ReDi: Efficient Learning-Free Diffusion Inference via Trajectory Retrieval

Kexun Zhang [1] **Xianjun Yang** [1] **William Yang Wang** [1] **Lei Li** [1]

Abstract

Diffusion models show promising generation capability for a variety of data. Despite their high generation quality, the inference for diffusion models is still time-consuming due to the numerous sampling iterations required. To accelerate the inference, we propose REDI, a simple yet learning-free ***Re***trieval-based ***Di***ffusion sampling framework. From a precomputed knowledge base, REDI retrieves a trajectory similar to the partially generated trajectory at an early stage of generation, skips a large portion of intermediate steps, and continues sampling from a later step in the retrieved trajectory. We theoretically prove that the generation performance of REDI is guaranteed. Our experiments demonstrate that REDI improves the model inference efficiency by 2× speedup. Furthermore, REDI is able to generalize well in zero-shot cross-domain image genreation such as image stylization. The code and demo for REDI is available at `https://github.com/zkx06111/ReDiffusion`.

1. Introduction

Deep generative models are changing the way people create content. Among them, diffusion models have shown great capability in a variety of applications including image synthesis (Ho et al., 2020; Dhariwal & Nichol, 2021), speech synthesis (Liu et al., 2021a), and point cloud generation (Zhou et al., 2021). Latent diffusion models (Rombach et al., 2022) such as Stable Diffusion are able to generate high-quality images given text prompts. However, the basic sampler for diffusion models proposed by Ho et al. (2020) requires a large number of function estimations (NFEs) during inference, making the generation process rather slow. For example, the basic sampler takes 336 seconds on average to run on an NVIDIA 1080Ti, where improving the efficiency of diffusion model inference is crucial.

[1]Department of Computer Science, University of California, Santa Barbara. Correspondence to: Kexun Zhang <kexun@ucsb.edu>.

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

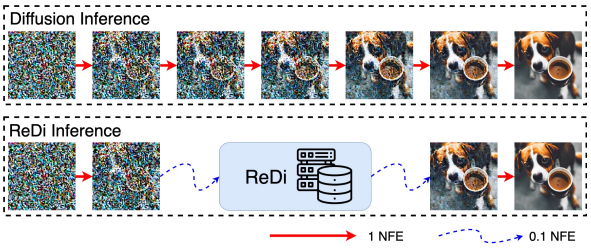


Figure 1. Diffusion Inference (upper) and REDI Inference (lower). REDI reduces the number of function estimations (NFEs) during inference by skipping several intermediate sampling steps. The overhead brought by REDI is minimal compared to the cost it reduced.

Previous studies on improving the efficiency of diffusion model inference can be categorized into two types, learning-based ones, and learning-free ones. Learning-based samplers (Salimans & Ho, 2021; Meng et al., 2022; Lam et al., 2022; Watson et al., 2021; Kim & Ye, 2022) require additional training to reduce the number of sampling steps. However, their training is expensive, especially for large diffusion models like Stable Diffusion. In contrast, learning-free samplers do not require training, and are, therefore, applicable to more scenarios. In this paper, we focus on learning-free approaches to speed up inference.

Existing efficient learning-free samplers for diffusion (Liu et al., 2021b; Lu et al., 2022a;b; Zhang & Chen, 2022; Karras et al., 2022) all try to find a more accurate numerical solver for the diffusion ODE (Song et al., 2021), but they do not utilize its sensitivity. The sensitivity of ODEs suggests that under certain conditions, a small perturbation in the initial value does not change the solution too much (Khalil, 2002). This observation motivates the assumption that a previously generated trajectory - if close enough to the current trajectory - can serve as an estimate for it.

In this paper, we propose REDI, a learning-free sampling framework based on trajectory retrieval. Figure 1 illustrates the original full diffusion inference and the REDI inference. Given a pre-trained diffusion model, REDI does not modify its weights or train new modules. Instead, we fix the model

and build a knowledge base $\mathcal{B}$ of trajectories upon a chosen dataset during the precomputation. During the inference, REDI first computes the early few steps in the trajectory as they are, and then retrieves a similar trajectory from $\mathcal{B}$. In this way, one later step in the retrieved trajectory can surrogate the actual one and serve as an initialization point for the model. By jumping from an early time step to a later time step V, REDI is able to save a larger portion of function estimations (NFEs) any numerical solver.

We first prove that REDI's performance is bounded by the distance between the query trajectory and the retrieved neighbor. Then we report results from in-domain experiments to show empirically that with a moderate-sized knowledge base, REDI is able to achieve comparable performance to recent efficient samplers with a $2\times$ speedup. To demonstrate that REDI generalizes well in cross-domain adaptation, we propose an extension to REDI that generates various stylistic images given the same single-style knowledge base. The stylized images generated by REDI are well-rated by human evaluators. Our ablation studies show that under different settings, the actual results of REDI correlate well with the theoretical bounds, indicating the bounds are tight enough to estimate the generation quality.

Our contributions are as follows:

- We propose REDI, a retrieval-based learning-free framework for diffusion models. REDI is orthogonal to previous learning-free samplers and reduces the number of function estimations (NFEs) by skipping some intermediate steps in the trajectory.
- We prove a theoretical bound for the generation quality of REDI that correlates well with the actual performance.
- We show empirically that REDI can improve the inference efficiency with precomputation and perform well in zero-shot domain adaptation.

2. Related Work

Retrieval-Based Diffusion Models Previous studies on retrieval-based diffusion (Blattmann et al., 2022; Sheynin et al., 2022) have different emphases including rare entity generation (Chen et al., 2022), out-of-domain adaptation, semantic manipulation, and parameter efficiency. However, they all need to train a new model instead of building upon a trained model, which requires much computing power and time. They retrieve images and/or text using pre-built similarity measures like CLIP embedding cosine similarity (Radford et al., 2021). But the pre-built measures they use are not specially designed for diffusion models and have no proven performance guarantee.

Learning-Free Diffusion Samplers Most learning-free diffusion samplers are based on the stochastic/ordinary differential equation (SDE/ODE) formulation of the denoising process proposed by Song et al. (2021). This formulation allows the use of better numerical solvers for larger step sizes and fewer model iterations. Under the SDE framework, previous works alter the numerical solver (Song et al., 2021), the initialization point (Chung et al., 2022), the step-size (Jolicoeur-Martineau et al., 2021), and the order of the solver (Karras et al., 2022). The SDE can also be reformulated as an ordinary differential equation which is deterministic and therefore easier to accelerate. Many works (Liu et al., 2021b; Lu et al., 2022a;b; Zhang & Chen, 2022; Karras et al., 2022) hence built upon the ODE formulation and propose better ODE solvers for the problem. Although existing studies have extensively explored how a better numerical solver can be used to accelerate diffusion inference. They have not taken the sensitivity of ODEs into consideration.

