Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Don't Start from Scratch: Behavioral Refinement via Interpolant-based Policy Diffusion

Kaiqi Chen[1], Eugene Lim[1], Kelvin Lin[1], Yiyang Chen[1], and Harold Soh[1,2]
[1]Dept. of Computer Science, National University of Singapore.
[2]Smart Systems Institute, NUS.
Contact Authors: {kaiqi, harold}@comp.nus.edu.sg

***Abstract*—Imitation learning empowers artificial agents to mimic behavior by learning from demonstrations. Recently, diffusion models, which have the ability to model high-dimensional and multimodal distributions, have shown impressive performance on imitation learning tasks. These models learn to shape a policy by diffusing actions (or states) from standard Gaussian noise. However, the target policy to learn is often significantly different from Gaussian and this mismatch can result in poor performance when using a small number of diffusion steps (to improve inference speed) and under limited data. The key idea in this work is that initiating from a more informative source than Gaussian enables diffusion methods to mitigate the above limitations. We contribute both theoretical results, a new method, and empirical findings that show the benefits of using an informative source policy. Our method, which we call BRIDGER, leverages the stochastic interpolants framework to bridge arbitrary policies, thus enabling a flexible approach towards imitation learning. It generalizes prior work in that standard Gaussians can still be applied, but other source policies can be used if available. In experiments on challenging simulation benchmarks and on real robots, BRIDGER outperforms state-of-the-art diffusion policies. We provide further analysis on design considerations when applying BRIDGER. Code for BRIDGER is available at https://github.com/clear-nus/bridger.**

I. Introduction

Imitation learning enables robots to learn policies from demonstrations and has been applied to a variety of domains including manipulation [34, 58, 13], autonomous driving [33, 5], and shared autonomy [39, 56]. Recently, there has been a significant interest in the adaptation of diffusion models for imitation learning [7, 37, 18]. These deep generative models, which progressively transform Gaussian noise to a policy over a number of diffusion steps, offer practical advantages over classical techniques [41, 42] — they scale well with the number of dimensions in the action/state spaces (e.g., for visuo-motor learning on a 7-DoF robot arm [7]) and are able to capture complex multimodal distributions. However, current diffusion methods also require large training datasets and typically have long inference times due to the number of diffusion steps needed to obtain effective action distributions for complex tasks [37].

An examination of existing diffusion-style imitation learning reveals a fundamental issue: these models learn to shape a policy starting from standard Gaussian noise, which is often starkly different from the intended policy or action distribution. The key insight in our work is that initiating from Gaussian noise isn't a prerequisite. To explore this, we move beyond the conventional diffusion framework and employ stochastic interpolants [2] for bridging arbitrary densities within finite time (Fig. 1). This approach allows us to leverage stochastic source policies, enabling the diffusion process to begin from a more informative starting point. Example source policies include policies hand-crafted using prior knowledge of the task or data-driven policies trained on a similar task. We find

