kNN-DIFFUSION: IMAGE GENERATION VIA LARGE-SCALE RETRIEVAL

Anonymous authors
Paper under double-blind review

Figure 1: (a) Samples of stickers generated from text inputs, (b) Semantic text-guided manipulations applied to the "Original" image without using edit masks. In both cases, our model was trained using only pre-trained multi-modal embeddings, but without an explicit text-image dataset

ABSTRACT

Recent text-to-image models have achieved impressive results. However, since they require large-scale datasets of text-image pairs, it is impractical to train them on new domains where data is scarce or not labeled. In this work, we propose using large-scale retrieval methods, in particular, efficient k-Nearest-Neighbors (kNN), which offers novel capabilities: (1) training a substantially small and efficient text-to-image diffusion model using only pre-trained multi-modal embeddings, but without an explicit text-image dataset, (2) generating out-of-distribution images by simply swapping the retrieval database at inference time, and (3) performing text-driven local semantic manipulations while preserving object identity. To demonstrate the robustness of our method, we apply our kNN approach on two state-of-the-art diffusion backbones, and show results on several different datasets. As evaluated by human studies and automatic metrics, our method achieves state-of-the-art results compared to existing approaches that train text-to-image generation models using images only (without an explicit paired text-image dataset).

1 INTRODUCTION

Large-scale generative models have been applied successfully to image generation tasks (Gafni et al., 2022; Ramesh et al., 2021; Nichol et al., 2021; Saharia et al., 2022; Yu et al., 2022), and have shown outstanding capabilities in extending human creativity using editing and user control. However, these models face several significant challenges: (i) **Large-scale paired data requirement**. To achieve high-quality results, text-to-image models rely heavily on large-scale datasets of (text, image) pairs collected from the internet. Due to the requirement of paired data, these models cannot be applied to new or customized domains with only unannotated images. (ii) **Computational cost and efficiency**. Training these models on highly complex distributions of natural images usually requires scaling the size of the model, data, batch-size, and training time, which makes them challenging to train and less accessible to

the community. Recently, several works proposed text-to-image models trained without an explicit paired text-image datasets. Liu et al. (2021) performed a direct optimization to a pre-trained model based on a CLIP loss (Radford et al., 2021). Such approaches are time-consuming, since they require optimization for each input. Zhou et al. (2021) proposed training with CLIP image embedding perturbed with Gaussian noise. However, to achieve high-quality results, an additional model needs to be trained with an annotated text-image pairs dataset.

In this work, we introduce a novel generative model, *kNN-Diffusion*, which tackles these issues and progresses towards more accessible models for the research community and other users. Our model leverages a large-scale retrieval method, k-Nearest-Neighbors (kNN) search, in order to train the model with only pre-trained multi-modal embeddings, but without an explicit text-image dataset. Specifically, our diffusion model is conditioned on two inputs: (1) image embedding (at training time) or text embedding (at inference), extracted using the multi-modal pre-trained CLIP encoder, and (2) kNN embeddings, representing the k most similar images in the CLIP latent space. During training, we assume that no paired text is available, hence condition only on CLIP image embedding and on k additional image embeddings, selected using the retrieval model. At inference, only text inputs are given, so instead of image embeddings, we use the text embedding that shares a joint embedding space with the image embeddings. Here, the kNN image embeddings are retrieved using the *text* embeddings.

The additional kNN image embeddings have three main benefits: (1) they extend the distribution of conditioning embeddings and ensure the distribution is similar in train and inference, thus helping to bridge the gap between the image and text embedding distributions (see Fig. 10 for visualization); (2) they teach the model to learn to generate images from a target distribution by using samples from that distribution. This capability allows generalizing to different distributions at test time and generating out-of-distribution samples; (3) they hold information that does not need to be present in the model, which allows it to be substantially smaller. We demonstrate the effectiveness of our kNN approach in Sec. 4.

To assess the performance of our method, we train our model on two large-scale datasets: the Public Multimodal Dataset (Singh et al., 2021) and an image-only stickers dataset collected from the Internet. We show state-of-the-art zero-shot capabilities on MS-COCO (Lin et al., 2014), LN-COCO (Pont-Tuset et al., 2020) and CUB (Wah et al., 2011). To further demonstrate the advantage of using retrieval methods in text-to-image generation, we train two diffusion backbones using our kNN approach: continuous (Ramesh et al., 2022) and discrete (Gu et al., 2021). In both cases we outperform the model that is trained without kNN. In comparison to alternative methods presented in Sec. 4, we achieve state-of-the-art results in both human evaluations and FID score, with 400 million parameters and 7 second inference time.

Lastly, we introduce a new approach for local and semantic manipulations that is based on CLIP and kNN, without relying on user-provided masks. Specifically, we fine-tune our model to perform local and complex modifications that satisfies a given target text prompt. For example, given the teddy bear's image in Fig. 4, and the target text "holds a heart", our method automatically locates the local region that should be modified and synthesizes a high-resolution manipulated image in which (1) the teddy bear's identity is accurately preserved and (2) the manipulation is aligned with the target text. We demonstrate our qualitative advantage by comparing our results with two state-of-the-art models, Text2Live (Bar-Tal et al., 2022) and Textual Inversion (Gal et al., 2022), that perform image manipulations without masks (Fig. 4, 20 and 21).

We summarize the contributions of this paper as follows: (1) We propose *kNN-Diffusion*, a novel and efficient model that utilizes a large-scale retrieval method for training a text-to-image model with only pre-trained multi-modal embeddings, but without an explicit text-image dataset.. (2) We demonstrate efficient out-of-distribution generation, which is achieved by substituting retrieval databases. (3) We present a new approach for local and semantic image manipulation, without utilizing masks. (4) We evaluate our method on two diffusion backbones, discrete and continuous, as well as on several datasets, and present state-of-the-art results compared to baselines trained using image-only datasets.

2 Related Work

Text-to-image models. Text-to-image generation is a well-studied task that focuses on generating images from text descriptions. While GANs (Xu et al., 2018; Zhu et al., 2019; Zhang et al., 2021) and VQ-VAE Transformer-based methods (Ramesh et al., 2021; Gafni et al., 2022; Yu et al., 2022; Ding et al., 2021) have shown remarkable results, recently impressive results have been attained with discrete (Gu et al., 2021) and continuous (Nichol et al., 2021; Saharia

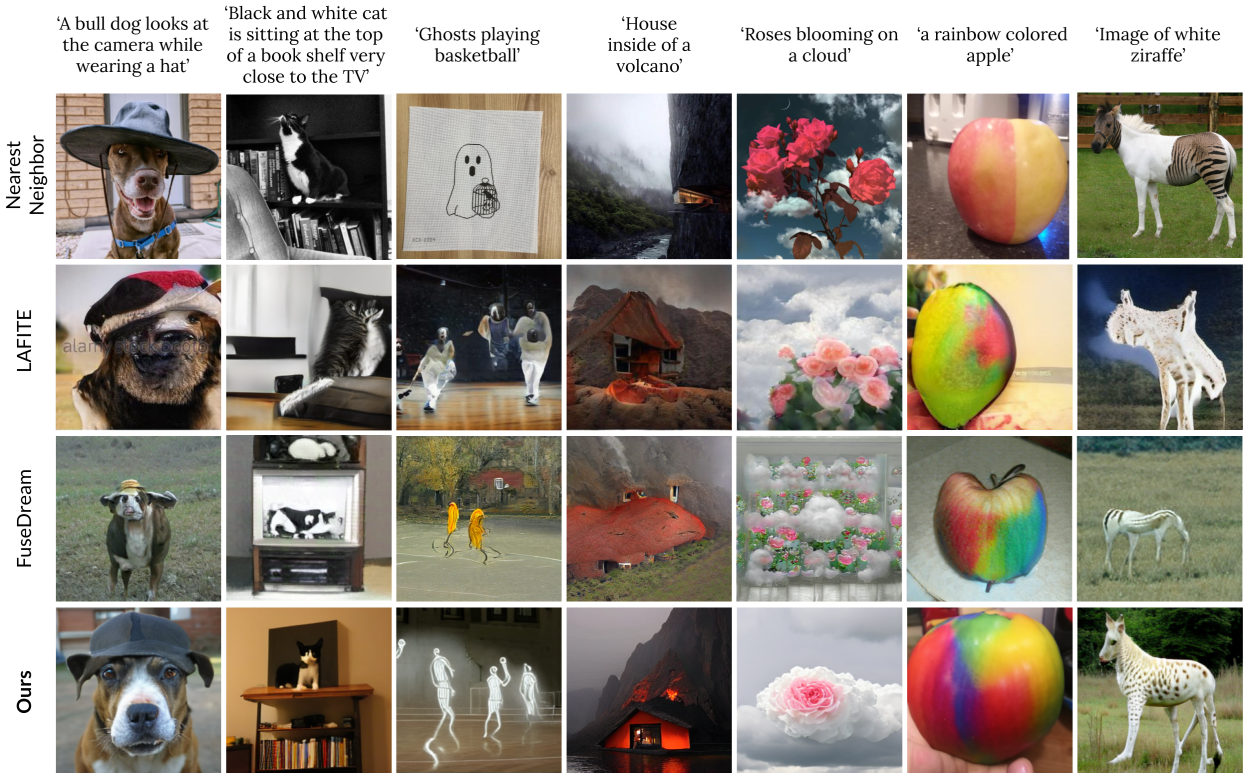


Figure 2: Qualitative comparisons with previous work. *Nearest Neighbor* is the first nearest neighbor of the text-input in the PMD dataset.

et al., 2022; Ramesh et al., 2022; Rombach et al., 2022) diffusion models. Most recent works trained diffusion models conditioned on text embeddings extracted using a pre-trained text encoder (Saharia et al., 2022; Yu et al., 2022) or image embedding extracted using CLIP (Ramesh et al., 2022). While producing impressive results, all previous works described above are supervised and trained with paired text-image datasets.