3. Background

3.1. Diffusion Models

Assuming data follow an unknown true distribution $p(\boldsymbol{x})$, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Kingma et al., 2021) define a generation process. For any $p(\boldsymbol{x})$, diffusion models learn a denoising process by iteratively adding noise to original data (denoted as $\boldsymbol{x}_0$) from time step 0 until time step T. The forward process adds noise to $\boldsymbol{x}_0$ such that at time step t, the distribution of $\boldsymbol{x}_t$ conditioned on $\boldsymbol{x}_0$ is

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\alpha(t)x_0, \sigma^2(t)\boldsymbol{I}\right), \tag{1}$$

where $\alpha(t), \sigma(t)$ are real-valued positive functions with bounded derivatives. The signal-to-noise ratio (SNR) is defined as $\gamma = \alpha^2(t)/\sigma^2(t)$. By making the SNR decreasing, the marginal distribution of $\boldsymbol{x}_T$ approximates a zero-mean Gaussian, i.e.

$$q(\boldsymbol{x}_T) \approx \mathcal{N}\left(\mathbf{0}, \tilde{\sigma}^2\boldsymbol{I}\right). \tag{2}$$

The noise-adding forward process transforms a data sample from the original distribution to the zero-mean Gaussian $q(\boldsymbol{x}_T)$, while the backward process randomly samples from $q(\boldsymbol{x}_T)$ and denoises the sample with a neural network parameterized conditional distribution $q(\boldsymbol{x}_s|\boldsymbol{x}_t)$ where $s < t$ until it reaches time step 0.

3.2. The Differential Equation Formulation

Kingma et al. (2021) proves that the forward process is equivalent (in distribution) to the following stochastic differential equation (SDE) in terms of the conditional distribution

$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$,

$$\mathrm{d}\boldsymbol{x}_t = f(t)\boldsymbol{x}_t\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}, \tag{3}$$

$$f(t) = \frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}, \tag{4}$$

$$g(t) = \frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t} + 2\frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}\sigma^2(t), \tag{5}$$

where $\boldsymbol{w}$ is the standard Wiener process.

Song et al. (2021) shows that the forward SDE has an equivalent reverse SDE starting from time T with the marginal distribution $q(\boldsymbol{x}_T)$,

$$\mathrm{d}\boldsymbol{x}_t = [f(t)\boldsymbol{x}_t - g^2(t)\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t, \tag{6}$$

where $\bar{\boldsymbol{w}}_t$ is the reverse Wiener process and t goes from 0 to T.

In practice, $\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t) \approx \boldsymbol{\epsilon}/\sigma(t)$ is estimated using a noise-predictor function $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$, where $\boldsymbol{\epsilon}$ is the Gaussian noise added to $\boldsymbol{x}_0$ to obtain $\boldsymbol{x}_t$, i.e.,

$$\boldsymbol{x}_t = \alpha(t)\boldsymbol{x}_0 + \sigma(t)\boldsymbol{\epsilon}, \quad \epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{7}$$

To learn $\boldsymbol{\epsilon}_\theta$, the following objective function is minimized (Ho et al., 2020; Song et al., 2021),

$$L(\theta) = \int_0^T \omega(t)\mathbb{E}_{q(\boldsymbol{x}_0)}\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|^2\right]\mathrm{d}t, \tag{8}$$

where $\omega(t)$ is a weighting function, and the integral is estimated using random samples.

Song et al. (2021) proves that the following ordinary differential equation (ODE) is equivalent to Equation 6,

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \left[f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)\right]. \tag{9}$$

When $\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)$ is replaced by its estimation, we obtain the neural network parameterized ODE,

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \left[f(t)\boldsymbol{x}_t - \frac{g^2(t)}{2\sigma^2(t)}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right]. \tag{10}$$

The inference process of diffusion models can be formulated as solving Equation 10 given a random sample from the Gaussian distribution $q(\boldsymbol{x}_T)$. For each iteration in solving the equation, the noise-predictor function $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is estimated with the trained neural network θ. Therefore, the inference time consumption can be approximated by the number of function estimations (NFEs), i.e., how many times $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is estimated.

4. The REDI Method

Given a starting sample $\boldsymbol{x}_T$, a guidance signal y, ReDi accelerates diffusion inference by skipping some intermediate samples in the sample trajectory to reduce NFEs.

4.1. The Sample Trajectory

Definition 4.1. Given a starting sample $\boldsymbol{x}_T$, the *sample trajectory* of a diffusion model is a sequence of intermediate samples generated in the iterative process from $\boldsymbol{x}_T$ to $\boldsymbol{x}_0$. For a particular time step size δ, the sample trajectory is the sequence $\{\boldsymbol{x}_T, \boldsymbol{x}_{T-\delta}, \boldsymbol{x}_{T-2\delta}, \ldots, \boldsymbol{x}_\delta, \boldsymbol{x}_0\}$, where T is the initial time step.

The sample trajectory describes the intermediate steps to generate the final data sample (e.g. an image), so the inference time linearly correlates with its length, i.e., the number of estimations used. While previous numerical solvers work towards enlarging the step size δ, REDI aims at skipping some intermediate steps to reduce the length of the trajectory. REDI is able to do so because the first few steps determine only the layout of the image which can be shared by many, and the following steps determine the details (Meng et al., 2021; Wu et al., 2022).

In the following section, we describe the REDI algorithm with this definition.

Algorithm 1 REDI Knowledge Base Construction

input A dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^N$ where $y^{(i)}$ is some guidance signal and $\boldsymbol{x}^{(i)}$ is a data sample;
A pretrained diffusion model θ that computes $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, y)$, the noise estimation of $\boldsymbol{x}_t$ at time step t;
Two constant time steps k and v, where $k > v$;
A numerical sampler S, $\boldsymbol{x}_{t-1} = S(\boldsymbol{x}_t, t, \theta)$.
output A knowledge base $\mathcal{B} = \{(key^{(i)}, val^{(i)})\}_{i=1}^N$
for $i \leftarrow 1$ **to** N **do**
  $\boldsymbol{x}_T \sim p(x_T|\boldsymbol{x}^{(i)})$
  // Note that the signal-to-noise ratio is close to 0 at time step T due to the noise ratio of diffusion models. Therefore sampling $\boldsymbol{x}_T$ from $p(\boldsymbol{x}_T|\boldsymbol{x}^{(i)})$ is almost the same as from $p(\boldsymbol{x}_T)$.
  // More discussions about the initial sample in Appendix A.5.
  for $j \leftarrow T$ **downto** 1 **do**
    $\boldsymbol{x}_{j-1} \leftarrow S_\theta\left(\boldsymbol{x}_j, j, y^{(i)}\right)$
  end for // Calculating the trajectories of the ith sample
  $key^{(i)} \leftarrow \boldsymbol{x}_k$ // An early of the trajectory as the key
  $val^{(i)} \leftarrow \boldsymbol{x}_v$ // An intermediate sample as the value
end for
return $\{(key^{(i)}, val^{(i)})\}_{i=1}^N$

4.2. The REDI Inference framework

Given a trained diffusion model, REDI does not require any change to the parameters. It only needs access to the sample trajectory of generated images. We show the inference procedure of REDI in Figure 2. Instead of generating all