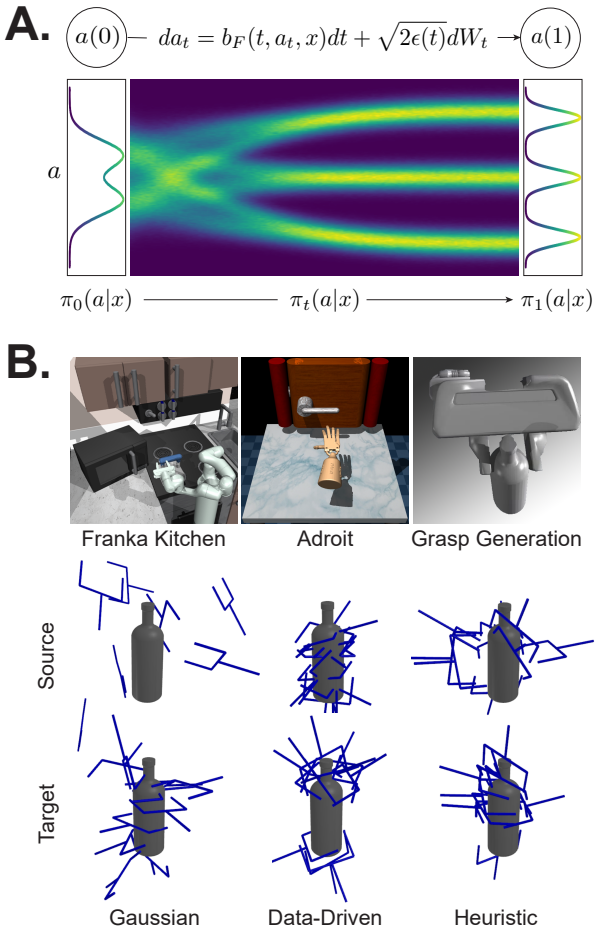

Figure 1: (**A**) Overview of action generation with BRIDGER. With trained velocity b and score s functions, BRIDGER transports the actions from source distribution $\pi_0(a|x)$ to the target distribution $\pi_1(a|x)$ via the forward SDE (Eq. 9). (**B**) We tested BRIDGER on challenging robot benchmark tasks and show that using informative source policies enhances performance. For example, in 6-DoF grasp generation, using heuristic or data-driven source policies results in more successful grasps compared to the conventional Gaussian.

that this shift retains the inherent advantages of diffusion-style imitation learning, but positively impacts inference time and performance. Practically, this leads to faster generation and more accurate robot actions. Our approach also generalizes prior work in diffusion-based imitation learning since if no source policy is available, simple distributions such as the Gaussian can be used.

In this paper, we first contribute a theoretical analysis of the impact of different source policies in diffusion. In brief, we find that under reasonable assumptions, selecting a better source policy results in better target policies. We then turn to a practical approach for incorporating source policies into diffusion methods. Applying the stochastic interpolants framework [2] to imitation learning, we derive a new method called BRIDGER (**B**ehavioral **R**efinement via **I**nterpolant-based **D**iffusion for **Ge**nerative **R**obotics). To our knowledge, our work is the first adaptation of this bridging methodology to imitation learning, contrasting with its previous use in simple synthetic tasks [60] and image generation [60, 25]. In addition to standard neural architecture design for the learnt forward model, the stochastic interpolant framework relies on several critical design choices including the source policy and interpolant. The interpolant dictates how a sampled point transitions from the source to the target distribution, with the transition modulated by noise introduced through time-dependent Gaussian latent variables [2]. Intuitively, the interpolant forms a "bridge" or "guide" between two policies (e.g., it gradually changes poor robot actions into better ones).

We contribute a systematic empirical study of the effects of using source policies (and other design elements) on a diverse set of robot tasks, including the Franka kitchen benchmark, grasp generation, and manipulation using a robot hand. Overall, the experimental results coincide with our theoretical findings; Gaussians were seldom the most effective source distribution and surprisingly, even simple heuristic distributions resulted in superior learnt policies compared to the Gaussian. We demonstrate that given a good source policy, BRIDGER surpasses existing state-of-the-art diffusion policies. Additionally, we discuss the effects of the interpolant function when learning highly multi-modal behaviors. Similar positive results were observed on real-world experiments using two robots: a Franka Emika Panda arm with a two-finger gripper for stable grasping, and a UR5e equipped with a Shadow Dexterous Hand Lite for synthetic wound cleaning. These tasks involved real-world high-dimensional observations (e.g., point clouds and images) and complex actions (22 action dimensions per time-step for the wound cleaning task).

In summary, our work connects distribution bridging to imitation learning, which results in improved performance and addresses inherent limitations of standard diffusion, such as lengthy inference times. We contribute:

- Theoretical results on the impact of diffusing from source policies of varying quality;
- A practical method that enables source policies to be used for diffusion-based imitation learning, which leads to better trade-offs between inference speed and performance;
- A comprehensive empirical study demonstrating the impact of source distributions and interpolant design on outcome quality across various robot tasks.