Several works have proposed training text-to-image models without an explicit text-image dataset. data. FuseDream (Liu et al., 2021) proposed a direct optimization to a pre-trained generative model based on CLIP loss. This method relies on a pre-trained GAN and requires a time-consuming optimization process for each image. LAFITE (Zhou et al., 2021) recently demonstrated text-to-image generation results without requiring paired text-image datasets. Here, the embeddings of CLIP are used interchangeably at train and test time to condition a GAN-based model. The joint text-image embedding enables inference given a text input, whereas in training the model is fed with the visual embedding only. However, the gap between the text and image distributions in the joint embeddings space leads to results with substantially lower quality, as we show in our experiments. To overcome this gap, LAFITE added noise to the image embeddings during training. Our remedy to this gap in distributions is to condition the model on the retrieval of an actual image embeddings, using a text-image joint space.

Retrieval for generation. The Information Retrieval (IR) literature tackles the challenge of retrieving a small amount of information from a large database, given a user's query. While it has been shown that adding more training data improves performance, this comes at a cost, as it requires collecting and training on more data. A simple, yet efficient retrieval mechanism is to retrieve the K nearest neighbors (kNN) between the query and the entities in the database in some pre-calculated embedding space (Bijalwan et al., 2014). The database allows the model to leverage extensive world-knowledge for its specific task Borgeaud et al. (2021). Recently, language models were augmented with a memory component, allowing them to store representations of past inputs (Wu et al., 2022). The latter were then queried using a lookup operation, improving performance in various benchmarks and tasks. Retrieval models have been used for various tasks in learning problems, for example, language modeling (Borgeaud et al., 2021), machine translation (Gu et al., 2018), question answering (Lee et al., 2019) and image generation (Tseng et al., 2020; Qi et al., 2018). RetrieveGAN (Tseng et al., 2020) uses a differentiable retrieval module for image generation from a scene description, RetrievalFuse (Siddiqui et al., 2021) proposed a neural 3D scene reconstruction based on a retrieval

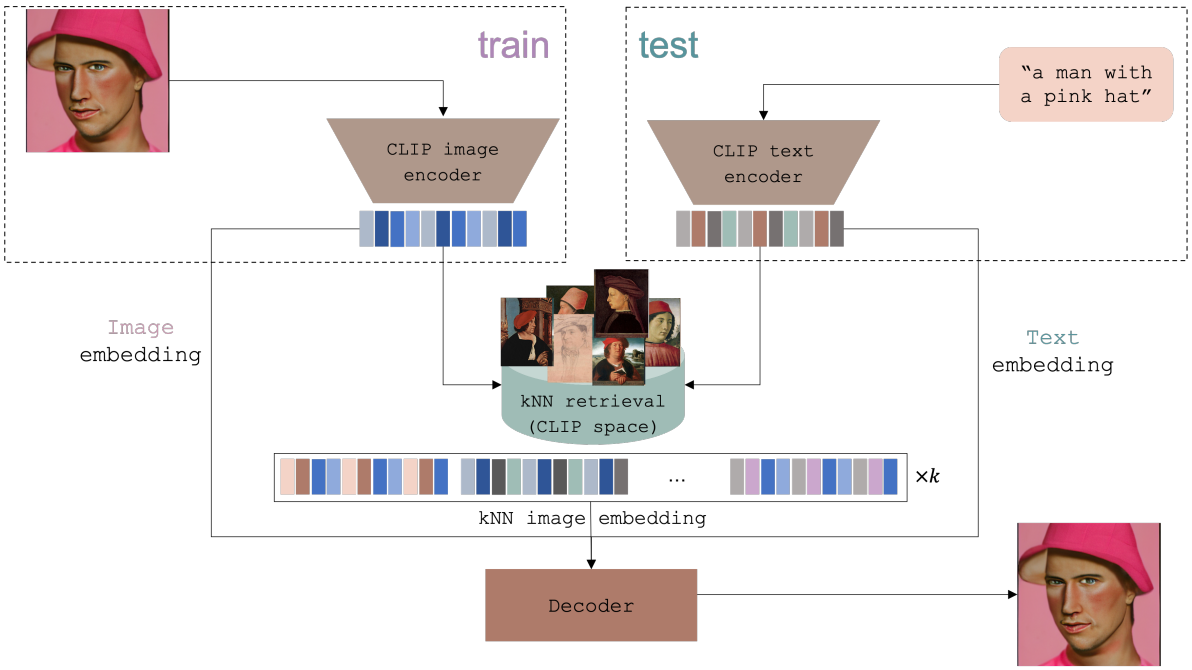


Figure 3: The overall framework of our *kNN-Diffusion* model. In both training and inference, the decoder is conditioned on CLIP embedding, and kNN image embedding. During training, we condition the model on image CLIP embedding, and its kNN image embeddings extracted using the retrieval method. At inference time, given an input text, the kNN image embeddings are retrieved based on the CLIP text embedding that shares a joint embedding space with the image embedding.

system. SIMS (Qi et al., 2018) proposed generating an image using semantic layout and compatible image segments that are retrieved from retrieval database of image segments. (Iskakov, 2018) showed that the use of retrieval database in inpainting task significantly boosts visual quality. In this work we utilize the kNN retrieval mechanism over the shared text-image embedding space, CLIP (Radford et al., 2021). Using extensive ablation studies, we show the importance of the retrieval model both for training and inference, and demonstrate its large impact on performance.

Multi-modal feature learning. Learning a joint and aligned feature space for several modalities is challenging, as it requires alignment between the modalities (paired datasets), whose distributions may vary. Specifically, the joint feature space of vision-and-language has been a long-standing problem in artificial intelligence. CLIP (Radford et al., 2021) is a method that successfully tackled this challenge, by leveraging contrastive learning over a large dataset of aligned text-image pairs. Several works, such as BLIP (Li et al., 2022), (Mu et al., 2021) and FLAVA (Singh et al., 2021), followed this idea and further improved the joint representation. The joint representation was shown to hold a strong semantic alignment between the two modalities, enabling image generation (Liu et al., 2021; Wang et al., 2022), image manipulation (Patashnik et al., 2021; Avrahami et al., 2022b), and image captioning (Mokady et al., 2021). In this work we leverage the joint representation in two ways: (i) enabling textless training with only visual data, while using text at inference time, and (ii) creating an efficient embedding space for the use of the retrieval model.

3 METHOD

Our main goal is to facilitate language-guided generation of user-specified concepts while using an images-only dataset (without an explicit paired text-image dataset) during training. A possible way to achieve this goal is to use a shared text-image encoder that will map text-image pairs into the same latent space, thus allowing training with an image embedding, and inferring from text embedding. A candidate for this encoder is CLIP, which has been trained with a contrastive loss on a large-scale dataset of text-image pairs. However, as we show quantitatively in Tab. 1, 2 and qualitatively in Fig. 5, 14, 10, CLIP embeddings alone cannot accurately bridge the gap between the text and image distributions. In order to reduce this gap, several methods have been proposed. The closest work to ours is LAFITE, which perturbs the CLIP image embedding with adaptive Gaussian noise. Under the assumption that there is a large

Figure 4: Results for text-guided image manipulations without using masks. The original image is shown in the left column, our manipulated images are shown in the center. The images of Bar-Tal et al. (2022); Gal et al. (2022) were generated using the authors' official code. The full comparison is available in the supplement.

paired text-image dataset, Ramesh et al. (2022) have proposed a prior that is used during inference, and is trained to generate possible CLIP image embeddings from a given text caption.

In this regard, we propose using a large-scale and non-trainable image embedding index as an integral part of the diffusion process. Our proposed method, *kNN-Diffusion*, assumes that only image data is provided during training. As shown in Fig. 3, our model is comprised of three main components: (1) A multi-modal text-image encoder (CLIP); (2) A retrieval model - A data structure containing image embeddings, which is indexed for a fast kNN search; (3) An image generation network - A trainable diffusion-based image generation model, conditioned on the projected retrievals. For both training and inference, the image generation network is conditioned on K additional image embeddings, chosen using the retrieval model to ensure a similar distribution of the condition in training and inference. The following sections describe these components.

Retrieval model. Our retrieval model has three non-trainable modules: a pre-trained text encoder f_{txt} (CLIP text encoder), a pre-trained image encoder f_{img} (CLIP image encoder) and an index $\mathcal{H}$. The encoders map text descriptions and image samples to a joint multi-modal d-dimensional feature space $\mathbb{R}^d$. The index stores an efficient representation of the images database - $\mathcal{H} := \{f_{img}(i) \in \mathbb{R}^d | i \in \mathcal{I}\}$ where $\mathcal{I}$ denotes the dataset of images. During training, we use the index to efficiently extract the k nearest neighbors in the feature space of the image embedding $f_{img}(\mathrm{I}) \in \mathbb{R}^d$ - $\mathrm{knn}_{img}(\mathrm{I}, k) := \arg\min^k_{h \in \mathcal{H}} \mathbf{s}(f_{img}(\mathrm{I}), h)$ where $\mathbf{s}$ is a distance function and $\arg\min^k$ output the minimal k elements. The set $\{f_{img}(\mathrm{I}), \mathrm{knn}_{img}(\mathrm{I}, k)\}$ is used as the condition to the generative model. During inference, given a query text t, an embedding $f_{txt}(t)$ is extracted. The generative model is conditioned on this embedding and its k nearest neighbors from the database - $\mathrm{knn}_{txt}(t, k) := \arg\min^k_{h \in \mathcal{H}} \mathbf{s}(f_{txt}(t), h)$. During training, we add embeddings of real images, by applying the retrieval method to the input image embedding. The extracted kNN should have a large enough distribution to cover the potential text embedding. During inference, the kNN are retrieved using the text embedding (See Fig. 11). In all of our experiments we use the cosine similarity metric as the distance function $\mathbf{s}$, $k = 10$ for the number of nearest neighbors and $d = 512$. The full implementation details can be found in Sec.6.5 of the supplement.

