One Transformer Fits All Distributions in Multi-Modal Diffusion at Scale

Fan Bao [1 2] **Shen Nie** [3 2] **Kaiwen Xue** [3 2] **Chongxuan Li** [3] **Shi Pu** [1 2] **Yaole Wang** [1]
Gang Yue [1 2] **Yue Cao** [4] **Hang Su** [1 5] **Jun Zhu** [1 2 5]

Abstract

This paper proposes a unified diffusion framework (dubbed *UniDiffuser*) to *fit all distributions relevant to a set of multi-modal data in one model*. Our key insight is – learning diffusion models for marginal, conditional, and joint distributions can be unified as predicting the noise in the perturbed data, where the perturbation levels (i.e. timesteps) can be different for different modalities. Inspired by the unified view, UniDiffuser learns all distributions simultaneously with a minimal modification to the original diffusion model – perturbs data in all modalities instead of a single modality, inputs individual timesteps in different modalities, and predicts the noise of all modalities instead of a single modality. UniDiffuser is parameterized by a transformer for diffusion models to handle input types of different modalities. Implemented on large-scale paired image-text data, UniDiffuser is able to perform image, text, text-to-image, image-to-text, and image-text pair generation by setting proper timesteps without additional overhead. In particular, UniDiffuser is able to produce perceptually realistic samples in all tasks and its quantitative results (e.g., the FID and CLIP score) are not only superior to existing general-purpose models but also comparable to the bespoken models (e.g., Stable Diffusion and DALL·E 2) in representative tasks (e.g., text-to-image generation). Our code is available at https://github.com/thu-ml/unidiffuser.

[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua-Huawei Joint Center for AI, BNRist Center, State Key Lab for Intell. Tech. & Sys., Tsinghua University [2]ShengShu, Beijing, China [3]Gaoling School of AI, Renmin University of China; Beijing Key Lab of Big Data Management and Analysis Methods, Beijing, China [4]Beijing Academy of Artificial Intelligence [5]Pazhou Laboratory (Huangpu), Guangzhou, China. Correspondence to: Jun Zhu <dcszj@tsinghua.edu.cn>.

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

1. Introduction

Recently, we are witnessing a content-creation revolution driven by the rapid advances of generative modeling on multi-modal data. In particular, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021c) have shown an incredible ability to create high-fidelity and diverse data (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022; Ho et al., 2022a; Popov et al., 2021), whose content aligns well with the input text condition.

However, these generative models are designed as bespoke systems, which only allow a single task. Actually, humans can generate various multi-modal content simultaneously, with arbitrary conditioning types. For example, artists can create paintings conditioned on texts, scenes, or just imagination and employ language ability to generate the caption of a photo. Toward a general generative system on multi-modal data, a unified training framework that can cover all types of multi-modal generative tasks (see Figure 1) is one of the fundamental components.

The task is solved by fitting a corresponding distribution in the view of probabilistic modeling. For instance, text-to-image generation can be formulated as learning the conditional distribution $p(\text{Image}|\text{Text})$. A classical way to fit all relevant distributions is implicit – it first learns the joint distribution and then infers the marginal and conditional distributions by additional procedures (e.g., Markov Chain Monte Carlo (Srivastava & Salakhutdinov, 2012)), which is unaffordable on large-scale multi-modal data (Schuhmann et al., 2022).

In contrast, this paper presents a diffusion-based framework (dubbed *UniDiffuser*) that *explicitly fits all relevant distributions in one model* without introducing additional training or inference overhead. Our key insight is – learning diffusion models for all distributions can be unified as predicting the noise in the perturbed data, where the perturbation levels (i.e. timesteps) can be different for different modalities. For instance, a zero level indicates conditional generation given the corresponding modality, and a maximum level indicates unconditional generation of other modalities by ignoring the corresponding modality. Inspired by the unified view, UniDiffuser learns all distributions simultaneously with a minimal modification to the original diffusion model (Ho

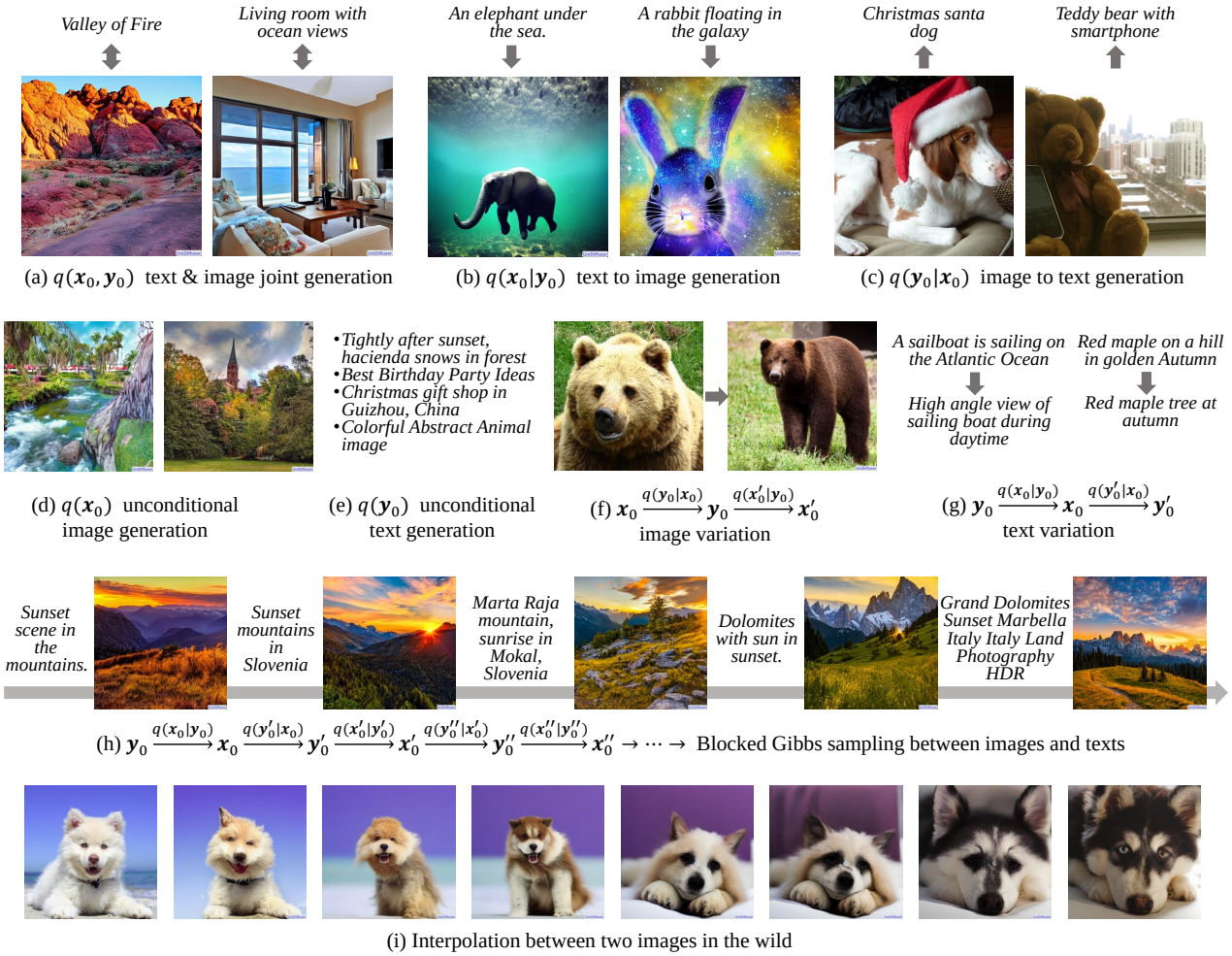


Figure 1. **UniDiffuser handles various tasks by fitting all distributions with one transformer.** (a-e) UniDiffuser can directly perform joint generation, conditional generation, and unconditional generation. (f-g) Image variation and text variation are direct applications by leveraging two conditional distributions modeled by UniDiffuser. (h) Furthermore, UniDiffuser can perform blocked Gibbs sampling to see how images and texts are translated to each other. (i) UniDiffuser can also perform interpolation between two images in the wild.

et al., 2020) (see Figure 2) – perturbs data in all modalities instead of a single modality, inputs individual timesteps in different modalities, and predicts the noise of all modalities instead of a single modality. Naturally, UniDiffuser is able to perform all kinds of generation (see Figure 1) in the same way as bespoken diffusion models. Moreover, UniDiffuser can perform the classifier-free guidance (Ho & Salimans, 2021) for free to improve the sample quality in both conditional and joint generation because UniDiffuser already models marginal distributions.

Besides the probabilistic modeling framework, a unified architecture that can handle input types of different modalities is another fundamental component in a general generative system. Notably, the emergence of Transformer (Vaswani et al., 2017; Dosovitskiy et al., 2021) and its applications on generative modeling (Bao et al., 2023a) provide a promising solution to capture interactions between modalities. Naturally, UniDiffuser employs a transformer-based backbone.

We implement UniDiffuser in the latent space (Rombach et al., 2022) with an additional CLIP encoder (Radford et al., 2021) for images and GPT-2 (Radford et al., 2019) decoder for texts on large-scale image-text data (Schuhmann et al., 2022). UniDiffuser is able to perform image, text, text-to-image, image-to-text, and image-text pair generation by setting proper timesteps without additional overhead. In particular, UniDiffuser is able to produce perceptually realistic samples in all tasks and its quantitative results (e.g., the FID and CLIP score) are not only superior to existing general-purpose models but also comparable to the corresponding bespoken models (e.g., Stable Diffusion and DALL·E 2) in

representative tasks (e.g., text-to-image generation).

2. Background

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) perturb the data by gradually injecting noise to data $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, which is formalized by a Markov chain:

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}),$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{\alpha_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}),$$

where β_t is the noise schedule and $\alpha_t = 1 - \beta_t$.

The data can be generated by reversing this process, where the reverse transition $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is approximated by a Gaussian model $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}|\boldsymbol{\mu}_t(\boldsymbol{x}_t), \sigma_t^2 \boldsymbol{I})$. As shown by Bao et al. (2022b), the optimal mean under maximal likelihood estimation is

$$\boldsymbol{\mu}_t^*(\boldsymbol{x}_t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t]\right), \quad (1)$$

where $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\boldsymbol{\epsilon}^x$ is the standard Gaussian noise injected to $\boldsymbol{x}_t$. To estimate the conditional expectation $\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t]$, a noise prediction network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is adopted to minimize the regression loss as follows:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}^x} \|\boldsymbol{\epsilon}^x - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2, \quad (2)$$

where t is uniformly sampled from $\{1, 2, \ldots, T\}$ and $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}^x$. According to the property of the l_2 regression loss, the optimal noise prediction network satisfies $\boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t, t) = \mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t]$. Since Eq. (2) is also equivalent to the denoising score matching loss (Vincent, 2011), the optimal noise prediction network also satisfies $\boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t, t) = -\sqrt{\overline{\beta}_t}\nabla \log q(\boldsymbol{x}_t)$, where $q(\boldsymbol{x}_t)$ is the distribution of the perturbed data at timestep t.

