READ: Retrieval-Enhanced Asymmetric Diffusion for Motion Planning

Takeru Oba†, Matthew Walter‡, Norimichi Ukita†
† Toyota Technological Institute, ‡ Toyota Technological Institute at Chicago
sd21502@toyota-ti.ac.jp, mwalter@ttic.edu, ukita@toyota-ti.ac.jp

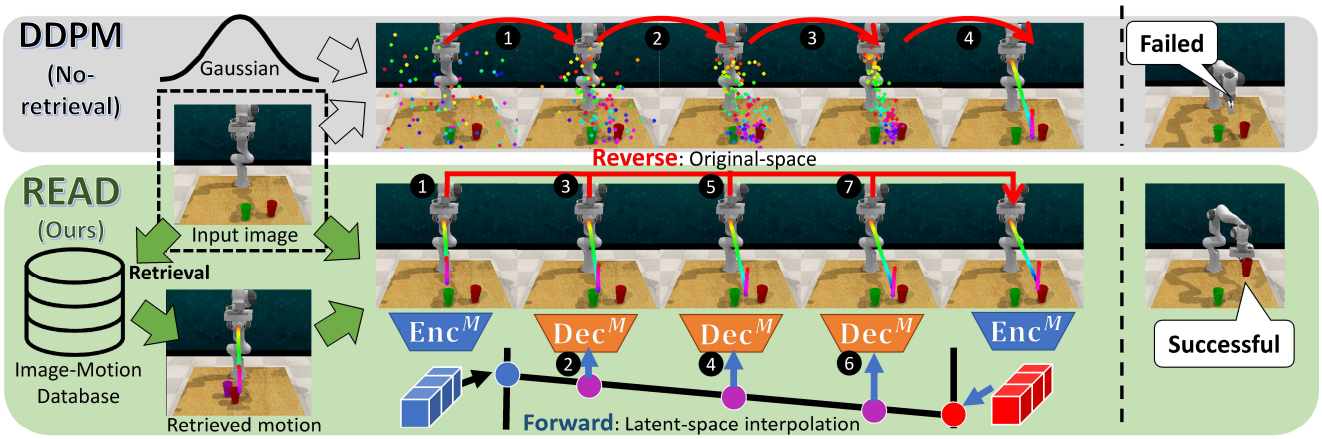


Figure 1. Given an image of the scene, READ retrieves a candidate initial motion from a database of image-motion pairs based on image similarity. READ then refines this motion through asymmetric diffusion that models the forward process as interpolation in a latent space and then performs one-step refinement via reverse diffusion in the original space. Alternating iterative forward and reverse processes, indicated by encircled numbers, improve the final prediction (i.e., motion). While READ refines only **motions**, images are also displayed for visualization. Temporal poses composing each motion are visualized as colored dots overlaid on the image.

Abstract

This paper proposes Retrieval-Enhanced Asymmetric Diffusion (READ) for image-based robot motion planning. Given an image of the scene, READ retrieves an initial motion from a database of image-motion pairs, and uses a diffusion model to refine the motion for the given scene. Unlike prior retrieval-based diffusion models that require long forward-reverse diffusion paths, READ directly diffuses between the source (retrieved) and target motions, resulting in an efficient diffusion path. A second contribution of READ is its use of asymmetric diffusion, whereby it preserves the kinematic feasibility of the generated motion by forward diffusion in a low-dimensional latent space, while achieving high-resolution motion by reverse diffusion in the original task space using cold diffusion. Experimental results on various manipulation tasks demonstrate that READ outperforms state-of-the-art planning methods, while ablation studies elucidate the contributions of asymmetric diffusion. Code: https://github.com/Obat2343/READ

1. Introduction

Image-based motion planning is the problem of producing feasible motions that enable an agent, such as a robotic arm [9, 20] or self-driving car [1, 11], to successfully perform a task (e.g., reach a goal) based on an image of the scene. Typically, there is a diverse set of successful motions (homotopies), e.g., an arm can take different motions to reach the goal, which requires us to model the distribution over successful motions. Generative models provide a promising means to represent these distributions. Among them, diffusion models [19, 34] have been applied to achieve high sample diversity in various domains [13, 22, 32, 48, 49, 52]. For motion planning, however, high sample diversity is insufficient—it is critical that the generated motion is kinematically feasible and successfully performs the task [9, 35, 36]. To achieve both, we propose Retrieval-Enhanced Asymmetric Diffusion (READ), a framework that efficiently refines a candidate motion retrieved from an image-motion database towards a motion that is both feasible and successful in the target scene. READ does so through four key contributions.

(1) **Retrieval enhancement:** Typical stochastic diffusion-based motion planning methods generate motions by running reverse diffusion from a random (e.g., Gaussian) sample, which may result in motion that is infeasible and/or does not reach the goal (Fig. 1 (top) and Fig. 2(a)). Retrieval-based methods use a diffusion process to refine a candidate (retrieved) motion that is assumed to be near the target motion (Fig. 1 (bottom) and Fig. 2(b), 2(c), 2(d)). However, prior retrieval-based methods employ forward diffusion to move towards the initial distribution (e.g., a zero-mean Gaussian) along the path indicated by each black line in Figs. 2(b) and 2(c) and then refine the motion towards the target, resulting in a longer, roundabout path. Instead, READ performs diffusion directly from the retrieved motion to the target (Fig. 2(d)).

(2) **Latent space interpolation:** Forward diffusion has difficulty in preserving the feasibility of the motion in the original high-dimensional task space. To enhance the retrieved motion, READ instead performs interpolation in a latent space (Fig. 2(d)) in which the semantics of the motion, including feasibility, are preserved [4, 6, 10, 53].

(3) **Asymmetric diffusion:** While the latent space supports the generation of feasible motions, the lower dimensionality makes it difficult to model high-resolution motions such as those needed for manipulation. To address this, we propose asymmetric diffusion that performs the forward process in the latent space and the reverse process in the original space to achieve high-resolution refinement.

