ImageBART: Bidirectional Context with Multinomial Diffusion for Autoregressive Image Synthesis

Anonymous Author(s)

Abstract

Autoregressive models and their sequential factorization of the data likelihood have recently demonstrated great potential for image representation and synthesis. Nevertheless, they incorporate image context in a linear 1D order by attending only to previously synthesized image patches above or to the left. Not only is this unidirectional, sequential bias of attention unnatural for images as it disregards large parts of a scene until synthesis is almost complete. It also processes the entire image on a single scale, thus ignoring more global contextual information up to the gist of the entire scene. As a remedy we incorporate a coarse-to-fine hierarchy of context by combining the autoregressive formulation with a multinomial diffusion process: Whereas a multistage diffusion process successively compresses and removes information to coarsen an image, we train a Markov chain to invert this process. In each stage, the resulting autoregressive ImageBART model progressively incorporates context from previous stages in a coarse-to-fine manner. Experiments demonstrate the gain over current autoregressive models, continuous diffusion probabilistic models, and latent variable models. Moreover, the approach enables to control the synthesis process and to trade compression rate against reconstruction accuracy, while still guaranteeing visually plausible results.

1 Introduction

Spurred by the increasingly popular attention mechanism, a remarkably simple principle has driven progress in deep generative modeling over the past few years: Factorizing the likelihood of the data in an autoregressive (AR) fashion

$$p(x) = \prod_i p_\theta(x_i|x_{<i}) \tag{1}$$

and subsequently learning the conditional transition probabilities with an expressive neural network such as a transformer [63]. The success of this approach is evident in domains as diverse as language modeling [3], music generation [10], neural machine translation [36, 64], and (conditional) image synthesis [44, 4]. However, especially for the latter task of image synthesis, which is also the focus of this work, the high dimensionality and redundancy present in the data challenges the direct applicability of this approach.

Missing Bidirectional Context Autoregressive models which represent images as a sequence from the top-left to the bottom-right have demonstrated impressive performance in sampling novel images and completing the lower half of a given image [4, 12]. However, the unidirectional, fixed ordering of sequence elements not only imposes a perceptually unnatural bias to attention in images by only considering context information from left or above. It also limits practical applicability to image modification: Imagine that you only have the lower half of an image and are looking for a completion of the upper half then these models fail at this minor variation of the completion task. The importance of contextual information from both directions [27] has also been recognized in the context of language modeling [9, 35]. However, simply allowing bidirectional context as in [9] does not provide a valid factorization of the density function for a generative model. Furthermore, the sequential sampling strategy introduces a gap between training and inference, as training relies on so-called

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

teacher-forcing [1] (where ground truth is provided for each step) and inference is performed on previously sampled tokens. This *exposure bias* can introduce significant accumulations of errors during the generation process, affecting sample quality and coherence [47].

Global Context & Control via Multinomial Diffusion We propose a coarse-to-fine approach that addresses the unidirectional bias of generative autoregressive models and their exposure bias as well as the lacking global context. We formulate learning the data density as a hierarchical problem. A coarser stage provides compressed contextual side information about the *entire* image for the autoregressive process on the next finer stage. We utilize a diffusion process to gradually eliminate information and compress the data, yielding a hierarchy of increasingly abstract and compact representations. The first scale of this approach is a discrete representation learning task (cf. [62, 48, 10, 12, 66, 46]). Subsequently, we further compress this learned representation via a fixed, multinomial diffusion process [54, 21]. We then invert this process by training a Markov chain to recover the data from this hierarchy. Each Markovian transition is modeled autoregressively but it simultaneously attends to the preceding state in the hierarchy, which provides crucial global context to each individual autoregressive step. As each of this steps can also be interpreted as learning a denoising cloze task [35], where missing tokens at the next finer stage are "refilled" with a bidirectional encoder and an autoregressive decoder, we dub our approach *ImageBART*.

Contributions of our work Our approach tackles high-fidelity image synthesis with autoregressive models by learning to invert a fixed multinomial diffusion process in a discrete space of compact image representations to successively introduce context. This reduces both the often encountered exposure bias of AR models and also enables locally controlled, user-interactive image editing. Additionally, our model effectively handles a variety of conditional synthesis tasks and our introduced hierarchy corresponds to a successively compressed image representation. We observe that our model sample visually plausible images while still enabling a trade-off between reconstruction capability and compression rate.

2 Related Work

Latent Variable Models Among likelihood-based approached, latent variable models represent a data distribution with the help of unobserved latent variables. For example, Variational Autoencoders (VAEs) [29, 49] encode data points into a lower dimensional latent variable with a factorized distribution. This makes them easy to sample, interpolate [34, 28] and modify [65]. In a conditional setting [30], latent variables which are independent from the conditioning lead to disentangled representations [22, 58, 38, 50]. A hierarchy of latent variables [55] gives mutli-scale representations of the data. Unfortunately, even the deepest instantiations of these models [37, 60, 6] lack in sample quality compared to other generative models and are oftentimes restricted to highly regular datasets.

Autoregressive Models AR models represent a distribution as a product of conditional, learnable factors via the chain rule of probability densities. While this makes them powerful models for density estimation [59, 15], their samples often lack global consistency. Especially on image data modeled with convolutional architectures [61, 51], this has been attributed to a locality bias of convolutional neural networks (CNNs) which biases the model towards strong local correlations between neighboring pixels at the expense of a proper modeling of coherence [31, 13]. This leads to samples resembling texture patterns without discernible global structure. Attempts to fix this properties by including explicit latent variables [18, 5, 13] have not been overly successful, mainly due the expressiveness of AR models, providing little incentive for learning additional latent variables.