Conditional generation with diffusion models. In the case of conditional generation, we have paired data $(\boldsymbol{x}_0, \boldsymbol{y}_0) \sim q(\boldsymbol{x}_0, \boldsymbol{y}_0)$, and we want to model the conditional data distribution $q(\boldsymbol{x}_0|\boldsymbol{y}_0)$. The Gaussian model of the reverse process conditioned on $\boldsymbol{y}_0$ is $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}|\boldsymbol{\mu}_t(\boldsymbol{x}_t, \boldsymbol{y}_0), \sigma_t^2 \boldsymbol{I})$. Similarly to Eq. (1), the optimal mean under maximal likelihood estimation is

$$\boldsymbol{\mu}_t^*(\boldsymbol{x}_t, \boldsymbol{y}_0) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t, \boldsymbol{y}_0]\right). \quad (3)$$

To estimate $\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t, \boldsymbol{y}_0]$, a noise prediction network conditioned on $\boldsymbol{y}_0$ is adopted to minimize the regression loss

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{y}_0,\boldsymbol{\epsilon}^x} \|\boldsymbol{\epsilon}^x - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t)\|_2^2.$$

Classifier-free guidance (CFG) (Ho & Salimans, 2021) is proposed to improve the sample quality of a conditional diffusion model. Specifically, it samples by linearly combining a conditional model and an unconditional one:

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t) = (1+s)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t) - s\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \quad (4)$$

where s is the guidance scale. The conditional and unconditional models share parameters by introducing a null token $\varnothing$, i.e., $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0 = \varnothing, t)$.

3. Method

Section 3.1 presents *UniDiffuser*, a single diffusion model to capture the marginal, conditional, and joint distributions determined by multi-modal data simultaneously. Section 3.2 demonstrates how to perform classifier-free guidance (CFG) for free in conditional and joint sampling of UniDiffuser. For simplicity, we focus on two-modal data in this paper but UniDiffuser can be easily extended to more modalities.

3.1. UniDiffuser: One Diffusion Fits All Distributions

Formally, suppose we have two modalities of data sampled from distribution $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$. We aim to design a diffusion-based model that is able to capture all relevant distributions determined by $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$, i.e., the marginal distributions $q(\boldsymbol{x}_0)$ and $q(\boldsymbol{y}_0)$, the conditional distributions $q(\boldsymbol{x}_0|\boldsymbol{y}_0)$ and $q(\boldsymbol{y}_0|\boldsymbol{x}_0)$, and the joint distribution $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$.

We notice that learning a distribution with diffusion models is equivalent to estimating a conditional expectation over the noise. In particular, modeling the marginal distribution $q(\boldsymbol{x}_0)$ is equivalent to estimating the conditional expectation of the noise injected to $\boldsymbol{x}_t$, i.e., $\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t]$, according to Eq. (1). Similarly, the key quantities to be estimated in modeling the conditional distribution $q(\boldsymbol{x}_0|\boldsymbol{y}_0)$ and the joint distribution $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$ are $\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_t, \boldsymbol{y}_0]$ (see Eq. (3)) and $\mathbb{E}[\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y|\boldsymbol{x}_t, \boldsymbol{y}_t]$ respectively.

A key observation is that all above conditional expectations can be unified in the general form of $\mathbb{E}[\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y|\boldsymbol{x}_{t^x}, \boldsymbol{y}_{t^y}]$, where t^x and t^y are two timesteps that can be different, and $\boldsymbol{x}_{t^x}$ and $\boldsymbol{y}_{t^y}$ are the corresponding perturbed data. In particular, a maximum timestep T means marginalizing it. Namely, by setting $t^y = T$, we have $\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_{t^x}, \boldsymbol{y}_T] \approx \mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_{t^x}]$[1], which corresponds to the marginal distribution $q(\boldsymbol{x}_0)$. Similarly, a zero timestep means conditioning on the corresponding modality and a tied timestep means sampling two modalities jointly. Formally, $\mathbb{E}[\boldsymbol{\epsilon}^x|\boldsymbol{x}_{t^x}, \boldsymbol{y}_0]$ corresponds to the conditional distribution $q(\boldsymbol{x}_0|\boldsymbol{y}_0)$ by setting $t^y = 0$ and $\mathbb{E}[\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y|\boldsymbol{x}_t, \boldsymbol{y}_t]$ corresponds to the joint distribution $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$ by setting $t^x = t^y = t$. Moreover, we can characterize $q(\boldsymbol{x}_0|\boldsymbol{y}_{t^y})$ and $q(\boldsymbol{y}_0|\boldsymbol{x}_{t^x})$ for all t^y and t^x and

[1] There is a negligible gap between $\boldsymbol{y}_T$ and the standard Gaussian noise $\boldsymbol{\epsilon}^y$ for a large T (e.g., 1000 by default (Ho et al., 2020)).

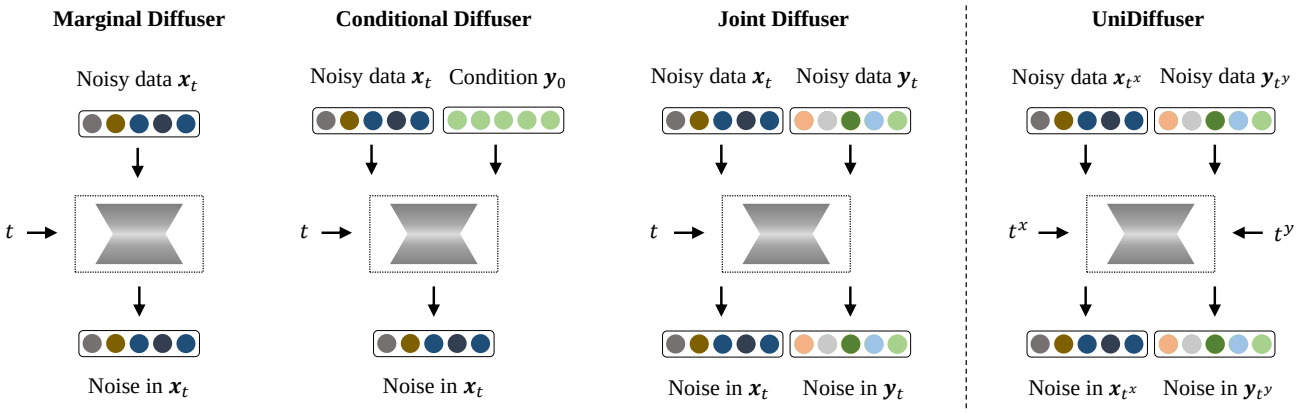


Figure 2. **Comparison with bespoken diffusers.** UniDiffuser fits all distributions simultaneously with a minimal modification of Ho et al. (2020). In particular, it degenerates to bespoken diffusion models by setting the timesteps (or noise levels) properly.

generate data conditioned on noisy input, by estimating $\mathbb{E}[\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y | \boldsymbol{x}_{t^x}, \boldsymbol{y}_{t^y}]$ in general.

Inspired by the unified view, we learn $\mathbb{E}[\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y | \boldsymbol{x}_{t^x}, \boldsymbol{y}_{t^y}]$ for all $0 \leq t^x, t^y \leq T$ to model all relevant distributions determined by $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$. Specifically, we employ a *joint noise prediction network*[2] $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t^x}, \boldsymbol{y}_{t^y}, t^x, t^y)$ to predict the noise injected to $\boldsymbol{x}_{t^x}$ and $\boldsymbol{y}_{t^y}$ together by minimizing the following regression loss similarly to Ho et al. (2020):

$$\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y, t^x, t^y} \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t^x}, \boldsymbol{y}_{t^y}, t^x, t^y) - [\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y]\|_2^2, \quad (5)$$

where $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is a random data point, $[,]$ denotes concatenation, $\boldsymbol{\epsilon}^x$ and $\boldsymbol{\epsilon}^y$ are sampled from standard Gaussian distributions, and t^x and t^y are uniformly sampled from $\{1, 2, \ldots, T\}$ independently. We call our method *UniDiffuser* because it captures multiple distributions in a unified way. We present the training algorithm in Appendix B.

The objective in Eq. (5) is as simple as the original DDPM in Eq. (2). Besides, for a single update of parameters, UniDiffuser only requires a single forward-backward calculation for multiple tasks (i.e., distributions), which is as efficient as the original DDPM. Although the gradient estimate of UniDiffuser has a slightly higher variance than the original DDPM due to two independent timesteps, we do not observe that UniDiffuser suffers from slower convergence.

UniDiffuser attempts to fit all distributions by one joint noise prediction network, requiring that the backbone can handle the mutual interaction between modalities and is scalable for large-scale data and multiple tasks. Inspired by the excellent performance of transformers on multi-modal representation learning at scale (Kim et al., 2021; Wang et al., 2022), we employ a transformer-based network in UniDiffuser, as detailed in Section 4.2.

[2]UniDiffuser can be easily reparameterized to data prediction or velocity prediction (Salimans & Ho, 2022) as well.

Given a single joint noise prediction network, UniDiffuser can perform unconditional, conditional, and joint sampling according to a certain sampler (see Appendix B for the sampling algorithm). Notably, by setting the timesteps properly, the inference procedure of UniDiffuser is the same as the bespoken models. In comparison, learning a single joint distribution (Srivastava & Salakhutdinov, 2012; Hu et al., 2022) over multi-modal data requires additional procedures (e.g., Markov Chain Monte Carlo) to sample from the marginal or conditional distributions, which is unaffordable on large-scale multi-modal data (Schuhmann et al., 2022).

3.2. Classifier-Free Guidance for Free

Classifier-free guidance (CFG) (Ho & Salimans, 2021) combines a conditional and an unconditional model linearly during sampling (see Eq. (4)). It is simple yet effective to improve the sample quality and image-text alignment in diffusion models. Notably, CFG is directly applicable to the conditional and joint sampling of UniDiffuser without modifying the training process (see Figure 3 for results).

Formally, we denote the output of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ as the concatenation of $\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}$ and $\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}$, i.e. $\boldsymbol{\epsilon}_{\boldsymbol{\theta}} = [\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}, \boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}]$, where we omit the input for simplicity. UniDiffuser can perform CFG for free in conditional sampling because it captures both the conditional and unconditional models. For example, we can generate $\boldsymbol{x}_0$ conditioned on $\boldsymbol{y}_0$ similarly to Eq. (4) as follows:

$$\hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t) = (1+s)\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t, 0) - s\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T),$$

where $\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t, 0)$ and $\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T)$ represent the conditional and unconditional models respectively, and s is the guidance scale. In contrast to the original CFG, UniDiffuser does not need to specify a null token for parameter sharing.

