Semi-Parametric Neural Image Synthesis

Anonymous Author(s)
Affiliation

Abstract

Novel architectures have recently improved generative image synthesis leading to excellent visual quality in various tasks. Much of this success is due to the scalability of these architectures and hence caused by a dramatic increase in model complexity and in the computational resources invested in training these models. Our work questions the underlying paradigm of compressing large training data into ever growing parametric representations. We rather present an orthogonal, semi-parametric approach. We complement a comparably small generative image model, e.g., a diffusion or autoregressive model, with a separate image database and a retrieval-based approach. During training we retrieve a set of nearest neighbors from this external database for each training instance and condition the generative model on these informative samples. While the retrieval approach is providing the (local) content, the model is focusing on learning the composition of scenes based on this content. As demonstrated by our experiments, simply swapping the database for one with different contents transfers a trained model post-hoc to a novel domain. The evaluation shows competitive performance on tasks which the generative model has not been trained on, such as class-conditional or text-to-image synthesis and zero-shot stylization. With negligible memory and computational overhead for external database and retrieval we can significantly reduce the parameter count of the generative model and still outperform the state-of-the-art.

1 Introduction

Deep generative models has made tremendous leaps; especially in language modeling as well as in generative synthesis of high-fidelity images and other data types. In particular for images, astounding results have recently been achieved [18, 12, 47, 50], and three main factors can be identified as the driving forces behind this progress: First, the success of the transformer [74] has caused an architectural revolution in many vision tasks [16], for image synthesis especially through its combination with autoregressive modeling [18, 49]. Second, since their rediscovery, diffusion models have been applied to high-resolution image generation [63, 65, 27] and, within a very short time, set new standards in generative image modeling [12, 28, 54, 50]. Third, these approaches *scale* well [49, 50, 30, 68]; in particular when considering the model- and batch sizes involved for high-quality models [12, 47, 49, 50] there is evidence that this scalability is of central importance for their performance.

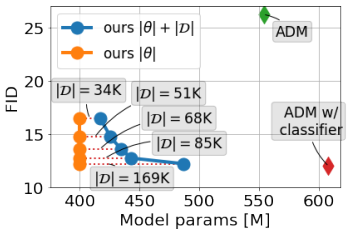


Figure 1: Our semi-parametric diffusion model outperforms the unconditional SOTA model ADM [12] on ImageNet [11] and even reaches the class-conditional ADM (ADM w/ classifier), while reducing parameter count. $|\mathcal{D}|$: Number of instances in database at inference; $|\theta|$: Number of trainable parameters

However, the driving force underlying this training paradigm are models with ever growing numbers of parameters [68] that require huge computational resources. Besides the enormous demands in energy consumption and training time, this paradigm renders future generative modeling more and more exclusive to privileged institutions, thus hindering the democratization of research. Therefore, we here present an orthogonal approach. Inspired by recent advances in retrieval-augmented NLP [3, 75], we question the prevalent approach of expensively compressing visual concepts shared between distinct training examples into large numbers of trainable

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

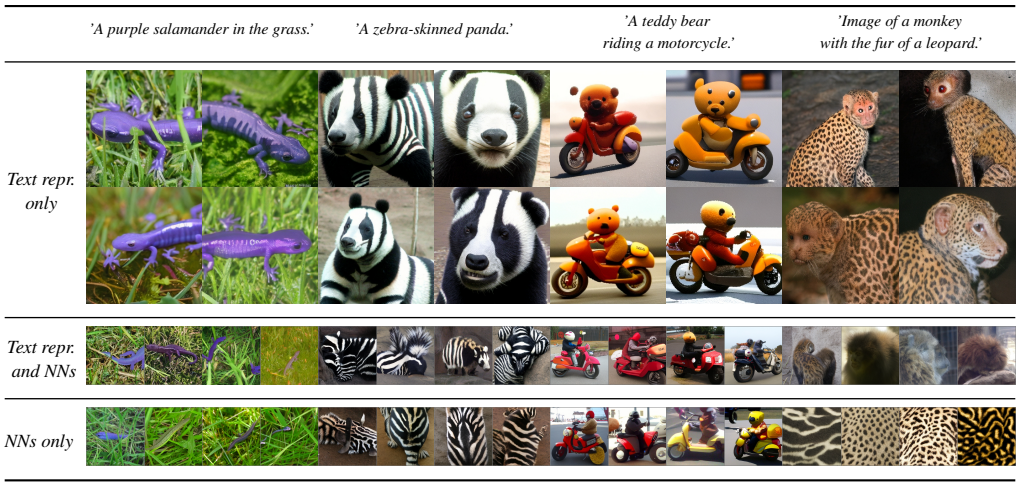


Figure 2: As we retrieve nearest neighbors in the shared text-image space provided by CLIP, we can use text prompts as queries for exemplar-based synthesis. We observe our *RDM* to readily generalize to unseen and fictional text prompts when building the set of retrieved neighbors by directly conditioning on the CLIP text encoding $\phi_{\text{CLIP}}(c_{\text{text}})$ (top row). When using $\phi_{\text{CLIP}}(c_{\text{text}})$ together with its $k-1$ nearest neighbors from the retrieval database (middle row) or the k nearest neighbors alone without the text representation, the model does not show these generalization capabilities (bottom row).

parameters and equip a comparably small generative model with a large image database. During training, our resulting *semi-parametric* generative models access this database via a nearest neighbor lookup and, thus, need not learn to generate data 'from scratch'. Instead, they learn to *compose* new scenes based on retrieved visual instances. This property not only increases generative performance with reduced parameter count (see Fig. 1), and lowers compute requirements during training. Our proposed approach also enables the models during inference to generalize to new knowledge in form of alternative image databases without requiring further training, what can be interpreted as a form of post-hoc model modification [3]. We show this by replacing the retrieval database with the WikiArt [57] dataset after training, thus applying the model to zero-shot stylization.