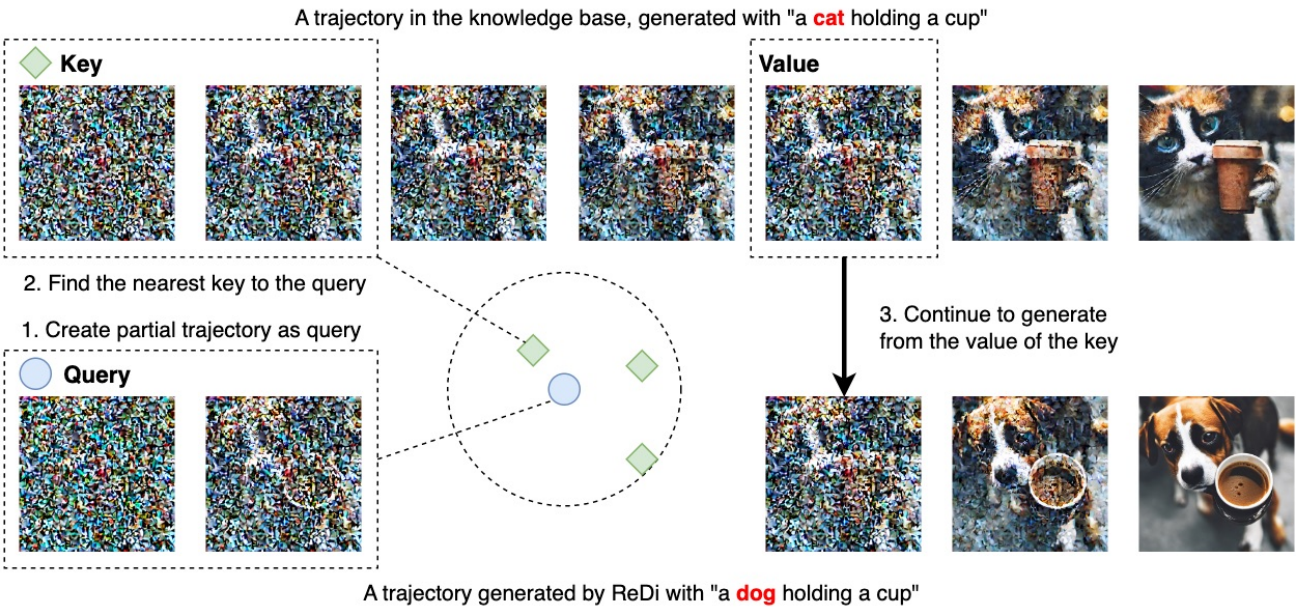


Figure 2. The inference procedure of REDI. The upper part is a complete trajectory generated by Stable Diffusion to build the knowledge base. The lower part is a trajectory generated by REDI with some intermediate steps skipped.

T/δ intermediate samples in the trajectory, REDI skips some of them to reduce number of function estimations (NFEs) and improve efficiency. This is done by generating the first few samples $\boldsymbol{x}_{T..k}$, using them as a query to retrieve a similar trajectory $\boldsymbol{x}'_{T..k}$, and then starting from $\boldsymbol{x}'_v$ of the trajectory retrieved. This way, the NFEs spent to go from time step k to time step v would be unnecessary.

As shown in Figure 2, the retrieval of the similar trajectory $\boldsymbol{x}'$ depends on a precomputed knowledge base $\mathcal{B}$. We formally describe the construction of $\mathcal{B}$ in Algorithm 1. REDI first computes the sample trajectories for the data samples in a dataset $\mathcal{D}$. For every sample trajectory $\{\boldsymbol{x}_{n\delta}\}_{n=T/\delta,\ldots,1,0}$, a sample early in generation $\boldsymbol{x}_k$ is chosen as the key, while a later sample $\boldsymbol{x}_v$ is chosen as the value. The time consumption for computing one key-value pair is similar to that of one generation of the model. The total time consumption is proportional to $|\mathcal{D}|$. With a moderate-sized $\mathcal{D}$, not only can we achieve comparable performance with much less time, we can also perform zero-shot domain adaptation.

The inference process of REDI is formally described in Algorithm 2. Given a guidance signal y (in our case, a text prompt), we want to generate a data sample (in our case, an image) $\boldsymbol{x}$ from it. We first generate the first few steps $\boldsymbol{x}_{T..k}$ to query the knowledge base $\mathcal{B}$ with $\boldsymbol{x}_k$. We find the top-H keys that are closest to q in terms of the distance measure d. Then we find out the set of weights w^* that make the linear combination of the top-H keys approximate q the best. With w^*, we linearly combine the value and use that combination as the initialization point for the remaining steps of the sampling process.

Algorithm 2 REDI Inference

input θ, k, v, S are the same as defined in Algorithm 1;
$\mathcal{B}$ is the knowledge base computed by Algorithm 1;
The guidance signal y;
The number of nearest neighbors H we want to retrieve;
A distance measure $d(\cdot,\cdot)$ between a query and a key.
output A data sample x conditioned on the guidance signal y.
$\boldsymbol{x}_T \sim p(\boldsymbol{x}_T)$
for $i \leftarrow T$ **to** $k+1$ **do**
  $\boldsymbol{x}_{i-1} \leftarrow S_\theta(\boldsymbol{x}_i, i, y)$
end for
$q \leftarrow \boldsymbol{x}_k$ // Computing the first few samples as the query
Find the top-H neighbors $j_1, j_2, \ldots, j_H$ from $\mathcal{B}$ that are nearest to q (measured by d)
$w^* \leftarrow \arg\min_w d\left(q, \sum_{i=1}^{H} w_i key^{(j_i)}\right)$
$\hat{x}_v \leftarrow \sum_{i=1}^{H} w_i^* val^{(j_i)}$
for $i \leftarrow v$ **downto** 1 **do**
  $\hat{x}_{i-1} \leftarrow S_\theta(\hat{x}_i, i, y)$
end for
return $\hat{x}_0$

4.3. Extending REDI for Zero-Shot Domain Adaptation

One limitation of the REDI framework described in subsection 4.2 is its generalizability. When the guidance signal y contains out-of-domain information that does not exist in $\mathcal{B}$, it is difficult to retrieve a similar trajectory from $\mathcal{B}$ and

run REDI. Therefore, We propose an extension to REDI in order to generalize it to out-of-domain guidance. For an out-of-domain guidance signal y, we break it into 2 parts - the domain-agnostic y^{in}, and the domain-specific y^{out}. We use y^{in} to generate the partial trajectory as the retrieval key. After retrieval, we start from the retrieved value with both y^{in} and y^{out} as guidance signal.

For example, under the image synthesis setting, when $\mathcal{B}$ contains style-free images that are generated without any style specifier in the prompt, it is difficult for REDI to synthesize images from a stylistic prompt y because finding a stylistic trajectory from a style-free knowledge base is hard. However, with the proposed extension, REDI is able to synthesize stylistic images.

When a stylistic guidance signal y is given, the part of y describing the content of the image is the domain-agnostic y^{in}, and the part of y specifying the style of the image is the domain-specific y^{out}. Although it is difficult to find a trajectory similar to one generated by $y = (y^{\text{in}}, y^{\text{out}})$, it is feasible to retrieve a trajectory similar to one generated by y^{in}. Therefore, we first use the content description to generate the retrieval key and then use the whole prompt for the following sampling steps to make the image stylistic.