Image generation network. In order to demonstrate the robustness of our method, we apply our kNN approach on two different diffusion backbones: Discrete (Gu et al., 2021) and Continuous (Nichol et al., 2021; Sohl-Dickstein et al., 2015; Ho et al., 2020; Dhariwal & Nichol, 2021). Although very different in practice, these models share the same theoretical idea. Let $x_0 \sim q(x_0)$ be a sample from our images distribution. A forward diffusion process is a Markov chain that adds noise at each step $q(x_n|x_{n-1})$. The reverse process, $p_\theta(x_{n-1}|x_n, x_0)$, is a denoising process

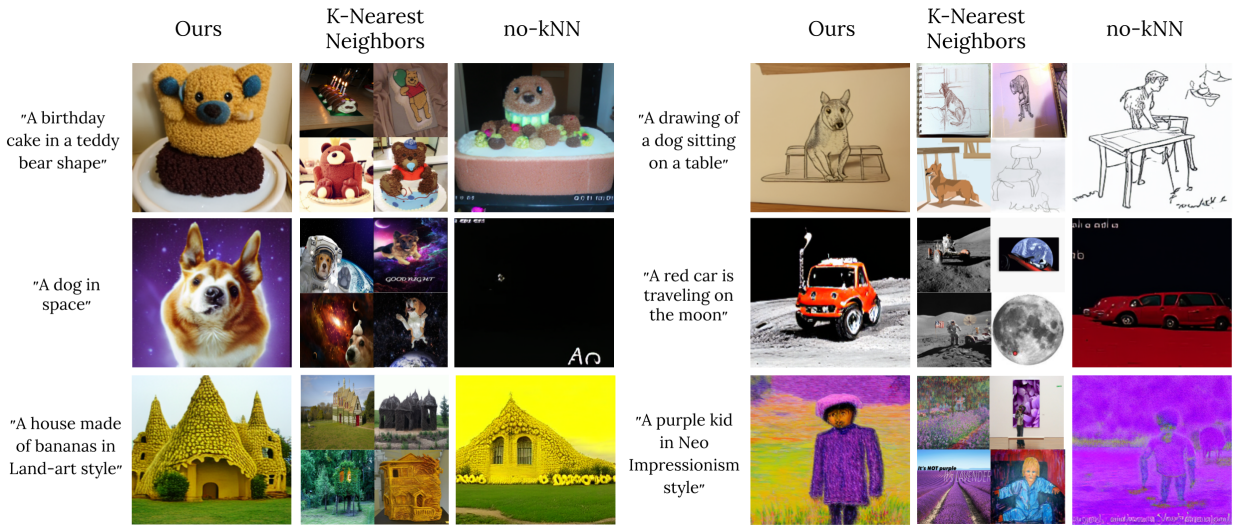


Figure 5: Comparison of our model, trained on PMD with (1) kNN extracted in inference, (2) the same model without using kNN in inference. While the kNN lack information regarding text semantics, our model considers both text semantics and the kNN, thus proving the advantage of using both the text and the kNN embeddings.

that removes noise from an initialized noise state. At inference time, the model can generate an output, starting with noise and gradually removing it using p_θ. For additional background on diffusion models, please refer to Sec. 6.1 in the supplement.

In the discrete diffusion model, $q(x_n|x_{n-1}) := v^T(x_n)\mathbf{Q}_n v(x_{n-1})$ where $v(x_n)$ is a one-hot vector with entry 1 at x_n, and $\mathbf{Q}_n$ is a transition matrix, modeling the probability to move from state x_{n-1} to x_n, using uniform probability over the vocabulary and a pre-defined probability for additional special *[MASK]* token. We can compute the reverse transition distribution according to: $p_\theta(x_{n-1}|x_n, y) := \sum_{\hat{x}_0=1}^{k} q(x_{n-1}|x_n, \hat{x_0}) p_\theta(\hat{x_0}|x_n, x_0, y)$ where x_0 is a discrete vector, tokenized by the VQGAN (Esser et al., 2021) encoder and y is the conditioning signal. For modeling p_θ we have followed (Gu et al., 2021) and used a conditional Transformer (Vaswani et al., 2017). Full implementation details of the discrete model can be found in Sec. 6.6 of the supplement.

In the continuous diffusion model, $q(x_n|x_{n-1}) := \mathcal{N}(x_n; \sqrt{\alpha_t} x_{n-1}, (1-\alpha_n)x_0)$ and $p_\theta(x_{n-1}|x_n, y) := \mathcal{N}(\mu_\theta(x_n, y), \Sigma_\theta(x_n, y))$. Here, the noise function is Gaussian noise. Following (Ho et al., 2020; Nichol et al., 2021) we trained a model ϵ_θ to predict the added noise using a standard mean-squared error loss: $L := E_{n\sim[1,N], x_0\sim q(x_0), \epsilon\sim\mathcal{N}(0,\mathbf{I})}[||\epsilon - \epsilon_\theta(x_n, n, y||^2]$ where ϵ_θ is a U-net model and y is the conditioning signal. Additional implementation details of the continuous model can be found in Sec. 6.7 in the supplement.

In both cases, we condition our model on $y = (f_{img}(x_0), \text{knn}_{img}(x_0, k))$ where $f_{img}(x_0)$ is the CLIP image embedding, $\text{knn}_{img}(x_0, k)$ is the k nearest neighbors in the feature space of the image embedding. Following (Ramesh et al., 2022; Rombach et al., 2022) conditional injection, we condition our model on the image CLIP embedding, and the kNN clip embeddings by applying cross attention in the attention layers of the architecture. We study various approaches for the kNN conditioning in Sec. 6.4 in the supplement.

We sample both our models using Classifier Free Guidance (CFG) (Nichol et al., 2021; Ho & Salimans, 2021). Since CFG was originally proposed for continuous models, we propose a method for using it with discrete models as well. The full details can be found in Sec. 6.6 of the supplement.

3.1 Text-only image manipulation

The majority of previous works in the task of image manipulation either rely on user-provided masks (Nichol et al., 2021; Avrahami et al., 2022b;a), or are limited to global editing (Crowson et al., 2022; Kim et al., 2022). Recently, several works (Bar-Tal et al., 2022; Hertz et al., 2022; Gal et al., 2022) have made progress with local manipulations without relying on user edited masks. Nevertheless, most of the techniques suffer from several shortcomings: (1) They

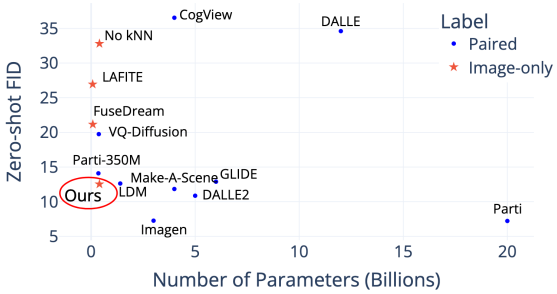


Figure 6: FID comparison on MS-COCO, including models trained on image-only datasets and text-image datasets.

Table 1: Results for zero-shot Text-to-Image generation on the MS-COCO, CUB and LN-COCO test sets. Image-quality and Text-alignment report the percentage of majority human raters votes in favor of our method when comparing between a certain model and ours.

Model	MS-COCO			CUB			LN-COCO		
	FID↓	Im. qual.	Txt align.	FID↓	Im. qual.	Txt align.	FID↓	Im. qual.	Txt align.
LAFITE	26.9	72.1	65.3	89.7	74.0	59.6	42.8	68.4	61.9
FuseDream	21.2	64.0	79.3	50.2	79.1	60.9	37.5	71.1	59.0
no-kNN	32.8	70.8	68.3	95.1	81.0	61.2	65.0	61.4	59.8
Ours	**12.5**	-	-	**42.9**	-	-	**35.6**	-	-

enable local texture changes, yet cannot modify complex structures, (2) they struggle to preserve the identity of the object, for example, when manipulating humans, (3) they require optimization for each input.

We address these issues by extending *kNN-Diffusion* to perform local and semantic-aware image manipulations without any provided mask. Illustration of the approach is provided in Fig. 12 and Fig. 13 in the supplement. For this task, the model is trained to predict the original image from a manipulated version. Specifically, we create a manipulated version of the image, which differs from the original image only in some local area. Given a random local area M in the image I, the manipulated image I_{manip} is constructed by replacing the area with the corresponding nearest neighbor: $\mathrm{I}_{manip} = \mathrm{I} \cdot (1 - M) + \mathrm{nn}_{img}(\mathrm{I}, 1) \cdot M$, where $\mathrm{nn}_{img}(\mathrm{I}, 1)$ is the the nearest neighbor obtained after aligning it with I using the ECC alignment algorithm (Evangelidis & Psarakis, 2008). The model then receives as input the manipulated image, together with the CLIP embedding of the original image *only* in the local area: $f_{img}(\mathrm{I} \cdot M)$. This CLIP embedding represents the required modification that should be applied to the manipulated image in order to predict the original image. During inference, instead of using the CLIP embedding of the local area, the desired modification is represented using the CLIP embedding of the user text query. We modified the model to be capable of receiving as a condition both the manipulated image and the CLIP embedding of the local area.

4 Experiments

First, we conduct qualitative and quantitative comparisons on MS-COCO, LN-COCO and CUB datasets to compare with previous work. To further demonstrate the advantage of our method, we provide comparison on an image-only stickers dataset, where we apply our approach on two diffusion backbones. Next, we demonstrate image manipulation and out-of-distribution capabilities. Finally, to better assess the effect of each contribution, an ablation study is provided.

Datasets and Metrics. For photo-realistic experiments, our model was trained only on the images (omitting the text) of a modified version of the Public Multimodal Dataset (PMD) used by FLAVA (Singh et al., 2021). More information about the dataset is available in Sec. 6.3 of the supplement. To further demonstrate the capabilities of our method, we collected 400 million sticker images from the web, containing combinations of concepts such as objects, characters/avatars and text. The collected stickers do not have paired text, and are substantially different from photo-realistic data. Furthermore, since they have no paired text, they were not part of CLIP's training data, which makes the text-to-image generation task more challenging.
Evaluation metrics are based on objective and subjective metrics: (i) FID (Heusel et al., 2017) is an objective metric used to assess the quality of synthesized images, (ii) human evaluation - we ask human raters for their preference, comparing two methods based on image quality and text alignment. We used 600 image pairs; five raters rated each pair. The results are shown as a percentage of majority votes in favor of our method over the baselines. We report the full human evaluation protocol in the supplement. We chose to omit Inception-Score, since it is shown by Barratt & Sharma (2018) to be a misleading metric for models that were not trained on Imagenet.