From a broader perspective, our research demonstrates the potential of bridging models in imitation learning. We hope that this work lays the foundation for future imitation learning methods that leverage past policies for lifelong robot learning.

II. Preliminaries: Background & Related Work

A. Problem Formulation

In imitation learning [20, 59, 51], we wish to learn a policy from expert demonstrations. Let $\pi_1(a|x)$ be an expert policy; it captures the probability of an expert selecting an action a given an observation x. Suppose we have a dataset $\mathcal{D} = \{x^{(i)}, a^{(i)}\}_{i=1}^{N}$ drawn from $\pi_1(a|x)$ and let $\pi_0(a|x)$ be a distribution that we can easily sample a from.

Our goal is to learn a model for transporting actions drawn from π_0 to π_1. More concretely, given an observation x, let $\pi_t(a|x)$ over time $t \in [0, 1]$ be distributions on the *bridge* that links $\pi_0(a|x)$ and $\pi_1(a|x)$. Let $\hat{\pi}_t$ be the density of the learned distribution over time $t \in [0, 1]$. We want the generated target density $\hat{\pi}_1$ to match the ground truth density π_1.

Similar to recent research [7], our approach primarily generates *action sequences* rather than single-step actions. Upon completing (or partially completing) an action sequence, the model assimilates a new observation to generate the subsequent action sequence. In the following, a denotes an action sequence or a target pose, depending on the context.

B. Diffusion-based Policy Learning

Diffusion-based policy methods [7, 37] are largely based on Denoising Diffusion Probabilistic Models (DDPM) [17]. These methods operate by progressively adding noise to an action a_1 during a forward process and subsequently, employing a reverse process to learn how to denoise. This is achieved by training a neural network $g_\upsilon(a_{t_k}, t_k)$ to predict the noise $z \sim \mathcal{N}(0, I)$ added to the data sample, within a defined temporal sequence $0 = t_0 < t_1 < \cdots < t_{K-1} < t_K = 1$. The training utilizes a regression loss,

$$\mathcal{L}(\upsilon) = \mathbb{E}\left[\|z - g_\upsilon(\sqrt{\bar{\alpha}_k}a_1 + \sqrt{1-\bar{\alpha}_k}z, t_k)\|^2\right]. \quad (1)$$

Upon completion of training, DDPM utilizes initial samples a_0 drawn from a standard Gaussian distribution and applies a K-step denoising procedure to synthesize the desired output a_1,

$$a_{t_k} = \frac{1}{\sqrt{\alpha_k}}\left(a_{t_{k-1}} - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}}g_\upsilon(a_{t_k}, t_k)\right) + \sigma_k\mathcal{N}(0, I). \quad (2)$$

One limitation of the original DDPM is slow sampling; the number of steps K required is typically large (e.g., hundreds to thousands). To improve sample efficiency, diffusion-based policy methods [7, 37] employ Denoising Diffusion Implicit Models (DDIM) [9], which apply a *deterministic* non-Markovian process to trade-off between sample quality and inference speed.

C. Related Work

BRIDGER builds upon a large body of work in imitation learning, specifically behavior cloning. Behavior cloning has had a long history, with early methods adopting supervised learning techniques, principally regression [5, 34, 34, 58, 13, 38, 48]. Although efficient, regression-based methods were generally unable to capture multi-modal behavior. Later methods used classification methods, by discretizing the action space [41, 42, 4, 52], to address this limitation but were sensitive to hyperparameters (such as the level of discretization) and struggled with high-precision tasks [12].

Modern imitation methods employ generative models, starting with Gaussian Mixture Models [28], then transitioning to more powerful Energy-Based Models (EBMs) [8, 12, 21, 44] and Diffusion Models [40, 7, 32, 31]. Compared to EBMs, diffusion models have demonstrated better training stability [7] and effectiveness in generating consistent, multi-modal action sequences for visual-motor control [40, 7, 32, 31] and hierarchical, long-horizon planning tasks [36, 16, 30, 6, 53]. Recent progress has been made in reducing the computational costs of diffusion methods by adapting the number of diffusion steps [18] or leveraging DDIM [37].