CFG is also applicable to joint sampling. By setting $t^x = t^y = t$, note that the joint score model can be equivalently

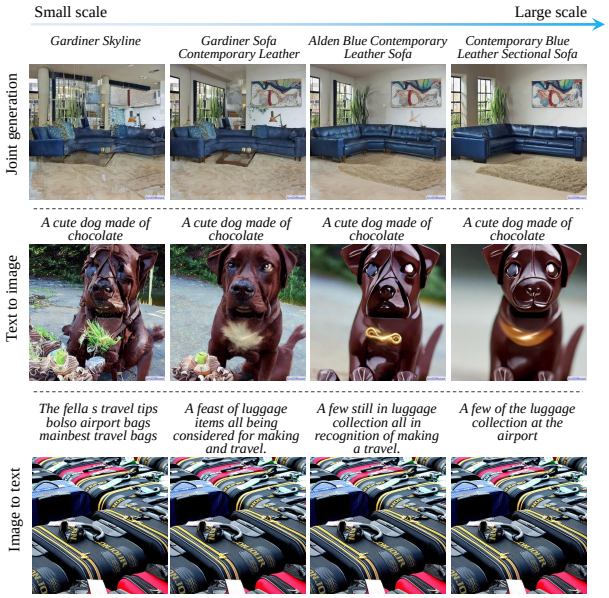


Figure 3. **Effects of CFG.** UniDiffuser employs CFG for free in joint and conditional sampling, improving the sample quality and image-text alignment with a large scale of around 6.

expressed in the form of conditional models as follows:

$$\begin{aligned}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t, t) \approx & -\sqrt{\bar{\beta}_t}[\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t, \boldsymbol{y}_t), \nabla_{\boldsymbol{y}_t} \log q(\boldsymbol{x}_t, \boldsymbol{y}_t)] \\ = & -\sqrt{\bar{\beta}_t}[\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{y}_t), \nabla_{\boldsymbol{y}_t} \log q(\boldsymbol{y}_t|\boldsymbol{x}_t)],\end{aligned}$$

where $q(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is the joint distribution of perturbed data at the same noisy level t. Inspired by the above relationship between score functions, $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t, t)$ can be viewed as approximating a pair conditional scores $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{y}_t)$ and $\nabla_{\boldsymbol{y}_t} \log q(\boldsymbol{y}_t|\boldsymbol{x}_t)$. In the same spirit of CFG, we can replace each conditional score by interpolating the joint model with the corresponding unconditional model as follows:

$$\begin{aligned}&\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t) \\ =&(1+s)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t, t) - s[\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T), \boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}^x, \boldsymbol{y}_t, T, t)] \\ \approx & -\sqrt{\bar{\beta}_t}[(1+s)\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{y}_t) - s\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t), \\ & \qquad (1+s)\nabla_{\boldsymbol{y}_t} \log q(\boldsymbol{y}_t|\boldsymbol{x}_t) - s\nabla_{\boldsymbol{y}_t} \log q(\boldsymbol{y}_t)],\end{aligned}$$

where $\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T)$ and $\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}^x, \boldsymbol{y}_t, T, t)$ represent unconditional models. We summarize the formulation of CFG in UniDiffuser for all tasks in Appendix C.

4. UniDiffuser on Images and Texts

Images and texts are two of the most common modalities in daily life. Thus, it is representative to validate the effectiveness of UniDiffuser on the two modalities.

Our implementation is two-staged following (Rombach et al., 2022) (see Figure 4). First, we convert images and texts to continuous latent embeddings $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ via image and text encoders and introduce two decoders for reconstruction, as presented in Section 4.1. Second, we train UniDiffuser parameterized by a transformer on the latent embeddings $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$, as presented in Section 4.2.

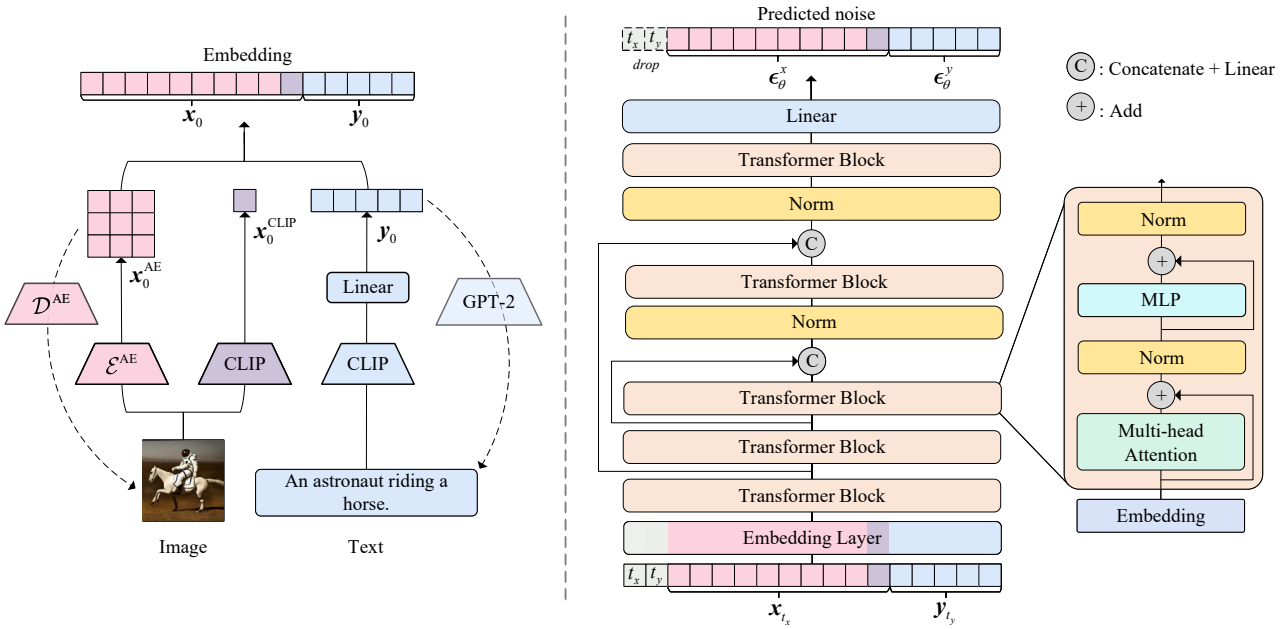
4.1. Encoding Images and Texts into Latent Space

The image and text encoder-decoders are illustrated in Figure 4 (a). Below we provide their details.

Image encoder-decoder. The image encoder consists of two parts. The first part is the image autoencoder employed in Stable Diffusion (Rombach et al., 2022). We use its encoder $\mathcal{E}^{\text{AE}}$ to obtain an embedding for image reconstruction $\boldsymbol{x}_0^{\text{AE}}$. The second part is the image CLIP (Radford et al., 2021) (ViT-B/32). It extracts a semantic embedding $\boldsymbol{x}_0^{\text{CLIP}}$ of dimension 512. The final latent embedding for images is the concatenation of the outputs from two parts, i.e., $\boldsymbol{x}_0 = [\boldsymbol{x}_0^{\text{AE}}, \boldsymbol{x}_0^{\text{CLIP}}]$. Empirically, we found that $\boldsymbol{x}_0^{\text{AE}}$ is sufficient for image reconstruction via the image decoder $\mathcal{D}^{\text{AE}}$ from Stable diffusion and the additional $\boldsymbol{x}_0^{\text{CLIP}}$ helps understand the semantics of images in image-to-text generation. We hypothesize that the different roles of the two embeddings are inherently caused by the original objectives, i.e. reconstruction versus semantics alignment with text.

Text encoder-decoder. As for the text encoder, we employ the same text CLIP as Stable Diffusion (Rombach et al., 2022). The text CLIP outputs 77 vectors and each is 768-dimensional. To facilitate training, we add an extra linear layer, which reduces the dimension of each vector to 64 to obtain the final text embedding $\boldsymbol{y}_0$. We construct the text decoder $\mathcal{D}^{\text{text}}$ based on GPT-2 (Radford et al., 2019). Specifically, GPT-2 takes $\boldsymbol{y}_0$ as a prefix embedding (Mokady et al., 2021) and reconstructs the text autoregressively. Freezing the parameters in CLIP, we train the linear layer and finetune GPT-2 to reconstruct the input texts, which performs well on reconstruction. We present more training details and the reconstruction results in Appendix E.

Remark. We observe that the latent embeddings of both image and text already have similar and reasonable numerical ranges. Specifically, they are concentrated within the range of $[-2, 2]$ and exhibit approximately normal distributions with comparable mean and variance values (image modality: mean = 0.0269, standard deviation = 0.7919; text modality: mean = 0.0127, standard deviation = 0.5957). As a result, we did not apply additional normalization to them. For more modalities, we can similarly convert them to continuous latent features through encoders that have regularization on the latent space. This makes it easy for all modalities to have similar ranges after normalization. Besides, obtaining high-quality encoders and decoders is relatively straightfor-



(a) Encode images & texts into latent space

(b) The U-ViT backbone of the joint noise prediction network

Figure 4. **Implementation of UniDiffuser on image-text data.** (a) First, we encode images and texts into latent space. (b) Second, we train UniDiffuser parameterized by a transformer (Bao et al., 2023a) in the way illustrated in Figure 2 on the latent embeddings.

ward and can be achieved with a smaller amount of data. For example, the dataset size of the image encoder and decoder is less than 1% of UniDiffuser's. Therefore, in practice, we can efficiently train high-quality encoders and decoders for each modality at a modest cost if needed.

4.2. Transformer as Joint Noise Prediction Network

We train a joint noise prediction network on the embeddings obtained in Section 4.1 according to Eq. (5). It is natural to employ a transformer-based backbone in UniDiffuser to handle inputs from different modalities. In particular, we adopt U-ViT (Bao et al., 2023a), a recently proposed transformer for conditional diffusion models. The original U-ViT is characterized by treating all inputs including the data, the condition, and the timestep as tokens, and employing long skip connections between shallow and deep layers. In UniDiffuser, we slightly modify U-ViT by treating the two modalities of data and their corresponding timesteps as tokens. Besides, we empirically find that the pre-layer normalization (Xiong et al., 2020) in the original U-ViT causes overflow easily when trained with mixed precision. A simple solution is to use the post-layer normalization (Vaswani et al., 2017) and add a layer normalization after concatenating a long skip connection, which stabilizes the training of UniDiffuser. We illustrate the backbone in Figure 4 (b) and present more details in Appendix D.

5. Related Work

Multi-modal generative modeling. Many prior work on multi-modal generative modeling can be formalized as learning a conditional distribution. Representative applications include text-to-image generation (Ramesh et al., 2021; Ding et al., 2021; Ramesh et al., 2022; Nichol et al., 2022; Saharia et al., 2022; Yu et al., 2022; Gu et al., 2022; Xu et al., 2018; Rombach et al., 2022), text-to-video generation (Ho et al., 2022a), text-to-speech generation (Chen et al., 2021; Popov et al., 2021) and image captioning (i.e., image-to-text generation) (Mokady et al., 2021; Chen et al., 2022). Such models are specially designed for a single task. In addition to learning a conditional distribution, Hu et al. (2022) aims to learn the joint distribution of image and text data via a discrete diffusion model (Gu et al., 2022). However, its scalability is unexplored.

The most related prior work is Versatile Diffusion (VD) (Xu et al., 2022), which employs a multi-flow architecture and is trained for multiple generation tasks in the traditional multi-task framework, which requires multiple feed-forward to compute losses for all tasks and carefully tuned gradient multipliers for different layers during training. In contrast, UniDiffuser provides an elegant solution based on the insightful unified view of training diffusion models. As a result, UniDiffuser is simpler (with a single training loss), more efficient to train (with a single forward-backward per

update), and can handle more tasks (able to perform joint sampling) without the need for complex tricks. Besides, UniDiffuser outperforms VD in both image-to-text and text-to-image generation tasks in terms of the FID and CLIP scores in our experiments (see Section 6), suggesting that the time-condition strategy in UniDiffuser is statistically more efficient than the multi-task one in VD.