5. Theoretical Analysis

We analyze in this section whether REDI is guaranteed to work. Our theorem is based on two assumptions.

Assumption 5.1. The noise predictor model $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is L_0-Lipschitz.

This assumption is used in Lu et al. (2022a) to prove the convergence of DPM-solver. Salmona et al. (2022) also argues that diffusion models are more capable of fitting multimodal distributions than other generative models like VAEs and GANs because of its small Lipschitz constant. The fact that $\boldsymbol{\epsilon}$ is an estimate of the Gaussian noise added to the original image suggests that a small perturbation in $\boldsymbol{x}_t$ does not change the noise estimation very much.

Assumption 5.2. The distance between the query and the nearest retrieved key is bounded, i.e. $d(q, \text{key}) \leq \varepsilon$.

This assumes that the nearest neighbor that REDI retrieves is "near enough", which we show empirically in section 6.

Given these assumptions, we can prove a theorem saying the distance between the retrieved value and the true sample generated retrieved value is an estimate near enough to the actual $\boldsymbol{x}_v$.

Theorem 5.3. *If* $d(\boldsymbol{x}_k, \textit{key}) \leq \varepsilon$, *then* $d(\boldsymbol{x}_v, \textit{val}) \leq e^{\mathcal{O}(k-v)}\varepsilon$.

Here, x_k is the generated early sample used to retrieve from the knowledge base. key is the k-th sample from a trajectory stored in the knowledge base. val is the v-th sample from the same trajectory as key. $\boldsymbol{x}_v$ is the true v-th sample if we generate the full trajectory using the original diffusion sampler starting from $\boldsymbol{x}_k$.

Proof: We first define $\boldsymbol{h}(\boldsymbol{x}, t) := \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}$ and prove it's Lipschitz continuous. This is equivalent to proving $\left|\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|$ is bounded:

$$\left|\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right| = \left|f(t) - \frac{g^2(t)}{2\sigma(t)}\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{x}}\right| \leq |f(t)| + \left|\frac{g^2(t)}{2\sigma(t)}\right|\left|\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{x}}\right| \tag{11}$$

$$\leq |f(t)| + \left|\frac{g^2(t)}{2\sigma(t)}\right| L_0. \tag{12}$$

Note that Equation 12 follows from the Lipschitz continuity of $\boldsymbol{\epsilon}$. $|f(t)| + \left|\frac{g^2(t)}{2\sigma(t)}\right| L_0$ is bounded by the bounds of f and g, which is determined by the range of t.

Since $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}$ is L-Lipschitz, the sensitivity of ODE (Khalil, 2002) suggests that for any two solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ to Equation 10 satisfies

$$|\boldsymbol{x}_v - \boldsymbol{y}_v| \leq e^{L(k-v)}|\boldsymbol{x}_k - \boldsymbol{y}_k|. \tag{13}$$

We summarize the proof of ODE sensitivity in Appendix A.1 to provide a more thorough perspective to our proof. With 13, the theorem is proven because key and val are y_v and y_k from a solution to Equation 10 according to Algorithm 1. □

This theorem can be interpreted as that if the retrieved trajectory is near enough, the retrieved $\boldsymbol{x}'_v$ would be a good-enough surrogate for the actual $\boldsymbol{x}_V$. Note that multiple factors affect the proven bound, namely the Lipschitz constant L, the distance to the retrieved trajectory d, and the choice of key and value steps k and v.

6. Experiments

This section presents experiments for REDI applied in both the inference acceleration setting and the stylization setting. We conduct these experiments to investigate the following research questions:

- Is REDI capable of keeping the generation quality while speeding up the inference process?
- Is the sample trajectory a better key for retrieval-based generation than text-image representations?
- Is REDI able to generate data from a new domain without re-constructing the knowledge base?

Table 1. The FID scores ↓ of REDI when it's applied to PNDM. With 20 NFEs, REDI is able to achieve better quality than the 40-step PNDM, making the inference process 2× faster. The NFE of the 20-step REDI is comparable to a 1000-step DDIM sampler.

	NFEs			
SAMPLER	20	30	40	1000
DDIM	-	-	-	0.255
PNDM	0.274	0.268	0.262	-
+REDI, H=1	0.265	0.264	**0.262**	-
+REDI, H=2	**0.255**	**0.247**	**0.252**	-

We conduct our experiments on Stable Diffusion v1.5[1], an implementation of latent diffusion models (Rombach et al., 2022) with 1000M parameters. Our inference code is based on Huggingface Diffusers[2]. To show that REDI is orthogonal to the choice of numerical solver, we conduct experiments with two widely used numerical solvers - pseudo numerical methods (PNDM) (Liu et al., 2021b) and multi-step DPM solvers (Lu et al., 2022a;b). To obtain the top-H neighbors, we use ScaNN (Guo et al., 2020), which adds a neglectable overhead to the generation process for retrieving the nearest neighbors.

6.1. REDI is efficient in precompute and inference

Figure 3. The image samples generated by Stable Diffusion and REDI with the same set of prompts.

To show REDI's ability to sample from trained diffusion models efficiently, we use REDI to sample from Stable Diffusion, with two different numerical solvers, PNDM and DPM-Solver. We build the knowledge base $\mathcal{B}$ upon MS-COCO (Lin et al., 2014) training set (with 82k image-caption pairs) and evaluate the generation quality on 4k random samples from the MS-COCO validation set. We use the Fréchet inception distance (FID) (Heusel et al., 2017) of the generated images and the real images. In practice, we choose L^2-norm as our distance measure d and calculate the optimal combination of neighbors using least square estimation.

[1]https://github.com/CompVis/stable-diffusion

[2]https://github.com/huggingface/diffusers

Table 2. The FID scores of REDI when it's applied to DPM-Solver.

	NFEs			
SAMPLER	10	12	15	17
DPM-SOLVER	0.268	0.265	0.263	0.258
+REDI, H=1	0.265	0.259	0.256	0.255
+REDI, H=2	**0.261**	**0.257**	**0.253**	**0.252**

Note that although the training of Stable Diffusion costs as much as 17 GPU years on NVIDIA A100 GPUs, constructing $\mathcal{B}$ with PNDM only requires 1 GPU day, which is much more efficient compared to learning-based methods like progressive distillation (Meng et al., 2022).

For PNDM, we generate 50-sample trajectories to build the knowledge base. We choose the key step k to be 40, making the first 10 samples in the trajectory the key, and alternate the value step v from 30 to 10. Different combinations of key and value steps determine how many NFEs are reduced. For DPM-solver, we choose the length of the trajectory to be 20 and conduct experiments on $k = 5, v = 8/10/13/15$. To compare the performance of REDI with the existing numerical solvers. We use them to generate images with the same NFE budget and compare their FIDs with REDI.

