stage which directly exposes them to distortion minimization, followed by learning the remaining representations $z_{K+1:T}$ in the second stage in accordance to the ELBO objective. This procedure can easily be extended to a multi-stage training when enforcing progressive coding over more than two groups of latent variables.

In practice, however, the typical architectural choices for HVAEs (and for us in particular, the VDVAE) often have parameter-sharing within and across p and q that make subcomponent freezing a challenge. We break the parameter-sharing by instead training two fully-fledged VDVAEs and considering the composition of the two VDVAEs to be our full HVAE. In the first stage, we train an unconditional VDVAE to minimize rate while targeting a particular mean-squared error distortion.[4] In the second stage, we then train a conditional VDVAE that learns to conditionally generate the original image based on the reconstructed image from the first-stage VDVAE.[5] The sampling procedure for the full HVAE thus involves first unconditional sampling from the first-stage VDVAE, followed by conditional sampling second-stage VDVAE conditioned on the first-stage sample. For fair comparison against a conventionally-trained HVAE, we made sure that the conventionally-trained HVAE has the exact same functional parameterization and number of trainable parameters as our full model—differing only in the *choice* of parameter values due to different training procedures. Similar to Section 4, we conduct our experiments on SVHN, CelebA, and LSUN-Bedrooms. To test whether progressive coding improves sampling quality, we compare the FID scores (Heusel et al., 2017) for the conventionally-trained HVAE versus our progressively-coded HVAE.

[4] To do so, we replaced the discretized mixture of logistics head with a linear head.

[5] We condition on the reconstructed image rather than directly on the first-stage VDVAE's representations in part because of architectural convenience, and in part because the reconstructed image can be seen as a summarization of the first-stage representations. This gives our second-stage VDVAE the interpretation as a probabilistic *deblurring* autoencoder, since it conditions on the blurry images constructed by the first-stage VDVAE.

5.2. Results

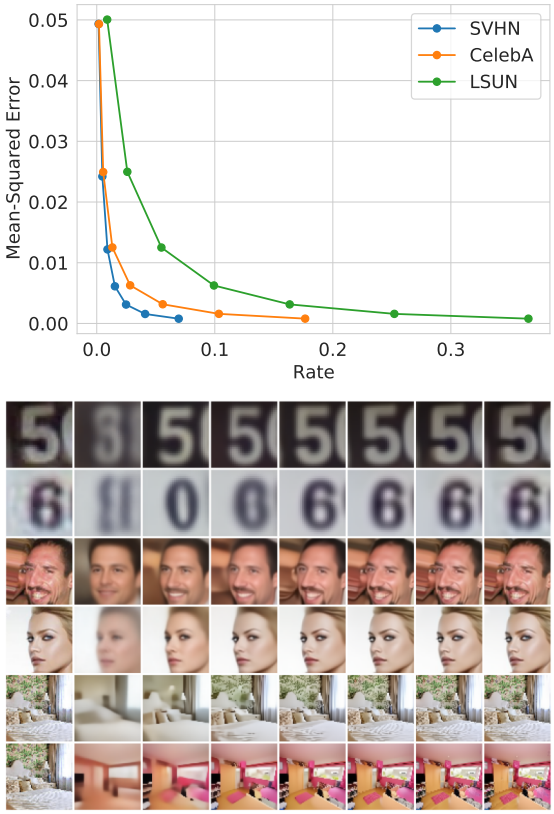

Figure 2. Rate versus distortion curves for the first-stage VDVAE models trained with six different distortion targets. All values are per-dimension. We also show representative reconstructions from all six VDVAEs (in order from highest to lowest distortion target), where the first column is the original image.

Table 2. Quantitative evaluation of conventionally-trained versus our progressively-coded HVAEs. ELBOs are per-dimension (i.e., normalized by the data dimensionality).