Generative Models on Improved Representations Another successful line of work first learn an improved image representation and subsequently learn a generative model for this representation [62, 7]. Most works [48, 12, 46] learn a discrete representation which is subsequently modeled autoregressively. Learning a compact representation enables the use of transformers for autoregressive modeling [4], which avoids the locality bias of CNNs and, when combined with adversarial learning [16], enables sampling of coherent high-resolution images [12]. However, AR modeling of a learned representation still limits applications compared to latent variable models. Their samples can still exert artefacts resulting from a sequential modeling of components, and, since these models are always trained by "teacher-forcing", they are susceptible to an exposure bias [1, 47, 17, 52, 33].

Diffusion Probabilistic Models Diffusion probabilistic models revert a fixed, diffusion process with a learned Markov Chain [54]. Being directly applied in pixel space, however, downstream analysis reveals that these models tend to optimize subtle details of the modeled data, which have little contribution to the sample quality [20], particularly hindering applications on high-resolution and -complexity datasets. By using a multinomial diffusion process on a compressed, discrete representation of images, we circumvent these issues. Diffusion probabilistic models require a very

large number of diffusion steps in order to model the reverse process with a model distribution that factorizes over components. Because our approach uses autoregressively factorized models for the reverse process, we can reduce the required number of steps and obtain significant improvements in sampling speed and the ability to model complex datasets.

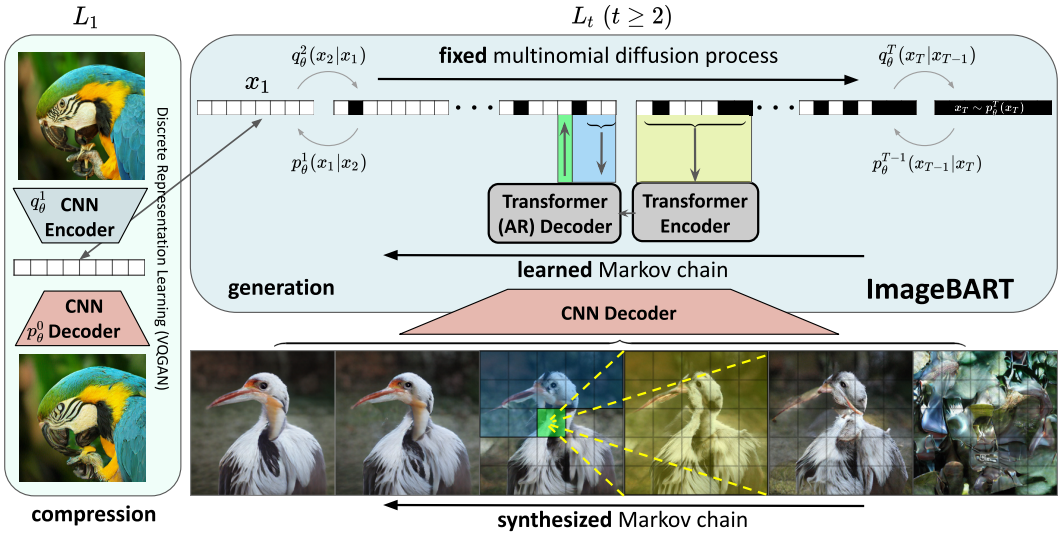

Figure 1: Overview over our approach: We first learn a compressed, discrete image representation x_1 and subsequently our generative ImageBART model reverts a fixed multinomial diffusion process via a Markov Chain, where the individual transition probabilities are modeled as independent autoregressive encoder-decoder models. This introduces a coarse-to-fine hierarchy such that each individual AR model can attend to global context from its preceding scale in the hierarchy.

3 Method

3.1 Hierarchical Generative Models

To tackle the difficult problem of modeling a highly complex distribution $p(x)$ of high-dimensional images x, we (i) introduce bidirectional context into an otherwise unidirectional autoregressive factorization of $p(x)$ as in Eq. (1) and (ii) reduce the difficulty of the learning problem with a hierarchical approach. To do so, we learn a sequence of distributions $(p_\theta^t)_{t=0}^T$, such that each distribution p_θ^{t-1} models a slightly more complex distribution with the help of a slightly simpler distribution p_θ^t one level above. This introduces a coarse-to-fine hierarchy of image representations $x_{0:T} := (x_t)_{t=0}^T$, such that an x_{t-1} is modeled conditioned on x_t, i.e. $x_{t-1} \sim p_\theta^{t-1}(x_{t-1}|x_t)$ and defines a reverse Markov Chain for $x =: x_0$ as $p_\theta(x_0) = p_\theta^T(x_T)\prod_{t=1}^T p_\theta^{t-1}(x_{t-1}|x_t)$. Since our goal is to approximate the original distribution $p(x)$ with $p_\theta(x_0)$, we introduce a forward Markov Chain, $q_\theta(x_{1:T}|x_0) = \prod_{t=1}^T q_\theta^t(x_t|x_{t-1})$, to obtain a tractable upper bound on the Kullback-Leibler (KL) divergence between p and p_θ, $\mathbb{KL}(p(x_0)\|p_\theta(x_0)) =: \mathcal{KL}$, using the evidence lower bound (ELBO). With $q_\theta^T(x_T|x_{T-1}) := p_\theta^T(x_T)$, we obtain

$$\mathcal{KL} \leq \underbrace{\mathbb{E}_{x_0,x_1} \log \frac{p(x_0)}{p_\theta^0(x_0|x_1)}}_{=:L_1 \rightarrow \text{ discrete repr. learning}} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{x_0,x_t}\mathbb{KL}(q_\theta^{t-1}(x_{t-1}|x_t,x_0)\|p_\theta^{t-1}(x_{t-1}|x_t))}_{=:L_t \rightarrow \text{ decoupled with diffusion process}} \tag{2}$$

We use L_1 to learn a compressed and discrete representation of images, such that subsequent stages of the hierarchy do not need to model redundant information (Sec. 3.2). With $L_t, t > 1$ we learn a model that can rely on global context from a coarser representation x_t to model the representation x_{t-1} (Sec. 3.3). See Fig. 1 for an overview of the proposed model.