We show the results of our experiments in Table 1 and Table 2. Some samples generated by ReDi are listed in Figure 3. For both PNDM and DPM-Solver, we report the FID scores of the sampler without REDI, REDI with top-1 retrieval, and REDI with top-2 retrieval. The results indicate that with the same number of NFEs, REDI is able to generate images with a better FID. They also indicate that when REDI is combined with numerical solvers, we can achieve better performance with only 40% to 50% of the NFEs. In terms of clock time, 50-step PNDM takes 2.94 seconds, while the 30-step REDI takes 1.75 seconds to generate one sample. The precomputation time for building the vector retrieval index and retrieving top-h neighbors, when amortized by 4000 inferences, is 0.0077 seconds, which only takes 0.4% of the inference time. Therefore, REDI is capable of keeping the generation quality while improving inference efficiency.

We further validate the effectiveness of REDI with more ablation studies and baselines in Appendix A.6.

6.2. Early samples from trajectories are better retrieval keys than text-image representations

One major difference between REDI and previous retrieval-based samplers is the keys we use. Instead of using text-image representations like CLIP embeddings, we use an early step from the sample trajectory as the key. Theoretically, this is the optimal solution, because we are di-

Table 3. The L2 distance and FID scores of using FID and trajectory as the retrieval key.

NFE	KEY	L2 NORM ↓	FID ↓
40	CLIP	9.03	0.2709
	TRAJECTORY	7.95	0.2626
30	CLIP	8.53	0.2784
	TRAJECTORY	7.28	0.2643

rectly minimalizing the bound from 5.35.3 by minimalizing $|\boldsymbol{x}_k - \boldsymbol{y}_k|$. In this section, we show empirically that sample trajectory is indeed the better retrieval key.

We compare CLIP embeddings and trajectories by using them alternatively as the key. To use CLIP embeddings as key, we build a knowledge base similar to the one we use in subsection 6.1 with them, where the keys are CLIP embeddings and the values are later samples in the sample trajectory. We run the same inference process except with a different knowledge base.

To evaluate the performance of retrieval keys, we use two criteria - the distance between the actual value and the retrieved value $d(\boldsymbol{x}_v, \hat{\boldsymbol{x}}_v)$, and the FID scores of the generated images. We report the results in Table 3. Under all NFE settings, the trajectories serve a better purpose as the key for REDI than CLIP embeddings. This corresponds well to Theorem 5.3.

6.3. REDI can perform zero-shot domain adaptation without a domain-specific knowledge base

We use the extended REDI framework from subsection 4.3 to generate domain-specific images. In particular, we conduct experiments on image stylization (Fu et al., 2022; Feng et al., 2022) with the style-free knowledge base $\mathcal{B}$ from 6.1. To generate an image with a specific style, we do not build a knowledge base from stylistic images. Instead, we use the content description y^{content} to generate the partial trajectory as key. After retrieval, we change the prompt from y^{content} to the combination of y^{content} and y^{style}, where y^{style} is the style description in the prompt.

In our experiments, we transfer the validation set of MS COCO to three different styles. The content description y^{content} directly comes from the image captions, while the style description y^{style} is appended as a suffix to the prompt. The style descriptions for the three styles are "*a Chinese painting*", "*by Claude Monet*", and "*by Vincent van Gogh*".

Unlike other experiments, in this experiment, we choose $K = 47$ and $V = 40$. Because from the preliminary experiments, we find that the style of the image is determined much earlier than its detailed content. For 100 randomly sampled captions from MS COCO validation, we generate the corresponding images in all three styles. To evaluate REDI's performance, we compare them with images generated by the original Stable Diffusion with PNDM solver.

We asked human evaluators from Amazon MTurk to evaluate the generated images. They are paid more than the local minimum wage. Every generated image is rated in three aspects, text faithfulness, style consistency, and layout similarity. Every aspect is rated on a scale of 1 to 5 where 5 stands for the highest level. Textual faithfulness represents how faithful the image is depicting the content description. Style consistency represents how consistent the image is to the specified style. Layout similarity represents how similar the layout of the stylistic image is to the layout of the style-free image.

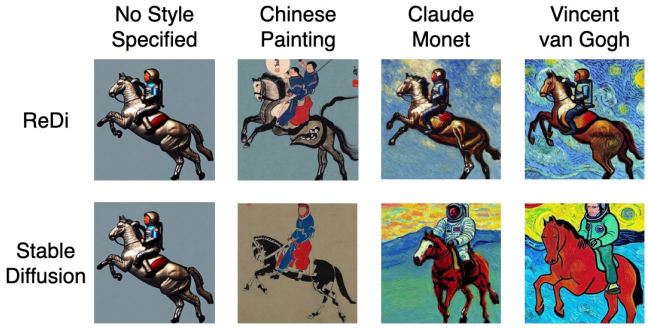


Figure 5. The image samples generated by Stable Diffusion and REDI with the prompt "*an astronaut riding a horse*" in different styles. Without extra precomputation, REDI is able to control the style and keep the same layout, while speeding up the inference.

We report the results of the human evaluation of the 3 different styles in Figure 4. In terms of text faithfulness and style, REDI's evaluation is comparable to Stable Diffusion. In terms of layout similarity, the images generated by REDI have significantly more similar layouts. This indicates that by retrieving the early sample in the trajectory, REDI is able to keep the layout unchanged while transferring the style. This finding is also demonstrated by the qualitative examples in Figure 5.

Furthermore, for stylistic prompts that can not be explicitly decomposed into y^{content} and y^{style}, we propose to use prompt engineering techniques from the natural language processing community by asking a large language model to conduct the decomposition for us. We include an example of such method in Appendix A.7.

7. The Tightness of the Theoretical Bound

In this section, we conduct experiments of ReDi on PNDM to show that the proven bound is tight enough to be an

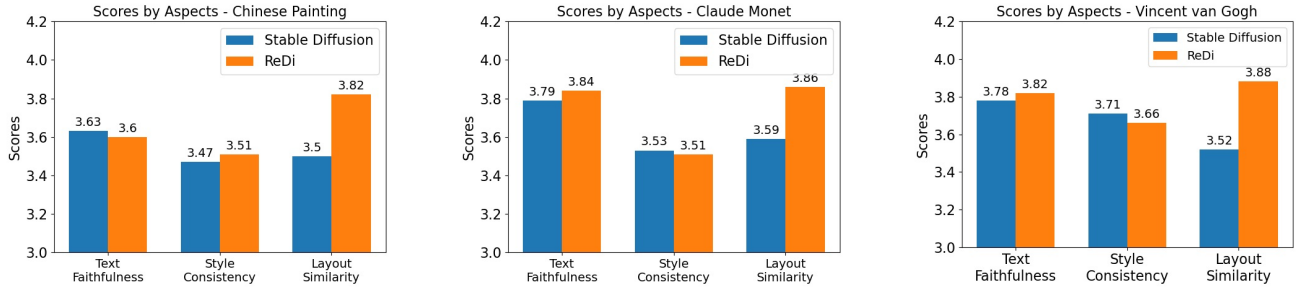
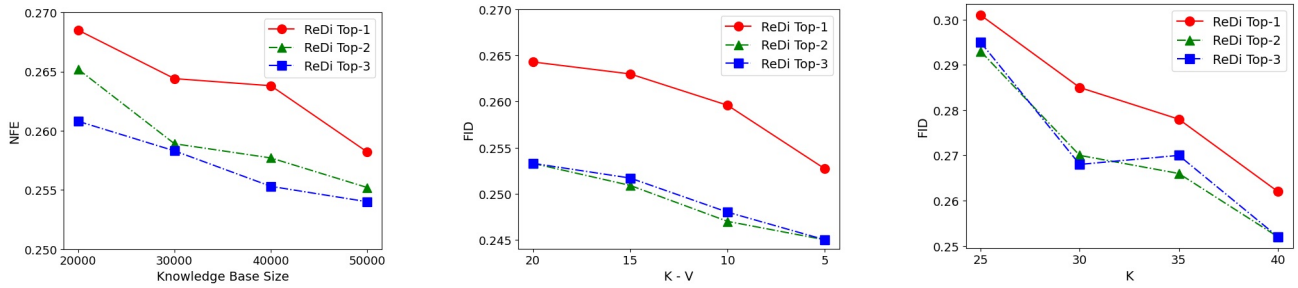