	Conventional Training		Progressive Coding	
Dataset	ELBO	FID	ELBO	FID
SVHN	$\mathbf{-1.31}$	18.92	-1.46	$\mathbf{10.07}$
CelebA	$\mathbf{-1.44}$	13.33	-1.58	$\mathbf{8.54}$
LSUN	$\mathbf{-1.72}$	40.71	-1.78	$\mathbf{36.87}$

We show the results of our first-stage VDVAE training (with a series of exponentially smaller distortion-targets $\{0.0512, \ldots, 0.0032, 0.0016, 0.0008\}$) in Figure 2. Our visualization of how the reconstructions change across models further show that below a distortion-target of $\sim$0.0032, the any further improvements in mean-squared error become nearly-imperceptible despite incurring a large rate. To build the second-stage VDVAE, we used the first-stage VDVAE with distortion-target 0.0032 for SVHN and CelebA. Due to LSUN's generally higher rate, we instead used the VDVAE with distortion-target 0.0064. We compare the full

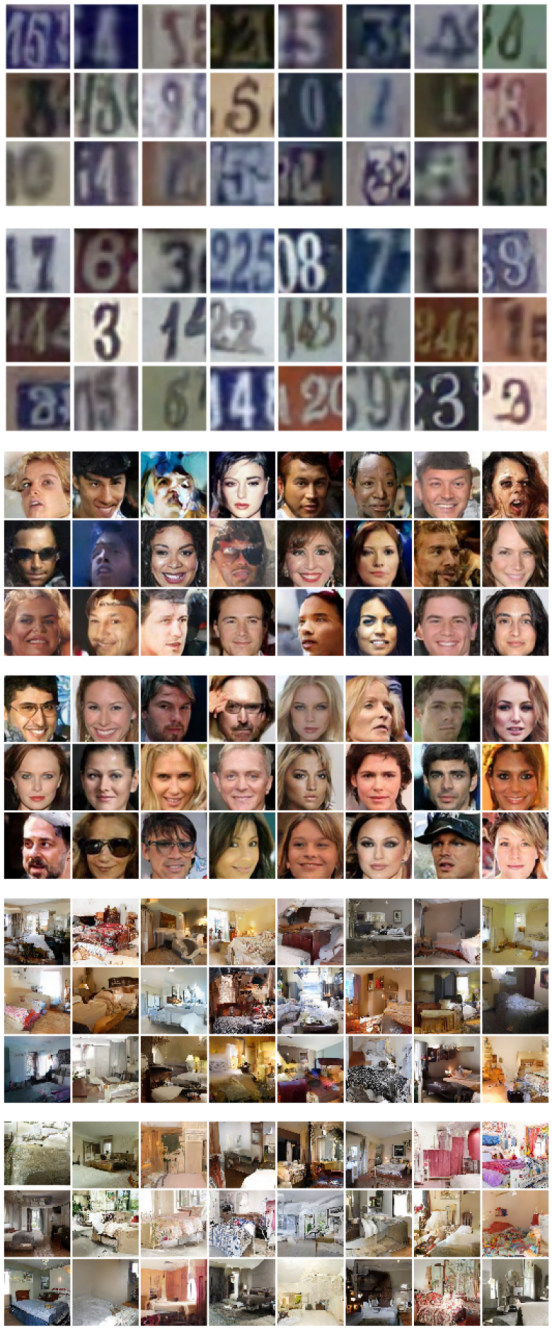

Figure 3. For each dataset (in order: SVHN, CelebA, LSUN), we compare non-cherry picked samples from conventional training (top row), versus the samples progressive training. We do not use reduced-temperature sampling (Kingma & Dhariwal, 2018). In SVHN and CelebA, we see greater global structural consistency of the samples from the progressively-coded HVAE.

HVAE using conventional training versus our two-stage progressive-coding-based training procedure. Table 2 shows our quantitative evaluation of these two models on ELBO and FID. Our progressively-coded HVAE achieves better FID scores with slightly worse ELBO (to be expected since conventional training is unconstrained optimization of the ELBO). Visual inspection of the samples in Figure 3 further supports the hypothesis that progressive coding yields better sampling quality. This is especially evident in CelebA and SVHN, where we see the greatest percent FID improvement.

6. Mean-Squared Error and Perceptual Similarity

We showed in Section 5 that progressive coding can significantly improve HVAE's sampling quality. This phenomenon naturally raises an important question: why does progressive coding improve sample quality—let alone progressive coding with respect to mean-squared error distortion? In this section, we shall shed some additional light on this phenomenon.