(4) **Cold diffusion:** Standard forward and reverse diffusion processes operate in the same space. The same is true of applications of cold diffusion [3]; however, because a one-step reverse process in cold diffusion is independent of the forward process, the forward and reverse processes can be designed independently. READ takes advantage of this to perform asymmetric diffusion in the original and latent spaces.

While READ works for a variety of image-based motion planning problems, we evaluate it on robot manipulation. Experiments reveal that READ outperforms contemporary baselines, while a series of ablation studies demonstrate the contributions of the different components of READ.

2. Related work

2.1. Diffusion models

With their sequential application of denoising autoencoders, diffusion models [19, 34, 46, 47] achieve stable learning and produce high-fidelity outputs. Latent space diffusion [8, 41, 50] achieve gains in computational efficiency by learning a distribution over a lower-dimensional latent space.

Generalized diffusion models use various task-specific initializations and perturbations instead of Gaussian noise [2, 28, 29]. For image restoration tasks, IRSDE [28] designs the initialization and perturbations as the interpolation from the source (input) to the target images.

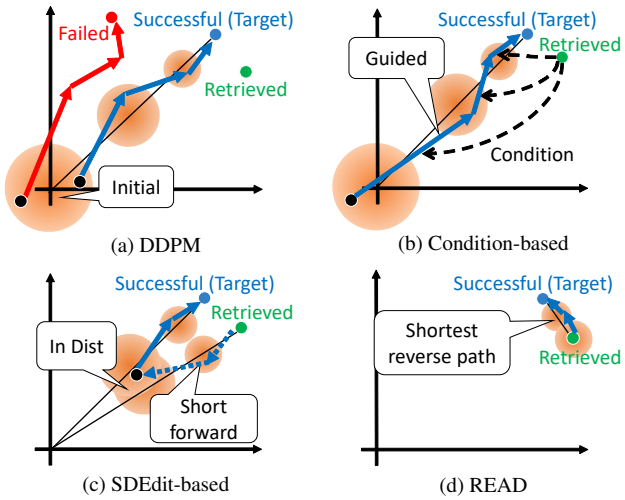


Figure 2. Various reverse processes, each of which is depicted by a solid arrow. Orange circles represent a noise distribution.

Continuous diffusion methods [24, 46, 47] model the diffusion process as a stochastic differential equation (SDE). Drawing on the large body of work in SDE optimization, continuous diffusion models are capable of high-quality generation in fewer steps [14, 23, 24, 30].

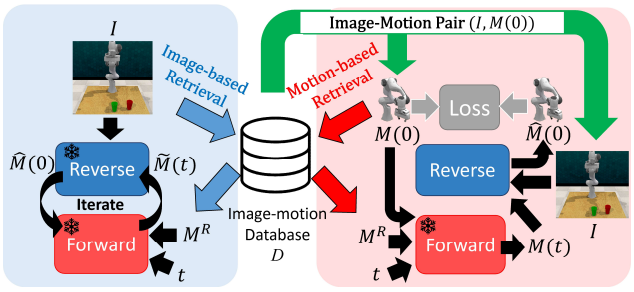
Based on these insights, READ adopts a task-specific initialization with perturbations in the latent space, and employs continuous diffusion.

2.2. Retrieval-based diffusion models

Retrieval is often utilized to improve the performance of generative models [26, 54]. In retrieval-based diffusion models, retrieval data guides the diffusion processes. These models are categorized into condition- [5, 7, 43, 55] and SDEdit-based [31, 36] methods, shown in Figures 2(b) and 2(c), respectively.

During the reverse diffusion process, condition-based methods guide each reverse step, as depicted by each dashed arrow in Fig. 2(b), so that given a sample from the initial noise distribution, the denoised data gradually becomes similar to the retrieved data distribution. For example, RDM [5] and ReMoDiffuse [55] retrieve images and human motions, each of which matches a given text, and use them as conditions for text-to-image generation and text-to-motion generation, respectively.

As shown in Fig. 2(c), SDEdit-based methods [31, 36] replace the initial sample of the reverse process with the retrieved data, allowing the reverse process to start from near-target data. R2-Diff [36] applies SDEdit to image-based motion planning so that the initial motion is retrieved based on image similarity. Our method belongs to the SDEdit-based category in terms of replacing the initial sample with the retrieved motion.



(a) Inference (b) Training

Figure 3. Overview of the training and inference procedures. $\tilde{M}(t)$ is M^R and $M(t)$ if $t = 1$ and $t < 1$, resepctively.

Algorithm 1: Inference Procedure

Input: I
Output: $\hat{M}(0)$

$M^R \leftarrow$ Image-based Retrieval(D, I) ;
$\hat{M}(0) \leftarrow \text{Reverse}(I, M^R, 1)$;
$t \leftarrow 1 - \frac{1}{N}$;
while $t > 0$ **do**
  $M(t) \leftarrow \text{Forward}(\hat{M}(0), M^R, t)$;
  $\hat{M}(0) \leftarrow \text{Reverse}(I, M(t), t)$;
  $t \leftarrow t - \frac{1}{N}$;
end
return $\hat{M}(0)$

3. Retrieval-Enhanced Asymmetric Diffusion

3.1. Preliminaries

We consider image-motion pairs (I, M), where I is an RGB-D image of the robot and its surrounding environment and M is the sequence of robotic end-effector poses $\boldsymbol{p}_{1,\cdots,L}$ from time 1 to L. Each pose $\boldsymbol{p}_l$ at time l consists of its three-dimensional position, six-dimensional orientation [56], and its open/close grasp state. The position is expressed as a combination of the image coordinates (u_l, v_l) and the depth (z_l) of the robotic hand from the camera. The rotation $\boldsymbol{r}_l$ is expressed by a 6D vector, as proposed in [56]. The grasping state g_l is a real value between 0 and 1, representing the closed and open states, respectively. Through the forward process, M is perturbed along the diffusion step t. The perturbed motion at t is denoted by $M(t)$. $M(0)$ is a target motion with no perturbation. Our goal is to predict $M(0)$ from I. The predicted motion is denoted by $\hat{M}(0)$.