Figure 4. Textual faithfulness, style consistency, and layout similarity scores for Stable Diffusion and ReDi, in 3 different styles. In terms of layout similarity, ReDi is significantly better than Stable Diffusion, indicating that ReDi is able to control the style without changing the layout.



(a) ReDi with different KB sizes (b) ReDi with different skip sizes (c) ReDi with key steps

Figure 6. The FID scores of ReDi applied to two numerical solvers for diffusion models, PNDM and DPM-Solver.

estimator of the actual performance. The bound proven in Theorem 5.3 is affected by the following factors:

- The distance to the retrieved neighbor $|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k|$ which depends on the size of the knowledge base $|\mathcal{B}|$ and the number of nearest neighbors H used.
- The difference between the key step and the query step $k - v$, which depends on k and v.
- The Lipschitz constant L which depends on k and v.

Therefore, we conduct ablation studies on the knowledge base size, the choice of k and v, and the number of neighbors H to check if the performance ReDi correlates well with the theoretical bound. In all ablation experiments, we use PNDM as the numerical sampler and evaluate the performance using FID scores on samples generated from MSCOCO validation. Our findings indicate that the performance of ReDi correlates well with the theoretical bound. Therefore the proven bound can be a good estimator for model performance.

Knowledge base size We control the size of the knowledge base by randomly sampling from the complete knowledge base described in subsection 6.1. We experiment with knowledge bases of sizes 20K, 30K, 40K and 50K. As shown in Figure 6a, as the knowledge base size gets smaller, the performance of REDI drops. This finding is consistent with the theoretical bound since the bound is proportional to $|\boldsymbol{x}_k - \boldsymbol{x}'_k|$.

Key and query steps There are two questions to be studied about how key and query steps influence the performance of ReDi - the difference between them $k - v$, and the choice of k. The bound from Theorem 5.3 is proportional to the exponential of $k - v$. Therefore, we control the difference between the key and the query steps by fixing the key step $k = 40$, and alternating v to be $35, 30, 25$. As shown in Figure 6b, as $K - V$ gets bigger, the performance of ReDi drops. This finding is consistent with the theoretical bound since the bound is proportional to e^{k-v}. The bound is also proportional to e^L. Even if the difference between k and v is fixed, the choice of k alone can affect L and the performance. We keep $k - v = 15$ and alternate k to be $40, 30, 20$. As shown in Figure 6c, as k gets smaller, the performance of ReDi drops. This finding is consistent with the theoretical bound since the bound of Stable Diffusion explodes when $t \to 0$ (Liu et al., 2021b).

Number of neighbors retrieved Increasing the number of neighbors can make the approximation of the query more accurate. Therefore, we control the number of neighbors

in every ablation experiment to evaluate how H affects the performance. As shown in Figure 6, the performance of ReDi rises as the number of neighbors increases. This correlates well with the theoretical bound. We also find the difference between two Hs converges as H gets to 3.

8. Discussion

Generalizability with respect to guidance scale The knowledge base for REDI is built with the guidance scale of 7.5. We investigate whether REDI works for other guidance scales. The results and generated sample images are listed in Appendix A.2. The results show that REDI is still able to function and produce similar-quality results when the guidance weight is not the same as the one used in the knowledge base.

Complicated and compositional prompts Due to the retrieval nature of REDI, it is possible that when prompted with complicated and compositional prompts, there may not be a trajectory in $\mathcal{B}$ that's close enough to the prompt. To qualitatively evaluate REDI's performance on complicated prompts, we pick the 4 longest prompts in the test set and compare the generated images of REDI and PNDM. The samples are shown in Appendix A.3. From the image samples with the longest prompts, we observe that ReDi does not show an inferior ability of compositionality compared with Stable Diffusion. Sometimes it shows better compositionality than Stable Diffusion. However, to enable better compositionality, it's better to have a larger knowledge base.

Impact on sample diversity When the knowledge base is small, it's possible that the diversity of the generated samples can be limited to a smaller space around the data points in the knowledge base. We investigate how REDI affects the sample diversity by computing the Inception Score (Salimans et al., 2016) for the generated images and the ground truth images in Appendix A.4. The results indicate that REDI is capable of generating diverse images.

Beyond the image modality We have only utilized REDI under the text-to-image setting. However, it may be extended to other modalities. We argue that the extension is possible because DPM-solver, which has the same Lipschitz assumption as ours, is used in other domains. Our theoretical analysis (Theorem 5.3) is based on the Lipschitz assumption (Assumption 5.1), which is also the theoretical basis for the performance of the DPM-Solver (Lu et al., 2022a). Variations of DPM-Solver have demonstrated effectiveness in diverse domains, including audio, video (Ruan et al., 2023), and molecular graph (Huang et al.). Therefore, we contend that REDI may also be applicable in these cases.

9. Conclusion

This paper proposes REDI, a learning-free diffusion inference framework via trajectory retrieval. Unlike previous learning-free samplers that extensively explore more efficient numerical solvers, we focus on utilizing the sensitivity of the diffusion ODE, which leads to our choice of using partial trajectories as query and key for retrieval. We prove a bound for REDI with the Lipschitz continuity of the diffusion model. The experiments on Stable Diffusion empirically verify REDI's capability to generate comparable images with improved efficiency. The zero-shot domain adaptation experiments shed light on further usage of ReDi and call for a better understanding of diffusion inference trajectories. We look forward to future studies built upon ReDi and the properties of the diffusion ODE. To make the best out of the limited knowledge base, it is also desirable to study ReDi in a compositionality setting so that the combination of different trajectories can be more controllable.

Acknowledgements

This work is partially supported by unrestricted gifts from IGSB and Meta via the Institute for Energy Efficiency.

We thank Weixi Feng, Tsu-Jui Fu, Jiabao Ji, Yujian Liu and Qiucheng Wu for helpful discussions and proof-reading.

References

Bellman, R. The stability of solutions of linear differential equations. *Duke Math. J.*, 10(1):643–647, 1943.