Furthermore, our approach is formulated indepently of the underlying generative model, allowing us to present both retrieval-augmented diffusion (*RDM*) and autoregressive (*RARM*) models. By searching in and conditioning on the latent space of CLIP [48] and using scaNN [23] for NN-search, we the nearest neighbor causes negligible overheads in training/inference time (0.95 ms to retrieve 20 nearest neighbors from a database of 20M examples) and storage space (2GB per 1M examples). We show that semi-parametric models yield high fidelity and diverse samples: *RDM* surpasses recent state-of-the-art diffusion models in terms of FID and diversity while requiring less trainable parameters. Furthermore, the shared image-text feature space of CLIP allows for various conditional applications such as text-to-image or class-conditional synthesis, despite being trained on images only (as demonstrated in Fig. 2). Finally, we present additional truncation strategies to control the synthesis process which can be combined with model specific sampling techniques such as classifier-free guidance for diffusion models [26] or top-k sampling [19] for autoregressive models.

2 Related Work

Generative Models for Image Synthesis. Generating high quality novel images has long been a challenge for deep learning community due to their high dimensional nature. Generative adversarial networks (GANs) [20] excel at synthesizing such high resolution images with outstanding quality [4, 32, 33, 60] while optimizing their training objective requires some sort of tricks [1, 22, 45, 44] and their samples suffer from the lack of diversity [67, 1, 46, 41]. On the contrary, likelihood-based methods have better training properties and they are easier to optimize thanks to their ability to capture the full data distribution. While failing to achieve the image fidelity of GANs, variational autoencoders (VAEs) [36, 52] and flow-based methods [13, 14] facilitate high resolution image generation with fast sampling speed [70, 38]. Autoregressive models (ARMs) [9, 71, 73, 59] succeed in density estimation like the other likelihood-based methods, albeit at the expense of computational efficiency. Starting with the seminal works of Sohl-Dickstein et al. [63] and Ho et al. [27], diffusion-based generative models have improved generative modeling of artificial visual systems [12, 37, 76, 29, 78, 56]. Their good performance, however, comes at the expense of high training costs and slow sampling. To circumvent the drawbacks of ARMs and diffusion models,

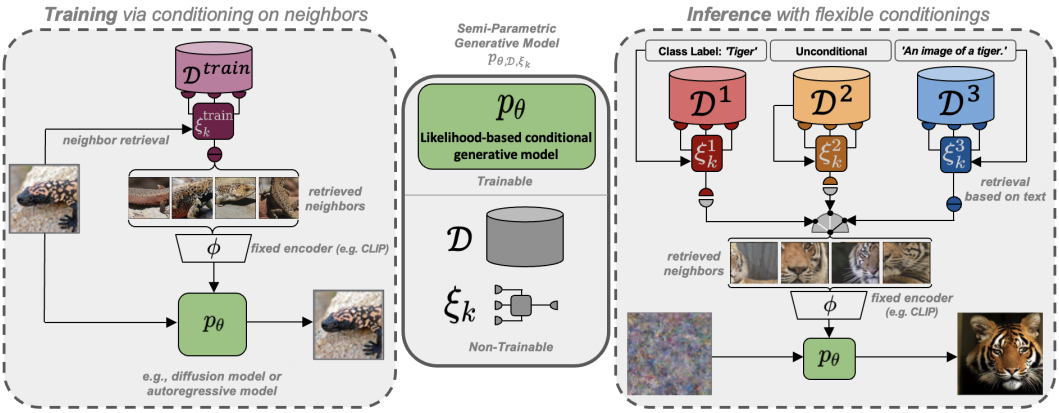


Figure 3: A semi-parametric generative model consists of a trainable conditional generative model (decoding head) $p_\theta(x|\cdot)$, an external database $\mathcal{D}$ containing visual examples and a sampling strategy ξ_k to obtain a subset $\mathcal{M}_\mathcal{D}^{(k)} \subseteq \mathcal{D}$, which serves as conditioning for p_θ. During training, ξ_k retrieves the nearest neighbors of each target example from $\mathcal{D}$, such that p_θ only needs to learn to compose consistent scenes based on $\mathcal{M}_\mathcal{D}^{(k)}$, *cf.* Sec 3.2. During inference, we can exchange $\mathcal{D}$ and ξ_k, thus resulting in flexible sampling capabilities such as post-hoc conditioning on class labels (ξ_k^1) or text prompts (ξ_k^3), *cf.* Sec. 3.3, and zero-shot stylization, *cf.* Sec. 4.3.

several two-stage models are proposed to scale them to higher resolutions by training them on the compressed image features [72, 51, 18, 79, 54, 62, 17]. However, they still require large models and significant compute resources, especially for unconditional image generation [12] on complex datasets like ImageNet [11] or complex conditional tasks such as text-to-image generation [47, 49, 21, 54]. To address these issues, given limited compute resources, we propose to trade trainable parameters for an external memory which empowers smaller models to achieve high fidelity image generation.

Retrieval-Augmented Generative Models. Using external memory to augment traditional models has recently drawn attention in natural language processing (NLP) [34, 35, 43, 24]. For example, RETRO [3] proposes a retrieval-enhanced transformer for language modeling which performs on par with state-of-the-art models [5] using significantly less parameters and compute resources. These retrieval-augmented models with external memory turn purely parametric deep learning models into semi-parametric ones. Early attempts [42, 61, 69, 77] in retrieval-augmented visual models do not use an external memory and exploit the training data itself for retrieval. In image synthesis, IC-GAN [7] utilizes the neighborhood of training images to train a GAN and generates samples by conditioning on single instances from the training data. However, using training data itself for retrieval potentially limits the generalization capacity, and thus, we favor an external memory in this work.

3 Image Synthesis with Retrieval-Augmented Generative Models

Our work considers data points as an explicit *part of the model*. In contrast to common neural generative approaches for image synthesis [4, 33, 60, 51, 18, 9, 8], this approach is not only parameterized by the learnable weights of a neural network, but also a (fixed) set of data representations and a non-learnable *retrieval* function, which, given a query from the training data, retrieves suitable data representations from the external dataset. Following prior work in natural language modeling [3], we implement this retrieval pipeline as a nearest neighbor lookup.