Table 2: Results on the stickers dataset. For both the continuous and the discrete models, we report the percentage of human raters prefer our method over the baselines with respect to image quality and text alignment. Discrete *no-kNN* refers to VQ-diffusion, trained without an explicit text-image dataset, and Continuous *no-kNN*, to DALL·E2 decoder, trained without an explicit text-image dataset.

Model	FID↓	Ours Discrete		Ours Continuous	
		Image quality	Text alignment	Image quality	Text alignment
DALL·E2+ClipCap	55.5	71.6	69.2	67.0	68.3
LAFITE	58.7	63.5	59.9	76.0	71.2
no-kNN	52.7	72.1	67.6	66.8	69.4
Ours	**40.8**	-	-	-	-

4.1 Qualitative and Quantitative Results

We begin by comparing our model, trained on the PMD dataset, with the previous works LAFITE and FuseDream, that trained on image-only datasets. To demonstrate the advantage of using a retrieval method in text-to-image generation, we trained a model variant, *no-kNN* . This baseline was trained solely on image embeddings (omitting the kNN), while during inference, the images were generated using the text embedding. Tab. 1 displays zero-shot results on three different datasets: MS-COCO, CUB and LN-COCO. We follow the evaluation protocol of LAFITE, reporting our results on 30,000 images from MS-COCO validation set without training, nor using it's training partition in the kNN index. Similarly, we follow LAFITE for CUB and LN-COCO evaluation. As can be seen, our model achieves the lowest FID score in all scenarios. In addition, human evaluations rate our method as better aligned to text and with the highest images quality. In Fig. 2, 5 and 16 we present a qualitative comparison between the methods. One can observe that while the simple retrieval baseline outputs non-generated images with high-quality, the images generated by our method are more faithful to the input text.

To further demonstrate the effectiveness of our method, we present in Fig. 6 a comparison of our model with the latest text-to-image models trained on paired text-image datasets: DALL·E, CogView, VQ-Diffusion, GLIDE, Latent Diffusion(LDM), Make-A-Scene, DALL·E2, Parti and Imagen. As can be seen, our model achieves comparable results to recent models trained with full text-image pairs (e.g LDM, GLIDE), despite being trained on an image-only dataset, with significantly lower computational costs. The results demonstrate that leveraging an external retrieval database allows our method to compensate for different trade-offs, in particular, reducing the number of parameters in the model. Additional samples are provided in Fig. 15 in the supplement.

Text-to-sticker generation. As the sticker dataset does not have paired text, and is substantially different from photo-realistic data, it allows us to illustrate the advantage of our model on an image-only dataset. A selection of stickers generated by our model is presented in Fig. 1 and Fig. 18, 17. To demonstrate the importance of using kNN on image-only datasets, we evaluate our approach on two diffusion backbones. To this end, we trained a continuous diffusion model (Ramesh et al., 2022) and a discrete diffusion model (Gu et al., 2021), both conditioned on the kNN image embeddings. For each backbone, we compare our method with the following baselines: (1) *no-kNN* - this baseline was trained using both the continuous and the discrete methods conditioned only on image CLIP embedding, without using kNN. In the discrete case, we trained a VQ-diffusion model, while in the continuous case, we trained a re-implementation of DALL·E2's decoder (without prior). (2) *DALL·E2+ClipCap* baseline - here, we first captioned the entire sticker dataset using ClipCap (Mokady et al., 2021), then trained DALL·E2 decoder on the captioned dataset. (3) LAFITE - we trained LAFITE language-free model on our stickers dataset using the authors' published code.

We present the results in Tab. 2. The FID is calculated over a subset of $3,000$ stickers, generated from the ClipCap captioned dataset. As can be seen, our model achieves the lowest FID score. In addition, it outperforms all baselines in human evaluation comparison, using continuous and discrete backbones. In particular, compared with the same model trained without kNN, our model achieves significantly higher favorability in both text alignment and image quality.

4.2 Applications

Text-only image manipulation. We demonstrate the manipulation capabilities of our model in Fig. 1, 4 and 19. Furthermore, we qualitatively compare our model with Text2LIVE (Bar-Tal et al., 2022) and Textual Inversion (Gal et al., 2022), using the authors' published code. Text2LIVE proposed generating an edit layer that is composed over

Figure 7: A qualitative comparison between different indexes used by the same model. In each row we have conditioned the model with kNN of a different index. (1) **Aesthetic.** Using images from the first quantile of an aesthetic classifier, (2) **Unaesthetic.** Using images from the last quantile of an aesthetic classifier, (3) **Image search engine.** Using images retrieved from Google Images, (4) **Stickers.** Using the stickers index.

the original input, using a generator trained for each training image. Textual Inversion utilized the pre-trained Latent Diffusion model to invert the input image into a token embedding. The embedding is then used to compose novel textual queries for the generative model. Fig. 4 shows representative results, and the rest are included in Fig. 20 and 21 in the supplement. In contrast to our model, baseline methods lack text correspondence or they do not preserve the identity of the object. Since Text2LIVE is optimized to perform local changes, it has the difficulty changing the structure of the object (e.g. the "raising his hand" example in Fig. 4). Textual Inversion baseline changes the identity of the object because it struggles reconstructing the textual representation of the source image. Our model, on the other hand, can perform challenging manipulations that are aligned with the text, while preserving the object identity.

Out-of-distribution generation. Using the retrieval index as part of the generation process enables using different databases during inference, without fine-tuning. This allows the model to generate images from distributions that were not part of the training set, enabling out-of-distribution generation. This novel capability is demonstrated with the same model trained on PMD, using three different retrieval databases: *(i) A stickers database* presented in Sec. 4. (ii) *Aesthetic database:* This database is constructed by filtering images according to a classifier score. Let C be a classifier that for each image $i \in I$ outputs a score $s = C(i)$. This classifier enables filtering the kNN using $L \leq s < H$, where L and H are low and high thresholds, respectively. We demonstrate this capability using an open source pre-trained aesthetics classifier A (Christoph Schuhmann, 2022): For each text input $t \in T$, we apply A on the kNN, and then divide the kNN into five equal quantiles based on A score. As can be seen in Fig. 8, using kNN with higher aesthetics score result in generated images with higher aesthetics mean score. *(iii) Image search engine:* Generative models are stationary in the sense that they are unable to learn new concepts after being trained, hence fine-tuning is required to represent

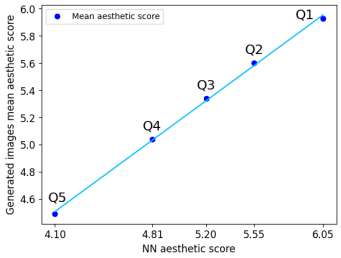


Figure 8: Mean aesthetics score of the generated images as a function of the conditioned kNN mean aesthetics score. As can be seen, the generation can be controlled by filtering the kNN.

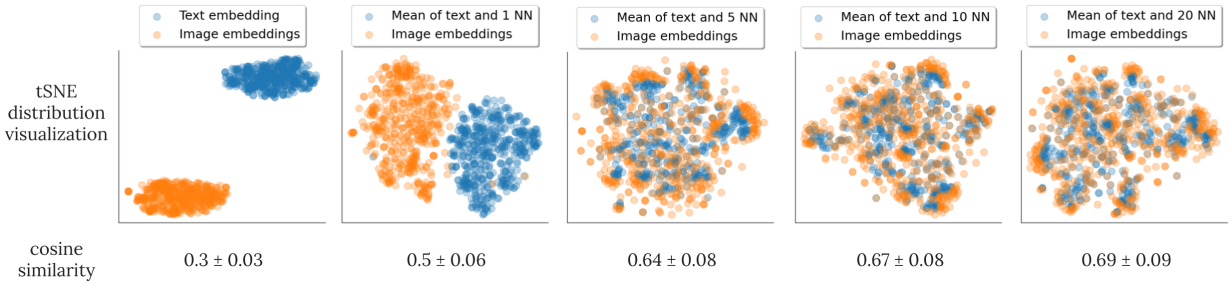


Figure 10: ***tSNE* visualization of 500 random (text, image) CLIP embeddings pairs taken from MS-COCO validation set.** As can be seen in the leftmost figure, there is a gap between the text and image distributions. By gradually adding kNN to the mean CLIP embedding of the text, the gap decreases, demonstrating the importance of the kNN in bridging the distributions. We also report the mean cosine similarity between the distributions. *tSNE* was calculated with perplexity of a 100 and 5000 iterations.

new styles and concepts. Here, we use an online image search engine, which allows the model to adapt to new data without additional fine-tuning. A qualitative comparison of all three methods is shown in Fig.7.

4.3 Ablation Study

We conclude our experiments with an ablation study, to quantify the contribution of our different components. The experiments are performed on the PMD dataset, and FID scores are reported on the MS-COCO dataset. We provide ablation analysis on index size, different kNN conditioning approaches and the model scalability in Sec. 6.4 of the supplement.

Number of nearest neighbors. The results in Fig. 9 demonstrate the importance of applying the retrieval mechanism during training and inference. Here, we evaluate our *kNN-Diffusion* model, trained on PMD dataset, with different numbers of kNN during inference - $1, 5, 10, 20, 100, 1000$. Furthermore, we examined the baseline *no-kNN*, in which during inference, the model is conditioned only on the text input embedding $f_{txt}(t)$, without using kNN. As can be seen, best performance is achieved using 10 neighbors.

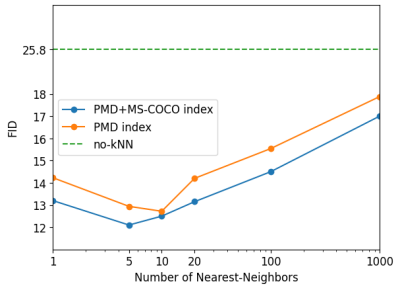


Figure 9: MS-COCO test FID score on various K's in two settings: (1) Zero-Shot (2) Index includes MS-COCO train subset. **No index inference** was trained with kNN, but did not employ kNN in inference.