Multi-modal representation learning aims to learn features for different modalities that can be transferred to downstream tasks. Vision-and-language pretraining (VLP) is at the front. VLP can employ different strategies, such as contrastive learning (Radford et al., 2021), masked data modeling (Wang et al., 2022), and a combination of multiple losses (Kim et al., 2021; Li et al., 2022; Bao et al., 2022c). A transformer is often employed to fuse the two modalities. This work implies that a transformer is also effective for multi-modal generative modeling.

Diffusion models are initially proposed by Sohl-Dickstein et al. (2015). Recently, Ho et al. (2020) introduce a noise prediction formulation, and Song et al. (2021c) introduce a stochastic differential equation formulation for learning diffusion models. Diffusion models are able to generate high-quality images (Dhariwal & Nichol, 2021),audios (Chen et al., 2021; Kong et al., 2021), videos (Ho et al., 2022b), point clouds (Luo & Hu, 2021) and molecular conformations (Hoogeboom et al., 2022; Bao et al., 2023b). Other improvements in diffusion models include fast sampling (Song et al., 2021a; Bao et al., 2022b; Salimans & Ho, 2022; Lu et al., 2022b;c) and improved training and sampling techniques (Nichol & Dhariwal, 2021; Song et al., 2021b; Kingma et al., 2021; Vahdat et al., 2021; Zhao et al., 2022; Bao et al., 2022a; Lu et al., 2022a; Karras et al., 2022).

6. Experiments

We present the experimental setup in Section 6.1. We show the ability of UniDiffuser to perform multiple generation tasks and directly compare it with existing large models in Section 6.2. We further demonstrate that UniDiffuser naturally supports applications like data variation, blocked Gibbs sampling between modalities (see Section 6.3), and interpolation between images in the wild (see Section 6.4).

6.1. Setup

Dataset. We use three subsets of LAION-5B (Schuhmann et al., 2022) following Stable Diffusion (Rombach et al., 2022). The first one is laion2B-en, which contains around 2B image-text pairs with English captions. The second one is laion-high-resolution, which contains around 170M image-text pairs with image resolution $\geq$1024 and multi-lingual captions. The third one is laion-aesthetics v2 5+, which is a subset of laion2b-en containing around 600M image-text pairs with high visual quality. Following Stable Diffusion, we additionally filter laion-aesthetics v2 5+ to images with resolution $\geq$512 and an estimated watermark probability $<$0.5, leading to around 193M preserved pairs. For image normalization, we follow the standard practice in diffusion models by normalizing the image values from the range of $[0, 255]$ to $[-1, 1]$. Since the texts in LAION-5B are quite noisy, we further clean texts in the laion-aesthetics v2 5+ subset by removing URLs, HTML tags, emails, contents in brackets, quotes except `'s`, and symbols except `, . ? !`. Before inputting the text into CLIP, we tokenize the preprocessed text using CLIP's built-in tokenizer, which is based on byte-level Byte-Pair-Encoding (Radford et al., 2021).

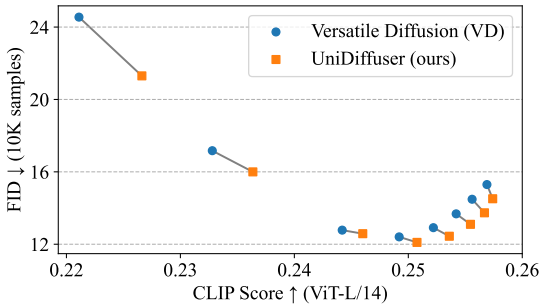


Figure 5. **Comparing UniDiffuser and VD in text-to-image generation.** We connect the results with the same scale in CFG. UniDiffuser consistently outperforms VD in all settings w.r.t. both the CLIP score ↑ (horizontal axis) and FID ↓ (vertical axis).

Training and Sampling. The training is multiple-staged following Stable Diffusion (Rombach et al., 2022). In the first stage, we train 250K steps at 256×256 resolution on laion2B-en with a batch size of 11264 and 5K warm-up steps. In the second stage, we fine-tune the model with 200K steps at 512×512 resolution on laion-high-resolution with a batch size of 2112 and 5K warm-up steps. In the last stage, we resume from the last checkpoint of the second stage (including both weights of the model and states of the optimizer), and train 220K steps at 512×512 resolution on laion-aesthetics v2 5+ with a batch size of 2112. Following Bao et al. (2023a), we use the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 2e-4, a weight decay of 0.03 and running coefficients of $(\beta_1, \beta_2) = (0.9, 0.9)$ in all stages. We reduce the learning rate by a factor of 10 and continue training whenever the validation loss does not decrease. We train with mixed precision for efficiency. When U-ViT is trained at 256×256 resolution, we interpolate the positional embeddings related to images via bilinear interpolation. The training takes around 28 days on 88 A100 (80GB) GPUs. We use DPM-Solver (Lu et al., 2022b;c) with 50 steps in all experiments.

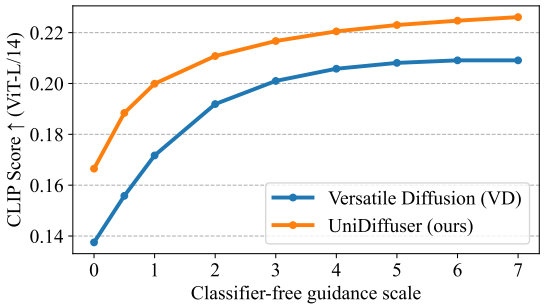


Figure 6. **Comparing UniDiffuser and VD in image-to-text generation.** UniDiffuser consistently outperforms VD with the same CFG scale (horizontal axis) w.r.t. the CLIP score ↑ (vertical axis).

Table 1. **Zero-shot FID ↓ on the MS-COCO validation set.** † marks results produced by us upon official implementation and other results are taken from the corresponding references. We report the results of UniDiffuser and VD with a scale of 3 in CFG, which is the best choice for both models according to Figure 5.

Model	FID ↓
Bespoken models	
GLIDE (Nichol et al., 2022)	12.24
Make-A-Scene (Gafni et al., 2022)	11.84
DALL·E 2 (Ramesh et al., 2022)	10.39
Stable Diffusion† (Rombach et al., 2022)	8.59
Imagen (Saharia et al., 2022)	7.27
Parti (Yu et al., 2022)	**7.23**
General-purpose models	
Versatile Diffusion† (Xu et al., 2022)	10.09
UniDiffuser (**ours**)	**9.71**

Baseline. To our knowledge, Versatile Diffusion (VD) (Xu et al., 2022) is the most direct competitor for general-purpose multi-modal generation (see details in Section 5). We directly compare to VD in all experiments if possible. The results of VD are reproduced by us upon official code because there is no quantitative result in the original paper.

Evaluation. For text-to-image generation, we report the FID (Heusel et al., 2017) and CLIP score (Radford et al., 2021) on the MS-COCO validation set (Lin et al., 2014) to measure the image fidelity and image-text alignment respectively. Following the literature, we randomly draw 10K and 30K prompts from the MS-COCO validation set to calculate FID and the CLIP score on generated images. For image-to-text generation, we report the CLIP score to measure the image-text alignment. Specifically, we randomly draw 10K images to calculate the score on generated texts.

Figure 7. **Random samples of UniDiffuser and VD on text-to-image generation.** UniDiffuser produces semantically correct images given representative prompts while VD does not.

6.2. Main Results

We first systematically compare with the most direct baseline Versatile Diffusion (VD), which is a general-purpose generative model, in both text-to-image and image-to-text generation. Quantitatively, UniDiffuser outperforms VD consistently in both tasks under all metrics and guidance CFG scales, as presented in Figure 5 and Figure 6. The empirical results demonstrate the effectiveness (in addition to the simplicity, efficiency, and generality) of UniDiffuser compared to VD (see details in Section 5). Qualitatively, Figure 7 presents samples in text-to-image generation, and UniDiffuser aligns image and text better than VD. See more results including image-to-text generation in Appendix G.

We also compare with bespoken systems designed for text-to-image generation w.r.t. zero-shot FID on MS-COCO in Table 1. Although UniDiffuser is designed to handle multiple generation tasks, its performance on the single text-to-image generation task is comparable to bespoken diffusion models such as Stable Diffusion and outperforms famous diffusion models like DALL·E 2.

Finally, we present examples of joint, conditional, and unconditional generation in Figure 1 (a-e) to show the generality of UniDiffuser. See more examples in Appendix A.

6.3. Data Variation and Gibbs Sampling

UniDiffuser naturally supports applications such as image variation and text variation. For example, given a source image, we can firstly perform image-to-text generation to obtain a description of the image, and then perform text-to-image generation with this description as input to obtain a new image with similar semantics but different contents. In Figure 1 (f-g), we present examples on image and text variation. Furthermore, we can perform blocked Gibbs sampling to see how images and texts are translated to each other by chaining conditional distributions modeled by UniDiffuser. We present examples in Figure 1 (h). More samples on data variation and blocked Gibbs can be found in Appendix A.

6.4. Interpolation between Two Images in the Wild

UniDiffuser can also perform interpolation between two images in the wild. Specifically, we firstly perform image-to-text generation to obtain the latent text embeddings of the two images via the deterministic DPM-Solver with the same Gaussian noise as the initial state for both images. Then we perform a noise injection process via DPM-Solver to get a noisy version of the latent image embeddings given the two latent text embeddings. We perform spherical linear interpolation between the latent text embeddings and the noisy version of the latent image embeddings to obtain intermediate states. Finally, with the text intermediate states as the condition and the image intermediate states as the initial state, we generate the final images by DPM-solver. See Appendix F for a formalized algorithm of the interpolation procedure. We present examples in Figure 1 (i) and more examples can be found in Appendix A.

7. Conclusion

We propose UniDiffuser, a general-purpose multi-modal probabilistic framework based on insights of unifying training of diffusion models for different distributions. UniDiffuser is able to perform various generation tasks via one model with minimal modification of the original diffusion models. Empirical results on image-text data show the effectiveness of UniDiffuser compared to large existing models. UniDiffuser also enables semi-supervised learning and learning on more modalities, which are left as future work. Currently, the text generated by our implementation is not that smooth, mainly because the text data is noisy.

UniDiffuser has high potential to improve multiple tasks: by fitting multiple tasks with one single transformer network, UniDiffuser can be much easier to further improve all tasks simultaneously (e.g., by increasing parameter scale and data scale) and maintain under the large-scale pre-training regime. Any further improvement/optimization of the underlying single network can seamlessly benefit all tasks.

Social Impact: We believe UniDiffuser can advance real-world applications with generated content due to its generality. However, it is worth noting that large-scale multi-modal generative models may have consequences like "deepfakes". We watermark all images sampled from the model and will provide a systematical protocol to relieve the problem before releasing the code and model.