3.2. Overview

As depicted in Fig. 3, READ takes as input an image I of the scene and retrieves a similar image from the dataset D. D contains images paired with a valid motion that performs the task depicted in the image. High similarity between the input and retrieved images serves as a proxy to indicate that the retrieved motion (i.e., M^R) is close to a motion that can achieve the task in the target scene depicted in I. While any image similarity criteria can be used, we follow previous work [36] and compute similarity based on differences in image features obtained from the image encoder Enc^I. Then, the retrieved motion M^R is refined by iterating the latent space forward and original space reverse processes. For the forward process in the latent space, READ uses a motion encoder-decoder pair Enc^M and Dec^M.

Inference: Algorithm 1 outlines the inference procedure. As shown in Fig. 3(a), given a query image I, READ retrieves M^R from the database D based on image similarity. M^R is refined by following cold-diffusion-like optimization. First, M^R is fed into the reverse process, which is described in detail in Section 3.4, to predict a successful motion via one-step direct refinement. Its precision is, however, insufficient. For further optimization by iterative refinements the forward process, which is described in detail in Section 3.3, obtains $M(t)$ that is then fed into the reverse process instead of M^R. During this iteration, the continuous step t is decreased from 1 to 0 to progressively approach a successful motion. This iterative refinement scheme works essentially similarly to general discrete diffusion steps. While this iterative refinement scheme improves the image-motion consistency of the final output (denoted by $\hat{M}(0)$) in the high-resolution motion space in the reverse process, the forward process in the latent space maintains motion semantics (i.e., motion feasibility).

Training: Figure 3(b) and Algorithm 2 outline the training procedure of the reverse process. Initially, the method samples each pair I and $M(0)$ from the image-motion training dataset D, as depicted by the green arrow in Fig. 3(b). Then, we retrieve an initial motion M^R similar to $M(0)$. Note that this similarity is evaluated on motion because image feature F^I is under training.

We found that it is important to retrieve not only the most similar motion but also the k-nearest-neighbor motions for data augmentation. That is, each of k retrieved motions is used as M^R for training. k is a hyperparameter; a larger k may enhance robustness in case of retrieval failure, but an excessively large k may introduce a gap between the distribution of retrieved motions during inference and training.

The reverse process is trained so that M^R is refined toward $M(0)$. Given M^R and $M(0)$, $M(t)$ at a continuous step t is obtained by the forward process. t is randomly selected in accordance with the training procedure of a general diffusion model with discrete steps. Then, $M(t)$, I, and t are fed into the reverse process to predict $\hat{M}(0)$. Finally, by minimizing the MSE loss between $M(0)$ and $\hat{M}(0)$, the parameters of the reverse process are updated.

Algorithm 2: Training Procedure

Sample $I, M(0)$ from D ;
$t \sim$ uniform$[0, 1]$;
$M^R \leftarrow$ Motion-based Retrieval(D, $M(0)$) ;
$M(t) \leftarrow$ Forward$(M(0), M^R, t)$;
$\hat{M}(0) \leftarrow$ Reverse$(M(t), I, t)$;
Update Reverse() on ∇Loss$(\hat{M}(0), M(0))$;

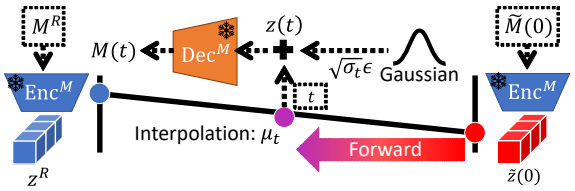


Figure 4. Illustration of our forward process. Black dashed frames are inputs of the forward process.

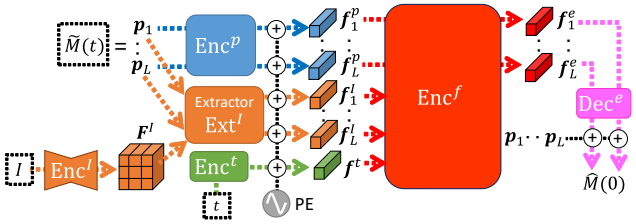


Figure 5. Illustration of our reverse process. Black dashed frames are inputs of the reverse process.

3.3. Forward SDE via latent space interpolation

As shown in Fig. 4 and Algorithm 3, our forward SDE obtains $M(t)$ by interpolation between M^R and $\tilde{M}(0)$, which is $M(0)$ and $\hat{M}(0)$ in the training and inference procedures, respectively, along the continuous diffusion step $t \in [0, 1]$ as follows. $\tilde{M}(0)$ and M^R are embedded into $\tilde{\boldsymbol{z}}(0)$ and $\boldsymbol{z}^R$, respectively, in the latent space by Enc^M. $\tilde{\boldsymbol{z}}(0)$ is then perturbed with the following SDE based on IR-SDE [28]:

$$d\tilde{\boldsymbol{z}} = \theta_t(\boldsymbol{z}^R - \tilde{\boldsymbol{z}}(t))dt + \omega_t d\boldsymbol{w}, \tag{1}$$

where θ_t and ω_t are hyperparameters, and $\boldsymbol{w}$ is the standard Wiener process. θ_t controls the speed to approach from $\tilde{M}(0)$ to M^R, while ω_t controls the stochasticity of perturbation. $\tilde{\boldsymbol{z}}(t)$ can be acquired from μ_t, computed by interpolation between $\boldsymbol{z}^R$ and $\tilde{\boldsymbol{z}}(0)$, by satisfying $\frac{\omega_t^2}{\theta_t} = 2\lambda^2$ for all t, as proven in [28]:

$$\boldsymbol{z}(t) = \boldsymbol{\mu}_t(\tilde{\boldsymbol{z}}(0), \boldsymbol{z}^R) + \sqrt{\sigma_t}\boldsymbol{\epsilon}_t, \tag{2}$$

$$\boldsymbol{\mu}_t(\tilde{\boldsymbol{z}}(0), \boldsymbol{z}^R) := \boldsymbol{z}^R + (\tilde{\boldsymbol{z}}(0) - \boldsymbol{z}^R)e^{-\bar{\theta}_t}, \tag{3}$$

$$\sigma_t := \lambda^2(1 - e^{-2\bar{\theta}_t}), \tag{4}$$

Algorithm 3: Forward SDE

Input: $M(0), M^R, t$
Output: $M(t)$

$\boldsymbol{z}(0), \boldsymbol{z}^R \leftarrow$ Enc$^M(M(0))$, Enc$^M(M^R)$;
$\boldsymbol{z}(t) \leftarrow \boldsymbol{\mu}_t(\boldsymbol{z}(0), \boldsymbol{z}^R) + \sqrt{\sigma_t}\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$;
$M(t) \leftarrow$ Dec$^M(z(t))$;
return $M(t)$;

where $\bar{\theta}_t = \int_0^t \theta_s ds$ and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, I)$. Finally, $\boldsymbol{z}(t)$ is converted to $M(t)$ by Dec^M.