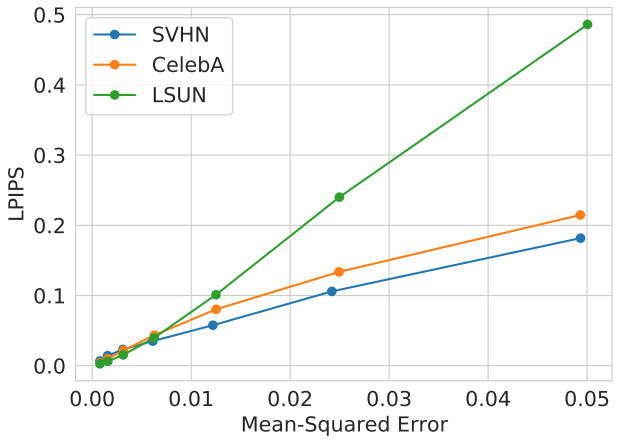

Figure 4. Comparison of LPIPS versus mean-squared error for our first-stage VDVAE models. In all three datasets, we see a clear correlation between LPIPS and mean-squared error.

We begin by remarking on the relation between pixel-space mean-squared error and perceptual similarity for natural images. It is well-known that, in general, mean-squared error is a poor measure of perceptual similarity (Blau et al., 2018). We can see this by considering two hypothetical compressors—one which translates the image during reconstruction, and another which deletes the middle of the image during reconstruction. It is possible for both compressors to achieve the same mean-squared error distortion, even though deletion is a perceptually more aggressive operation. This means that if we were to plot the set of all possible compressors by comparing each compressor's average mean-squared error distortion against its average perceptual similarity distortion (i.e., using Zhang et al. (2018)'s LPIPS), we will likely find no meaningful correlation.

Crucially, however, the aforementioned reasoning does not consider the *rate* of each compressor. If we restrict our hypothetical plot solely to the set of compressors (d, p, q) that are *rate-minimal* for any choice of mean-squared error target,

$$\{(d, p, q) \mid \exists \lambda, (d, p, q) \text{ is } \lambda\text{-optimal}\},$$

it is no longer clear what the resulting LPIPS versus mean-squared error plot will look like. We therefore conduct this experiment using the first-stage VDVAE models that we trained in Section 5, by measuring each model's image reconstructions based on both mean-squared error and LPIPS.

Figure 4 demonstrates a surprising monotonicity between LPIPS and pixel-space mean-squared error even in the large-error regime.[6] This suggests that rate-minimization with a targeted mean-squared error distortion is in fact prioritizing perceptually relevant information. If so, then we can interpret progressive coding with respect to mean-squared error as prioritizing the modeling of more perceptually important information first. We further speculate that modeling perceptually-relevant information first may be important because it makes the perceptually-relevant information less susceptible to cascading error during sampling—a potential issue if the information had been modeled further downstream in the hierarchical model. While subjecting this conjecture to rigorous testing is beyond the scope of this paper, we hope that this perspective will serve as a promising candidate hypothesis for future research on this phenomenon.

7. Conclusion

In this work, we reframed the task of image synthesis as a representation learning problem and sought to regularize HVAEs toward representations that will improve its sampling quality. We formalized the progressive coding hypothesis presented in Ho et al. (2020) and proposed a simple progressing coding-based training procedure that can be applied to HVAEs. Our experiments show that conventionally-trained HVAEs are poor progressive coders, and that training the HVAE to be a better progressive coder improves

[6] This finding seems to be at odds with existing literature that stipulates a trade-off between mean-squared error distortion and perception (Blau & Michaeli, 2018; 2019). We attribute our contrary finding to our differing definitions of perceptual quality—we rely on LPIPS, which measures perceptual similarity between the reconstructed image and the original reference image, versus the "no-reference" perceptual quality index used in Blau & Michaeli (2019), which measures the divergence between the original versus reconstructed distributions. Understanding why mean-squared error for λ-optimal models tracks with LPIPS remains an open research problem.

its sampling quality. Our work lends further support to the progressive coding hypothesis and suggests promising future research directions into the theory of progressive coding (such as why the mean-squared error distortion for rate-minimal compressors appears to track with perceptual similarity) as well as its application to variational autoencoders (such as scaling the application of progressive coding to more levels of hierarchy in HVAEs).