Sec. 3.1 and Sec. 3.2 formalize this approach for training retrieval-augmented diffusion and autoregressive models for image synthesis, while Sec. 3.3 introduces sampling mechanisms that become available once such a model is trained. Fig. 3 provides an overview over our approach.

3.1 Retrieval-Enhanced Generative Models of Images

Unlike common, fully parametric neural generative approaches for images, we define a *semi-parametric* generative image model $p_{\theta,\mathcal{D},\xi_k}(x)$ by introducing trainable parameters θ *and* non-trainable model components $\mathcal{D}, \xi_k$, where $\mathcal{D} = \{y_i\}_{i=1}^N$ is a *fixed* database of images $y_i \in \mathbb{R}^{H_\mathcal{D} \times W_\mathcal{D} \times 3}$ that is disjoint from our train data $\mathcal{X}$. Further, ξ_k denotes a (non-trainable) sampling strategy to obtain a subset of $\mathcal{D}$ based on a query x, i.e. $\xi_k \colon x, \mathcal{D} \mapsto \mathcal{M}_\mathcal{D}^{(k)}$, where $\mathcal{M}_\mathcal{D}^{(k)} \subseteq \mathcal{D}$ and $|\mathcal{M}_\mathcal{D}^{(k)}| = k$. Thus, only θ is actually learned during training.

Importantly, $\xi_k(x, \mathcal{D})$ has to be chosen such that it provides the model with beneficial visual representations from $\mathcal{D}$ for modeling x and the entire capacity of θ can be leveraged to *compose* consistent

scenes based on these patterns. For instance, considering query images $x \in \mathbb{R}^{H_x \times W_x \times 3}$, a valid strategy $\xi_k(x, \mathcal{D})$ is a function that for each x returns the set of its k nearest neighbors, measured by a given distance function $d(x, \cdot)$.

Next, we propose to provide this retrieved information to the model via *conditioning*, i.e. we specify a general semi-parametric generative model as

$$p_{\theta,\mathcal{D},\xi_k}(x) = p_\theta(x \mid \xi_k(x, \mathcal{D})) = p_\theta(x \mid \mathcal{M}_{\mathcal{D}}^{(k)}) \tag{1}$$

In principle, one could directly use image samples $y \in \mathcal{M}_{\mathcal{D}}^{(k)}$ to learn θ. However, since images contain many ambiguities and their high dimensionality involves considerable computational and storage cost[1] we use a *fixed*, pre-trained image encoder ϕ to project all examples from $\mathcal{M}_{\mathcal{D}}^{(k)}$ onto a low-dimensional manifold. Hence, Eq. (1) reads

$$p_{\theta,\mathcal{D},\xi_k}(x) = p_\theta(x|\{\phi(y)|y \in \xi_k(x, \mathcal{D})\}). \tag{2}$$

where $p_\theta(x|\cdot)$ is a conditional generative model with trainable parameters θ which we refer to as *decoding head*. With this, the above procedure can be applied to any type of generative decoding head and is not dependent on its concrete training procedure.

3.2 Instances of Semi-Parametric Generative Image Models

During training we are given a train dataset $\mathcal{X} = \{x_i\}_{i=1}^M$ of images whose distribution $p(x)$ we want to approximate with $p_{\theta,\mathcal{D},\xi_k}(x)$. Our train-time sampling strategy ξ_k uses a query example $x \sim p(x)$ to retrieve its k nearest neighbors $y \in \mathcal{D}$ by implementing $d(x, y)$ as the cosine similarity in the image feature space of CLIP [48]. Given a sufficiently large database $\mathcal{D}$, this strategy ensures that the set of neighbors $\xi_k(x, \mathcal{D})$ shares sufficient information with x and, thus, provides useful visual information for the generative task. We choose CLIP to implement ξ_k, because it embeds images in a low dimensional space ($\dim = 512$) and maps semantically similar samples to the same neighborhood, yielding an efficient search space. Fig. 4 visualizes examples of nearest neighbors retrieved via a ViT-B/32 vision transformer [16] backbone.

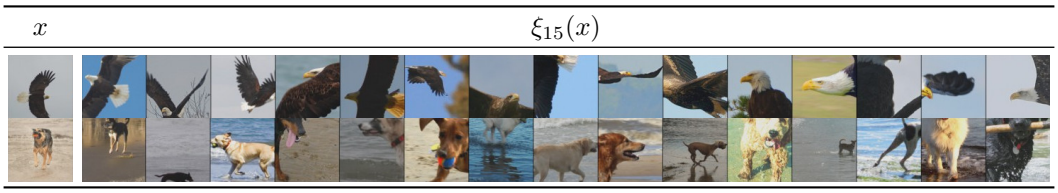


Figure 4: $k = 15$ nearest neighbors from $\mathcal{D}$ for a given query x when parameterizing $d(x, \cdot)$ with CLIP [48].

Note that this approach can, in principle, turn any generative model into a semi-parametric model in the sense of Eq. (2). While this formulation supports various explicit likelihood approaches, in this work we focus on models where the decoding head is either implemented as a diffusion or an autoregressive model, motivated by the success of these models in image synthesis [27, 12, 54, 47].

To obtain the image representations via ϕ, different encoding models are conceivable in principle. Again, the latent space of CLIP offers some advantages since it is (i) very compact, which (ii) also reduces memory requirements. Moreover, the contrastive pretraining objective (iii) provides a shared space of image and text representations, which is beneficial for text-image synthesis, as we show in Sec. 4.2. Unless otherwise specified, $\phi \equiv \phi_{\text{CLIP}}$ is set in the following. We investigate alternative parameterizations of ϕ in the appendix.

Note that with this choice, the additional database $\mathcal{D}$ can also be interpreted as a fixed *embedding layer*[2] of dimensionality $|\mathcal{D}| \times 512$ from which the nearest neighbors are retrieved.