3.2 Learning a compact, discrete representation for images

Since the first stage of the hierarchical process is the one that operates directly on the data, we assign it a separate role. To avoid that the optimization of L_t $(t = 1, \ldots, T)$ in Eq. (2) unnecessarily wastes capacity on redundant details in the input images—which is an often encountered property

of pixel-based likelihood models [62, 12, 40]—we take $L_1 = \mathbb{E}_{p(x_0)q^1_\theta(x_1|x_0)} \log \frac{p(x_0)}{p^0_\theta(x_0|x_1)}$ to be the reconstruction term for a discrete autoencoder model. This has the advantage that we can directly build on work in neural discrete representation learning, which has impressively demonstrated that discrete representations can be used for high-quality synthesis of diverse images while achieving strong compression. In particular, [39] and [12] have shown that adding an adversarial realism prior to the usual autoencoder objective helps to produce more realistic images at higher compression rates by locally trading reconstruction fidelity for realism.

More specifically, we follow [12] to encode images into a low-dimensional representation which is then vector-quantized with a learned codebook of size K to obtain $\{0, \dots, K-1\}^{h \times w} \ni x_1 \sim q^1_\theta(x_1|x_0)$ deterministically as the index of the closest codebook entry. The encoder is a convolutional neural network (CNN) with four downsampling steps, such that $h = H/16$ and $w = W/16$ for any input image $x_0 \in \mathbb{R}^{H \times W \times 3}$. For downstream autoregressive learning, this representation is then unrolled into a discrete sequence of length $N = h \cdot w$. To recover an image from x_1, we utilize a CNN decoder G, such that the reverse model is specified as

$$-\log p^0_\theta(x_0|x_1) \propto f_{rec}(x_0, G_\theta(x_1)) + \log D_\phi(G_\theta(x_1)) =: L_{rec}(x_0, x_1; \theta) + L_{adv}(x_1; \theta, \phi) \quad (3)$$

Here, f_{rec} denotes the perceptual similarity metric [14, 23, 11, 68] (known as LPIPS) and D_ϕ denotes a patch-based adversarial discriminator [16]. D_ϕ is optimized to differentiate original images x_0 from their reconstruction $G_\theta(x_1)$ using simultaneous gradient ascent, such that the objective for learning the optimal parameters $\{\theta^*, \phi^*\}$ reads:

$$\{\theta^*, \phi^*\} = \arg\min_\theta \max_\phi \Big(L_{rec}(x_0, x_1; \theta) - L_{adv}(x_1; \theta, \phi) + \log D_\phi(x_0) \Big) \quad (4)$$

Note that the optimization of θ via this objective includes the parameters of the learned codebook in addition to the parameters of the encoder and decoder.

3.3 Parallel learning of hierarchies

Under suitable choices for p_θ, q_θ, one can directly optimize these chains over $\sum_t L_t$. However, the objectives L_t of the hierarchy levels are coupled through the forward chain q_θ, which makes this optimization problem difficult. With expressive reverse models p^{t-1}_θ, the latent variables x_t are often ignored by the model [13] and the scale of the different level-objectives can be vastly different, resulting in a lot of gradient noise that hinders the optimization [42]. In the continuous case, reweighting schemes for the objective can be derived [19] based on a connection to score matching models [56]. However, since we are working with a discrete x_1, there is no analogue available.

While we could follow the approach taken for the first level and sequentially optimize over the objectives L_t, this is a rather slow process since each level $t-1$ needs to be converged before we can start solving level t. However, this sequential dependence is only introduced through the forward models q^t_θ and since q^1_θ already learns a strong representation, we can choose simpler and fixed, predefined forward processes for $q^t_\theta, t > 1$. Since the goal of these processes is to generate a hierarchy of distributions, the main objective is to reduce information in each transition. This is readily achieved by, e.g., randomly masking [9], removing [35] or replacing [21] a fraction of the components of x_{t-1}.

Multinomial diffusion This process of randomly replacing a fraction β_t of the components with random entries can be described as a multinomial diffusion process [21], a natural generalization of binomial diffusion [54]. The only parameter θ of q^t_θ is therefore β_t, which we consider to be fixed. Using the standard basis $e(k) = (\delta_{jk})_{j=1}^K$, the forward process can be written as a product of categorical distributions $\mathcal{C}$ specified in terms of the probabilities over the codebook indices:

$$q^t_\theta(x_t|x_{t-1}) = \prod_{i=1}^N \mathcal{C}(x^i_t|(1-\beta_t)e(x^i_{t-1}) + \beta_t \mathbb{1}/K), \quad t > 1 \quad (5)$$

where $\mathbb{1} = (1)_{j=1}^K$ is the all one vector. It then follows that after $t-1$ steps, on average, a fraction of $\bar{\alpha}_t := \prod_{l=2}^t (1-\beta_t)$ entries from x_1 remain unchanged in x_t, i.e.

$$q^t_\theta(x_t|x_1) = \prod_{i=1}^N \mathcal{C}(x^i_t|\bar{\alpha}_t e(x^i_1) + (1-\bar{\alpha}_t)\mathbb{1}/K), \quad t > 1. \quad (6)$$