8. Acknowledgements*

We are thankful to Kristy Choi and Casey Chu for their supportive discussions and feedback. This research is supported by NSF(#1651565), AFOSR (FA95501910024), ARO (W911NF-21-1-0125) and the Sloan Fellowship.

References

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. Expressive speech synthesis via modeling expressions with variational autoencoder. *arXiv preprint arXiv:1804.02135*, 2018.

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. Fixing a broken elbo. In *International Conference on Machine Learning*, pp. 159–168. PMLR, 2018.

Anonymous. Progressive distillation for fast sampling of diffusion models. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI. under review.

Blau, Y. and Michaeli, T. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237, 2018.

Blau, Y. and Michaeli, T. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pp. 675–685. PMLR, 2019.

Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., and Zelnik-Manor, L. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Child, R. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Dockhorn, T., Vahdat, A., and Kreis, K. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.

Donahue, C., McAuley, J., and Puckette, M. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Hsu, W.-N., Zhang, Y., and Glass, J. Learning latent representations for speech generation and transformation. *arXiv preprint arXiv:1704.04222*, 2017.

Jalal, A., Arvinte, M., Daras, G., Price, E., Dimakis, A. G., and Tamir, J. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34, 2021.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

Levine, N., Chow, Y., Shu, R., Li, A., Ghavamzadeh, M., and Bui, H. Prediction, consistency, curvature: Representation learning for locally-linear control. *arXiv preprint arXiv:1909.01506*, 2019.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016b.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Rezende, D. J. and Viola, F. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Backpropagation And Approximate Inference In Deep Generative Models. *arXiv preprint arXiv:1401.4082*, 2014.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. *Advances in neural information processing systems*, 29:3738–3746, 2016.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Sultan, M. M., Wayment-Steele, H. K., and Pande, V. S. Transferable neural networks for enhanced sampling of protein dynamics. *Journal of chemical theory and computation*, 14(4):1887–1894, 2018.

Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Uria, B., Côté, M.-A., Gregor, K., Murray, I., and Larochelle, H. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17 (1):7184–7220, 2016.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

Van Hoof, H., Chen, N., Karl, M., van der Smagt, P., and Peters, J. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 3928–3934. IEEE, 2016.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 7444–7453. PMLR, 2019.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhao, S., Song, J., and Ermon, S. The information autoencoding family: A lagrangian perspective on latent variable generative models. *arXiv preprint arXiv:1806.06514*, 2018.

A. Training Information

A.1. VDVAE Hyperparameters

Across all experiments, we use the default official VDVAE implementation with the following specifications applied to the hyperparameter file `hps.py`

```
svhnx.dec_blocks = "1x1,4m1,4x2,8m4,8x5,16m8,16x10,32m16,32x21"
svhnx.enc_blocks = "32x11,32d2,16x6,16d2,8x6,8d2,4x3,4d4,1x3"
svhnx.n_batch = 12

celeba.dec_blocks = "1x1,4m1,4x2,8m4,8x5,16m8,16x10,32m16,32x21,64m32,64x5"
celeba.enc_blocks = "64x5,64d2,32x11,32d2,16x6,16d2,8x6,8d2,4x3,4d4,1x3"
celeba.n_batch = 6

lsun.dec_blocks = "1x1,4m1,4x2,8m4,8x5,16m8,16x10,32m16,32x21,64m32,64x5"
lsun.enc_blocks = "64x5,64d2,32x11,32d2,16x6,16d2,8x6,8d2,4x3,4d4,1x3"
lsun.n_batch = 6

# The following is applied to all datasets:
parser.add_argument('--width', type=int, default=384)
parser.add_argument('--zdim', type=int, default=16)
parser.add_argument('--lr', type=float, default=0.0002)
parser.add_argument('--skip_threshold', type=float, default=500.0)
parser.add_argument('--skip_threshold', type=float, default=500.0)
parser.add_argument('--ema_rate', type=float, default=0.9999)
parser.add_argument('--adam_beta1', type=float, default=0.9)
parser.add_argument('--adam_beta2', type=float, default=0.9)
parser.add_argument('--warmup_iters', type=float, default=0)
```

We shall refer to this setup as the "original" VDVAE setup. We apply experiment-specific minor modifications, which we shall describe below.