3.2.1 Retrieval-Augmented Diffusion Models

In order to reduce computational complexity and memory requirements during training, we follow [54] and build on latent diffusion models (LDMs) which learn the data distribution in the latent space $z = E(x)$ of a pretrained autoencoder. We dub this retrieval-augmented latent diffusion model *RDM* and train it with the usual reweighted likelihood objective [27], yielding the objective [63, 27]

$$\min_\theta \mathcal{L} = \mathbb{E}_{p(x), z \sim E(x), \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(z_t, t, \{\phi_{\text{CLIP}}(y)|y \in \xi_k(x, \mathcal{D})\})\|_2^2 \right], \tag{3}$$

[1] Note that $\mathcal{D}$ is essentially a part of the model weights

[2] For a database of 1M images and using 32-bit precision, this equals approximately 2.048 GB

where the expectation is approximated by the empirical mean over training examples. In the above equation, ϵ_θ denotes the UNet-based [55] denoising autoencoder as used in [12, 54] and $t \sim \text{Uniform}(1, T)$ denotes the time step [63, 27]. To feed the set of nearest neighbor encodings $\phi_{\text{CLIP}}(y)$ into ϵ_θ, we use the cross-attention conditioning mechanism proposed in [54].

3.2.2 Retrieval-Augmented Autoregressive Models

Our approach is applicable to several types of likelihood-based methods. We show this by augmenting diffusion models (Sec. 3.2.1) as well as autoregressive models with the retrieved representations. To implement the latter, we follow [18] and train autoregressive transformer models to model the distribution of the discrete image tokens $z_q = E(x)$ of a VQGAN [18, 72]. Specifically, as for ***RDM***, we train retrieval-augmented autoregressive models ***(RARMs)*** conditioned on the CLIP embeddings $\phi_{\text{CLIP}}(y)$ of the neighbors y, so that the objective reads

$$\min_\theta \mathcal{L} = -\mathbb{E}_{p(x), z_q \sim E(x)} \Big[\sum_i \log p(z_q^{(i)} | z_q^{(<i)}, \{\phi_{\text{CLIP}}(y) | y \in \xi_k(x, \mathcal{D})\}) \Big] , \tag{4}$$

where we choose a row-major ordering for the autoregressive factorization of the latent z_q. We condition the model on the set of neighbor embeddings $\phi_{\text{CLIP}}(\xi_k(x, \mathcal{D}))$ via cross-attention [74].

3.3 Inference for Retrieval-Augmented Generative Models

During inference we have no access to query instances from the train set. However, since $\mathcal{D}$ and ξ_k are not learned, we can change them at test time as long as they still provide $p_\theta(x|\cdot)$ with a suitable set of visual representations.

Conditional Synthesis without Conditional Training During synthesis, additional conditional information such as text prompts or class labels, which has not been available during training, can provide more fine-grained control. The exchangeability of $\mathcal{D}$ and ξ_k at test time offers additional flexibility compared to standard generative approaches: Depending on the application, it is possible to extent/restrict $\mathcal{D}$ for particular exemplars; or to skip the retrieval via ξ_k altogether and provide a set of representations $\{\phi_{\text{CLIP}}(y_i)\}_{i=1}^k$ directly.

For text-to-image generation, for example, our model can be conditioned in several ways: Given a text prompt c_{text} and using the text-to-image retrieval ability of CLIP, we can retrieve k neighbors from $\mathcal{D}$ and use these as an implicit text-based conditioning. However, since we condition on CLIP representations ϕ_{CLIP}, we can also condition directly on the *text* embeddings obtained via CLIP's language backbone (since CLIP's text-image embedding space is shared). Accordingly, it is also possible to combine these approaches and use text and image representations simultaneously. We show and compare the results of using these sampling techniques in Fig. 2.

Given a class label c_c, we define a text such as *'An image of a* $t(c_c)$*.'* based on its textual description $t(c_c)$ or apply the embedding strategy for text prompts and sample a pool $\xi_l(c_c)$, $k \leq l$ for each class. By randomly selecting k adjacent examples from this pool for a given query c_c, we obtain an inference-time class-conditional model and analyze these post-hoc conditioning methods in Sec. 4.2.

For **unconditional generative modeling**, we randomly sample a pseudo-query $\tilde{x} \in \mathcal{D}$ to obtain the set $\xi_k^{\text{test}}(\tilde{x}, \mathcal{D})$ of its k nearest neighbors. Given this set, Eq. (2) can be used to draw samples, since $p_\theta(x|\cdot)$ itself is a generative model. However when generating all samples from $p_{\theta,\mathcal{D},\xi_k}(x)$ only from one particular set $\xi_k^{\text{test}}(\tilde{x})$, we expect $p_{\theta,\mathcal{D},\xi_k}(x)$ to be unimodal and sharply peaked around $\tilde{x}$. When intending to model a complex multimodal distribution $p(x)$ of natural images, this choice would obviously lead to weak results. Therefore, we construct a proposal distribution based on $\mathcal{D}$ where

$$p_{\mathcal{D}}(\tilde{x}) = \frac{|\{x \in \mathcal{X} \mid \tilde{x} \in \xi_k(x, \mathcal{D})\}|}{k \cdot |\mathcal{X}|} , \quad \text{for } \tilde{x} \in \mathcal{D} . \tag{5}$$

This definition counts the instances in the database $\mathcal{D}$ which are useful for modeling the training dataset $\mathcal{X}$. Note that, $p_{\mathcal{D}}(\tilde{x})$ only depends on $\mathcal{X}$ and $\mathcal{D}$, what allows us to precompute it. Given $p_{\mathcal{D}}(\tilde{x})$, we can obtain a set

$$\mathcal{P} = \Big\{ x \sim p_\theta(x | \{\phi(y) | y \in \xi_k(\tilde{x}, \mathcal{D})\}) \Big| \, \tilde{x} \sim p_{\mathcal{D}}(\tilde{x}) \Big\} \tag{6}$$

of samples from the our model. We can thus draw from the unconditional modeled density $p_{\theta,\mathcal{D},\xi_k}(x)$ by drawing $x \sim \text{Uniform}(\mathcal{P})$.