This enables computation of the posterior $q_\theta(x_{t-1}|x_t, x_1) = \frac{q_\theta^t(x_t|x_{t-1})q_\theta(x_{t-1}|x_1)}{q_\theta(x_t|x_1)}$ for $t > 2$, and, using the fact that q_θ^1 is deterministic, we can rewrite L_t as

$$\mathbb{E}_{p(x_0)q_\theta(x_t|x_1)}\mathbb{KL}(q_\theta^{t-1}(x_{t-1}|x_t, x_1)\|p_\theta^{t-1}(x_{t-1}|x_t)), \quad t > 2 \tag{7}$$

such that the KL term can now be computed analytically for $t > 2$. For $t = 2$, we use a single sample Monte-Carlo estimate for the maximum likelihood reformulation, i.e.

$$\arg\min L_2 = \arg\max \mathbb{E}_{p(x_0)}\mathbb{E}_{q_\theta^2(x_2|x_1)} \log p_\theta^1(x_1|x_2). \tag{8}$$

Finally, we set p_θ^T to be a uniform distribution. This completes the definition of the reverse chain p_θ, which can now be started from a random sample for $x_T \sim p_\theta^T(x_T)$, denoised sequentially through $x_{t-1} \sim p_\theta^{t-1}(x_{t-1}|x_t)$ for $t = T, \ldots, 2$, and finally be decoded to a data sample $x_0 = G(x_1)$.

Reverse diffusion models Under what conditions can we recover the true data distribution? By rewriting $\sum_t L_t$, we can see from

$$\mathbb{KL}(p(x_0)\|p_\theta(x_0)) \leq \sum_{t=1}^{T} \mathbb{KL}(q_\theta(x_{t-1}|x_t)\|p_\theta^{t-1}(x_{t-1}|x_t)) \tag{9}$$

that this is possible as long as all reverse models are expressive enough to represent the true reverse processes defined by q_θ. For the first level, we can ensure this by making x_1 large enough such that the reconstruction error becomes negligible. For the diffusion process, previous image models [54, 19, 57, 21] relied on the fact that, in the limit $\beta_t \to 0$, the form of the true reverse process has the same functional form as the forward diffusion process [54, 32]. In particular, this allows modeling of the reverse process with a distribution factorized over the components. However, to make q_θ^{T-1} close to a uniform distribution requires a very large T (in the order of 1000 steps) with small β_t. Training such a large number of reverse models is only feasible with shared weights for the models, but this requires a delicate reweighting [19] of the objective and currently no suitable reweighting is known for the discrete case considered here.

Thus, to be able to recover the true data distribution with a modest number of reverse models that can be trained fully parallel, and without weight-sharing, we model each reverse process autoregressively. We use an encoder-decoder transformer architecture [63], such that the decoder models the reverse process for x_{t-1} autoregressively with the help of global context obtained by cross-attending to the encoder's representation of x_t. Note that the need for autoregressive modeling gets reduced for small β_t, which we can adjust for by reducing the number of decoder layers compared to encoder layers.

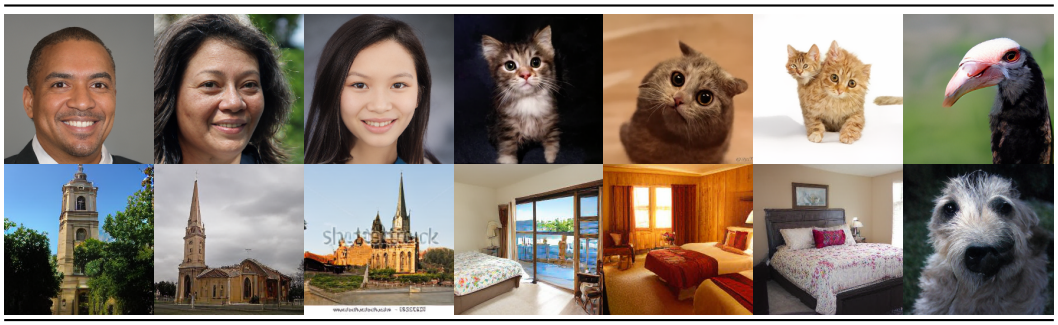

Figure 2: Samples from our models. Top row: FFHQ, LSUN-Cats, ImageNet, LSUN-Churches, LSUN-Bedrooms, ImageNet.

4 Experiments

Sec. 4.1 evaluates the quality ImageBART achieves in image synthesis. Since we are especially interested in better controllable of the generative process, we evaluate the performance of ImageBART on class- and text-conditional image generation in Sec. 4.2. The ability of our approach to attend to global context enables a new level of localized control which is not possible with previous, purely autoregressive approaches as demonstrated in Sec. 4.3. Finally, Sec. 4.4 presents ablations on model and architecture choices.

Method	Cats	Beds	Churches	FFHQ
VDVAE [6]	–	–	–	28.5
DDPM [19]	19.75	4.90	7.89	–
StyleGAN2 [25]	7.25	2.35	3.86	3.8
BigGAN [2]	–	–	–	12.4
DCT [40]	–	6.40	–	–
TT [12]	17.31	6.35	–	11.4
ImageBART	15.09	5.51	7.32	9.57

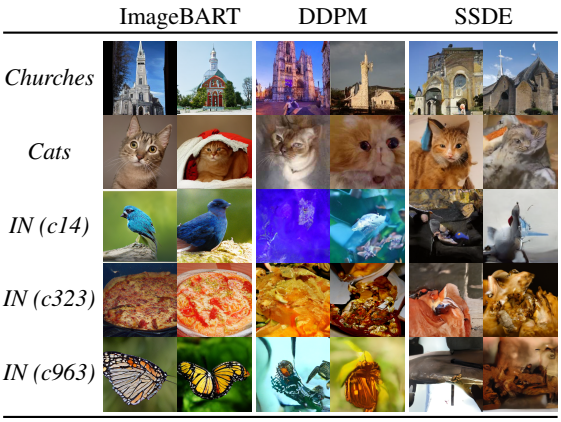

Table 1: Left: FIDs on the LSUN-{Churches,Beds,Cats} [67] and on the FFHQ [24] datasets. All images have resolution 256×256. Right: Corresponding qualitative comparisons.