Acknowledgements

This work was supported by NSF of China Projects (Nos. 62061136001, 61620106010, 62076145, U19B2034, U1811461, U19A2081, 6197222); Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098; a grant from Tsinghua Institute for Guo Qiang; the High Performance Computing Center, Tsinghua University; the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (22XNKJ13). C. Li was also sponsored by Beijing Nova Program. J.Z was also supported by the New Cornerstone Science Foundation through the XPLORER PRIZE.

References

Bao, F., Li, C., Sun, J., Zhu, J., and Zhang, B. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *ICML*, 2022a.

Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *ICLR*, 2022b.

Bao, F., Li, C., Cao, Y., and Zhu, J. All are worth words: a vit backbone for score-based diffusion models. In *CVPR*, 2023a.

Bao, F., Zhao, M., Hao, Z., Li, P., Li, C., and Zhu, J. Equivariant energy-guided sde for inverse molecular design. In *ICLR*, 2023b.

Bao, H., Wang, W., Dong, L., Liu, Q., Mohammed, O. K., Aggarwal, K., Som, S., and Wei, F. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In *NeurIPS*, 2022c.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. In *ICLR*, 2021.

Chen, T., Zhang, R., and Hinton, G. Analog bits: Generating discrete data using diffusion models with self-conditioning. *ArXiv preprint*, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *ArXiv preprint*, 2021.

Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. Cogview: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022.

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS*, 2021.

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

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *ArXiv preprint*, 2022a.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022b.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *ICML*, 2022.

Hu, M., Zheng, C., Zheng, H., Cham, T.-J., Wang, C., Yang, Z., Tao, D., and Suganthan, P. N. Unified discrete diffusion for simultaneous vision-language generation. *arXiv preprint arXiv:2211.14842*, 2022.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *NeurIPS*, 2021.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. In *ICLR*, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.

Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., and Zhu, J. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *ICML*, 2022a.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022b.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022c.

Luo, S. and Hu, W. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021.

Mokady, R., Hertz, A., and Bermano, A. H. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. A. Grad-tts: A diffusion probabilistic model for text-to-speech. In *ICML*, 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *ICML*, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *ArXiv preprint*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICLR*, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *ICLR*, 2021a.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. In *NeurIPS*, 2021b.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021c.

Srivastava, N. and Salakhutdinov, R. R. Multimodal learning with deep boltzmann machines. In *NeurIPS*, 2012.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *ArXiv preprint*, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 2011.

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O. K., Singhal, S., Som, S., et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *ArXiv preprint*, 2022.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *ICML*, 2020.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

Xu, X., Wang, Z., Zhang, E., Wang, K., and Shi, H. Versatile diffusion: Text, images and variations all in one diffusion model. *arXiv preprint arXiv:2211.08332*, 2022.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *ArXiv preprint*, 2022.

Zhao, M., Bao, F., Li, C., and Zhu, J. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. In *NeurIPS*, 2022.

A. More Examples

Below we present more samples of UniDiffuser on joint generation (Figure 8), text-to-image generation (Figure 9), image-to-text generation (Figure 10), unconditional image generation (Figure 11), unconditional text generation (Figure 13), image variation (Figure 12), text variation (Figure 14), blocked Gibbs sampling (Figure 15) and interpolation (Figure 16).

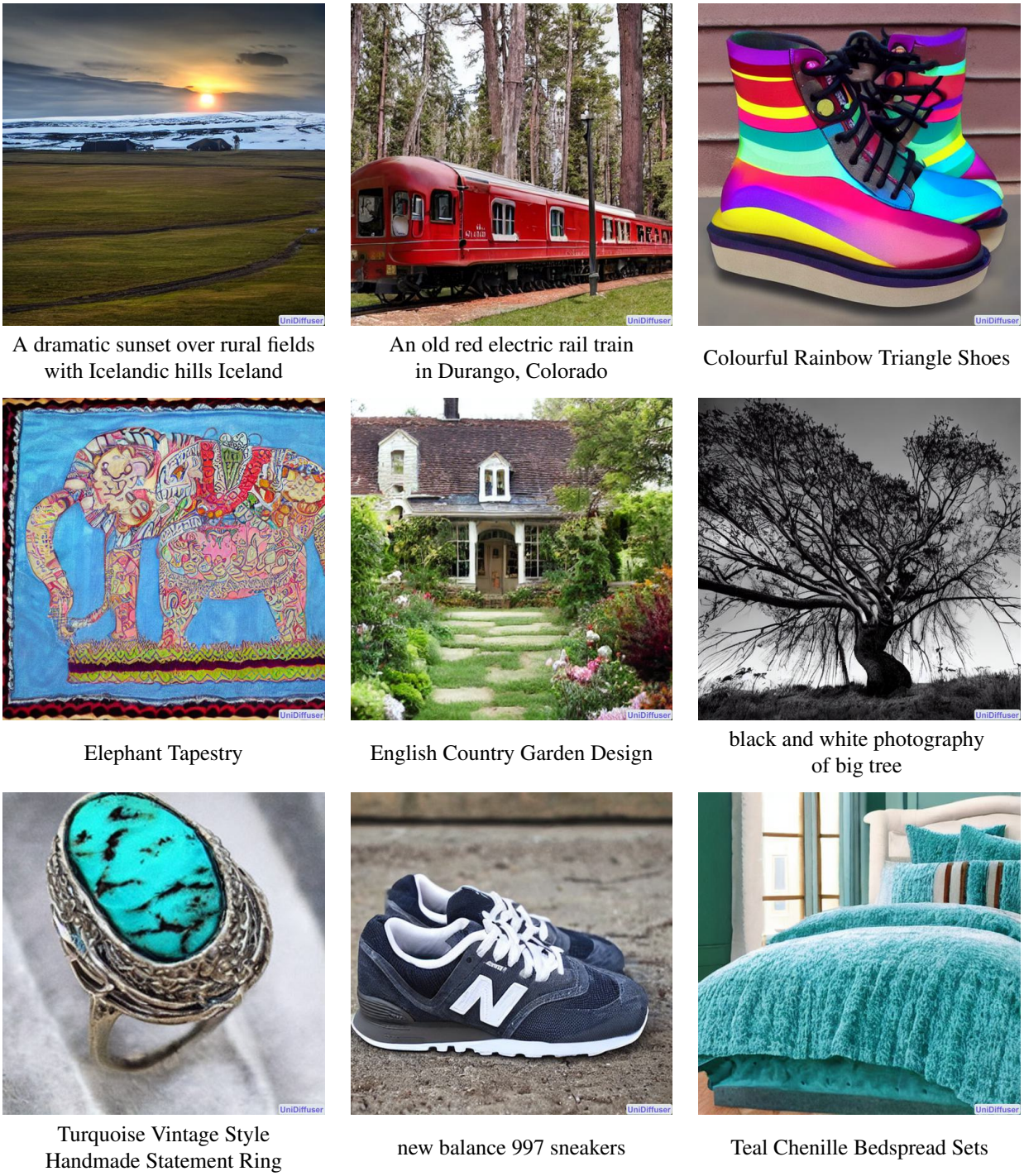
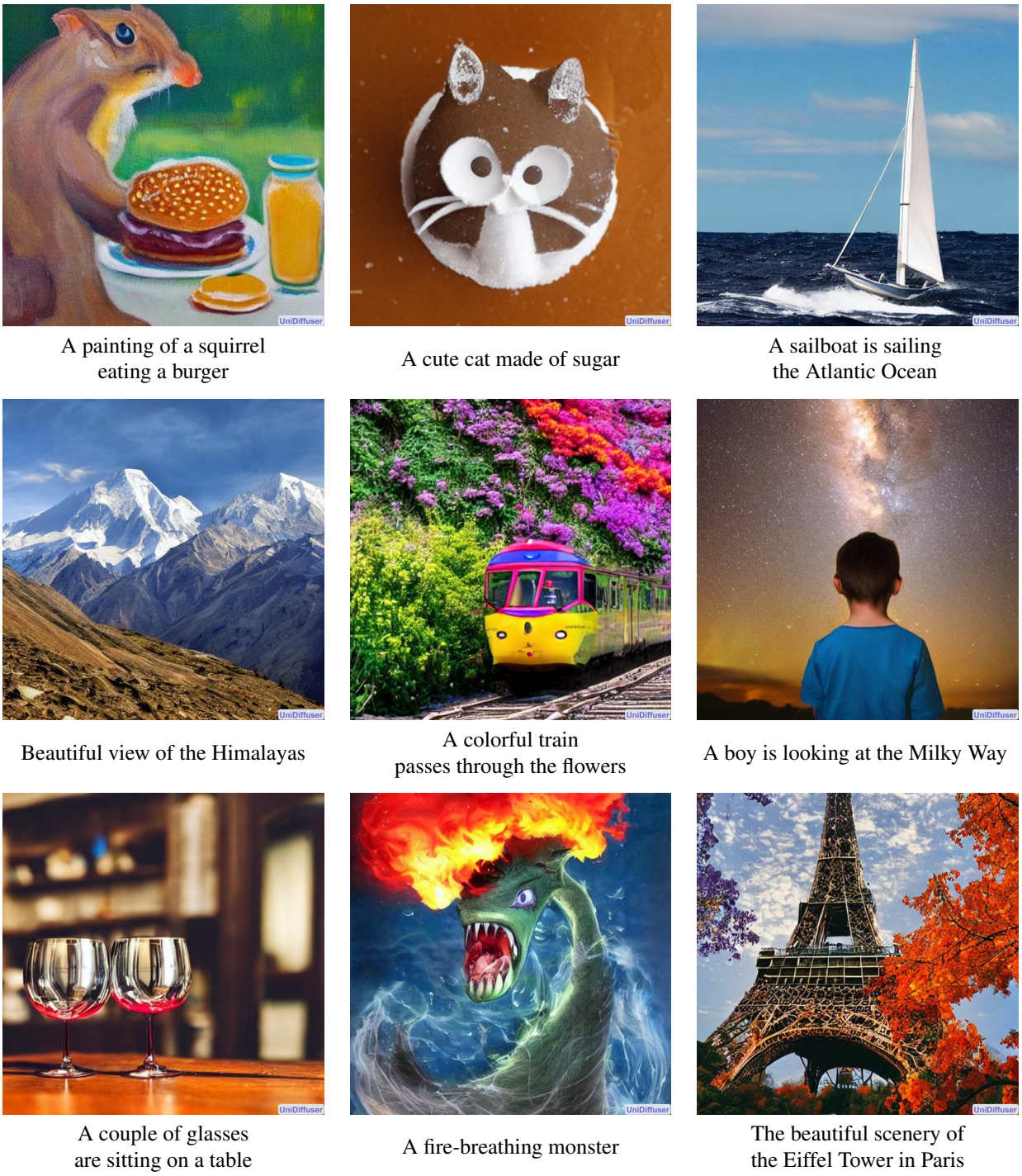


Figure 8. Selected samples of UniDiffuser on joint generation of image and text.