A.2. (R_K, D_K) Pareto-Optimality Experiment

We first trained the original VDVAE for 800K steps. We then created a masked VDVAE decoder which takes as inputs only the first $z_{1:K}$ latent variables via the `Decoder.forward_manual_latents` functionality. We mask out the latent variables $z_{K+1:T}$ in the second decoder. We train the second decoder for 400K steps.

To perform γ-optimal training of the VDVAE with only the first $z_{1:K}$ latent variables, we simply paired the original encoder with our masked VDVAE decoder (where we replaced the discreteized mixture of logistics head (`DmolNet`) with a 1×1 convolutional head to do mean prediction) and perform joint training on objective in Equation (1) for 400K steps.

A.3. First-Stage VAE

In order to perform distortion-targeting, we removed the discretized mixture of logistics head and simply replaced it with a 1×1 convolutional head to do mean prediction. We train all models for 800K steps.

A.4. Second-Stage VAE

Our second-stage VAE conditions on the output of the first-stage model to predict the original data point. We use exactly the same architecture as the first-stage VDVAE with the following architectural modifications:

1. Our encoder takes as input both the original data point and the first-stage model prediction.
2. Our decoder must also conditional now condition on the first-stage model prediction. To do so, we create a second encoder (same architecture, except for taking only the first-stage model prediction as input), and supply the hidden

representations from this encoder to the existing VDVAE decoder, thus achieving a U-Net architecture (Ronneberger et al., 2015).

3. We preserve the use of the discretized mixture of logistics head from the original VDVAE in order to optimize for ELBO.

To perform progressive coding, we train the first-stage VDVAE based on the λ-optimal training (i.e. distortion-targted training) objective, freeze the first-stage VDVAE, and then training the second-stage VDVAE (for 800K steps). To perform conventional training, we simply jointly trained both the first and second-stage VDVAEs end-to-end via ELBO maximization for 800K steps. Note that every trainable parameter in both the two-stage training process and the end-to-end conventional training process is thus updated exactly 800K times.

A.5. Constrained Optimization via Lagrangian Multiplier Optimization

Letting $R(p, q)$ and $D(d, q)$ denote the rate and distortion (as measured by mean-squared error) respectively, we can follow Zhao et al. (2018) and perform λ-optimal training by jointly optimizing the following two objectives with gradient descent.

$$\underset{d,p,q}{\text{minimize}}\, R(p, q) + D(d, q) + \eta \cdot (D(d, q) - \lambda) \tag{8}$$

$$\underset{\eta}{\text{maximize}}\, \eta \cdot (D(d, q) - \lambda), \tag{9}$$

where η denotes the the Lagrange multiplier for our constraint. In practice, we instead perform

$$\underset{p,q}{\text{minimize}}(1 - \eta) \cdot R(p, q) + \eta \cdot D(d, q) \tag{10}$$

$$\underset{\eta}{\text{maximize}}\, \eta \cdot (D(d, q) - \lambda). \tag{11}$$

The general intuition still holds: if $D > \lambda$, then Equation (11) is maximized by increasing η, which in turn encourages Equation (10) to reduce D. We restrict $\eta \in [0.0001, 0.9999]$ via projected gradient descent and initialize at $\eta = 0.9999$ to mimic KL annealing Sønderby et al. (2016). We use the same technique described above to also perform γ-optimal training (simply swap the targeted term from D to R).

We optimize the Lagrange multiplier with Adam as well, using a learning rate of 0.0002 and $(\beta_1, \beta_2) = (0, 0.999)$, as well as an exponential moving average of 0.99 to slow down and smooth out the adjustment of the Lagrange multiplier in our weighted rate-distortion objective in Equation (8).