4.1 High-Fidelity Image Synthesis with ImageBART

In this section we present both qualitative and quantitative results on images synthesized by our approach. We train models for unconditional generation on FFHQ [24], LSUN -Cats, -Churches and -Bedrooms [67] and on class-conditional synthesis on ImageNet [8].

Effective Discrete Representations Learning the full hierarchy as described in Eq. (2) and without unnecessary redundancies in the data requires to first learn a strong compression model via the objective in Eq. (4). [12] demonstrated how to effectively train such a model and we directly utilize the publicly available pretrained models. For training on LSUN, we finetune an ImageNet pretrained model for one epoch on each dataset. As the majority of codebook entries remains unused, we shrink the codebook to those entries which are actually used (evaluated on the validation split of ImageNet) and assign a random entry for eventual outliers. This procedure yields an effective, compact representation on which we subsequently train ImageBART.

Training Details As described in Sec. 3.3, we use an encoder-decoder structure to model the reverse Markov Chain $p_\theta^{t-1}(x_{t-1}|x_t),\ t < T$, where the encoder is a bidirectional transformer model and decoder is implemented as an AR transformer. As the context for the last scale is pure noise, we employ a decoder-only variant to model $p_\theta^{T-1}(x_{T-1}|x_T)$. Furthermore, to account for the different complexity of the datasets, we adjust the number of multinomial diffusion steps for each dataset accordingly. For FFHQ we choose a chain of length $T = 3$, such that the total model consists of (i) the compression stage and (ii) $n = 2$ transformer models trained in parallel via the objective described in Eq.(7). Similarly, we set $n = 3$ for each of the LSUN models and $n = 5$ for the ImageNet model.

Results For each of the aforementioned settings, Fig. 2 depicts samples of size 256×256 generated with ImageBART and with a single pass through the learned Markov Chain, demonstrating that our model is able to produce realistic and coherent samples. This is further confirmed by a quantitative analysis in Tab. 1, where we compare FID scores of competing likelihood-based and score-based methods such as TT [12] and DDPM [19]. Regarding other works on diffusion models such as [19] and [57] operating directly in pixel space, we observe that these approaches perform roughly equivalently well in terms of FID for datasets of low complexity (e.g. LSUN-Bedrooms and-Churches). For more complex datasets (LSUN-Cats, ImageNet), however, our method outperforms these pixel-based approaches, which can also be seen qualitatively on the right in Tab. 1. Additional hyperparameters of these experiments can be found in the supplementary.

4.2 Conditional Markov Chains for Controlled Image Synthesis

Being a sequence-to-sequence model, our approach allows for flexible and arbitrary conditioning by simply prepending tokens, similar to [12]. More specifically, each learned transition $p_\theta^{t-1}(x_{t-1}|x_t, c),\ t > 1$ of the Markov chain is then additionally conditioned on a representation c, e.g. a single token in the case of the class-conditional model of Sec. 4.1. Note that the compression model p_θ^0 remains unchanged.

Text-to-Image Synthesis Besides class-conditional modeling on ImageNet, we also learn a text-conditional model on *Conceptual Captions* [53, 41]. We obtain c by using the publicly available

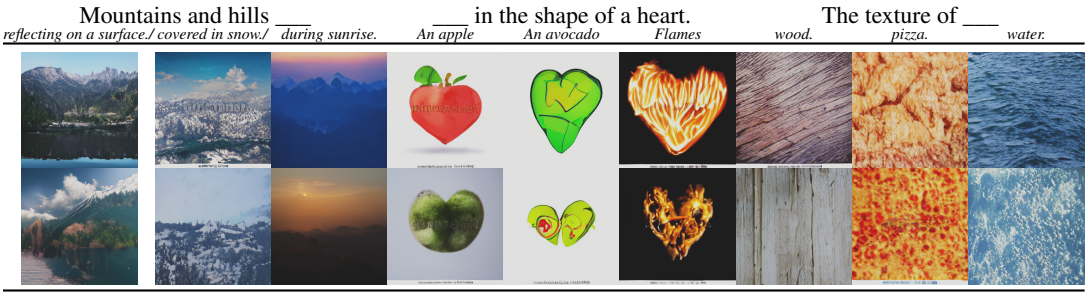

Figure 3: Samples from text-conditional ImageBART. Best 2 of 32 with reranking as in [46].

tokenizer of the CLIP model [45], yielding a conditioning sequence of length 77. To model the dataset, we choose $T = 5$ and thus train $n = 4$ transformer models independently. For the p_θ^0, we directly transfer the compression model from Sec. 4.1, trained on the ImageNet dataset.

Fig. 3 visualizes synthetic samples obtained with this model for various "image-cloze" tasks. Our resulting model is able to attend to semantic variations in the conditioning sentence (e.g. a change of weather for imagery of mountains) and renders the corresponding images accordingly. Please see the supplementary material for additional samples.

Resolutions Beyond 256×256 Pixels. Our approach is not restricted to generating images of size 256×256 pixels. Although trained on a fixed resolution, we can apply our models in a patch-wise manner, where we use the sliding attention window of [12] for each scale $t > 0$. As we now incorporate more and more global context while decoding with the Markov chain (which can be thought of as widening a noisy receptive field), ImageBART is able to render consistent images in the megapixel regime. See for example Fig. 4, where we use our text-conditional model to render an image of size 300×1800 pixel and interpolate between two different text prompts. More examples, especially also for semantically guided synthesis, can be found in the supplement.