By choosing only a fraction $m \in (0, 1]$ of most likely examples $\tilde{x} \sim p_{\mathcal{D}}(\tilde{x})$, we can artificially truncate this distribution and trade sample quality for diversity. See the appendix for a detailed description of this mechanism which we call *top-m sampling* and Sec. 4.5 for an empirical demonstration.

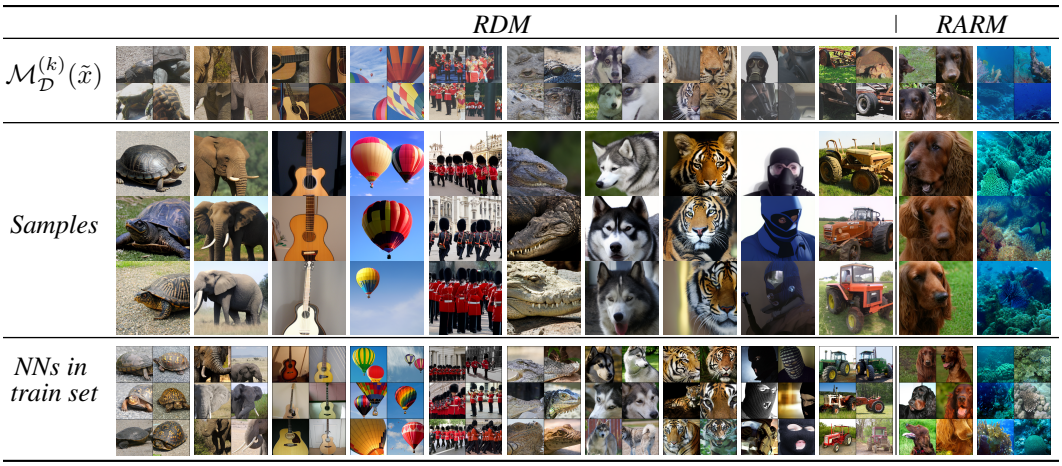


Figure 5: Samples from our unconditional *RDM* and *RARM* together with the sets of $\mathcal{M}_{\mathcal{D}}^{(k)}(\tilde{x})$ of retrieved neighbors based on the pseudo query $\tilde{x}$, *cf.* Sec. 3.3 and nearest neighbors from the train set, measured in CLIP [48] feature space. Samples are generated with $m = 0.01$, guidance with $s = 2.0$ and 100 DDIM steps for *RDM* and $m = 0.05$, guidance scale $s = 3.0$ and top-$k = 2048$ for *RARM* .

4 Experiments

This section presents experiments for both retrieval-augmented diffusion and autoregressive models. During training, we use a database $\mathcal{D}$ of 20M examples from the OpenImages [39] dataset. We extract 2-3 patches per image (see the supplement) to use each image at least once. To obtain nearest neighbors we apply the ScaNN search algorithm [23] in the feature space of a pretrained CLIP-ViT-B/32 [48]. Using this setting, retrieving 20 nearest neighbors from the database described above takes approximately 0.95 ms. For more details on our retrieval implementation, see the supplementary. For quantitative performance measures we use FID [25], Inception Score (IS) [58] and Precision-Recall [40], and, for the diffusion models, generate samples with the DDIM sampler [64] with 100 steps and $\eta = 1$. For hyperparameters, implementation and evaluation details *cf.* the appendix.

4.1 Semi-Parametric Image Generation

Drawing pseudo-queries from the proposal distribution proposed in Sec. 3.3 and Eq. (6) enables semi-parametric unconditional image generation. However, before the actual application, we determine an appropriate choice for the value k of the retrieved neighbors during training.

How many neighbors to retrieve during training? As the number k_{train} of retrieved nearest neighbors during training has a strong influence on the properties of the resulting model after training, we first identify hyperparameters obtain a model with optimal synthesis properties. Hence, we parameterize p_θ with a diffusion model and train five models for different $k_{\text{train}} \in \{1, 2, 4, 8, 16\}$ on ImageNet [11]. All models use identical generative backbones and computational resources (details in (details in the appendix) Fig. 6 shows resulting performance metrics assessed on 1000 samples. For FID and IS we do not observe significant trends. Considering precision and recall, however, we see that increasing k_{train} trades consistency for diversity. Large k_{train} causes recall, i.e. sample diversity, to deteriorate again.

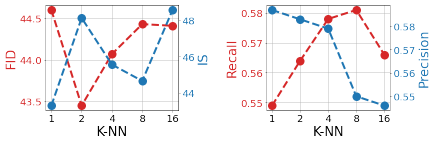


Figure 6: Effect of k_{train}.

We attribute this to a regularizing influence of non-redundant, additional information beyond the single nearest neighbor, which is fed to the respective model during training, when $k_{\text{train}} > 1$. For $k_{\text{train}} \in \{2, 4, 8\}$ this additional information is beneficial and the corresponding models appropriately mediate between quality and diversity. Thus, we use $k = 4$ for our main *RDM* . Furthermore, the numbers of neighbors has a significant effect on the generalization capabilities of our model for *conditional* synthesis, *e.g.* text-to-image synthesis as in Fig. 2. We provide an in-depth evaluation of this effect in Sec. 4.2 and conduct a similar study for *RARM* in the supplement.