Unlike the above methods, BRIDGER is the first to generalize diffusion-type policy learning to exploit source stochastic policies. As we will see, this leads to better performance compared to strong baselines (such as DDIM) when informative policies are available, especially when using a small number of diffusion steps. Since BRIDGER adapts existing distributions, it can be seen as a few-shot learner, specifically for imitation learning [10, 51]. Existing work on few-shot imitation learning requires either specific prior policies [11, 57, 24] or a hierarchical structure [54]. In contrast, BRIDGER only requires that we are able to sample from the source policy. For fair comparison against few-shot methods, we adopt residual learning which is used for few-shot reinforcement learning [43, 1, 22], planning [29], and shared autonomy [39, 56]. BRIDGER is also related to recent gradient-flow methods [3, 47] but the SDE is over finite time and does not use classifiers to approximate the drift.

III. Bridging Policies: Theoretical Considerations

The central premise of this work is the use of source policies for diffusion-based imitation learning: we posit that starting with a more informative source density facilitates the shaping of the target density. We examine this hypothesis theoretically in this section. Under reasonable assumptions, we show that a "good" source policy can enhance the resultant the target policy up to an addictive factor. Note that these results apply to any diffusion-type model whereby a source distribution is gradually adapted over time to match a target.

Formally, we denote the "difference" between the action distribution $\hat{\pi}$ at time t (conditioned upon observation x) and the expert policy π_1 as

$$\phi_{F,\hat{\pi}}(t,x) = F(\hat{\pi}_t(\cdot|x), \pi_1(\cdot|x)).$$

where $F(\cdot,\cdot)$ is a measure of difference between two distributions (e.g. KL divergence $\mathbb{KL}(\cdot,\cdot)$ and cross-entropy $\mathbb{H}(\cdot,\cdot)$). In our setup, we diffuse from a source distribution $\hat{\pi}$ from time $t=0$ to $t=1$, and would like $\phi_{F,\hat{\pi}}(1,x)$ to be small.

Assumption 1. *There exist constants $\epsilon_{\max} > \epsilon_{\min} > 0$ such that for all t, x,*

$$0 \geq -\epsilon_{\min} \geq \partial_t \phi_{F,\hat{\pi}}(t,x) \geq -\epsilon_{\max}.$$

We can interpret $\epsilon_{\min}\,\mathrm{d}t$ and $\epsilon_{\max}\,\mathrm{d}t$ as the minimum and maximum improvement in the differences towards π_1 after diffusing for some infinitesimal $\mathrm{d}t$ time. We believe is this reasonable given a trained model with limited capacity.

Theorem 1. *Let $\hat{\pi}_0$ and $\hat{\rho}_0$ be two source distributions and given that Assumption 1 holds. Then the improvement of the generated target distribution is bounded by the improvement of the source distribution*

$$\begin{aligned}\phi_{F,\hat{\pi}}(1,x) - \phi_{F,\hat{\rho}}(1,x) \\ \leq \phi_{F,\hat{\pi}}(0,x) - \phi_{F,\hat{\rho}}(0,x) + \epsilon_{\max} - \epsilon_{\min}.\end{aligned}$$

The proof can be found in the Appendix A. Intuitively, Theorem 1 states that if $\hat{\pi}_0$ is a better source distribution than $\hat{\rho}_0$ (i.e., $\phi_{F,\hat{\pi}}(0,x) < \phi_{F,\hat{\rho}}(0,x)$), then after diffusion, $\hat{\pi}_1$ is better than $\hat{\rho}_1$ up to an additive factor of $\epsilon_{\max} - \epsilon_{\min}$. To elaborate, let us rewrite the bound as

$$\phi_{F,\hat{\pi}}(1,x) + d(\hat{\pi}_0, \hat{\rho}_0) - (\epsilon_{\max} - \epsilon_{\min}) \leq \phi_{F,\hat{\rho}}(1,x)$$