5 Conclusion

"We shall always find, that every idea which we examine is copied from a similar impression", Hume (1748). In this paper, we propose using a large-scale retrieval method in order to train a novel text-to-image model, with only pre-trained multi-modal embeddings, but without an explicit text-image dataset.. Our extensive experiments demonstrate that using an external knowledge-base alleviates much of the model's burden of learning novel concepts, enabling the use of a relatively small model. In addition, it provides the model the capability of learning to adapt to new samples, which it only observes during test time. Lastly, we present a new technique utilizing the retrieval method for text-driven semantic manipulations without user-provided masks. As evaluated by human studies and automatic metrics, our method is significantly preferable to the baselines in terms of image quality and text alignment.

References

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022a.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18208–18218, 2022b.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Artem Babenko and Victor Lempitsky. The inverted multi-index. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1247–1260, 2014.

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *arXiv preprint arXiv:2204.02491*, 2022.

Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari, and Jordan Pascual. Knn based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, 7(1):61–70, 2014.

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.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Romain Beaumont Christoph Schuhmann. Aesthetic predictor. `https://github.com/LAION-AI/aesthetic-predictor`, 2022.

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*, 2022.

Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

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*, pp. 12873–12883, 2021.

Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1858–1865, 2008.

Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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*, abs/2111.14822, 2021.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

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. *Advances in neural information processing systems*, 30, 2017.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

David Hume. An enquiry concerning human understanding, 1748.

Karim Iskakov. Semi-parametric image inpainting. *arXiv preprint arXiv:1807.02855*, 2018.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2426–2435, 2022.

Ivan Krasin, Tom Duerig, Neil Alldrin, Andreas Veit, Sami Abu-El-Haija, Serge Belongie, David Cai, Zheyun Feng, Vittorio Ferrari, and Victor Gomes. Openimages: A public dataset for large-scale multi-label and multi-class image classification., 01 2016.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL `https://arxiv.org/abs/1602.07332`.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*, 2019.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Xingchao Liu, Chengyue Gong, Lemeng Wu, Shujian Zhang, Hao Su, and Qiang Liu. Fusedream: Training-free text-to-image generation with improved clip+ gan space optimization. *arXiv preprint arXiv:2112.01573*, 2021.

Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. *arXiv preprint arXiv:2112.12750*, 2021.

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.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL `https://proceedings.neurips.cc/paper/2011/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf`.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094, 2021.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *European Conference on Computer Vision*, pp. 647–664. Springer, 2020.

Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8808–8816, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL `https://arxiv.org/abs/2103.00020`.

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*, pp. 8821–8831. PMLR, 2021.

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.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

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*, pp. 12568–12577, 2021.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. *arXiv preprint arXiv:2112.04482*, 2021.

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

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2443–2449, 2021.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. *CoRR*, abs/1503.01817, 2015. URL `http://arxiv.org/abs/1503.01817`.

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*, pp. 242–257. Springer, 2020.

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

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.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1905–1914, 2021.

Zihao Wang, Wei Liu, Qian He, Xinglong Wu, and Zili Yi. Clip-gen: Language-free training of a text-to-image generator with clip. *arXiv preprint arXiv:2203.00386*, 2022.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *arXiv preprint arXiv:2203.08913*, 2022.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1316–1324, 2018.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 833–842, 2021.

Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. LAFITE: towards language-free training for text-to-image generation. *CoRR*, abs/2111.13792, 2021. URL `https://arxiv.org/abs/2111.13792`.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5802–5810, 2019.

6 APPENDIX

6.1 BACKGROUND

Continuous diffusion process Diffusion models are latent variable models that aim to model a distribution $p_\theta(x_0)$ that approximates the data distribution $q(x_0)$. Specifically, they model a forward process in the space of x_0 from data to noise. Given a sample from the data distribution $x_0 \sim q(x_0)$, this process produces a Markov chain of latent variables $x_1, \ldots, x_T$ by progressively adding Gaussion noise to the sample:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathcal{I}) \tag{1}$$

where β_t is a variance schedule. As presented previously by (Ho et al., 2020), the latent variable x_t can be expressed directly as a linear combination of noise and x_0:

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \epsilon\sqrt{1-\overline{\alpha}_t}, \quad \epsilon \sim \mathcal{N}(0, \mathcal{I}) \tag{2}$$

where $\alpha_t := \Pi_{i=1}^{t}(1-\beta_i)$. In order to sample from the data distribution $q(x_0)$, we define the "reverse process" $p(x_{t-1}|x_t)$ which samples first from $q(x_T)$ and then samples reverse steps $q(x_{t-1}|x_t)$ until x_0.

Since the data distribution is unknown, we need to train a model to approximate it. Note that when T is large enough, the noise vector x_T nearly follows an isotropic Gaussian distribution. This suggests learning a model $p_\theta(x_{t-1}|x_t)$ to predict mean μ_θ and covariance matrix Σ_θ:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{3}$$

To train this model, we can replace $\mu_\theta(x_t, t)$ by predicting the noise $\epsilon_\theta(x_t, t)$ added to x_0 using equation 2 and we get this objective function:

$$L := E_{t\sim[1,T], x_0\sim q(x_0), \epsilon\sim\mathcal{N}(0,\mathbf{I})}[||\epsilon - \epsilon_\theta(x_t, t, y||^2] \tag{4}$$

where y is an optional conditioning signal (such as text/image embedding or a low resolution image).

Discrete diffusion process Let $x_n \in \{1, \ldots, V\}^{h\times w}$ be the indices of the allocated codebook vectors extracted by a pre-trained VQGAN (Esser et al., 2021) encoder. The forward process of a diffusion model $q(x_n|x_{n-1})$ is a Markov chain that adds noise at each step. Moreover, the reverse process $q(x_{n-1}|x_n, x_0)$, is a denoising process that removes noise from an initialized noise state.As presented by (Gu et al., 2021), the forward diffusion process is given by:

$$q(x_n|x_{n-1}) = v^T(x_n)\mathbf{Q}_n v(x_{n-1}) \tag{5}$$

where $v(x_n)$ is a one-hot vector with entry 1 at x_n, and $\mathbf{Q}_n$ is the probability transition matrix from state x_{n-1} to x_n.

The reverse process is given by the posterior distribution:

$$q(x_{n-1}|x_n, x_0) = \frac{(v^T(x_n)\mathbf{Q}_n v(x_{n-1}))(v^T(x_{n-1})\bar{\mathbf{Q}}_{n-1}v(x_0))}{v^T(x_n)\bar{\mathbf{Q}}_n v(x_0)} \tag{6}$$

where $\bar{\mathbf{Q}}_n = \mathbf{Q}_n \cdots \mathbf{Q}_1$.

Inspired from mask language modeling (Devlin et al., 2018), they proposes corrupting the tokens by stochastically masking some of them. Specifically, an additional special token $[MASK]$ is proposed, so for each token there are (V+1) discrete states. The transition matrix is formulated as, By adding a small amount of unifrom noise to the categorial distribution, the transition matrix can be formulated as,

$$\mathbf{Q}_n = \begin{bmatrix} \alpha_n+\beta_n & \beta_n & \beta_n & \cdots & 0 \\ \beta_n & \alpha_n+\beta_n & \beta_n & \cdots & 0 \\ \beta_n & \beta_n & \alpha_n+\beta_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_n & \gamma_n & \gamma_n & \cdots & 1 \end{bmatrix} \tag{7}$$

where $\alpha_n \in [0, 1]$, $\beta_n = (1-\alpha_n-\gamma_n)/V$ and γ_n the probability of a token to be replaced with a $[MASK]$ token. Each token has a probability of γ_n to be replaced by the $[MASK]$ token, $V\beta_n$ to be resampled uniformly and $\alpha_n = (1 - V\beta_n - \gamma_n)$ to be unchanged.

6.2 Human Evaluation Protocol

For all of our human evaluation experiments, we used Amazon Mechanical Turk. For each experiment, we used 600 samples, each scored by five different people. The preferred sample was determined according to majority opinion. For each baseline comparison, we asked two questions (in different experiments): "Which image is of a higher quality?" and "Which image best matches the text?".

6.3 Datasets

The modified PMD dataset is composed of the following set of publicly available text-image datasets: SBU Captions (Ordonez et al., 2011), Localized Narratives (Pont-Tuset et al., 2020), Conceptual Captions (Sharma et al., 2018), Visual Genome (Krishna et al., 2016), Wikipedia Image Text (Srinivasan et al., 2021), Conceptual Captions 12M (Changpinyo et al., 2021), Red Caps (Desai et al., 2021), and a filtered version of YFCC100M (Thomee et al., 2015). In total, the dataset contains 69 million text-image pairs.

6.4 Ablation study

Index size As one can expect, increasing the index size at inference time improves performance. To demonstrate this hypothesis, we evaluated our model with an index containing 10%, 30%, 50% and 70% images of PMD dataset, and obtained FID scores of 13.92, 13.85, 13.72, and 13.65 respectively.

kNN conditioning We examined several different approaches to kNN input conditioning: (i) forwarding the kNN embeddings and the single image embedding through a self-attention layer before feeding the contextualized $K + 1$ embeddings to the model, (ii) feeding the model with one embedding, computed using cross-attention between the image embedding and the kNN embeddings, and, (iii) feeding the model with the image embedding concatenated with a learned linear projection of the kNN embeddings. These variants received FID scores of 18.3, 22.4, 34.1 respectively.

Scalability analysis As can be seen in Fig. 6, leveraging retrieval method as an integral part of the diffusion process enables the use of a relatively small model. In particular, we can observe that our 400M parameters model (1) achieves comparable results to recent state-of-the-art models trained with text-image pairs and without kNN (2) outperforms models trained with the same number of parameters, with text-image pairs, and without kNN, and (3) outperforms by a large margin recent models trained on an image-only dataset and without kNN. To further demonstrate this observation, we trained a small model of 250M parameters with kNN (***Ours-mini***) and compared it to our 400M parameters model trained without kNN (*no-kNN*). Both of the models were trained without text. While *Ours-mini* achieves FID score of 18.2, *no-kNN* achieves a score of 32.8. As can be seen, it is even lower than the FID (19.75) of (Gu et al., 2021) that was trained with 370M parameters, but with text-image pairs.