Qualitative results. Fig. 5 shows samples of *RDM* /*RARM* trained on the ImageNet and a subset of ImageNet for different sets $\mathcal{M}_{\mathcal{D}}^{(k)}(\tilde{x})$ of retrieved neighbors given a pseudo-query $\tilde{x} \sim p_{\mathcal{D}}(\tilde{x})$. We

Method	FID↓		IS↑	Precision↑		Recall↑		Nparams	
	train	*val*		*train*	*val*	*train*	*val*		
IC-GAN [7]	18.17	15.60*	59.00*	0.77	**0.73**	0.21	0.23	191M	conditioned on train set, add. aug.
ADM [12]	26.21	32.50*	39.70	0.61	-	0.63	-	554M	250 steps
ADM-G [12]	33.03	-	32.92	0.56	-	**0.65**	-	618M	250 steps, c.g., s=1.0
ADM-G [12]	**12.00**	-	**95.41**	0.76	-	0.44	-	618M	250 steps, c.g., s=10.0
RDM -4 (ours)	24.50	21.28	45.29	0.60	0.54	**0.65**	**0.66**	400M	100 steps, $m = 0.1$
RDM -4 (ours)	19.08	16.89	62.78	0.57	0.62	0.56	0.57	400M	100 steps, $m = 0.01$
RDM -4 (ours)	13.22	12.29	70.64	0.72	0.65	0.56	0.51	400M	100 steps, c.f.g., $s = 1.75$, $m = 0.1$
RDM -4 (ours)	13.60	13.11	87.58	**0.79**	**0.73**	0.51	0.50	400M	100 steps, c.f.g., $s = 1.5$, $m = 0.02$
RDM -4 (ours)	12.21	**11.31**	77.93	0.75	0.69	0.55	0.55	400M	100 steps, c.f.g., $s = 1.5$, $m = 0.05$

Table 1: Comparison of *RDM* with recent state-of-the-art methods for unconditional image generation on ImageNet [11]. While *c.f.g.* denotes classifier-free guidance with a scale parameter s as proposed in [26], *c.g.* refers to classifier guidance [12], what requires a classifier pretrained on the noisy representations of diffusion models to be available. *: numbers taken from [7].

also plot the nearest neighbors from the train set to show that this set is actually disjoint from the database $\mathcal{D}$ and that our model renders new, unseen samples.

Quantitative results. Tab. 1 compares our model with the recent state-of-the-art diffusion model ADM [12] and the semi-parametric GAN-based model IC-GAN [7] (which requires access to the *training set* examples during inference) in unconditional image synthesis on ImageNet [11] 256×256.

To boost performance, we use the sampling strategies proposed in Sec. 3.3 (which is also further detailed in the supplementary). With classifier-free guidance (c.f.g.), our model attains better scores than IC-GAN and ADM while being on par with ADM-G [12]. The latter requires an additional classifier and the labels of training instances during inference. Without any additional information about training data, e.g., image labels, *RDM* achieves the best overall performance.

For $m = 0.1$, our retrieval-augmented *RDM* surpasses unconditional ADM for FID, IS, precision and, with no guidance, also for recall. For $s = 1.75$, we observe bisected FID scores compared to our unguided model and even reach the guided model ADM-G, which, unlike *RDM* , requires a classifier that is pre-trained on noisy data representations. The optimal parameters for FID are $m = 0.05$, $s = 1.5$, as in the bottom row of Tab. 1. Overall, this shows the strong performance of *RDM* and the flexibility of top-m sampling and c.f.g., which we further analyze in Sec. 4.5.

4.2 Conditional Synthesis without Conditional Training

Text-to-Image Synthesis In Fig. 2, we show the zero-shot text-to-image synthesis capabilities of our ImageNet model for user defined text prompts. When building the set $\mathcal{M}_{\mathcal{D}}^{(k)}(c_{\text{text}})$ by directly using *i)* the CLIP encodings $\phi_{\text{CLIP}}(c_{\text{text}})$ of the actual textual description itself (top row), we interestingly see that our model generalizes to generating fictional descriptions and transfers attributions across object classes. However, when using *ii)* $\phi_{\text{CLIP}}(c_{\text{text}})$ together with its $k-1$ nearest neighbors from the database $\mathcal{D}$ as done in [2], the model does not generalize to these difficult conditional inputs (mid row). When *iii)* only using the k CLIP image representations of the nearest neighbors, the results are even worse (bottom row).

Similar to Sec. 4.1 we investigate the influence of k_{train} on the text-to-image generalization capability of *RDM* . To this end we evaluate the zero-shot transferability of the ImageNet models presented in the last section to text-conditional image generation and, using strategy i) from the last paragraph, evaluate their performance on 2000 captions from the validation set of COCO [6]. Fig. 7 compares the resulting FID and CLIP scores on COCO for the different choices of k_{train}. As a reference for the train performance, we furthermore plot the ImageNet FID. Similar to Fig. 6 we find that small k_{train} lead to weak generalization properties, since the corresponding models cannot handle misalignments between the text representation received during inference and image representations it is trained on. Increasing k_{train} results in sets $\mathcal{M}_{\mathcal{D}}^{(k)}(x)$ which cover a larger feature space volume, what regularizes the corresponding models to be more robust against such misalignments. Consequently, the generalization abilities increase with k_{train} and reach an optimum at $k_{\text{train}} = 8$. Further increasing k_{train} results in decreased information provided via the retrieved neighbors (*cf.* Fig. 4) and causes deteriorating generalization capabilities.

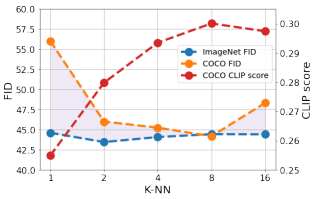


Figure 7: We observe that the number of neighbors k_{train} retrieved during training significantly impacts the generalization abilities of *RDM* . See Sec. 4.2.

We note the similarity of this approach to [50], which, by directly conditioning on the CLIP image representations of the data, essentially learns to invert the abstract image embedding. In our framework, this corresponds to $\xi_k(x) = \phi_{\text{CLIP}}(x)$ (i.e., no external database is provided). In order to fix the misalignment between text embeddings and image embeddings, [50] learns a conditional diffusion model for the generative mapping between these representations, requiring paired data. We argue that our retrieval-augmented approach provides an orthogonal approach to this task *without* requiring paired data. To demonstrate this, we train an "inversion model" as described above, i.e., use $\xi_k(x) = \phi_{\text{CLIP}}(x)$ with the same number of trainable parameters and computational budget as for the study in Fig. 7. When directly using text embeddings for inference, the model renders samples which generally resemble the prompt, but the visual quality is low (CLIP score 0.26 ± 0.05, FID ~ 87). Modeling the prior with a conditional normalizing flow [15, 53] improves the visual quality and achieves similar results in terms of text-consistency (CLIP score 0.26 ± 0.3, FID ~ 45), albeit requiring paired data. See Fig. 8 for a qualitative visualization and the supplement for implementation and training details.