In what follows, we emphasize the difference between our forward SDE and a general forward SDE, VPSDE [47], as follows. $\boldsymbol{\mu}_t$ in VPSDE is expressed by Eq. (5):

$$\boldsymbol{\mu}_t(\boldsymbol{z}(0)) = \boldsymbol{z}(0)e^{-\bar{\theta}_t} \tag{5}$$

In both READ and VPSDE, if $e^{-\bar{\theta}_t} = 1$, $\boldsymbol{\mu}_t = \boldsymbol{z}(0)$. However, if $e^{-\bar{\theta}_t} = 0$, $\boldsymbol{\mu}_t = \mathbf{0}$ in VPSDE, while $\boldsymbol{\mu}_t = \boldsymbol{z}^R$ in our forward SDE. Since $\boldsymbol{\mu}_t$ is linear with respect to $\tilde{\boldsymbol{z}}(0)$ in both SDEs (as expressed in Eq. (3) and Eq. (5) for our SDE and VPSDE, respectively), $e^{-\bar{\theta}_t}$ is regarded as an interpolation ratio. While this interpolation path is long (i.e., between 0 and $\boldsymbol{z}(0)$, which are "Initial" and "Target" in Fig. 2(a), respectively) in VPSDE, it is short (i.e., between $\boldsymbol{z}^R$ and $\boldsymbol{z}(0)$, which are "Retrieved" and "Target" in Fig. 2(d), respectively) in our SDE, making READ successful.

Algorithm 4: Reverse process

Input: $\tilde{M}(t), I, t$ # Note $\boldsymbol{p}_{1,\cdots,L} = \tilde{M}(t)$
Output: $\hat{M}(0)$

$F^I \leftarrow$ Enc$^I(I)$;
$\boldsymbol{f}^I_{1,\cdots,L} \leftarrow$ Ext$^I(F^I, \boldsymbol{p}_{1,\cdots,L}) +$ PE ;
$\boldsymbol{f}^p_{1,\cdots,L} \leftarrow$ Enc$^p(\boldsymbol{p}_{1,\cdots,L}) +$ PE ;
$\boldsymbol{f}^t \leftarrow$ Enc$^t(t) +$ PE;
$\boldsymbol{f}^e_{1,\cdots,L} \leftarrow$ Enc$^f(\boldsymbol{f}^p_{1,\cdots,L}, \boldsymbol{f}^I_{1,\cdots,L}, \boldsymbol{f}^t)$;
$\hat{M}(0) \leftarrow \boldsymbol{p}_{1,\cdots,L} +$ Dec$^e(\boldsymbol{f}^e_{1,\cdots,L})$;
return $\hat{M}(0)$

3.4. Reverse process in the original space

Figure 5 and Algorithm 4 show the procedure of the reverse process. I is encoded into image feature map F^I by Enc^I. The spatial dimension of F^I is the same as I for high-resolution representation. If this large feature map is fed into the following procedure for accurate refinement that requires a complex mechanism (e.g., Transformer), its computational cost becomes heavy. To avoid this problem, we downscale F^I into image feature vectors $\boldsymbol{f}^I_{1,\cdots,L}$ via an image feature extractor Ext^I. If Ext^I is implemented

with any naive downscaling, such as average pooling, high-resolution accurate motion prediction is impossible. For a high-resolution but size-reduced feature map, READ employs Spatially-aligned Temporal Embedding (STE) [35]. STE extracts the motion-relevant features along M in the image coordinate system (i.e., $u_1, v_1, \cdots, u_L, v_L$) from F^I. Note that asymmetric diffusion enables us to adopt STE because $u_1, v_1, \cdots, u_L, v_L$ are available only in the original space. $\boldsymbol{f}^I_{1,\cdots,L}$ is fed into an encoder, Enc^F, with temporal pose and time features (denoted by $\boldsymbol{f}^p_{1,\cdots,L}$ and $\boldsymbol{f}^t$) obtained by their encoders, Enc^p and Enc^t, respectively. Positional Encoding (PE) is applied to $\boldsymbol{f}^I_{1,\cdots,L}$, $\boldsymbol{f}^p_{1,\cdots,L}$ and $\boldsymbol{f}^t$. Finally, $\hat{M}(0)$ is predicted by feeding the output of Enc^F (denoted by $\boldsymbol{f}^e_{1,\cdots,L}$) into Dec^e with skip connections from $\boldsymbol{p}_{1,\cdots,L}$.

3.5. Implementation

While READ can employ arbitrary architectures for each network, we follow those used in Oba and Ukita [36], except Enc^M and Dec^M that do not appear in Oba and Ukita [36], as follows. Enc^I is ConvNext-based UNet [27, 42]. Enc^p, Enc^t and Dec^e are three-layer perceptrons with a gelu activation function [17]. Enc^f is a multi-head transformer encoder [51]. The numbers of heads and layers are four and eight, respectively. For PE, we adopt the sinusoidal positional encoding [51].

The architectures and training procedures of Enc^M and Dec^M follow [39] so that Transfomer-based Enc^M and Dec^M are pre-trained in an autoencoder manner with the VAE loss. Training data for Enc^M and Dec^M are borrowed from motion data included in the image-motion database. In accordance with [8, 41], Enc^M and Dec^M are frozen in the training procedure of the reverse process.