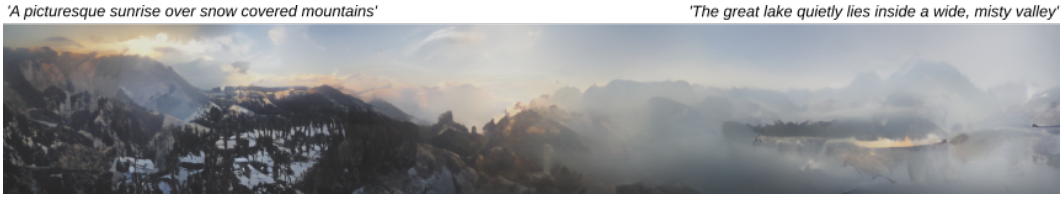

Figure 4: ImageBART is capable of generating high-resolution images. Here, we condition it on text prompts and interpolate between the two descriptions depicted above the image (see also Sec. 4.2).

4.3 Beyond Conditional Models: Local Editing with Autoregressive Models

Recent autoregressive approaches, which use a CNN to learn a discrete representation [62], partially alleviate the issues of pixel-wise autoregressive models by working on larger image patches. However, as we show in Fig. 4.3, even approaches which use adversarial learning to maximize the amount of context encoded in the discrete representation [12] cannot produce completions of the upper half of an image which are consistent with a given lower half.

While our approach also models each transition autoregressively from the top-left to the bottom-right, the ability to attend to global context from the previous scale enables consistent completions of also the upper half as shown in Fig. 4.3. To achieve this, we mask the diffusion process as described in the supplementary. For a user-specified mask (e.g. the upper half of the image in Fig. 4.3) m, this results in a forward-backward process $p_\theta^{t-1|t-1,m}$, which by definition always leaves the unmasked context intact. The reverse process then denoises the unmasked entries to make them consistent with the given context. The supplementary discusses the mixing time until $p_\theta^{t-1|t-1,m}$ samples from its steady state distribution of globally consistent samples.

Fig. 4.3 (bottom) visualizes this mixing process, where we use a model with $T = 3$. The first column shows the masked input. To start the process we set all masked entries to random entries. The first two columns then show (decoded) samples from the masked reverse processes $p_\theta^{2,m}$ and $p_\theta^{1,m}$, which

still display inconsistencies. The remaining columns show the trajectory of the process $p_\theta^{1|1,m}$, which demonstrates how the model iteratively adjusts its samples according to the given context until it converges to a globally consistent sample. For illustration, we show the analog trajectory obtained with [12], but because it can only attend to unidirectional context, this trajectory is equivalent to a sequence of independent samples and therefore fails to achieve global consistency.

The masked process can be used with arbitrary masks, which enables localized image editing with free, hand-drawn masks as shown in Fig. 4.3. Note that our model does not need to be trained specifically for this task, which also avoids generalization problems associated with training on masks [69]. Finally, especially in combination with the conditional models from Sec. 4.2, this allows for interesting novel applications, where local image regions are modified based on user specified class or text prompts, as shown in Fig. 4.3.

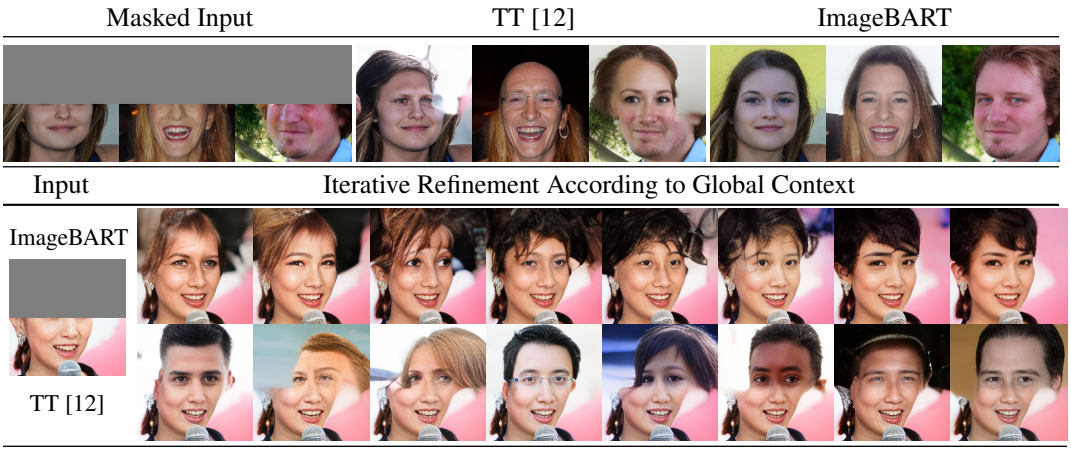

Figure 5: Without global context, AR models fail at completing upper halfs, contrasting ImageBART.

Figure 6: Local editing application using markov chain of length 16 on FFHQ. By incorporating bidirectional context ImageBART is able to solve this unconditional inpainting task (cf. Sec. 4.3).