Figure 8: Text-to-image generalization needs a generative prior or retrieval. See Sec. 4.2.

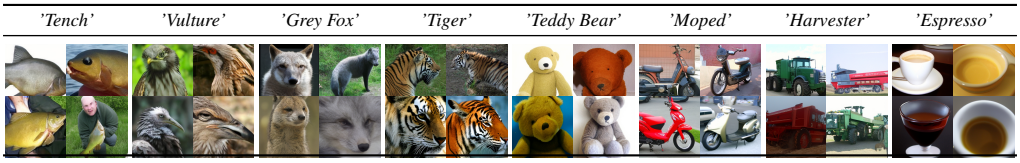


Figure 9: *RDM* can be used for class-conditional generation on ImageNet despite being trained without class labels. To achieve this during inference, we compute a pool of nearby visual instances from the database $\mathcal{D}$ for each class label based on its textual description, and combine it with its $k-1$ nearest neighbors as conditioning.

Class-Conditional Synthesis Similarly we can apply our model to zero-shot class-conditional image synthesis as proposed in Sec. 3.3. Fig. 9 shows samples from our model for classes from ImageNet. More samples for all experiments can be found in the supplement.

4.3 Zero-Shot Text-Guided Stylization by Exchanging the Database

In our semi-parametric model, the retrieval database $\mathcal{D}$ is an explicit part of the synthesis model. This allows novel applications, such as replacing this database after training to modify the model and thus its output. In this section we replace $\mathcal{D}_{\text{train}}$ of the ImageNet-*RDM* built from OpenImages with an alternate database $\mathcal{D}_{\text{style}}$, which contains all 138k images of the WikiArt dataset [57]. As in Sec. 4.2 we retrieve neighbors from $\mathcal{D}_{\text{style}}$ via a text prompt and use the text-retrieval strategy *iii)*. Results are shown in Fig. 10 (top row). Our model, though only trained on ImageNet, generalizes to this new database and is capable of generating artwork-like images which depict the content defined by the text prompts. To further emphasize the effects of this post-hoc exchange of $\mathcal{D}$, we show samples obtained with the same procedure but using $\mathcal{D}_{\text{train}}$ (bottom row).

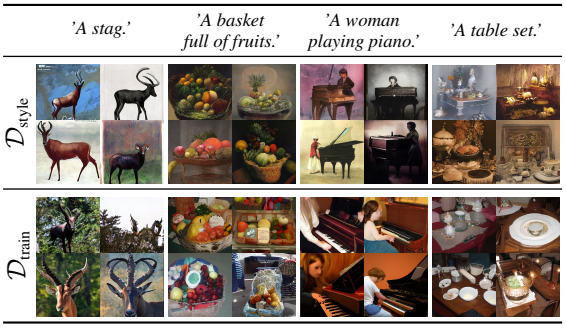


Figure 10: Zero-shot text-guided stylization with our ImageNet-*RDM* . Best viewed when zoomed in.

4.4 Increasing Dataset Complexity

To investigate their versatility for complex generative tasks, we compare semi-parametric models to their fully-parametric counterparts when systematically increasing the complexity of the training data $p(x)$. For both *RDM* and *RARM* , we train three identical models and corresponding fully parametric baselines (for details *cf.* the supplement) on the dogs-, mammals- and animals-subsets of ImageNet [11], *cf.* Tab. 3, until convergence.

Fig. 11 visualizes the results. Even for lower-complexity datasets such as *IN-Dogs*, our semi-parametric models improve over the baselines except for recall, where *RARM* performance slightly worse than a standard AR model. For more complex datasets, the performance gains become more significant. Interestingly, the recall scores of our models *improve* with increasing complexity, while

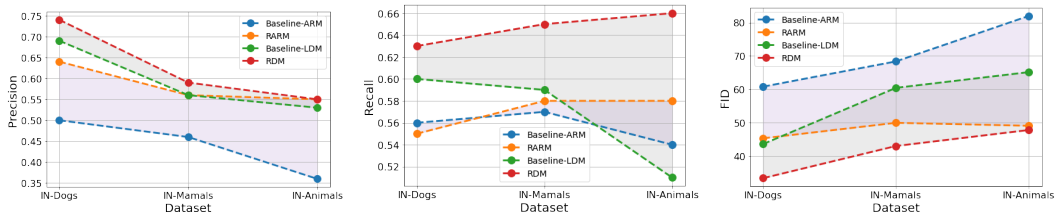
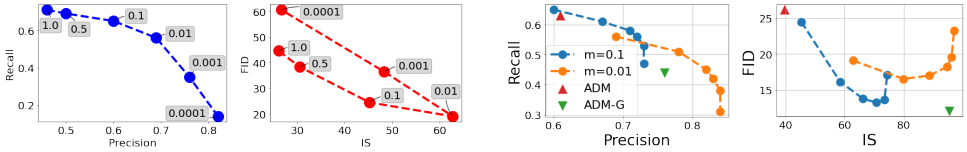


Figure 11: Assessing our approach when increasing dataset complexity as in Sec. 4.4. We observe that performance-gaps between semi- and fully-parametric models increase for more complex datasets.

those of the baselines strongly degrade. We attribute this to the explicit access of semi-parametric models to nearby visual instances for *all* classes including underrepresented ones via the proposal distribution $p_{\mathcal{D}}(\tilde{x})$, *cf.* Eq. (6), whereas a standard generative model might focus only on the modes containing the most often occurring classes (dogs in the case of ImageNet).