Blattmann, A., Rombach, R., Oktay, K., Müller, J., and Ommer, B. Semi-parametric neural image synthesis. In *Advances in Neural Information Processing Systems*, 2022.

Chen, W., Hu, H., Saharia, C., and Cohen, W. W. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022.

Chung, H., Sim, B., and Ye, J. C. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12413–12422, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Feng, W., He, X., Fu, T.-J., Jampani, V., Akula, A. R., Narayana, P., Basu, S., Wang, X. E., and Wang, W. Y. Training-free structured diffusion guidance for compositional text-to-image synthesis. *ArXiv*, abs/2212.05032, 2022.

Fu, T.-J., Wang, X. E., and Wang, W. Y. Language-driven artistic style transfer. In *European Conference on Computer Vision*, 2022.

Gronwall, T. H. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, pp. 292–296, 1919.

Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., and Kumar, S. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1908.10396.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Huang, H., Sun, L., Du, B., and Lv, W. Conditional diffusion based on discrete graph structures for molecular graph generation. In *NeurIPS 2022 Workshop on Score-Based Methods*.

Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., and Mitliagkas, I. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

Khalil, H. K. Nonlinear systems third edition. *Prentice Hall*, 115, 2002.

Kim, B. and Ye, J. C. Denoising mcmc for accelerating diffusion-based generative models. *arXiv preprint arXiv:2209.14593*, 2022.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Lam, M. W., Wang, J., Su, D., and Yu, D. Bddm: Bilateral denoising diffusion models for fast and high-quality speech synthesis. In *International Conference on Learning Representations*, 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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, J., Li, C., Ren, Y., Chen, F., and Zhao, Z. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *AAAI Conference on Artificial Intelligence*, 2021a.

Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021b.

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*, 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. *arXiv preprint arXiv:2206.00927*, 2022b.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022.

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 *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. 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.

Ruan, L., Ma, Y., Yang, H., He, H., Liu, B., Fu, J., Yuan, N. J., Jin, Q., and Guo, B. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10219–10228, 2023.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Salmona, A., de Bortoli, V., Delon, J., and Desolneux, A. Can push-forward generative models fit multimodal distributions? *arXiv preprint arXiv:2206.14476*, 2022.

Sheynin, S., Ashual, O., Polyak, A., Singer, U., Gafni, O., Nachmani, E., and Taigman, Y. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

Watson, D., Chan, W., Ho, J., and Norouzi, M. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2021.

Wu, Q., Liu, Y., Zhao, H., Kale, A., Bui, T., Yu, T., Lin, Z., Zhang, Y., and Chang, S. Uncovering the disentanglement capability in text-to-image diffusion models. *arXiv preprint arXiv:2212.08698*, 2022.

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Zhou, L., Du, Y., and Wu, J. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5826–5835, 2021.

A. Appendix

A.1. Proof of ODE Sensitivity

Theorem A.1. *(Grönwall–Bellman inequality (Gronwall, 1919; Bellman, 1943)) Suppose $\lambda : [a, b] \to \mathbb{R}$ be continuous and $\mu : [a, b] \to \mathbb{R}$ be continuous and non-negative. Let a continuous function $y : [a, b] \to \mathbb{R}$ and for $t \in [a, b]$ it satisfies*

$$y(t) \leq \lambda(t) + \int_a^t \mu(s)y(s)\,\mathrm{d}s,$$

then for t on the same interval

$$y(t) \leq \lambda(t) + \int_0^t \lambda(s)\mu(s)e^{\int_s^t \mu(\tau)\,d\tau}\,\mathrm{d}s$$

Theorem A.2. *(Sensitivity of ODE (Khalil, 2002))*

Let $f(t, x)$ be piecewise continuous in t and L-Lipschitz in x on $[t_0, T] \times R^n$, and let $y : \mathbb{R} \to \mathbb{R}^n$ is the solution of

$$\dot{y} = f(t, y), \quad y(t_0) = y_0$$

and $z : \mathbb{R} \to \mathbb{R}^n$ is the solution of

$$\dot{z} = f(t, z) + g(t, z), \quad z(t_0) = z_0$$

with respect to $y(t), z(t) \in \mathbb{R}^n$ for all $t \in [t_0, T]$. Suppose there exists $\mu > 0$ such that $|g(t, x)| \leq \mu$ for all $(t, x) \in [t_0, T] \times \mathbb{R}^n$ and further suppose $|y_0 - z_0| = \gamma$. Then for any $t \in [t_0, T]$,

$$\mid y(t) - z(t) \mid \leq \gamma e^{L(t-t_0)} + \frac{\mu}{L}(e^{L(t-t_0)} - 1).$$

Proof: First we can write the solutions of y and z by

$$y(t) = y_0 + \int_{t_0}^t f(s, y(s))\,\mathrm{d}s,$$

$$z(t) = z_0 + \int_{t_0}^t (f(s, z(s))) + g(s, z(s))\,\mathrm{d}s.$$

Subtracting the above two solutions yields

$$\mid y(t)-z(t) \mid \leq \mid y_0-z_0 \mid + \int_{t_0}^t \mid g(s, z(s)) \mid \,\mathrm{d}s + \int_{t_0}^t \mid f(s, y(s))-f(s, z(s) \mid \,\mathrm{d}s \leq \gamma+\mu(t-t_0)+\int_{t_0}^t L \mid y(s)-z(s) \mid \,\mathrm{d}s.$$

By letting $\lambda(t) = \gamma + \mu(t - t_0)$ and $\mu(s) = L$, then applying the A.1 to the function $|y(t) - z(t)|$ gives,

$$\mid y(t) - z(t) \mid \leq \gamma + \mu(t - t_0) + \int_{t_0}^t L(\gamma + \mu(s - t_0))e^{L(t-s)}\,\mathrm{d}s.$$

Finally, integrating the last term on the right hand gives,

$$\mid y(t) - z(t) \mid \leq \gamma + \mu(t - t_0) - \gamma - \mu(t - t_0) + \gamma e^{L(t-t_0)} + \int_{t_0}^t \mu e^{L(t-s)}\,\mathrm{d}s$$

$$= \gamma e^{L(t-t_0)} + \frac{\mu}{L}(e^{L(t-t_0)} - 1),$$

which finishes the proof.

In our setting, the $g(t, z)$ is simply equal to 0, so we have the following inequality

$$\mid y(t) - z(t) \mid \leq \gamma e^{L(t-t_0)}.$$

A.2. REDI's Performance with Different Guidance Signals

We list the FID scores of REDI with different guidance scales in Table 4. We also show generate samples using different scales in Figure 7.

Table 4. The FID scores ↓ of REDI with different guidance scales. Note that the knowledge base is built with a guidance scale of 7.5. the performance of ReDi experiences a slight drop when the guidance weight differs from the one used in the knowledge base.

GUIDANCE	5.5	6.5	7.5	8.5	9.5
FID	0.271	0.269	0.264	0.268	0.267

Figure 7. REDI-generated images with different guidance scales.