6.5 Retrieval Model

The retrieval model is implemented using FAISS (Johnson et al., 2019). FAISS is an efficient database, capable of storing billions of elements and finding their nearest neighbors in milliseconds. In the pre-processing phase, for each image in the dataset, we store the image index and its corresponding CLIP image embedding. During training, given a training image, we extract its CLIP image embedding and search for its 10 (see Fig. 9) nearest neighbors in the dataset based on the cosine similarity distance.

For an efficient search during training and inference, we use a non-exhaustive search: For this, we use an inverted file index. As in Babenko & Lempitsky (2014), we define Voronoi cells in the d-dimensional space (where $d = 512$ is the CLIP embedding dimensional space), s.t each database vector falls in one of the cells. During search time, only the embeddings contained in the cell the query falls in and a few neighboring ones are compared against the query vector. In addition, to fit the index of our large-scale datasets on a 128GB RAM server, we compress the code size from $512 \times 32/8 = 2048$ Bytes to 256 Bytes using optimized product quantization (Ge et al., 2013; Jegou et al., 2010). In Algorithm 1 we include pseudocode of the core of the implementation of the retrieval database.

6.6 Discrete kNN model

We provide additional implementation details for the discrete diffusion model. Additional training details can be found in Tab. 3.

Vector Quantization For token quantization, we use VQ-VAE and adapt the publicly available VQGAN(Esser et al., 2021) model, trained on the OpenImages(Krasin et al., 2016) dataset. The encoder downsamples images to 32×32 tokens and uses a codebook vocabulary with 2887 elements.

Image Tokenization In our discrete generative model we model images as a sequence of discrete tokens. To this end, we utilize a vector-quantized variational auto-encoder (VQ-VAE) (Van Den Oord et al., 2017) as image tokenizer. VQ-VAE consists of three components: (i) an encoder, (ii) a learned codebook, and, (iii) a decoder. Given an image, the encoder extracts a latent representation. The codebook then maps each latent vector representation to its nearest vector in the codebook. Finally, the decoder reconstructs the image from the codebook representation. VQ-VAE is trained with the objectives of reconstruction and codebook learning. VQ-GAN (Esser et al., 2021) adds an adversarial loss term that tries to determine whether the generated image is fake or real. This added term was shown to improve reconstruction quality.

Transformer We follow Gu et al. (2021) and train a decoder-only Transformer. The decoder module contains 24 transformer blocks, each containing full attention, cross-attention for the concatenated conditioner, and a feed-forward network. The timestamp n is injected using Adaptive Layer Normalization (Ba et al., 2016). The decoder contains 400 million parameters.

Classifier-free guidance We sample our diffusion models using classifier-free guidance (CFG) (Ho & Salimans, 2021; Nichol et al., 2021; Ramesh et al., 2022). CFG is performed by extrapolating an unconditional sample in the direction of a conditional sample. To support unconditional sampling, previous work had to fine-tune (Nichol et al., 2021) their models with 20% of the conditional features nullified. This enabled them to sample unconditional images from the model using the null condition, $y' = \overrightarrow{0}$, the null vector. We found that we can generate unconditional samples from our model using null conditioning without fine-tuning it. We hypothesize that by conditioning the model on a null vector, the cross-attention component is also nullified, resulting in no contribution to the diffusion process. During inference, in each step of the diffusion process we generate two images: conditional image logits, $p_\theta(x_{n-1}|x_n, y)$, conditioned on the desired multi-modal embedding y, and the unconditional image logits, $p_\theta(x_{n-1}|x_n, y')$, conditioned on the null embedding. Then, the final image for a diffusion step n is sampled from

$$\begin{aligned} p_\theta(x_{n-1}|x_n, y) =& p_\theta(x_{n-1}|x_n, y') + \\ & \lambda(p_\theta(x_{n-1}|x_n, y) - p_\theta(x_{n-1}|x_n, y')) \end{aligned}$$

where λ is a scale coefficient. In all of our experiments, we set $\lambda = 8$, which was found to yield the highest FID scores on the validation set. Note that the above extrapolation occurs directly on the logits output by p_θ, in contrast to GLIDE (Nichol et al., 2021), which extrapolates the pixel values.

Training Objective For completeness we are adding the training objective of the discrete model. The network is trained to minimize the variational lower bound (VLB):

$$\begin{aligned} \mathcal{L}_{\text{vlb}} &= \mathcal{L}_0 + \mathcal{L}_1 + \cdots + \mathcal{L}_{N-1} + \mathcal{L}_N, \\ \mathcal{L}_0 &= -\log p_\theta(x_0|x_1, f_{img}(I), \text{knn}_{img}(\text{I}, k)), \\ \mathcal{L}_{n-1} &= D_{KL}(q(x_{n-1}|x_n, x_0) \,||\, p_\theta(x_{n-1}|x_n, f_{img}(I), \text{knn}_{img}(\text{I}, k))), \\ \mathcal{L}_N &= D_{KL}(q(x_N|x_0) \,||\, p(x_N)) \end{aligned} \tag{8}$$

Where $p(\boldsymbol{x}_N)$ is the prior distribution of timestep $N = 100$, $f_{img}(I)$ is the CLIP image embedding, $\text{knn}_{img}(\text{I}, k)$ is the k nearest neighbors in the feature space of the image embedding. The full details can be found in Gu et al. (2021).

6.7 Continuous kNN model

We provide additional implementation details for the continuous diffusion model. Additional training details can be found in Tab. 3.

Decoder. We followed (Nichol et al., 2021; Ho et al., 2020; Ramesh et al., 2022) and re-implemented a diffusion *U-net* model. Specifically, we modify the architecture described in (Ramesh et al., 2022) by allowing multiple CLIP embeddings as the condition to the model. Since we do not have a paired text-image dataset, we removed the text transformer, and thus the text embedding. In particular, we use 512 convolution channels, 3 residual blocks, 64 heads

channels and attention resolution of 32, 16 and 8. Similarly to our discrete model, we trained two models (1) a *no-kNN* conditioned only on CLIP image embedding during training, (2) a kNN conditioned on CLIP image embedding and its kNN. Finally, we enable classifier-free guidance by randomly setting the CLIP embeddings to zero 10% of the time. As demonstrated in Tab. 2, we find that humans prefer our model over *no-kNN* 66.8% of the time for image quality and 69.4% of the time for text alignment.

Super-Resolution. As the decoder generates images with 64×64 resolution, we up-sampled the images to 256×256 using the open-source super resolution of (Nichol et al., 2021). To further up-sample the images to 512×512 and 1024×1024 we used the open-source super resolution provided by (Wang et al., 2021).

Training Objectives For completeness we are addding the training objective of our continuous model. Following Ho et al. (2020); Nichol et al. (2021) we are using mean-squared error loss to predict the noise:

$$L := E_{n \sim [1,N], x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0,\mathbf{I})}[||\epsilon - \epsilon_\theta(x_n, n, y)||^2]$$

where ϵ_θ is a $U - net$ model and $y = (f_{img}(x_0), \text{knn}_{img}(x_0, k))$.

Algorithm 1 Pseudo-code implementation for the construction of the retrieval model, training and sampling using conditioning kNN.

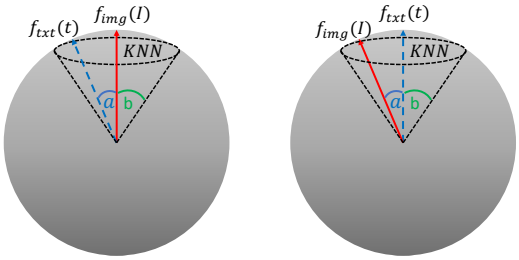
```
Retrieval model construction
def training(dataset: train image dataset):
    //inverted index of 50k centroids, with optimized product quantization to 256B
    index_cfg = "OPQ256_IVF50000_PQ256x8"
    index = faiss.index_factory(d, index_cfg, faiss.METRIC_INNER_PRODUCT)
    ivf = faiss.extract_index_ivf(index)
    clustering_index = faiss.index_cpu_to_all_gpus(faiss.IndexFlatIP(d))
    ivf.clustering_index = clustering_index
    train_dataset = []
    for image in random.sample(dataset, 1000000):
        train_dataset.append(CLIP_image_embedding(image))
    index.train(train_dataset)
    for image in dataset:
        index.add(CLIP_image_encoder(image))
    return index

Training
def training(I: FAISS index, image, k: Number of NN, t: timestamp [0, T-1]):
    image_encoding = CLIP_image_encoder(image)
    kNN = I.search(image_encoding, k)
    condition = concatenate([image_encoding, kNN])
    image_T = add_noise(image, t)
    image_0 = diffusion_model(image_T, t, condition)
    loss = criterion(image0, image)
    return loss

Sampling
def sampling(I: FAISS index, text, k : Number of NN):
    text_encoding = CLIP_text_encoder(text)
    kNN = I.search(text_encoding, k)
    condition = concatenate([text_encoding, kNN])
    image = sample_noise(T)
    for t in [T-1, T-2, ..., 0]:
        image = diffusion_model(image, t, condition)
    return image
```

	Discrete	Continuous
Number of nearest neighbors	10	10
Diffusion steps	100	1000
Noise schedule	-	cosine
Sampling steps	100	250
Model size	400M	1B
Sampling variance method	-	analytic
Dropout	-	0.1
Weight decay	4.5e-2	-
Batch size	512	1600
Iterations	150K	500K
Learning rate	4.05-4	1.4e-4
optimizer	AdamW	AdamW
Adam β_2	0.96	0.9999
Adam ϵ	1.0e-8	1.0e-8
EMA decay	0.99	0.9999
warmup	5000	25000
GPUs	128 A100	200 A100

Table 3: Training details of our models



(a) Training **(b)** Inference

Figure 11: During training, only the image I is given (red), whereas during inference only the text t is given (blue). In order to bridge the gap between the two distributions during training, we leverage the K nearest neighbors that should have a large enough distribution (dashed cone) to cover the potential text embedding (i.e. $cos(b) < cos(a)$). During inference, the opposite is applied.