$N = 100$, $k = 3$, and $\lambda = 0.5$. θ_t is linearly scheduled from 0.01 to 2.0. ω_t is derived to satisfy $\frac{\omega_t^2}{\theta_t} = 2\lambda^2$.

4. Experiments

We evaluate READ in comparison to contemporary baselines on 16 manipulation tasks along with a subset of 12 tasks (Fig. 6) from the RLBench [21] benchmark, implemented in the Coppelia simulator [40]. We use a database D with $|D| = 1000$ image-motion pairs for training and 100 pairs for testing. The training dataset is used as a retrieval dataset D in the test phase. All sequences in the database are scaled to length 100 ($L = 100$) via upsampling or downsampling. The size of the image is 256×256. Objects, including manipulation targets and obstacles, are randomly placed, and the camera and the robot's initial position are fixed. Since READ plans the task-space motion of the end-effector, we use inverse kinematics to solve for the corresponding joint angles. RLBench provides an indication of the binary success of each episode (indicated by reward > 1), while motions longer than 100 steps are labeled as having failed. We measure the performance of each method in terms of its average success rates on the sets of 12 (Avg12) and 16 (Avg16) tasks.

We compare READ to the following ten no-retrieval and retrieval baselines:

- **Deterministic**: Deterministic planner trained to minimize MSE between ground-truth and predicted motion.
- **DDPM** [19]
- **RVT** [16]: uses a multi-view transformer to aggregate information across virtual views obtained by re-rendering of the camera input. Note that RVT predicts only keyframe motion rather than the entire motion.
- **VPSDE** [47]: Continuous version of DDPM. using the Runge–Kutta–Fehlberg method [15] for sampling.
- **VINN** [37]: Uses self-supervised learning to obtain the retrieval model and predict the motion with non-parametric Locally Weighted Regression.
- **DMO-EBM** [35]: Refines the retrieved motion based on an energy-based model (EBM) score.
- **VPSDE+CG**: VPSDE [18] with Classifier-free Guidance (CG) [18], which guides the reverse process by conditioning on the retrieved motion M^R as in other work [7, 55].
- **R2-Diff** [36]: An SDEdit-based diffusion model that utilizes the retrieved motion as the initial motion (Fig. 2).

We consider the following variations of READ:

- **READ-O**: The forward and reverse processes are performed in the original task space using the Euler Maruyama method [25], which is a reverse process used in conventional continuous diffusion models.
- **READ-L**: The forward and reverse processes are performed in the latent space. It replaces STE, which cannot be used in the latent space, by downscaling F^I from 256×256 to 16×16. It then flattens the downscaled feature map for the transformer. Note that READ-L also adopts the Euler Maruyama method.

4.1. Main results

Table 1 compares the performance of READ to the no-retrieval and retrieval baselines on the 16 manipulation tasks. Performance numbers for DDPM, VINN, DMO-EBM, and R2-Diff are from Oba and Ukita [36], while we experimented with the other models by ourselves. See our code and supplemental material for implementation details.

READ vs. no-retrieval methods While conventional diffusion models (i.e., DDPM and VPSDE) outperform deterministic models (i.e., Deterministic, RVT) on tasks where the stochasticity of the motion is high (e.g., PC), they struggle on deterministic tasks (e.g., RT). In contrast, READ outperforms each no-retrieval baseline in terms of the average success rate on the 12- and 16-task sets (Avg12 and Avg16) and performs equal to or better on 8 tasks in the 12-task set.

Furthermore, we observed that RVT significantly decreases performance in tasks where motion planning should

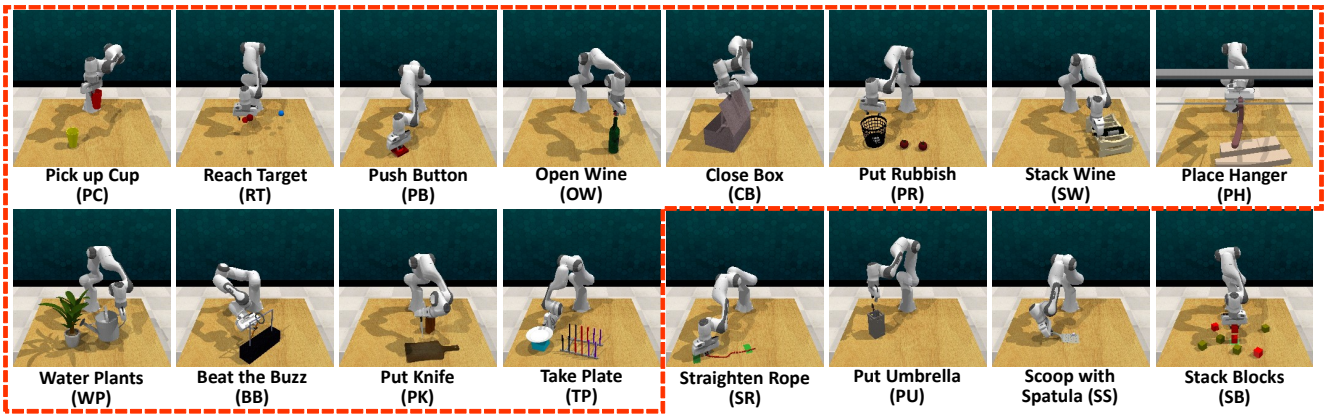


Figure 6. List of tasks. Their abbreviated names are in parentheses. The red dashed line encloses tasks included in Avg12.

Table 1. Success rates of 16 robot manipulation tasks. The best results of Avg16 and Avg12 are in red.