4.4 Ablations

Joint vs. Independent Training While it is possible to optimize Eq. (2) jointly across all scales, we found that training is more robust when training all scales independently. Besides the usual separation of training the compression model p_θ^0 and the generative model $p_\theta^{t\geq 1}$, training the latter in parallel over multiple scales avoids the tedious weighting of the loss contribution from different scales; an often encountered problem in other denoising diffusion probabilistic models [19]. **Efficiency with Less Decoder Layers** As we implement the conditional transition probabilities p_θ^{t-1} with an encoder-decoder transformer architecture, we are interested in the effect of altering the ratio of encoder and decoder layers in the model. Recent work has provided evidence that it is possible to significantly reduce the number of decoder layers and thus also decrease autoregressive decoding speed while maintaining high quality [26]. We perform an experiment on LSUN-Churches, where we analyze the effect of different layer-ratios on synthesis quality (measured by FID) and on decoding speed when fixing the total number of model parameters to 200M. The results in the right part of Fig. 8 confirm that it is indeed possible to reduce the number of decoder layers while maintaining

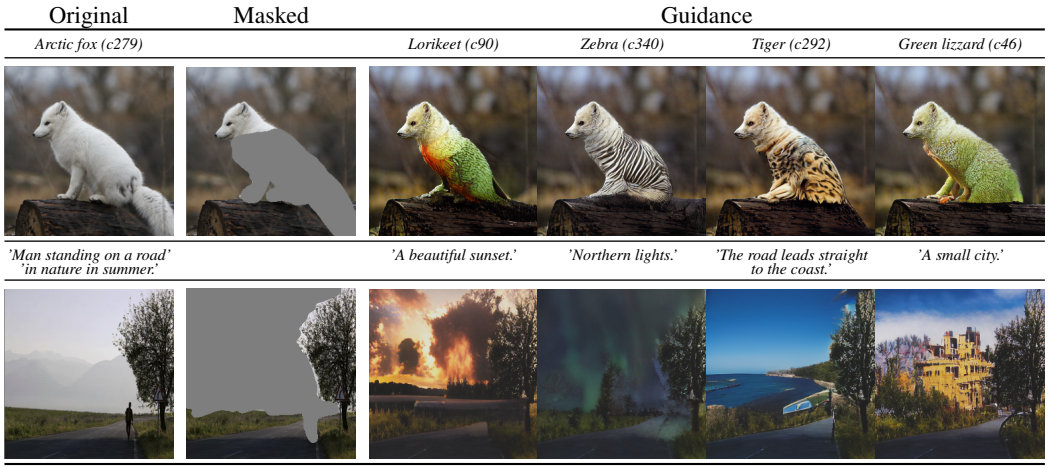

Figure 7: Conditionally guided inpainting results obtained from conditional ImageBART trained on the i) ImageNet (top row) and ii) Conceptual Captions (bottom row) datasets.

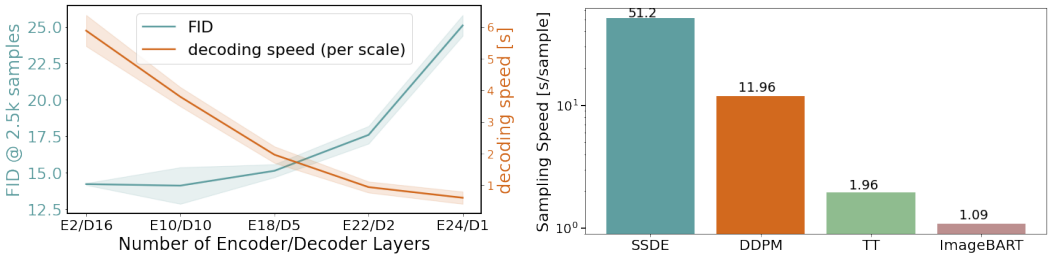

Figure 8: *Left:* Effect of number of encoder vs. decoder layers for a fixed total number of model parameters ($(195 \pm 5)M$), evaluated on LSUN-Churches. FIDs are evaluated w.r.t 3×2500k samples. The plot shows 3 standard deviations. All models are trained jointly over three scales. *Right:* Our model improves sampling speed upon recent state-of-the-art diffusion (SSDE [57],DDPM [19]) and AR models (TT [12]).

satisfactory FID scores with higher decoding efficiency. We identity a favorable trade-off between four and six decoder layers and transfer this setting to our other experiments. Finally, we compare our model in terms of sampling speed with the recent state-of-the-art generative diffusion [19, 57] and AR models [12]. The results are summarized in Fig, 8, revealing that our model achieves sampling times which are an *order of magnitude lower* than those of the competing diffusion models and two times as fast as the AR baseline [12].

5 Conclusion

We have proposed ImageBART, a hierarchical approach to introduce bidirectional context into autoregressive transformer models for high-fidelity controllable image synthesis. We invert a multinomial diffusion process by training a Markov chain to gradually incorporate context in a coarse-to-fine manner. Our study shows that this approach (i) introduces a natural hierarchical representation of images, with consecutive levels carrying more information than previous ones. (see also Fig. 9). (ii) It alleviates the unnatural unidirectional ordering of pure autoregressive models for image representation through global context from previous levels of the hierarchy. (iii) It enables global and local manipulation of a given input, a feat previously out-of-reach for ARMs. (iv) We additionally show that our model can be efficiently conditioned on various representations, allowing for a large class of conditional image synthesis tasks such as semantically guided generation or text-to-image synthesis.

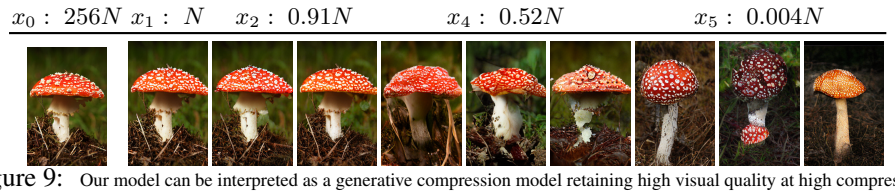

Figure 9: Our model can be interpreted as a generative compression model retaining high visual quality at high compression.

References

[1] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099, 2015.

[2] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.*, 2019.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[4] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 2020.

[5] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *CoRR*, abs/1611.02731, 2016.

[6] R. Child. Very deep vaes generalize autoregressive models and can outperform them on images. *CoRR*, abs/2011.10650, 2020.