6.8 Text-only image manipulation

Our approach is illustrated in Fig. 12. Additional manipulation examples are provided in Figs. 19. The full comparison with the baselines is provided in Fig. 20 and 21. We also provide in Fig. 13 several examples for the process of the manipulated images construction.

6.9 Additional Samples

Additional samples generated from challenging text inputs are provided in Figs. Fig. 15, 18 and Fig. 17.

Sticker generation In Fig. 14 we present a visual comparison of our discrete model, trained on the stickers dataset with (1) the kNN extracted during inference, (2) the same model without using kNN in inference. As can be seen, the stickers generated by our model are better aligned to the corresponding text compared to the baselines. While the baselines fail with challenging prompts, our model produces high-quality images that align with the text, and composes multiple concepts correctly.

COCO Validation Set Comparison Fig. 16 presents a qualitative comparison with FuseDream (Liu et al., 2021), CogView (Ding et al., 2021) and VQ-Diffusion (Gu et al., 2021) on the COCO validation set. Note that both CogView and VQ-Diffusion have been trained on an Image-Text *paired* dataset, whereas our model was not trained on the COCO dataset, nor used it in the retrieval model.

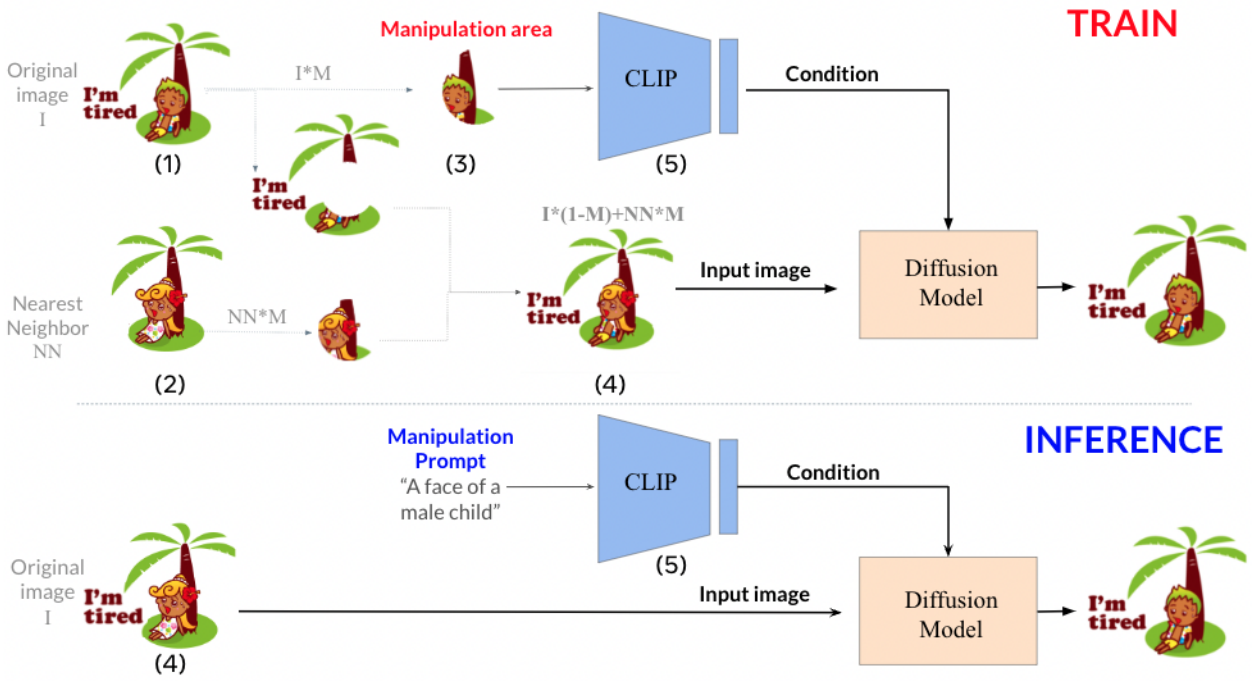


Figure 12: An illustration of our manipulation approach. **During training:** Given a training image (1), the model extracts its first nearest neighbor (2). Next, a random local area in the training image is selected (3), and the manipulated image is constructed by replacing the area with the corresponding nearest neighbor (4). The model then receives as input the manipulated image and the clip embedding of the local area that needs to be restored (5). **During inference:** Given an input image and a text query "A face of a male child", the model receives as input the image (4) and the clip embedding of the modifying text (5).

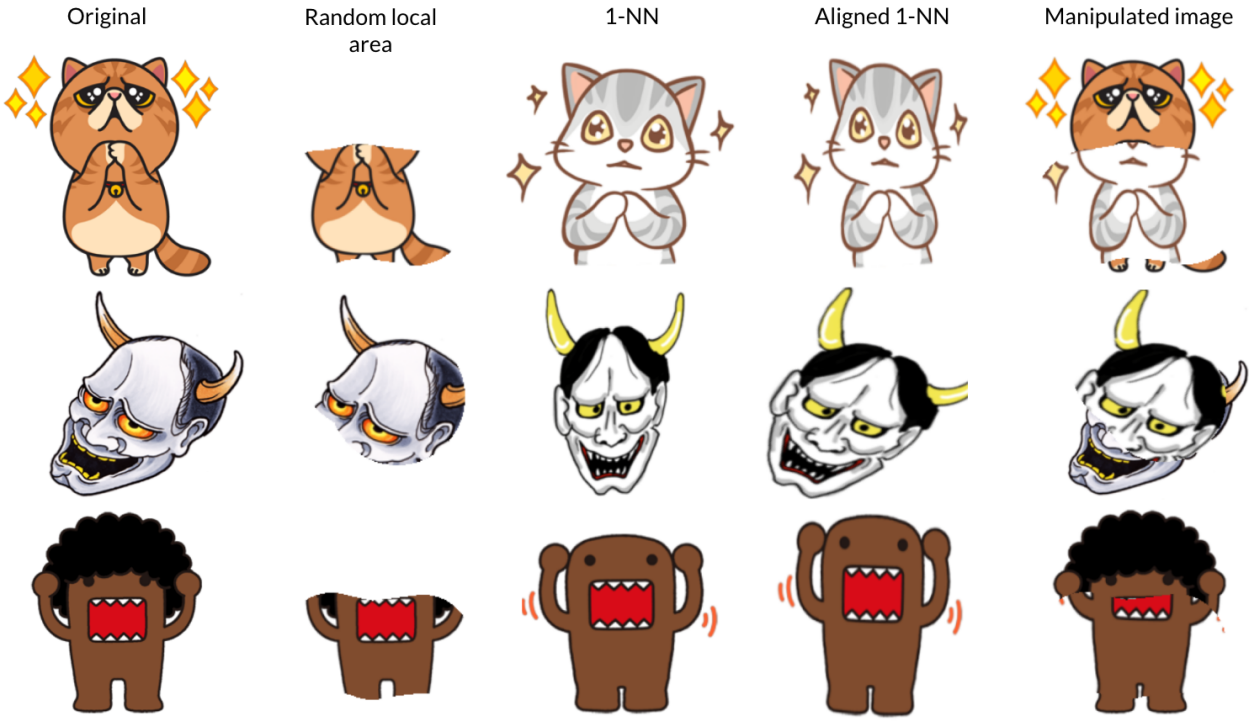


Figure 13: Illustration of the manipulated image construction process during training. Given an original image, we select a random local area, and extract the first nearest neighbor (1-NN). Using ECC alignment, we align the nearest neighbor with the original image and replace the random local area with it's corresponding nearest neighbor local area. The model then receives as input the manipulated image, together with the CLIP embedding of the local area, and tries to predict the original image.

Figure 14: Qualitative comparison of stickers generated using the discrete *kNN-Diffusion* model, 10 Nearest Neighbors to the text in the CLIP embedding and a discrete model that does not use kNN.