		Avg16	Avg12	PC	RT	PB	OW	CB	PR	SW	PH	WP	BB	PK	TP	SR	PU	SS	SB
no retrieval	Deterministic	56.4	72.7	64	74	88	65	85	77	99	39	95	61	35	91	3	24	3	0
	DDPM [19]	54.4	71.9	96	1	44	70	100	96	98	40	84	84	52	98	0	1	6	1
	VPSDE [47]	50.2	66.4	95	3	96	53	99	68	95	48	59	50	38	93	1	3	1	1
	RVT [16]	65.9	74.3	86	100	100	77	92	92	20	78	8	76	81	92	46	17	80	10
retrieval	VINN [37]	18.7	24.8	5	2	2	20	59	2	45	4	71	25	25	38	0	1	0	0
	DMO-EBM [35]	56.6	70.4	89	32	85	61	97	85	74	40	91	46	51	94	16	25	2	17
	VPSDE+CG [18]	14.6	19.5	26	10	5	6	72	6	30	1	51	7	5	15	0	0	0	0
	R2-Diff [36]	62.9	81.0	95	91	99	72	96	98	96	43	90	56	48	89	23	6	3	1
ours	**READ**	**70.2**	**88.8**	96	99	72	88	100	98	97	57	91	86	82	100	16	32	4	1
	READ-O	60.1	78.5	95	99	83	72	95	99	98	41	86	34	42	96	5	11	4	0
	READ-L	38.5	51.3	89	2	65	63	89	68	39	25	31	84	31	84	1	0	0	0

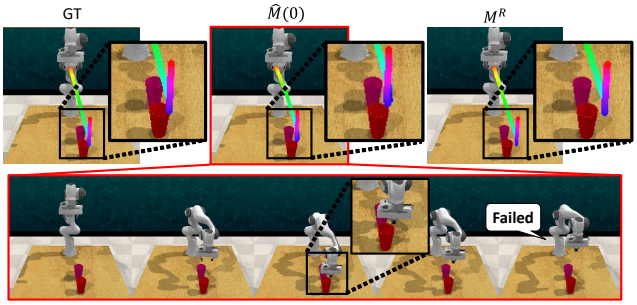


Figure 7. A visualization of a failure case for VPSDE+CG.

adhere to specific constraints. For example, WP requires carrying a watering can without spilling water to achieve the task. However, RVT fails to consider such constraints due to key-frame detection. Conversely, READ achieves high success rates even in such tasks.

READ vs. retrieval-based methods We see in Table 1 that READ achieves higher success rates on the 12- (Avg12) and 16-task (Avg16) sets, and matches if not exceeds their performance on 11 tasks in the 12-task set and 13 of the tasks in the 16-task set. Among all retrieval-based methods, VINN and VPSDE+CG exhibit noticeably lower success rates. In VINN, the retrieval model is trained only from the appearance of objects. In contrast, READ optimizes feature F^I for retrieval through motion refinement. Since the appearance of objects is almost the same for each task, VINN fails to retrieve the motion appropriately, while READ retrieves successful motions in most cases (Section 4.2). Meanwhile, VPSDE+CG fails during refinement rather than retrieval. As shown in Fig. 7, while the predicted motion $\hat{M}(0)$ seems feasible, classifier-free guidance strongly guides $\hat{M}(0)$ towards the retrieved motion M^R. Although the guidance makes small errors, the errors render the motion unsuccessful.

While DMO-EBM and R2-Diff outperform the no-retrieval methods on Avg16, they struggle to accurately refine M^R due to training difficulty [45] and roundabout diffusion path, respectively. READ avoids these issues through asymmetric diffusion. As shown in Fig. 8, READ accurately refines M^R to a successful motion $\hat{M}(0)$.

4.2. READ ablations and analyses

In the following, we perform a more detailed analysis of our READ framework.

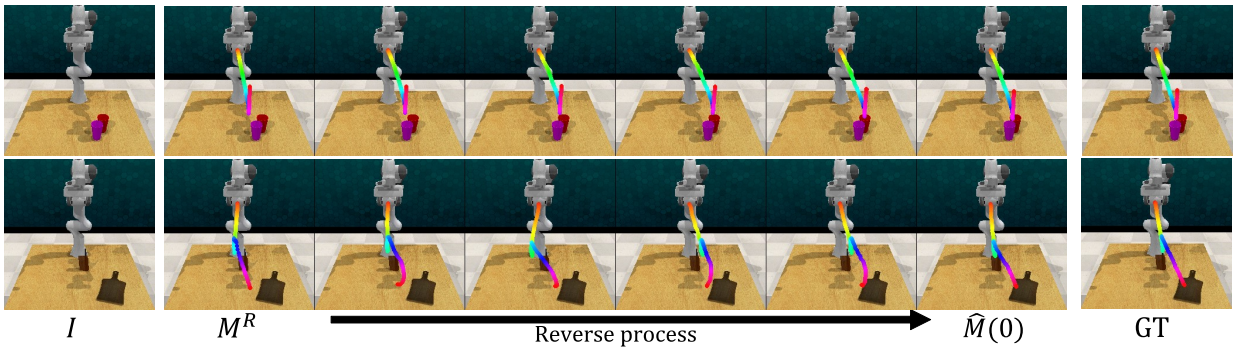


Figure 8. Visualization of motion refinement by READ.

Table 2. Detailed analysis

(a) Refinement

	i	Avg16	Avg12
READ	1	**70.4**	**88.6**
READ-O	1	61.4	78.3
READ-L	1	33.6	44.8
READ	3	**70.7**	**89.3**
READ-O	3	61.1	78.8
READ-L	3	34.1	45.3
READ	10	**71.0**	**89.8**
READ-O	10	60.1	77.3
READ-L	10	34.9	46.5

(b) Retrieval

	Avg16	Avg12
READ	56.6	73.5
READ-O	57.1	75.5
READ-L	52.4	68.9
Cheat	60.6	79.5

(c) EM vs. CD with READ-O

READ-O	Avg16	Avg12
w/ EM	60.1	78.5
w/ CD	58.8	77.4

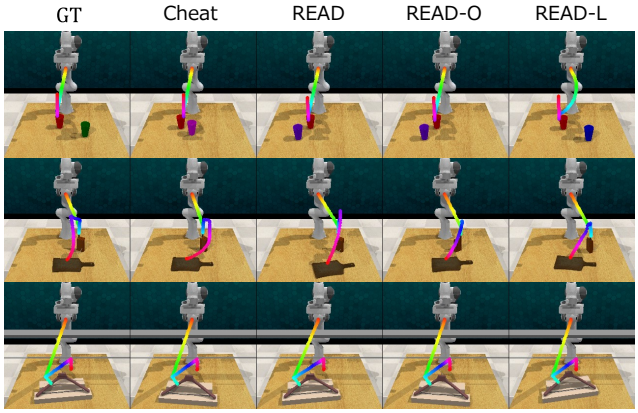


Figure 9. Examples of retrieved motion and image. Retrieved motions are overlaid on the retrieved images