[7] B. Dai and D. P. Wipf. Diagnosing and enhancing VAE models. In *ICLR (Poster)*. OpenReview.net, 2019.

[8] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[10] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A generative model for music. *CoRR*, abs/2005.00341, 2020.

[11] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, pages 658–666, 2016.

[12] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841, 2020.

[13] J. D. Fauw, S. Dieleman, and K. Simonyan. Hierarchical autoregressive image models with auxiliary decoders. *CoRR*, abs/1903.04933, 2019.

[14] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423. IEEE Computer Society, 2016.

[15] M. Germain, K. Gregor, I. Murray, and H. Larochelle. MADE: masked autoencoder for distribution estimation. *CoRR*, abs/1502.03509, 2015.

[16] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial networks. *CoRR*, 2014.

[17] A. Goyal, A. Lamb, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio. Professor forcing: A new algorithm for training recurrent networks. In *NIPS*, pages 4601–4609, 2016.

[18] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taïga, F. Visin, D. Vázquez, and A. C. Courville. Pixelvae: A latent variable model for natural images. *CoRR*, abs/1611.05013, 2016.

[19] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[20] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf`.

[21] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *CoRR*, abs/2102.05379, 2021.

[22] A. H. Jha, S. Anand, M. Singh, and V. S. R. Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *ECCV (3)*, volume 11207 of *Lecture Notes in Computer Science*, pages 829–845. Springer, 2018.

[23] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV (2)*, volume 9906 of *Lecture Notes in Computer Science*, pages 694–711. Springer, 2016.

[24] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. *CoRR*, abs/1912.04958, 2019.

[26] J. Kasai, N. Pappas, H. Peng, J. Cross, and N. A. Smith. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation, 2021.

[27] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *CoRR*, abs/2101.01169, 2021.

[28] Y. Kilcher, A. Lucchi, and T. Hofmann. Semantic interpolation in implicit models. *CoRR*, abs/1710.11381, 2017.

[29] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

[30] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014.

[31] A. Kolesnikov and C. H. Lampert. Pixelcnn models with auxiliary variables for natural image modeling. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1905–1914. PMLR, 2017.

[32] A. Kolmogoroff. Über die analytischen methoden in der wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104:415–458, 1931. URL `http://eudml.org/doc/159476`.

[33] R. Leblond, J. Alayrac, A. Osokin, and S. Lacoste-Julien. SEARNN: training rnns with global-local losses. *CoRR*, abs/1706.04499, 2017.

[34] D. Lesniak, I. Sieradzki, and I. T. Podolak. Distribution-interpolation trade off in generative models. In *ICLR (Poster)*. OpenReview.net, 2019.

[35] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. Association for Computational Linguistics, 2020.

[36] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. In *AAAI*, pages 6706–6713. AAAI Press, 2019.

[37] L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *NeurIPS*, pages 6548–6558, 2019.

[38] M. Mathieu, J. J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun. Disentangling factors of variation in deep representations using adversarial training. *CoRR*, abs/1611.03383, 2016.

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

[40] C. Nash, J. Menick, S. Dieleman, and P. W. Battaglia. Generating images with sparse representations. *CoRR*, abs/2103.03841, 2021.

[41] E. G. Ng, B. Pang, P. Sharma, and R. Soricut. Understanding guided image captioning performance across domains. *arXiv preprint arXiv:2012.02339*, 2020.

[42] A. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021.

[43] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[44] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 4052–4061. PMLR, 2018.

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[46] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021.

[47] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. In *ICLR (Poster)*, 2016.

[48] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, pages 14837–14847, 2019.

[49] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning, ICML*, 2014.

[50] R. Rombach, P. Esser, and B. Ommer. Network-to-network translation with conditional invertible neural networks. In *NeurIPS*, 2020.

[51] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *CoRR*, abs/1701.05517, 2017.

[52] F. Schmidt. Generalization in generation: A closer look at exposure bias. In *NGT@EMNLP-IJCNLP*, pages 157–167. Association for Computational Linguistics, 2019.

[53] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[54] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.

[55] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *NIPS*, pages 3738–3746, 2016.

[56] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, pages 11895–11907, 2019.

[57] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020.

[58] A. Szabó, Q. Hu, T. Portenier, M. Zwicker, and P. Favaro. Challenges in disentangling independent factors of variation. In *ICLR (Workshop)*. OpenReview.net, 2018.

[59] B. Uria, M. Côté, K. Gregor, I. Murray, and H. Larochelle. Neural autoregressive distribution estimation. *CoRR*, abs/1605.02226, 2016.

[60] A. Vahdat and J. Kautz. NVAE: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020.

[61] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. *CoRR*, abs/1606.05328, 2016.

[62] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *NIPS*, pages 6306–6315, 2017.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[64] C. Wang, Y. Tang, X. Ma, A. Wu, D. Okhonko, and J. Pino. fairseq s2t: Fast speech-to-text modeling with fairseq, 2020.

[65] T. White. Sampling generative networks: Notes on a few effective techniques. *CoRR*, abs/1609.04468, 2016.

[66] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. Videogpt: Video generation using VQ-VAE and transformers. *CoRR*, abs/2104.10157, 2021.

[67] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.

[68] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[69] C. Zheng, T. Cham, and J. Cai. Tfill: Image completion via a transformer-based architecture. *CoRR*, abs/2104.00845, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See supplementary.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Including the code while maintaining anonymity is difficult; it will be published after de-anonymization.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We include most relevant details in the supplementary and the full code release will contain the precise values.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Yes, wherever this does not introduce irresponsible waste of computational resources.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The supplementary contains an overview of used hardware.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] Licenses will be included along with the code release.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See supplementary.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]