where $d(\hat{\pi}_0, \hat{\rho}_0) = \phi_{F,\hat{\rho}}(0,x) - \phi_{F,\hat{\pi}}(0,x)$ is the difference in F between $\hat{\pi}$ and $\hat{\rho}$ at time 0. If $\phi_{F,\hat{\pi}}(0,x) < \phi_{F,\hat{\rho}}(0,x)$, then $d(\hat{\pi}_0, \hat{\rho}_0) > 0$. The positive factor $\epsilon_{\max} - \epsilon_{\min}$ accounts for the variability in improvements during the diffusion process; a greater disparity in the changes when starting from $\hat{\pi}_0$ versus $\hat{\rho}_0$ can influence the quality of the resulting target distributions. If we further assume that the improvements are equal regardless of the initial source, then this factor disappears and we obtain

$$\phi_{F,\hat{\pi}}(1,x) + d(\hat{\pi}_0, \hat{\rho}_0) \leq \phi_{F,\hat{\rho}}(1,x).$$

In practice, it is necessary to discretize time steps for sampling. Next, we extend our theoretical results to discrete time. Suppose we split the domain of time $[0,1]$ into $K+1$ discrete time steps $0 = t_0 < t_1 < \cdots < t_{K-1} < t_K = 1$. We make a similar assumption,

Assumption 2. *There exist constants $\epsilon_{\max} > \epsilon_{\min} > 0$ such that for all t, x,*

$$0 \geq -\epsilon_{\min}\delta t_k \geq \phi_{F,\hat{\pi}}(t_k,x) - \phi_{F,\hat{\pi}}(t_{k-1},x) \geq -\epsilon_{\max}\delta t_k$$

where $k \in \{1,\ldots,K\}$ and $\delta t_k = t_{k+1} - t_k$.

Here, $\epsilon_{\min}\,\delta t_k$ and $\epsilon_{\max}\,\delta t_k$ quantify the minimum and maximum improvement in F towards π_1 after diffusing with step size δt_k.

Theorem 2. *Let $\hat{\pi}_0$ and $\hat{\rho}_0$ be two source distributions and given that Assumption 2 holds. Then the improvement of the*

generated target distribution is bounded by the improvement of the source distribution

$$\phi_{F,\hat{\pi}}(1,x) - \phi_{F,\hat{\rho}}(1,x) \leq \phi_{F,\hat{\pi}}(0,x) - \phi_{F,\hat{\rho}}(0,x) + \epsilon_{\max} - \epsilon_{\min}.$$

Finally, we establish a similar result for the expected cost $\mathbb{E}[c(a|x)]$ when F is the cross-entropy and the expert is Boltzmann rational.

Assumption 3. *The density of the expert policy*

$$\pi_1(a|x) = \frac{1}{Z}\exp(-c(a|x))$$

where Z is a normalizing constant.

Theorem 3. *Let $\hat{\pi}_0$ and $\hat{\rho}_0$ be two source distributions and given that both Assumption 2 and 3 holds for $F = \mathbb{H}$ (cross-entropy). Then for any observation x, we have*

$$\mathbb{E}_{a\sim\hat{\pi}_1}[c(a|x)] - \mathbb{E}_{a\sim\hat{\rho}_1}[c(a|x)] \leq \mathbb{E}_{a\sim\hat{\pi}_0}[c(a|x)] - \mathbb{E}_{a\sim\hat{\rho}_0}[c(a|x)] + \epsilon_{\max} - \epsilon_{\min}.$$

Similar to Theorem 1, if we let $d_c(\hat{\pi}_0, \hat{\rho}_0) = \mathbb{E}_{a\sim\hat{\rho}_0}[c(a|x)] - \mathbb{E}_{a\sim\hat{\pi}_0}[c(a|x)])$ and $\hat{\pi}_0$ is a lower-cost source policy compared to $\hat{\rho}_0$, i.e., $d_c(\hat{\pi}_0, \hat{\rho}_0) > 0$, then under equal improvement $(\epsilon_{\max} - \epsilon_{\min}) = 0$,

$$\mathbb{E}_{\hat{\pi}_1}[c(a|x)] + d_c(\hat{\pi}_0, \hat{\rho}_0) < \mathbb{E}_{\hat{\rho}_1}[c(a|x)].$$

In words, the policy derived from the better source policy achieves a lower expected cost. In the next section, we discuss how we can practically bridge distributions for imitation learning.