Benefits of Asymmetric Diffusion From Table 1, we see that READ's use of asymmetric diffusion results in better Avg12 and Avg16 success rates compared to the variations of READ that perform diffusion solely in the original task space (READ-O) and latent space (READ-L). Comparing READ-O and READ-L, the results reveal that diffusion in the latent space performs demonstrably worse on average as well as on most of the individual tasks. We isolate the influence of asymmetric diffusion shortly, but together, the results above suggest that READ benefits from performing forward diffusion in the latent space and reverse diffusion in the original space.

Evaluation of retrieval Next, we evaluate the retrieval performance. Since the Enc^I weights differ between READ, READ-O, and READ-L, so will the retrieved motion M^R. Given the diversity of valid paths, which may be from different homotopy groups, it is not straightforward to define a metric that provides a suitable measure of the similarity between motion pairs. Instead, we measure the quality of the retrieved motion M^R in terms of its success rate when executed without refinement (i.e., $N = 0$). As an upper-bound on performance, we consider the motion in the database that is closest to the ground-truth motion, which we refer to as "cheat." We see in Table 2(b) that READ and READ-O perform similarly, with READ-O slightly better, and are only slightly worse than the "cheat" retrieval method, while READ-L is noticeably worse. This suggests that the reverse process in the original space has little effect on the image encoder and, in turn, retrieval. We also observe qualitatively, in Fig. 9, that the visual differences between the retrieved motions are negligible.

Evaluation of refinement via diffusion Next, we isolate the effect of READ's approach to motion refinement. To control for retrieval, we use i^{th} nearest neighbor to the gound-truth motion as the retrieved motion M^R for all methods. We then apply Algorithm 1 to refine M^R. Table 2(a) shows the success rates for READ, READ-O, and READ-L for different values for i. While M^R is the same for all methods, READ achieves higher performance for all i, suggesting that asymmetric diffusion improves the refinement rather than the retrieval.

To better understand the factors that contribute to the advantages of asymmetric diffusion, we consider (i) the space of the reverse process, (ii) the method of the reverse process, and (iii) the space of the forward process.

(i) Forward process in original vs. latent space: The reverse process is performed in the original space in READ and READ-O, while it occurs in the latent space in READ-L. Table 2(a) shows that the success rates of READ-L are

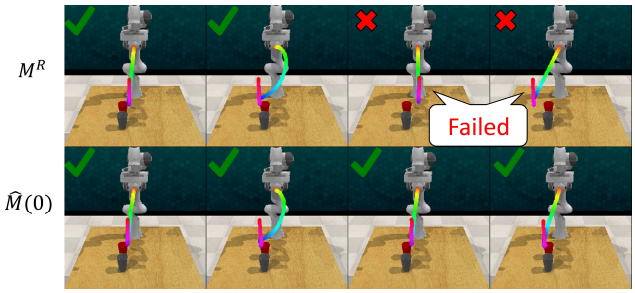


Figure 10. Generated motions from different retrieved motions.

Table 3. Success rates of READ with different hyperparameters.

Setting	N	k	λ	θ_{Max}	Avg16	Avg12
A	100	3	0.5	2.0	**70.2**	**88.8**
B	10	3	0.5	2.0	69.9	88.0
C	1	3	0.5	2.0	68.8	87.1
D	100	10	0.5	2.0	67.1	85.9
E	100	1	0.5	2.0	67.1	85.5
F	100	3	1.0	2.0	68.8	87.4
G	100	3	0.1	2.0	59.5	76.6
H	100	3	0.5	20.0	65.4	84.3
I	100	3	0.5	0.1	67.4	85.9

much lower than those of READ and READ-O, further suggesting that the reverse process in the original space is indispensable in READ.

(ii) Analysis of EM vs. CD: While READ-O and READ-L use the Euler Maruyama method (EM) for diffusion, READ employs cold diffusion (CD). For a fair comparison between the effects of EM and CD, we should compare READ with EM and READ with CD. Asymmetric diffusion in READ, however, can be designed only with CD. Therefore, we compare READ-O with EM to READ-O with CD. As shown in Table 2(c), EM slightly outperforms CD, suggesting that CD does not improve refinement quality.

(iii) Reverse process in original vs. latent space: The difference between READ and READ-O with CD is only the space of the forward process. Since READ achieves almost a 10% higher average success rate than READ-O with CD (Avg16 = 70.2 vs. 58.5), we conclude that performing the forward process in the latent space contributes to READ's performance gain.

4.3. Effect of retrieved motions

Since READ starts the refinement from each retrieved motion M^R, the predicted motion $\hat{M}(0)$ is expected to be influenced by M^R. Figure 10 visualizes examples of $\hat{M}(0)$ for different M^R. As we expected, $\hat{M}(0)$ is similar to M^R.

4.4. Hyperparameter studies

Table 3 shows the average success rates when changing only one of the hyperparameters (N, K, λ, or θ_t) from the best parameter combination (Setting A). We consider linear changes in θ_t from 0.01 to θ_{Max}.

N (the number of iterations): As shown in Table 3, READ achieves high success rates even with a quite small number of iterations (i.e., Avg16 = 68.8 for $N = 1$ in Setting C), demonstrating the effectiveness of our one-step reverse process. Yet, iterative refinement achieves better success rates (Avg16 = 69.9 for Setting B and 70.2 for Setting A).

k (the maximum rank of retrieved motions during training): k changes the degree of data augmentation. While appropriate values improve accuracy (Avg16 = 70.2 in (A) vs. 67.1 in (D)), excessively high values tend to decrease accuracy due to differences in distributions (Avg16 = 70.2 for Setting A vs. 67.1 for Setting E).