B. Additional Sample Visualizations

Figure 5. SVHN samples from conventionally-trained HVAE (top) versus progressively-coded HVAE (bottom).

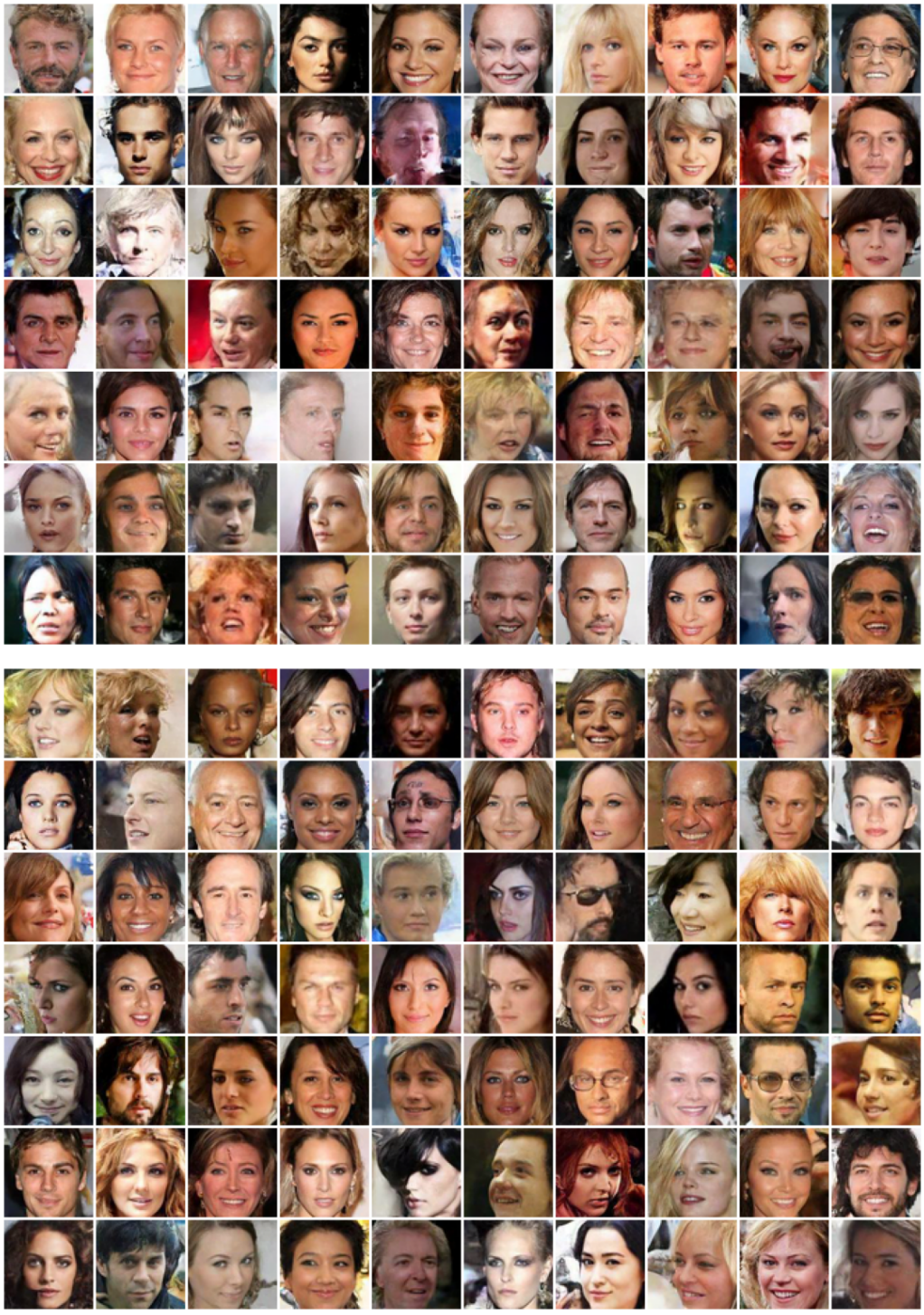

Figure 6. CelebA samples from conventionally-trained HVAE (top) versus progressively-coded HVAE (bottom).