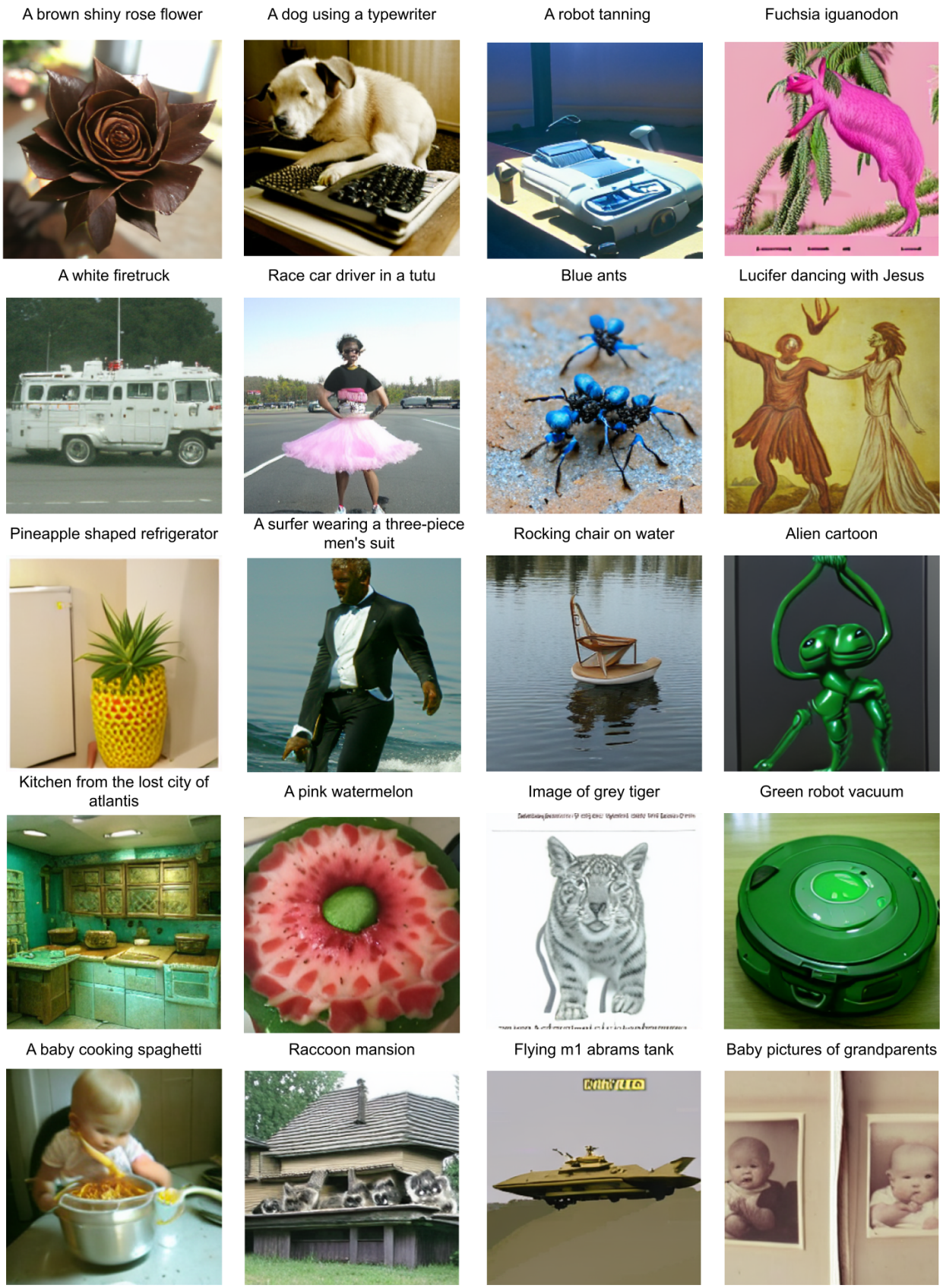


Figure 15: Additional samples generated from challenging text inputs using the photo-realistic model

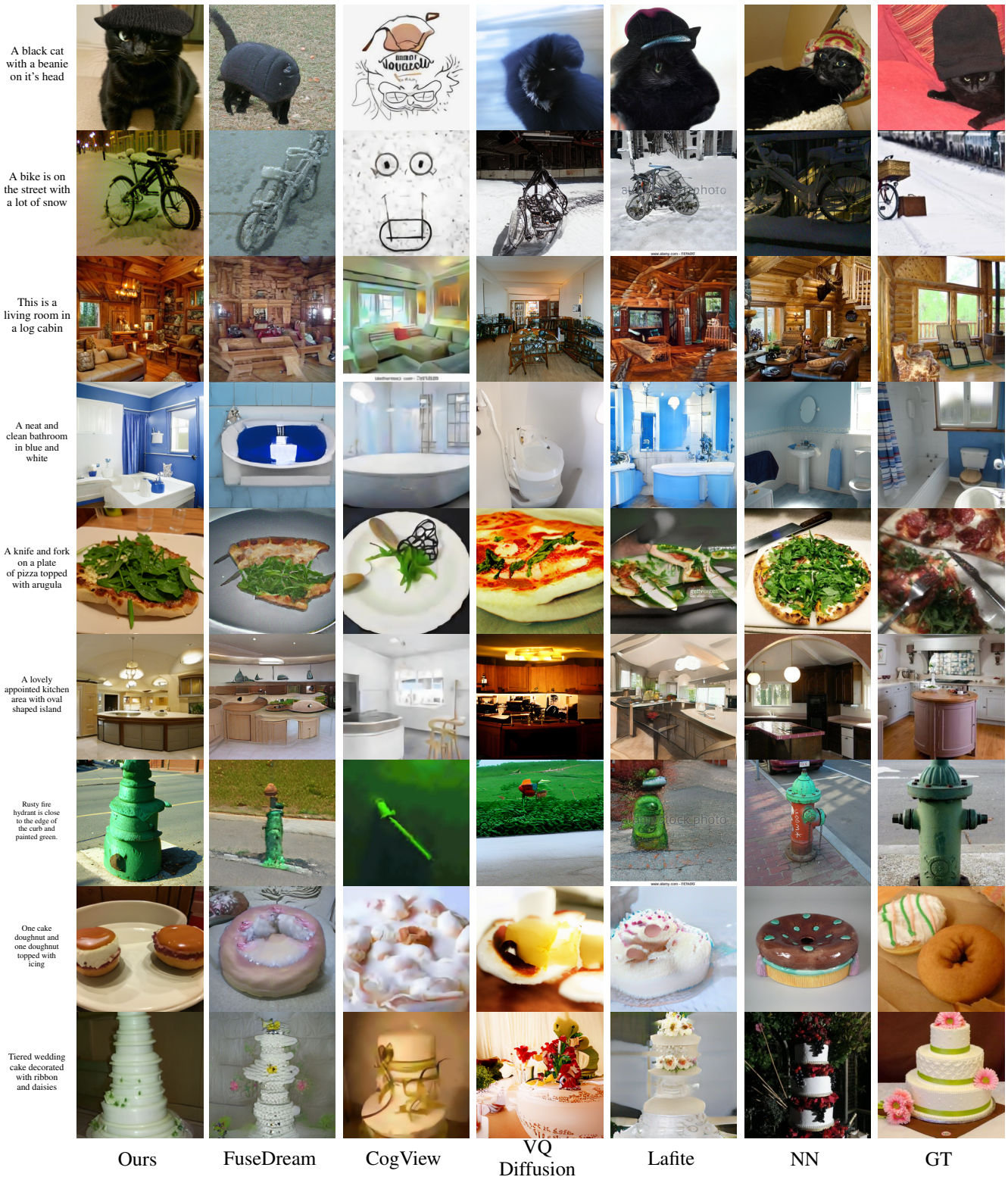


Figure 16: Samples from COCO validation set.

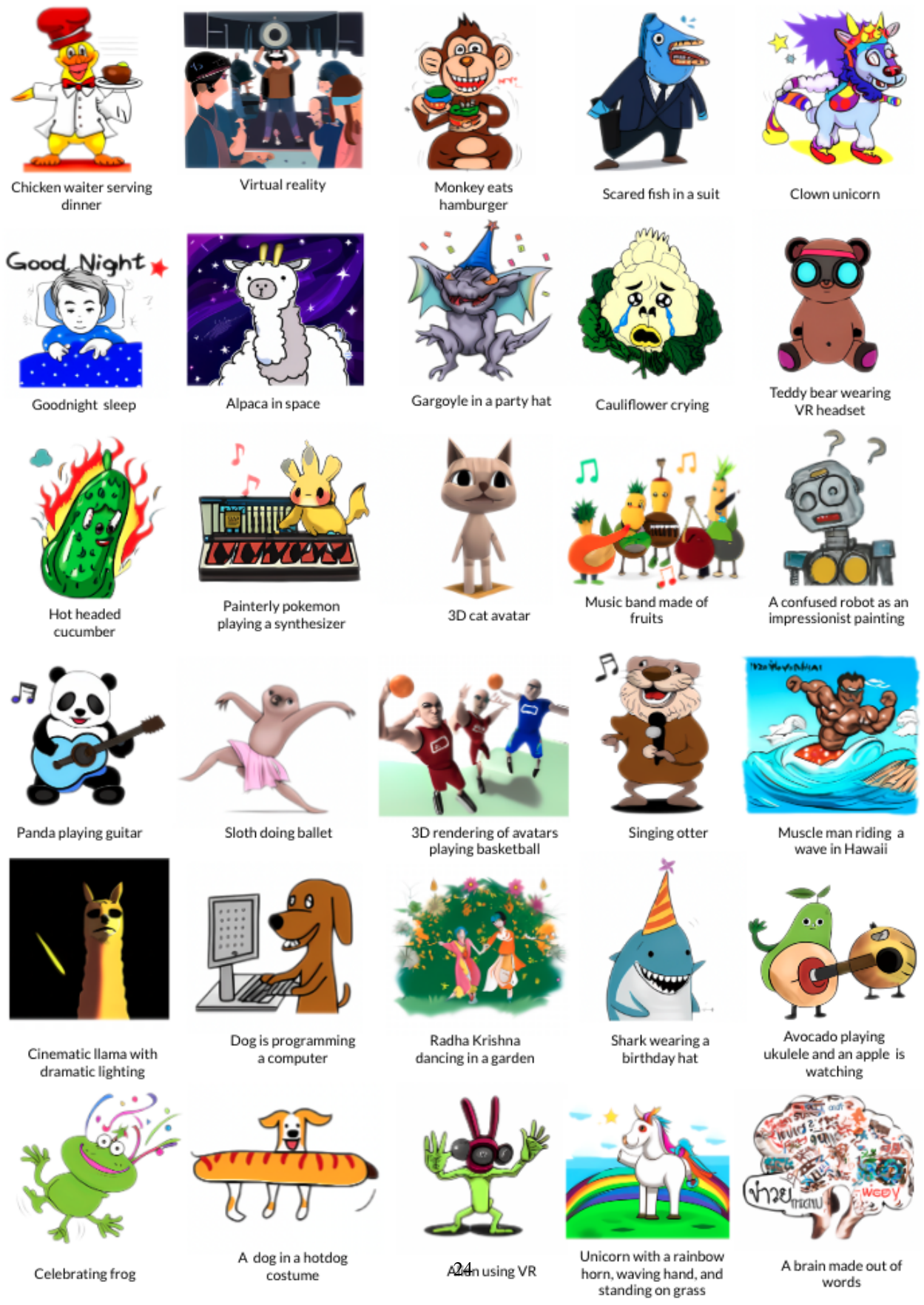


Figure 17: A selection of stickers generated using the continuous *kNN-Diffusion* model.

Figure 18: A selection of stickers generated using the discrete *kNN-Diffusion* model.

Figure 19: Additional manipulation examples, generated using our model.

Figure 20: comparison to Text2LIVE (Bar-Tal et al., 2022). For each input image, the bottom row corresponds to images generated by our model, and the top row corresponds to images generated by the Text2LIVE model.

Figure 21: comparison to Textual Inversion (Gal et al., 2022). For each input image, the bottom row corresponds to images generated by our model, and the top row corresponds to images generated by the Textual Inversion model.