IV. Method: BRIDGeR for Imitation Learning

In this section, we present a method that can learn to adapt a source policy to match observed demonstrations. The source policy could be a simple Gaussian, or an action distribution hand-crafted using prior knowledge, or a data-driven policy learned from data. We call our method **B**ehavioral **R**efinement via **I**nterpolant-based **D**iffusion for **Ge**nerative **R**obotics (BRIDGeR).

BRIDGeR is based on Stochastic Interpolants [2], a recently-proposed framework to bridge densities in finite time. At a high-level, a stochastic interpolant is a continuous-time stochastic process $\{y_t\}_t$ that "interpolates" between two arbitrary densities. As a concrete example, consider two densities p_0 and p_1. A simple linear stochastic interpolant is

$$y_t = (1-t)y_0 + ty_1 + \sqrt{2t(1-t)}z \tag{3}$$

where y_0 and y_1 are drawn from p_0 and p_1, respectively, and z is drawn from a standard Gaussian. By construction, the paths of y_t bridge samples from p_0 at time $t = 0$ and from p_1 at $t = 1$. Our goal is to learn how to transport samples along the paths of this interpolant.

In the following, we will present stochastic interpolants more formally and show how they can be adapted to imitation learning. We will then detail the key design elements explored in this work.

A. Stochastic Interpolants for Imitation Learning

Let $(C^r(\mathbb{R}^n))^m$ be the space of r continuously differentiable functions from $\mathbb{R}^m$ to $\mathbb{R}^n$ and $(C_0^r(\mathbb{R}^n))^m$ as the space of compactly supported and r continuously differentiable functions from $\mathbb{R}^m$ to $\mathbb{R}^n$. We extend the definition of stochastic interpolant in [2] to condition on observation x (e.g., an image or joint angles), similar to the concurrent work [19].

Definition 1 (Stochastic Interpolant)**.** Denote n_x and n_a as the dimension of observation and action respectively. Let $x \in \mathbb{R}^{n_x}$ be an observation. Given two probability density functions $\pi_0(a|x)$ and $\pi_1(a|x)$, a stochastic interpolant between $\pi_0(a|x)$ and $\pi_1(a|x)$ is a stochastic process $\{a_t\}_{t\in[0,1]}$ satisfying

$$a_t = I(t, a_0, a_1, x) + \gamma(t)z, \quad t \in [0,1] \tag{4}$$

where

1. $I \in C^2([0,1]\times\mathbb{R}^{n_a}\times\mathbb{R}^{n_a}\times\mathbb{R}^{n_x})^{n_a}$ satisfies the boundary conditions $I(0, a_0, a_1, x) = a_0$ and $I(1, a_0, a_1, x) = a_1$
2. There exist some $C_1 < \infty$ such that $|\partial_t I(t, a_0, a_1, x)| \leq C_1|a_0 - a_1|$ for $t \in [0,1]$.
3. $\gamma(t)$ satisfies $\gamma(0) = \gamma(1) = 0$, $\gamma(t) > 0$ for all $t \in (0,1)$, and $\gamma^2 \in C^2([0,1])$.
4. The pair (a_0, a_1) is drawn from a probability measure v that marginalizes on $\pi_0(\cdot|x)$ and $\pi_1(\cdot|x)$.
5. z is a standard Gaussian random variable independent of (a_0, a_1).