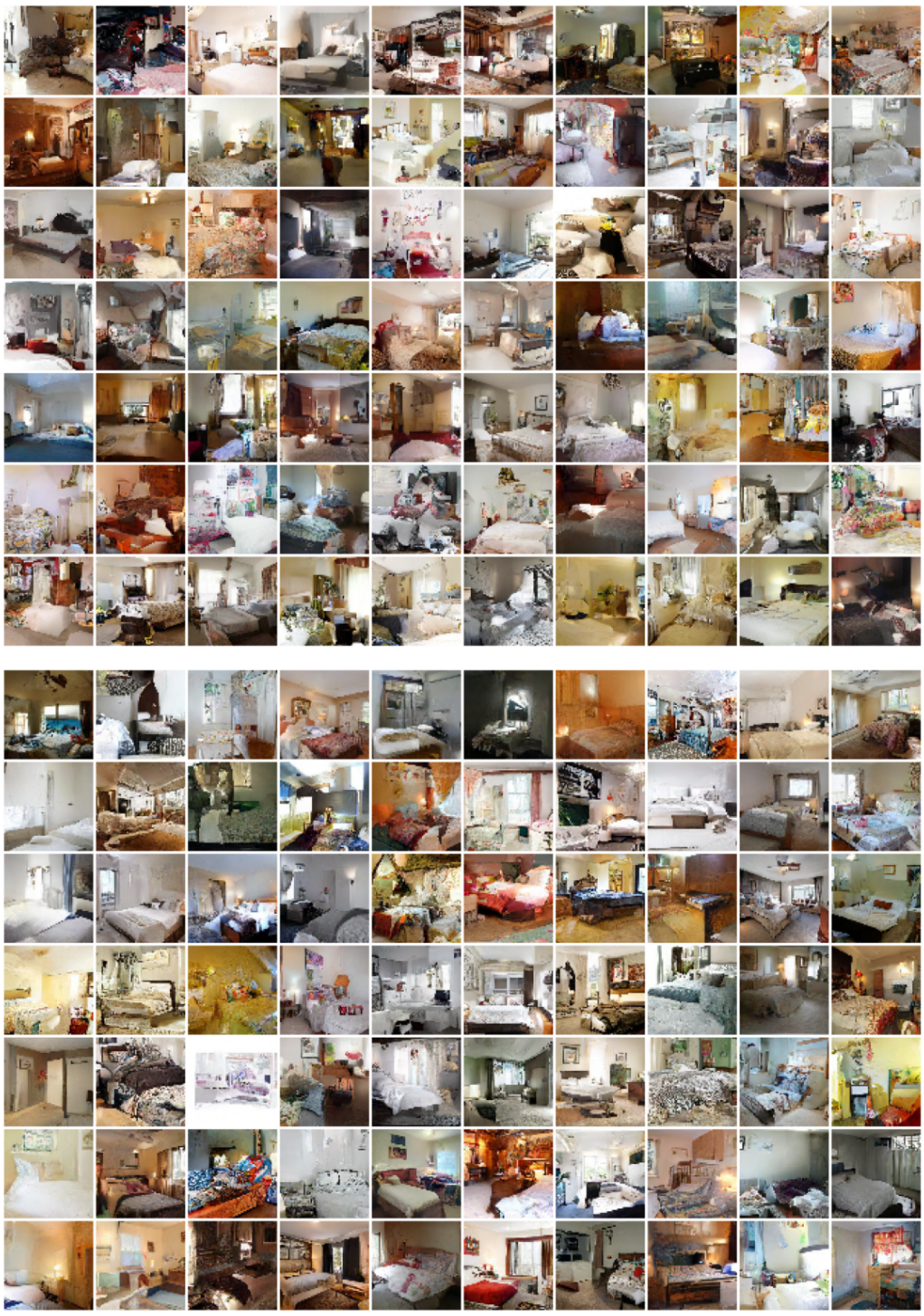

Figure 7. LSUN-Bedrooms samples from conventionally-trained HVAE (top) versus progressively-coded HVAE (bottom).