A painting of a squirrel eating a burger

A cute cat made of sugar

A sailboat is sailing the Atlantic Ocean

Beautiful view of the Himalayas

A colorful train passes through the flowers

A boy is looking at the Milky Way

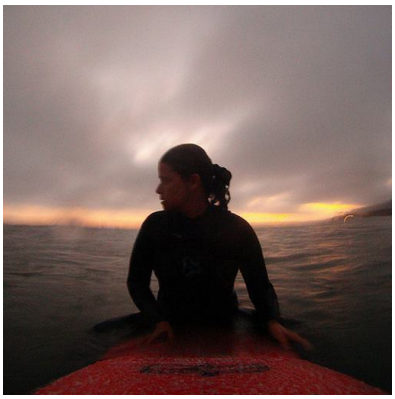
A couple of glasses are sitting on a table

A fire-breathing monster

The beautiful scenery of the Eiffel Tower in Paris

Figure 9. Selected samples of UniDiffuser on text-to-image generation.

A herd of cows standing inside a shed

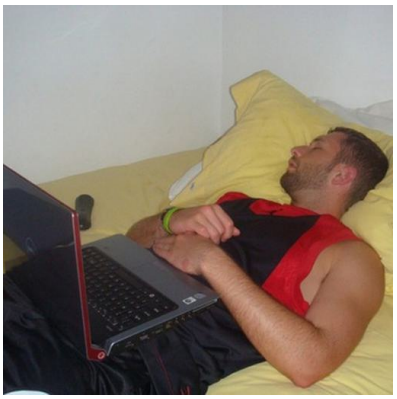


A woman sits on a kayak of Patagonia Argentina sea with a sunset clouds in the air after the rain.

Blue and orange boat at the port, Swansea.

Blue and yellow macaws sitting in a tree precariously

Cat resting on the rug



Man sleeping on bed with his laptop

Red London Bus

Two Grizzly

Train tracks with palm trees at night

Figure 10. Selected samples of UniDiffuser on image-to-text generation.

Figure 11. Selected samples of UniDiffuser on unconditional image generation.

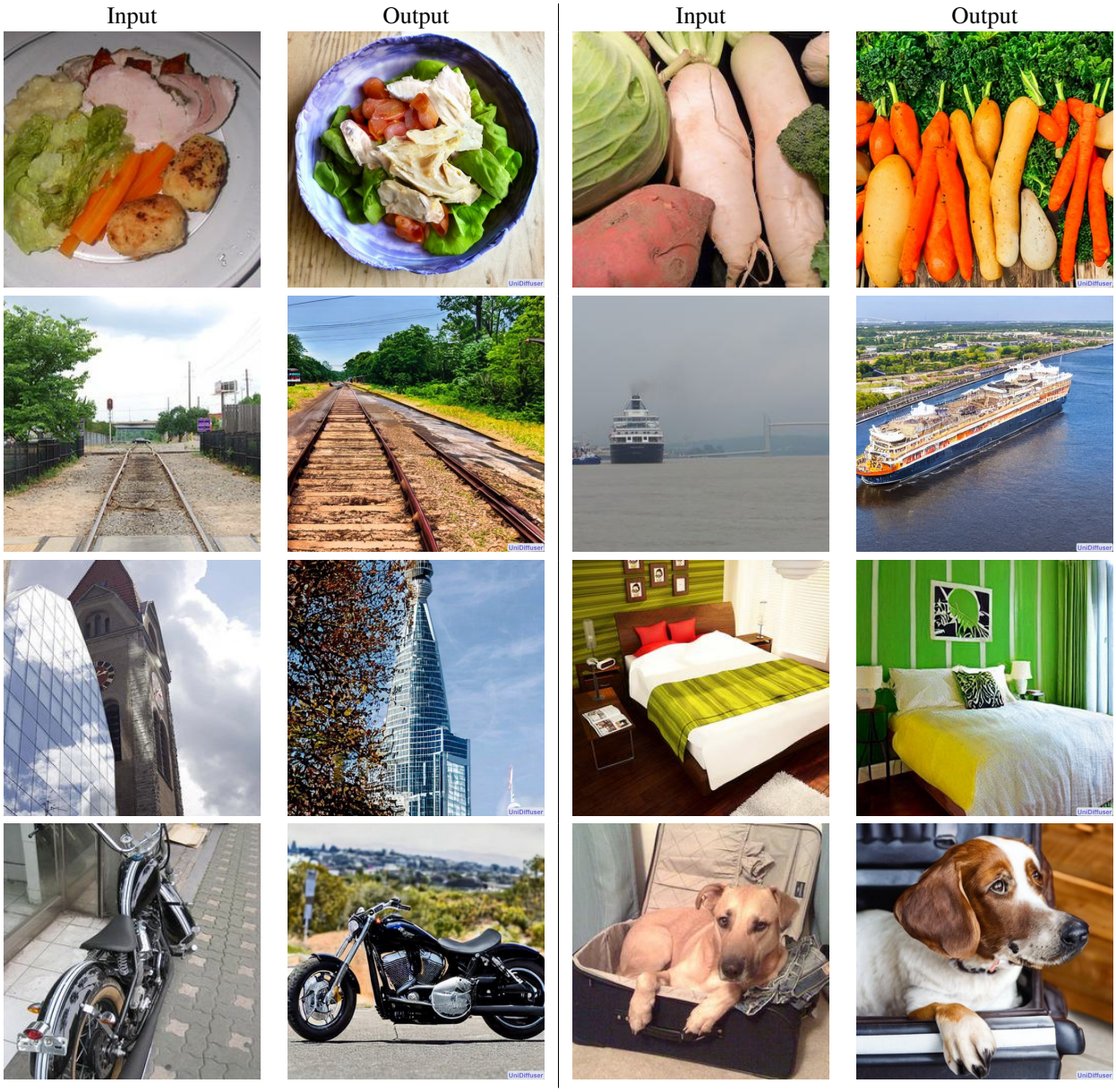


Figure 12. Selected samples of UniDiffuser on image variation.

- *Shoppers Crossing Bison Street*
- *Bend Butte Festival*
- *crapped slices on a bookshelf*
- *Waterfall over Montrose, Alabama Carnegie Trail, Adirondacks*
- *Espacefoil Water Mill for Commercial*
- *Liquid Watercolour*
- *Pizza Republic's Earth Day event*
- *Upgrade Vintage Office Décor*
- *Fire On The Coast Original Fine Art by Jenny Clinn*
- *French coast aerial view taken from Palermo*
- *brown farmhouse table*
- *What Color Is That Tree*
- *soldiers sits on tree after a breakfast break*
- *3 bedroom Condo for sale at 289 Cajundale Dr Lake Solitude Cay, FL, 346015*
- *women riding horses*
- *Kevin Durant Asks More Summer Terms With Xfinity*
- *Bicycle on the Wood Fence*
- *Rose Etched Blooms Petticoat*
- *Fishing Boats in Paintings*

Figure 13. Selected samples of UniDiffuser on unconditional text generation.

- *A woman walking while holding up a red umbrella*
- *woman in red outfit holding umbrella walking on road*

- *A man with a colored umbrella sits on a monument.*
- *Man with an umbrella on the steps in the rain*

- *A young child stands in the kitchen with an adult.*
- *Mother and daughter cooking together in kitchen*

- *A woman kisses a man as they sit on a motorcycle.*
- *couple kissing on motorcycle*

- *Children's toy animals are strewn across a floor.*
- *Collection of children's toys*

- *A woman is walking a dog in the city.*
- *Woman with dog walking on city street*

- *A cat at attention between two parked cars.*
- *Cat sitting on pavement between car*

- *A man is sitting on a bench next to a bicycle.*
- *handsome hipster guy on a city bench riding a bike*

- *Three road signs posted in a parking garage.*
- *red signs showing warning and road safety signs*

- *A couple of birds fly through a blue cloudy sky.*
- *Flock of pigeons over a blue sky*

- *A group of men standing in front of a bar having a conversation.*
- *A group of happy business people sitting at a table looking on, and having a drink in a bar.*

- *A small wooden table covered with delicious vegetables.*
- *Fresh vegetables with tomatoes, parsley and cilantro on wooden board*

- *A woman in a yellow bathroom is holding a camera.*
- *Woman in yellow dress holding a camera in her bathroom*

Figure 14. Selected samples of UniDiffuser on text variation.

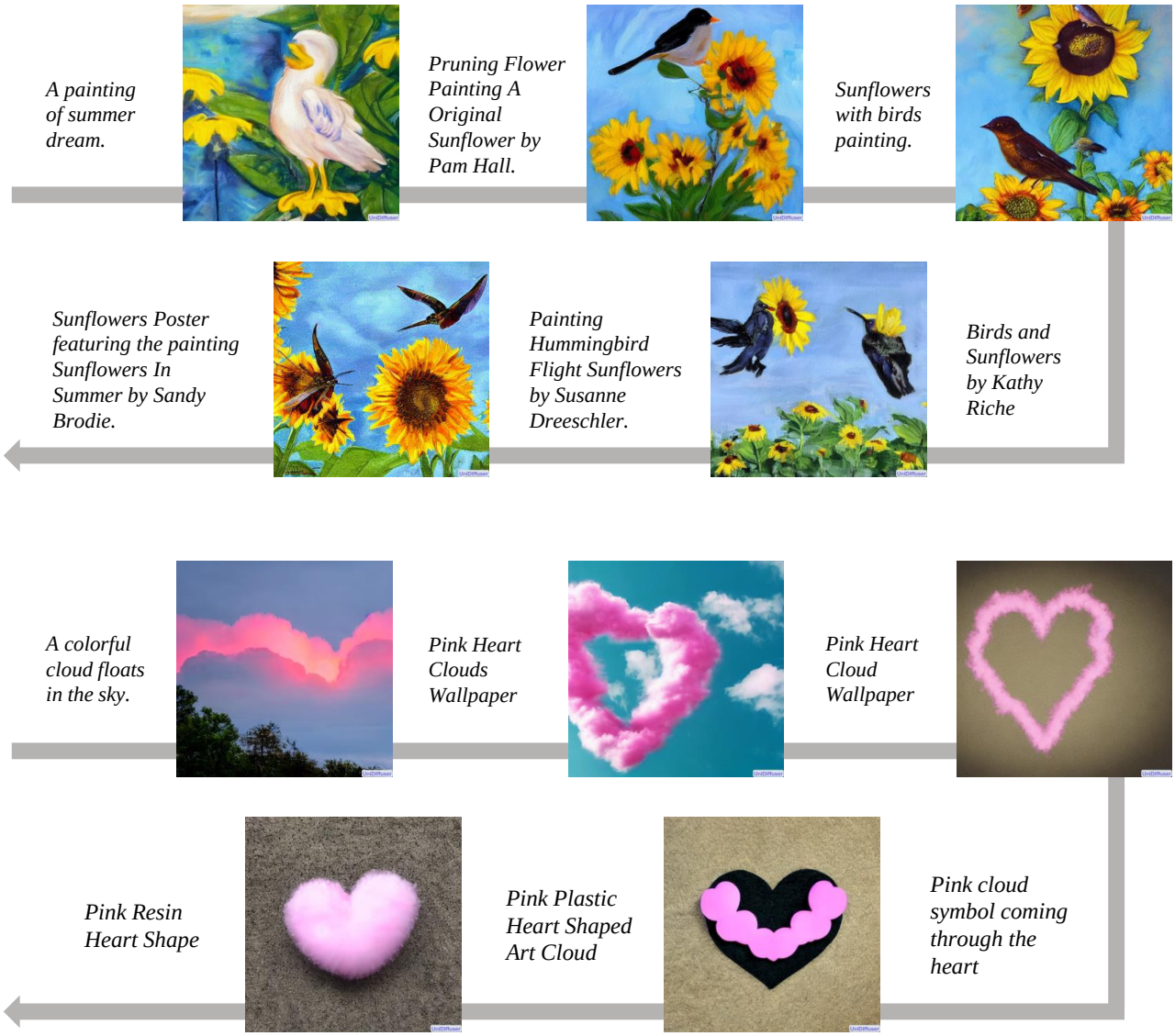


Figure 15. Selected samples of UniDiffuser on Blocked Gibbs sampling.