A.3. Samples with Complicated Prompts Generated by REDI

We show the image samples generated by REDI and PNDM from the longest prompts in Figure 8. ReDi does not show an inferior ability of compositionality compared with Stable Diffusion. Sometimes it shows better compositionality than Stable Diffusion. For example, In row 3, Stable Diffusion incorrectly generated a mask covering the rider's face when it should be covering the horse's face. In row 4, Stable Diffusion failed to generate the catcher and umpire in the background, whereas ReDi successfully did. These findings demonstrate the potential of ReDi to handle complex prompts.

Figure 8. REDI-generated images with the longest prompts. Red stands for the parts REDI missed. Green stands for the part PNDM missed. Blue stands for the parts both methods missed.

A.4. Evaluation of Sample Diversity

As shown in Table 5, the Inception Scores for REDI are comparable to those of the ground truth images in the MS-COCO validation set. Based on these results, we argue that REDI is capable of generating diverse images.

Table 5. The Inception Scores ↑ of REDI with different NFEs compared to the Inception Score of the ground truth images. The Inception Score is maximized when the images are evenly distributed and diverse.

NFE	20	30	40
REDI	29.49	30.79	31.23
GROUND TRUTH	30.79	30.79	30.79

A.5. The Initial Sample in Knowledge Base Construction

In Algorithm 1, the sample $\boldsymbol{x}_T$ to initialize every trajectory in the knowledge base is sampled from a conditional distribution $p(\boldsymbol{x}_T|\boldsymbol{x}^{(i)}$. However, the noise schedule in diffusion models causes the signal-to-noise ratio to approach 0 at time step T. This makes sampling from $p(\boldsymbol{x}_T|\boldsymbol{x}^{(i)})$ approximately the same as sampling from the unconditional distribution $p(\boldsymbol{x}_T)$. This makes it possible for REDI to construct the knowledge base without the ground truth images and with only the prompts. We investigate this possibility by rebuilding the knowledge base with initial samples from the unconditional distribution. The results reported in Table 6 indicate that whether the initial distribution is conditional does not significantly affect the performance of REDI.

Table 6. The FID Scores ↓ of REDI when the knowledge base is built with the unconditional initial distribution and the conditional one.

NFE	20	30	40
$\boldsymbol{x}_T \sim p(\boldsymbol{x}_T)$	0.268	0.263	0.265
$\boldsymbol{x}_T \sim p(\boldsymbol{x}_T\|\boldsymbol{x}_0)$	0.265	0.264	0.262

A.6. Validation of Effectiveness

Vanilla skipping from k to v We compare it REDI with a naive substitute - directly approximation of $\boldsymbol{x}_v$ from $\boldsymbol{x}_k$ using any numerical solver, without resorting to retrieval. The average L2 distances to the true value $\boldsymbol{x}_v$ for both the REDI retrieved $\hat{\boldsymbol{x}}_v$ and the vanilla skipped $\tilde{x}_v$ is reported in Table 7. It can be noted that REDI retrieved values are much closer to the true value, indicating the effectiveness of REDI. We also compute the FID for vanilla skipping under the PNDM $k = 40, v = 20$ setting. The FID of vanilla skipping is 1.58, which is much worse than the 0.267 from REDI.

Table 7. The L2 distances between the true value $\boldsymbol{x}_v$ and the estimated values.

NFE	20	30	40
$d(\boldsymbol{x}_v, \hat{\boldsymbol{x}}_v)$	7.95	18.73	7.95
$d(\boldsymbol{x}_v, \tilde{\boldsymbol{x}}_v)$	44.20	24.16	44.20

Similarity with Original Samples We compute the distance between REDI-generated samples and samples generated by the original sampler and list them in Table 8. The distance between ReDi and the original samples is very small (with an average 25% percentage) compared to the distance between the original samples and the ground truth. Therefore we argue that ReDi does not introduce much error to the original sampling trajectory.

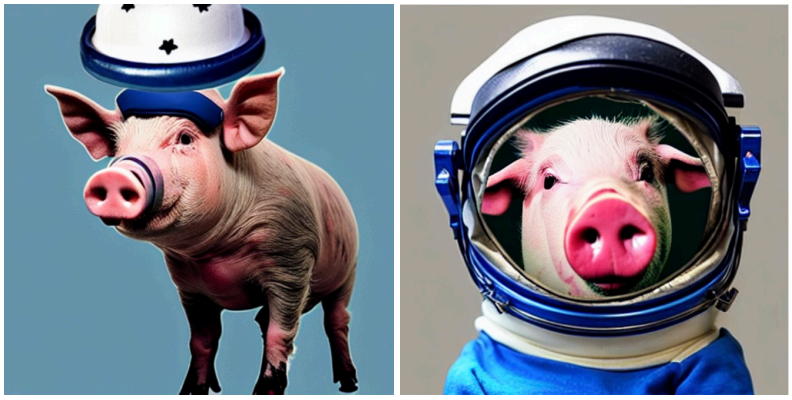
Table 8. The FID between the REDI and original samples, original samples and ground truth.

NFE	20	30	40
FID(REDI, ORIGINAL)	0.0745	0.0655	0.0632
FID(ORIGINAL, GROUND TRUTH)	0.274	0.268	0.272
PERCENTAGE	27.0%	24.4%	24.1%

A.7. Prompt Engineering for Explicit Style-Content Decomposition

We demonstrate how we can automatically decompose an out-of-domain prompt into y^{in} and y^{out} with the prompt *"A photo of a pig wearing an astronaut hat."*.

Unlike prompts with stylistic suffixes, this prompt cannot be easily split into an in-domain prefix and an out-of-domain suffix. Even with the original diffusion model, the generation quality is not satisfactory, as shown in Figure 9a.



(a) Sample from Stable Diffusion (b) Sample from ReDi

Figure 9. Generation samples from Stable Diffusion and ReDi using the prompt *"A photo of a pig wearing an astronaut hat"*.

We then asked GPT-4 to re-write the prompt with minimal changes to a more usual one (which is used as y^{in}) and then generate the image with ReDi. The response from GPT-4 is shown in Figure 10. GPT-4 is able to find out what is unusual about the prompt and re-write it to "a photo of a person wearing an astronaut hat". With that, REDI can generate much better samples as shown in Figure 9b.

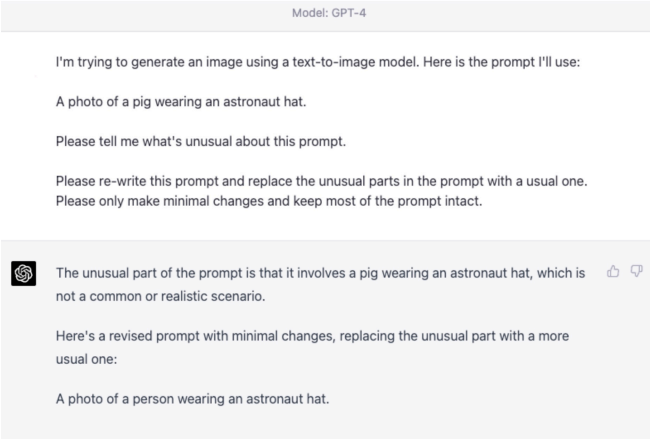


Figure 10. Response from GPT-4.