4.5 Quality-Diversity Trade-Offs

Top-m sampling. In this section, we evaluate the effects of the *top-m sampling* strategy introduced in Sec. 3.3. We train a *RDM* on the ImageNet [11] dataset and assess the usual generative performance metrics based on 50k generated samples and the entire training set [4]. Results are shown in Fig. 12a. For precision and recall scores, we observe a truncation behavior similar to other inference-time sampling techniques [4, 12, 26, 19]: For small values of m, we obtain coherent samples, which all come from a single or a small number of modes, as indicated by large precision scores. Increasing m, on the other hand, boosts diversity at the expense of consistency. For FID and IS, we find a sweet spot for $m = 0.01$, which yields optima for both of these metrics. Visual examples for different values of m are shown in the appendix. The supplement also contains similar experiments for *RARM* .



(a) Quality-diversity trade-offs when applying top-m sampling.

(b) Assessing the effects of classifier free guidance.

Figure 12: Analysis of the quality-diversity trade-offs when applying top-m sampling and classifier-free guidance.

Classifier-free guidance. Since *RDM* is a conditional diffusion model (conditioned on the neighbor encodings $\phi(y)$), we can apply classifier-free diffusion guidance [26] also for unconditional modeling. Interestingly, we find that we can apply this technique without adding an additional $\emptyset$-label to account for a purely unconditional setting while training ϵ_θ, as originally proposed in [26] and instead use a vector of zeros to generate an unconditional prediction with ϵ_θ. Additionally, this technique can be combined with *top-m sampling* to obtain further control during sampling. In Fig. 12b we show the effects of this combination for the ImageNet-model as described in the previous paragraph, with $m \in \{0.01, 0.1\}$ and classifier scale $s \in \{1.0, 1.25, 1.5, 1.75, 2.0, 3.0\}$, from left to right for each line. Moreover we qualitatively show the effects of guidance in the supplement, demonstrating the versatility of these sampling strategies during inference.

5 Conclusion

This paper questions the prevalent paradigm of current generative image synthesis: rather than compressing large training data in ever-growing generative models, we have proposed to efficiently store an image database and condition a comparably small generative model directly on meaningful samples from the database. To identify informative samples for the synthesis tasks at hand we follow an efficient retrieval-based approach. In the experiments our approach has outperformed the state of the art on various synthesis tasks despite demanding significantly less memory and compute. Moreover, it allows (i) conditional synthesis for tasks for which it has not been explicitly trained, and (ii) post-hoc transfer of a model to new domains by simply replacing the retrieval database. Combined with CLIP's joint feature space, our model achieves strong results on text-image synthesis, despite being trained only on images. In particular, our retrieval-based approach eliminates the need to train an explicit generative prior model in the latent CLIP space by directly covering the neighborhood of a given data point. While we assume that our approach still benefits from scaling, it shows a path to more efficiently trained generative models of images.

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 the supplemental material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the supplemental material.
   (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)? [Yes] The code will be released, the data is publicly available and the additional instructions are provided in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (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] See the supplemental material.
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? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code and pretrained models will be released.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
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]

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[2] Oron Ashual, Shelly Sheynin, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*, 2021.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. pages 1209–1218, 2018. doi: 10.1109/CVPR.2018.00132. URL `http://openaccess.thecvf.com/content_cvpr_2018/html/Caesar_COCO-Stuff_Thing_and_CVPR_2018_paper.html`.

[7] Arantxa Casanova, Marlène Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34, 2021.

[8] Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. *arXiv preprint arXiv:2204.07156*, 2022.

[9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[10] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *ArXiv*, abs/1604.06174, 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

[13] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=HkpbnH9lx`.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

[19] Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. *CoRR*, abs/1805.04833, 2018. URL `http://arxiv.org/abs/1805.04833`.

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

[21] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. *arXiv preprint arXiv:2111.14822*, 2021.

[22] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[23] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3887–3896. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/guo20h.html`.

[24] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020.

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Adv. Neural Inform. Process. Syst.*, pages 6626–6637, 2017.

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

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[28] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23 (47):1–33, 2022.

[29] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020.

[31] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[32] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[33] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[34] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

[35] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710*, 2020.

[36] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[37] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

[38] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[39] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *CoRR*, abs/1811.00982, 2018. URL `http://arxiv.org/abs/1811.00982`.

[40] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *CoRR*, abs/1904.06991, 2019. URL `http://arxiv.org/abs/1904.06991`.

[41] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[42] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. *arXiv preprint arXiv:2202.11233*, 2022.

[43] Yuxian Meng, Shi Zong, Xiaoya Li, Xiaofei Sun, Tianwei Zhang, Fei Wu, and Jiwei Li. Gnn-lm: Language modeling based on global contexts via gnn. *arXiv preprint arXiv:2110.08743*, 2021.

[44] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *Advances in neural information processing systems*, 30, 2017.

[45] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[46] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[47] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[49] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[50] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[51] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[52] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[53] Robin Rombach, Patrick Esser, and Björn Ommer. Network-to-network translation with conditional invertible neural networks. In *NeurIPS*, 2020.

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2021.

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI (3)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[56] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021.

[57] Babak Saleh and Ahmed M. Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *CoRR*, abs/1505.00855, 2015. URL `http://arxiv.org/abs/1505.00855`.

[58] Tim Salimans, I. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[59] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[60] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *arXiv preprint arXiv:2202.00273*, 2022.

[61] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Retrievalfuse: Neural 3d scene reconstruction with a database. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12568–12577, 2021.

[62] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2C: diffusion-denoising models for few-shot conditional generation. *CoRR*, abs/2106.06819, 2021. URL `https://arxiv.org/abs/2106.06819`.

[63] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[64] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[65] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[66] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020.

[67] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

[68] Rich Sutton. The bitter lesson, 2019. URL `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`.

[69] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. Retrievegan: Image synthesis via differentiable patch retrieval. In *European Conference on Computer Vision*, pages 242–257. Springer, 2020.

[70] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

[71] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[72] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[73] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[75] Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *CoRR*, abs/2203.08913, 2022. doi: 10.48550/arXiv.2203.08913. URL `https://doi.org/10.48550/arXiv.2203.08913`.

[76] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

[77] Rui Xu, Minghao Guo, Jiaqi Wang, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Texture memory-augmented deep patch-based image inpainting. *IEEE Transactions on Image Processing*, 30:9112–9124, 2021.

[78] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

[79] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.