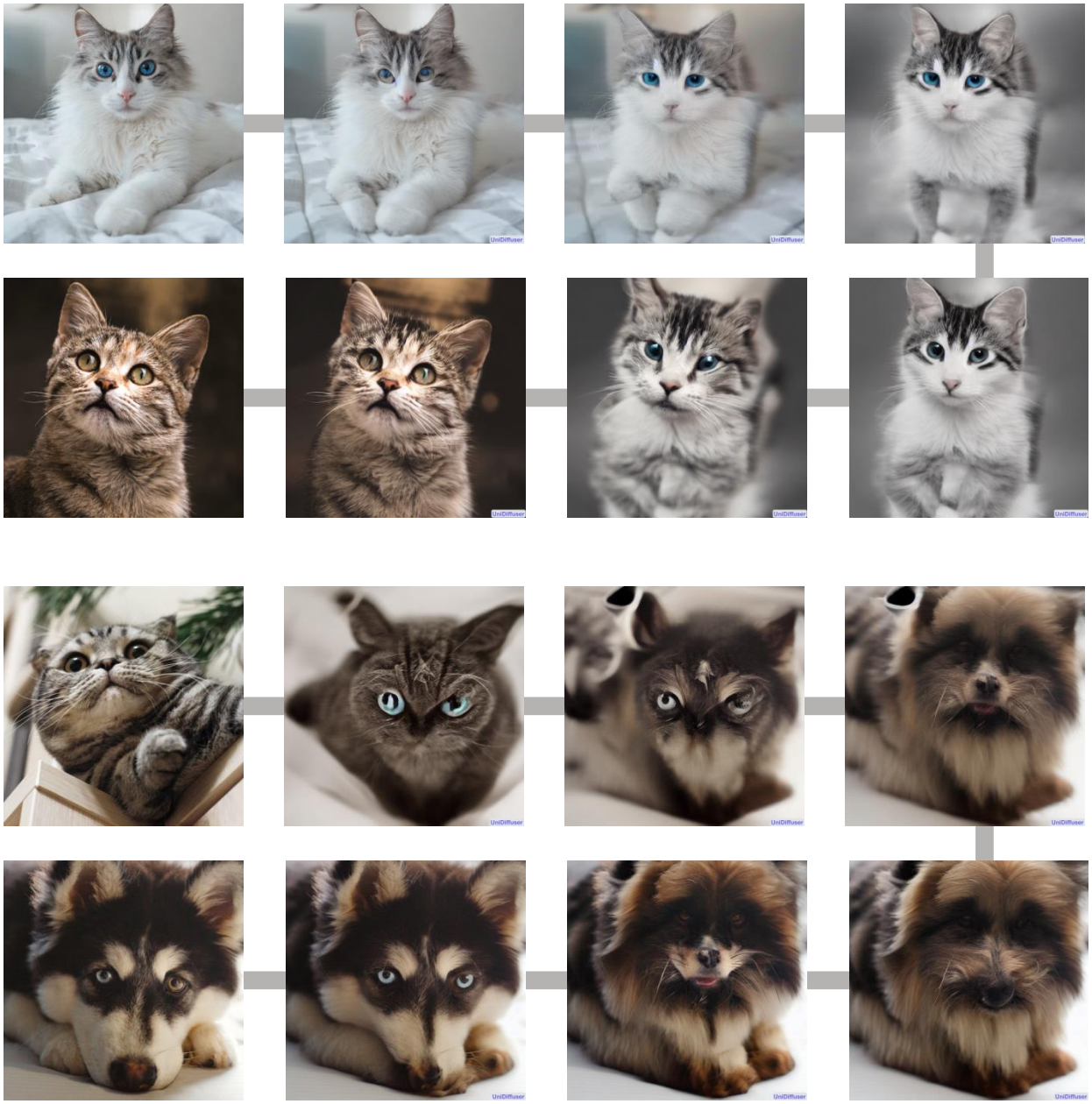


Figure 16. Selected samples of UniDiffuser on interpolation of two images in the wild.

B. The Training and Sampling Algorithms

In Algorithm 1, we present the training algorithm of UniDiffuser. In Algorithm 2,3,4, we present all sampling procedure of UniDiffuser by taking the DDPM sampler (Ho et al., 2020) as an example. Note that any other learning-free efficient sampler, such as DDIM (Song et al., 2021a), Analytic-DPM (Bao et al., 2022b) and DPM-Solver (Lu et al., 2022b;c), is directly applicable.

Algorithm 1 Training

repeat
  $\boldsymbol{x}_0, \boldsymbol{y}_0 \sim q(\boldsymbol{x}_0, \boldsymbol{y}_0)$
  $t^x, t^y \sim \text{Uniform}(\{1, 2, \ldots, T\})$
  $\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  Let $\boldsymbol{x}_{t^x} = \sqrt{\overline{\alpha}_{t^x}}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_{t^x}}\boldsymbol{\epsilon}^x$
  Let $\boldsymbol{y}_{t^y} = \sqrt{\overline{\alpha}_{t^y}}\boldsymbol{y}_0 + \sqrt{1-\overline{\alpha}_{t^y}}\boldsymbol{\epsilon}^y$
  Take gradient step on
    $\nabla_{\boldsymbol{\theta}} \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t^x}, \boldsymbol{y}_{t^y}, t^x, t^y) - [\boldsymbol{\epsilon}^x, \boldsymbol{\epsilon}^y]\|_2^2$
until converged

Algorithm 2 Unconditional sampling of $\boldsymbol{x}_0$ (similar for unconditional sampling of $\boldsymbol{y}_0$)

$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \ldots, 1$ **do**
  $\boldsymbol{z}^x \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{z}^x = \mathbf{0}$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}, t, T)\right) + \sigma_t \boldsymbol{z}^x$
end for
return $\boldsymbol{x}_0$

Algorithm 3 Sampling of $\boldsymbol{x}_0$ conditioned on $\boldsymbol{y}_0$ (similar for sampling of $\boldsymbol{y}_0$ conditioned on $\boldsymbol{x}_0$)

$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \ldots, 1$ **do**
  $\boldsymbol{z}^x \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{z}^x = \mathbf{0}$
  $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t, 0)\right) + \sigma_t \boldsymbol{z}^x$
end for
return $\boldsymbol{x}_0$

Algorithm 4 Joint sampling of $\boldsymbol{x}_0, \boldsymbol{y}_0$

$\boldsymbol{x}_T, \boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \ldots, 1$ **do**
  $\boldsymbol{z}^x, \boldsymbol{z}^y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{z}^x, \boldsymbol{z}^y = \mathbf{0}$
  $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t, t)\right) + \sigma_t \boldsymbol{z}^x$
  $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t, t)\right) + \sigma_t \boldsymbol{z}^y$
end for
return $\boldsymbol{x}_0, \boldsymbol{y}_0$

C. Summary of Classifier-Free Guidance Models

As mentioned in Section 3.2, UniDiffuser can perform classifier-free guidance for free for conditional and joint generation. We summarize models for classifier-free guidance in Table 2. We also include the models for unconditional sampling.

Table 2. **Models for different sampling tasks.** $\boldsymbol{\epsilon}^x$ and $\boldsymbol{\epsilon}^y$ are sampled from the standard Gaussian noise. s is the classifier-free guidance scale.

Task	Model
Joint sampling	$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t) = (1+s)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_t, t, t) - s[\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T), \boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}^x, \boldsymbol{y}_t, T, t)]$
Sample $\boldsymbol{x}_0$ conditioned on $\boldsymbol{y}_0$	$\hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t) = (1+s)\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t, 0) - s\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T)$
Sample $\boldsymbol{y}_0$ conditioned on $\boldsymbol{x}_0$	$\hat{\boldsymbol{\epsilon}}^y_{\boldsymbol{\theta}}(\boldsymbol{y}_t, \boldsymbol{x}_0, t) = (1+s)\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{x}_0, \boldsymbol{y}_t, 0, t) - s\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}^x, \boldsymbol{y}_t, T, t)$
Unconditional sampling of $\boldsymbol{x}_0$	$\hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T)$
Unconditional sampling of $\boldsymbol{y}$	$\hat{\boldsymbol{\epsilon}}^y_{\boldsymbol{\theta}}(\boldsymbol{y}_t, t) = \boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}^x, \boldsymbol{y}_t, T, t)$

D. Details of the U-ViT

We train a U-ViT for the joint noise prediction network, whose detailed configuration is presented in Table 3.

Table 3. The configuration of U-ViT.

Patch size	#Layers	Hidden size	MLP size	#Heads	#Params
2	31	1536	6144	24	952M

E. Details of the GPT-2 Text Decoders

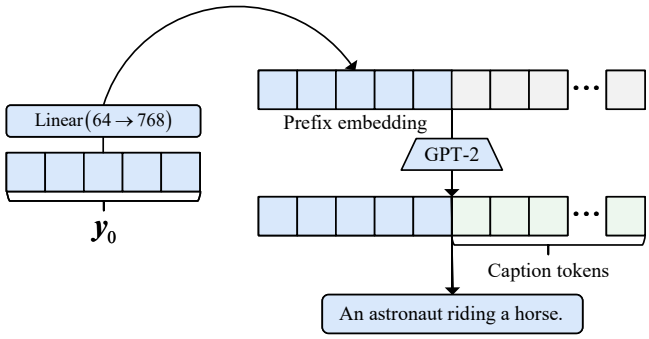


Figure 17. **GPT-2 Text Decoder.** $\boldsymbol{y}_0$ is fed into GPT-2 as a prefix embedding (Mokady et al., 2021), and reconstructs the text autoregressively based on the information of $\boldsymbol{y}_0$

As mentioned in Section 4.1, we firstly encode the text $\mathbf{T}$ into a low-dimensional embedding $\boldsymbol{y}_0$ through a CLIP and a linear layer, i.e., $\boldsymbol{y}_0 = \text{Linear}(\text{CLIP}(\mathbf{T}))$, and then reconstruct the original text using the GPT-2 by feeding $\boldsymbol{y}_0$ as a prefix embedding. The reconstruction pipeline of the GPT-2 decoder is illustrated in Figure 17. We finetune the GPT-2 and the linear layer with the following autoregressive loss:

$$\min_{\phi} \mathbb{E}_{\mathbf{T}}[\log p(\mathbf{T}|\boldsymbol{y}_0)] = \mathbb{E}_{\mathbf{T}}[\sum_{i=1}^{N} \log p(\mathbf{T}_i|\mathbf{T}_{1:i-1}, \boldsymbol{y}_0)],$$

where ϕ denotes the parameters of the linear layer and the GPT-2.