λ (scale of the noise in the forward process): λ affects the stochasticity of the forward process. During training, the stochasticity works as a form of data augmentation. During inference, λ handles the diversity of possible motions. While λ should be sufficiently small for retrieval-based inference (Avg16 = 70.2 for Setting A vs. 68.8 for Setting F), a smaller λ decreases the augmentation effect during training (Avg16 = 59.5 for Setting G).

θ_t (velocity of interpolation ratio): θ_t controls the interpolation ratio $e^{-\bar{\theta}_t}$. If θ_t is too small, perturbed motion $M(t)$ cannot reach the retrieved motion M^R, resulting in poor performance (Avg16 = 67.4 in Setting I) due to lack of data augmentation in training. In contrast, if θ_t is too large, the interpolated motion, $M(t)$, cannot be near $\tilde{M}(0)$, resulting in poor performance (Avg16 = 65.4 in Setting H) due to iterative refinement starting far from $\tilde{M}(0)$.

5. Conclusion

We present Retrieval-Enhanced Asymmetric Diffusion (READ) an image-based motion planning framework that generates motions that are both feasible and successful. Unlike the conventional diffusion models, READ adopts latent space interpolation as the forward process, enabling a shorter reverse path from the retrieved motion and preserving the semantics of the motion. Furthermore, we propose asymmetric diffusion to take advantage of the complementary nature of the latent and original task spaces. Experiments on a suite of simulated robot manipulation tasks reveal that READ outperforms contemporary retrieval and non-retrieval methods, while ablations elucidate the role of the different components of READ, particularly the critical nature of asymmetric diffusion for accurate planning.

References

[1] Joonwoo Ahn, Minsoo Kim, and Jaeheung Park. Autonomous driving using imitation learning with look ahead point for semi structured environments. *Scientific Reports*, 12(1):21285, 2022. 1

[2] Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. *arXiv preprint arXiv:2305.10699*, 2023. 2

[3] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022. 2

[4] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 2

[5] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:15309–15324, 2022. 2

[6] Nutan Chen, Alexej Klushyn, Francesco Ferroni, Justin Bayer, and Patrick Van Der Smagt. Learning flat latent manifolds with vaes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1587–1596, 2020. 2

[7] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022. 2, 5

[8] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18000–18010, 2023. 2, 5

[9] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1

[10] Ching-Yao Chuang and Youssef Mroueh. Fair mixup: Fairness via interpolation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 2

[11] Alexander Cui, Sergio Casas, Abbas Sadat, Renjie Liao, and Raquel Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 16107–16116, 2021. 1

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:8780–8794, 2021. 1

[14] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2

[15] J.R. Dormand and P.J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980. 5

[16] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. *arXiv:2306.14896*, 2023. 5, 6

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5, 1

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5, 6

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 1, 2, 5, 6

[20] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16750–16761, 2023. 1

[21] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics Autom. Lett.*, 5(2): 3019–3026, 2020. 5

[22] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022. 1

[23] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 2

[24] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:26565–26577, 2022. 2

[25] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer Berlin Heidelberg, 2011. 5

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. 2

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 5, 1

[28] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *arXiv preprint arXiv:2301.11699*, 2023. 2, 4

[29] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1680–1691, 2023. 2

[30] Hengyuan Ma, Li Zhang, Xiatian Zhu, and Jianfeng Feng. Accelerating score-based generative models with preconditioned diffusion sampling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[31] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2

[32] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4328–4338, 2023. 1

[33] Benjamin Newman, John Hewitt, Percy Liang, and Christopher D Manning. The eos decision and length extrapolation. *arXiv preprint arXiv:2010.07174*, 2020. 4

[34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8162–8171, 2021. 1, 2

[35] Takeru Oba and Norimichi Ukita. Data-driven stochastic motion evaluation and optimization with image by spatially-aligned temporal encoding. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1, 5, 6

[36] Takeru Oba and Norimichi Ukita. R2-diff: Denoising by diffusion as a refinement of retrieved motion for image-based motion prediction. *arXiv*, 2023. 1, 2, 3, 5, 6

[37] Jyothish Pari, Nur Muhammad (Mahi) Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022. 5, 6

[38] Karl Pertsch, Oleh Rybkin, Jingyun Yang, Shenghao Zhou, Konstantinos Derpanis, Kostas Daniilidis, Joseph Lim, and Andrew Jaegle. Keyframing the future: Keyframe discovery for visual prediction and planning. In *Learning for Dynamics and Control*, pages 969–979, 2020. 4

[39] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 10985–10995, 2021. 5

[40] E. Rohmer, S. P. N. Singh, and M. Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013. 5

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 2, 5

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 5, 1

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

[44] Lucy Xiaoyang Shi, Archit Sharma, Tony Z Zhao, and Chelsea Finn. Waypoint-based imitation learning for robotic manipulation. *arXiv preprint arXiv:2307.14326*, 2023. 4

[45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 6

[46] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1415–1428, 2021. 2

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2, 4, 5, 6, 1

[48] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 1

[49] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, 2023. 1

[50] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 5, 1

[52] Takuma Yoneda, Luzhe Sun, Bradly Stadie, Ge Yang, and Matthew Walter. To the noise and back: Diffusion for shared autonomy. *arXiv preprint arXiv:2302.12244*, 2023. 1

[53] Oguz Kaan Yüksel, Sebastian U Stich, Martin Jaggi, and Tatjana Chavdarova. Semantic perturbations with normalizing flows for improved generalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6619–6629, 2021. 2

[54] Kexun Zhang, Xianjun Yang, William Yang Wang, and Lei Li. Redi: efficient learning-free diffusion inference via trajectory retrieval. In *International Conference on Machine Learning*, pages 41770–41785. PMLR, 2023. 2

[55] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. *arXiv preprint arXiv:2304.01116*, 2023. 2, 5

[56] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3