We finetune the 124M parameter GPT-2 text decoder on texts of LAION-2B-en dataset (Schuhmann et al., 2022), which contains 2.3B image-text pairs. Following ClipCap (Mokady et al., 2021), we use the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 2e-5 and 5K warm-up steps. We train the decoder with 235K steps using a batch size of 768. When generating texts, we use the beam search strategy with a beam size of 5 and a maximum length of 67.

In our experiments, the embedding dimension of $\boldsymbol{y}_0$ is set to 64, and it still reconstructs the text well. Indeed, we get a BLEU-1 (Papineni et al., 2002) score of 0.969 and a BLEU-4 score of 0.894 between reference texts and reconstructed texts on MS-COCO test set (Karpathy split (Karpathy & Fei-Fei, 2015)). We present some examples in Figure 18, where the input texts are reconstructed very well.

Caption ID	Text	Reconstruction
71764	*A bathroom has swinging saloon style stall doors*	*A Bathroom Is Swing Springs Style Stall Doors*
640876	*A picture of a street sign with various posts on it.*	*A picture of a street sign with various posts on it.*
542566	*Two microwaves and a very old fashioned printer*	*Two microwaves and a very old fashioned printer*
35539	*A large truck parked across the street from another truck.*	*A large truck parked across the street from another trunk.*
802267	*A large group photo taken at a wedding.*	*A large group photo taken at a wedding.*
184192	*The street sign shows the names of two intersecting roads.*	*The street sign shows the names of two intersecting roads.*
341901	*A salad in a plastic bowl sitting on a table next to an apple.*	*A salad in a plastic bowl sitting on a table next to an apple.*
680092	*a couple of people are rowing in a boat*	*A couple of people are rowing in a boat*
395089	*a vase on a table with flowers inside*	*a vase on a table with flowers inside*
202966	*A Red Sox player preparing to throw a baseball.*	*A red Sox player preparing to throw a baseball.*

Figure 18. **Text reconstruction examples.** The GPT-2 text decoder reconstructs the text well.

F. The Interpolation Algorithm

We present the formalized procedure of interpolating two images using UniDiffuser in Algorithm 5.

Algorithm 5 Interpolate two images $\mathbf{I}^a$ and $\mathbf{I}^b$

Input: the spherical interpolation parameter $\theta \in [0, 1]$ ($\theta = 0$ leads to $\mathbf{I}^a$ and $\theta = 1$ leads to $\mathbf{I}^b$)
Let $\hat{\boldsymbol{\epsilon}}^y_{\boldsymbol{\theta}}(\boldsymbol{y}_t, \boldsymbol{x}_0, t) = (1 + s)\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{x}_0, \boldsymbol{y}_t, 0, t) - s\boldsymbol{\epsilon}^y_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}^x, \boldsymbol{y}_t, T, t)$ be the image-to-text model
Let $\hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t) = (1 + s)\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}_0, t, 0) - s\boldsymbol{\epsilon}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^y, t, T)$ be the text-to-image model
Encode $\mathbf{I}^a$ and $\mathbf{I}^b$ to get their latent embeddings $\boldsymbol{x}^a_0$ and $\boldsymbol{x}^b_0$
$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\boldsymbol{y}^a_0 = \text{DPM-Solver}(\text{initial_state} = \boldsymbol{y}_T, \text{start_time} = T, \text{end_time} = 0, \text{model} = \hat{\boldsymbol{\epsilon}}^y_{\boldsymbol{\theta}}(\boldsymbol{y}_t, \boldsymbol{x}^a_0, t))$
$\boldsymbol{y}^b_0 = \text{DPM-Solver}(\text{initial_state} = \boldsymbol{y}_T, \text{start_time} = T, \text{end_time} = 0, \text{model} = \hat{\boldsymbol{\epsilon}}^y_{\boldsymbol{\theta}}(\boldsymbol{y}_t, \boldsymbol{x}^b_0, t))$
$\boldsymbol{x}^a_T = \text{DPM-Solver}(\text{initial_state} = \boldsymbol{x}^a_0, \text{start_time} = 0, \text{end_time} = T, \text{model} = \hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}^a_0, t))$
$\boldsymbol{x}^b_T = \text{DPM-Solver}(\text{initial_state} = \boldsymbol{x}^b_0, \text{start_time} = 0, \text{end_time} = T, \text{model} = \hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}^b_0, t))$
$\boldsymbol{y}^\theta_0 = \text{slerp}(\boldsymbol{y}^a_0, \boldsymbol{y}^b_0, \theta)$
$\boldsymbol{x}^\theta_T = \text{slerp}(\boldsymbol{x}^a_T, \boldsymbol{x}^b_T, \theta)$
$\boldsymbol{x}^\theta_0 = \text{DPM-Solver}(\text{initial_state} = \boldsymbol{x}^\theta_T, \text{start_time} = T, \text{end_time} = 0, \text{model} = \hat{\boldsymbol{\epsilon}}^x_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{y}^\theta_0, t))$
Decode $\boldsymbol{x}^\theta_0$ to get the image $\mathbf{I}^\theta$
Return $\mathbf{I}^\theta$

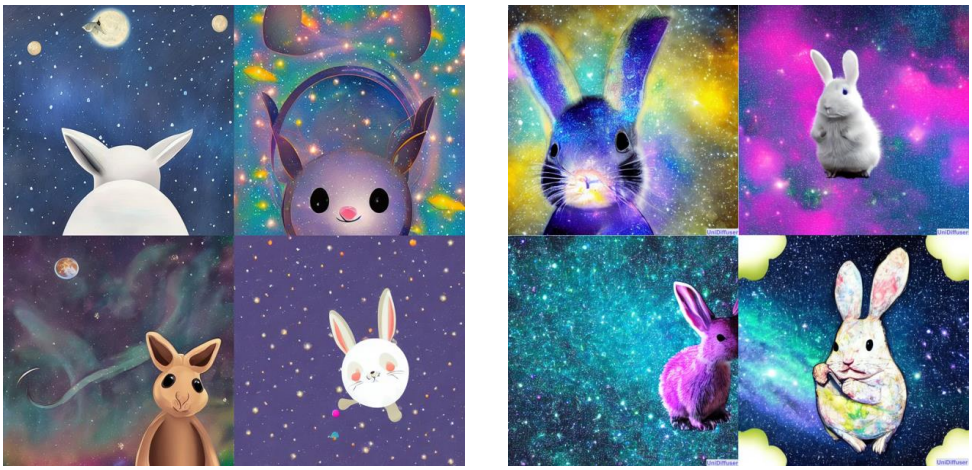
G. Comparison of Examples

In Figure 19 and Figure 20, we present more examples on text-to-image generation of UniDiffuser and Versatile Diffusion. Samples generated by UniDiffuser align better with the texts than VD. In Figure 21, we present examples on image-to-text generation of UniDiffuser and Versatile Diffusion. Samples generated by UniDiffuser align better with the images than VD.

Figure 19. Comparison of examples between UniDiffuser and VD on text-to-image generation. Samples generated by UniDiffuser align the texts better than VD.

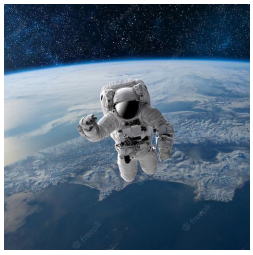
Versatile Diffusion (VD) UniDiffuser (ours)

An elephant under the sea.

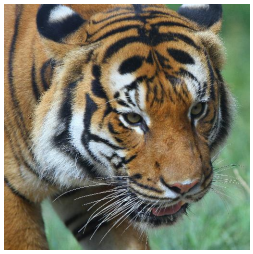


A rabbit floating in the galaxy

Figure 20. Comparison of examples between UniDiffuser and VD on text-to-image generation. Samples generated by UniDiffuser align the texts better than VD.



UniDiffuser (ours)	• *person looking at white light with stars and the trees in the sky* • *man standing by a lighted high trees in front of night sky*	• *Traffic signs stop* • *Traffic stop warning signs*	• *Astronaut diving in space.* • *Earth in space*
Versatile Diffusion	• *The night sky, with some milkys.* • *A person walking across a wooden bridge at night*	• *signs standing signs in a street* • *signs, stop signs on construction ground, at work.*	• *astronaut has space on earth in order to travel* • *a man soldier flying inside space*



UniDiffuser (ours)	• *Girl with the Pearl Earring* • *Lady in Blue, by Johannes Vermeer, Prestigious Studio.*	• *King Penguin in Long Island, Antarctica* • *King Penguin, New Zealand*	• *Bengal Tiger* • *Big Bengal Tiger*
Versatile Diffusion	• *young woman with a young girl wearing a hat, in a old library window* • *woman with a hat with cradle*	• *a penguaur is standing on a board on the ocean.* • *penguin on pengu island, king on board, boat on lake island mountain*	• *Tiger* • *tiger at circus*

Figure 21. Comparison of examples between UniDiffuser and VD on image-to-text generation. Samples generated by UniDiffuser align the images better than VD.

H. Efficiency Comparison

In Table 4, we compare the model size, inference time and memory (for generating 10 samples with 25 denoising steps on one A100 80GB GPU), and the training cost of UniDiffuser with other bespoke and general-purpose models. Results of other methods are obtained according to the official code or paper when they are available.

UniDiffuser is more efficient than both Stable Diffusion and Versatile Diffusion in terms of inference time and memory. Besides, compared to Stable Diffusion, UniDiffuser introduces only 10% extra parameters to support five tasks (i.e., image, text, text-to-image, image-to-text, and image-text pair generation) with comparable training cost. Compared to the general-purpose model Versatile Diffusion, UniDiffuser has fewer parameters, while achieving superior results (as presented in the main paper).

Table 4. Comparison of model size and computational cost.

	Model size	Inference time	Inference memory	Training cost
Bespoken models				
DALL·E 2 (Ramesh et al., 2022)	4.5B	–	–	–
Imagen (Saharia et al., 2022)	2B	–	–	–
Parti (Yu et al., 2022)	20B	–	–	–
Stable Diffusion† (Rombach et al., 2022)	860M	25.43s	67.83GB	150K (A100 40GB GPU hours)
General-purpose models				
Versatile Diffusion† (Xu et al., 2022)	2566M	23.89s	76.53GB	–
UniDiffuser (**ours**)	952M	19.77s	48.30GB	59K (A100 80GB GPU hours)

I. Licences

Datasets:

- LAION-5B (Schuhmann et al., 2022): Creative Common CC-BY 4.0 license
- MS-COCO (Lin et al., 2014): Creative Commons Attribution 4.0 License

Pretrained models:

- GPT-2 (Radford et al., 2019): MIT License
- CLIP (Radford et al., 2021): MIT License
- Image autoencoder from Stable Diffusion (Rombach et al., 2022): CreativeML Open RAIL-M License