From Definition 1 and Theorem 2.6 of [2], the transport equation between the source and target distribution is

$$\partial_t \pi + \nabla_a \cdot (b\pi) = 0 \tag{5}$$

where $\pi(t, a, x) \coloneqq \pi_t(a|x)$ and the velocity b is defined as

$$b(t, a, x) = \mathbb{E}\left[\partial_t I(t, a_0, a_1, x) + \dot{\gamma}(t)z\right]. \tag{6}$$

We can rewrite the transport equation (Eq. 5) as a forward Fokker-Planck equation, along with the corresponding forward stochastic process [2]. For any $\epsilon \in C_0([0,1])$ with $\epsilon(t) \geq 0$ for all $t \in [0,1]$, the probability density π in equation 4 satisfies the forward Fokker-Planck equation

$$\partial_t \pi + \nabla_a \cdot (b_F \pi) = \epsilon \nabla_a^2 \pi, \quad \pi(0) = \pi_0 \tag{7}$$

where

$$b_F(t, a, x) \coloneqq b(t, a, x) + \epsilon(t)s(t, a, x) \tag{8}$$

and $s(t, a, x) \coloneqq \nabla_a \log \pi(t, a, x)$ is the score of $\pi_t(\cdot|x)$. The solutions of the forward stochastic differentiable equation (SDE) associated with the Fokker-Planck (Eqn. 7) satisfy

$$\mathrm{d}a_t = b_F(t, a_t, x)\,\mathrm{d}t + \sqrt{2\epsilon(t)}\,\mathrm{d}W_t \tag{9}$$

solved forward in time from the initial action $a_0 \sim \pi_0$.

Action Sampling. Using above properties, we can sample actions by transporting samples from the source distribution π_0 to the target distribution π_1, i.e., we solve the forward SDE in Eq. 9 (See Fig. 1). The time interval $t \in [0,1]$ is discretized into points $t_0, \ldots, t_K$, with K denoting the total

number of diffusion steps and δt representing the uniform time step increment between t_k and t_{k+1}. The specifics of the sampling process are outlined in Algorithm 1.

Algorithm 1 BRIDGER Sampling

Input: Current observation x and a sample from the source policy $a_0 \sim \pi_0(a_0|x)$
  for $k \leftarrow 1$ `to` K **do**
    $z \sim \mathcal{N}(0, I)$
    $b_F(t_k, a_{t_k}, x) = b(t_k, x) + \epsilon(t_k)s(t_k, a_{t_k}, x)$
    $a_{t_{k+1}} = a_{t_{(k)}} + b_F(t, a_{t_k}, x)\delta t + \sqrt{2\epsilon(t_k)}z$
  end for
Output: a_1

Model Training. Sampling actions as above requires the velocity b and score s. Suppose that these two functions are parameterized by function approximators (e.g., neural networks), denoted as b_θ and s_η. The velocity b defined in equation 6 can be trained by minimizing,

$$L_b(\theta) = \int_0^1 \mathbb{E}\left[b_\theta(t, a_t, x) - (\partial_t I(t, a_0, a_1, x) + \dot{\gamma}(t)z)\right]^2 dt \tag{10}$$

where a_t is defined in Eqn. (4) and the expectation is taken independently over $\pi_0(a_0|x)$, $\pi_1(a_1|x)$, and $\mathcal{N}(z|0, I)$. Similarly, the score s of the probability density $\pi(t)$ can be trained by minimizing

$$L_s(\eta) = \int_0^1 \mathbb{E}\left[s_\eta(t, a_t, x) + \gamma^{-1}(t)z\right]^2 dt. \tag{11}$$

In practice, predicting $\gamma^{-1}(t)z$ can be difficult since $\gamma^{-1}(t)$ approaches infinity as t approaches 0 or 1. To address this issue, we re-parameterize the score as $s_\eta = \hat{s}_\eta\gamma^{-1}(t)$ in our model. To further improve training stability, we decompose the velocity b as suggested in [2],

$$b(t, a, x) = v(t, a, x) - \dot{\gamma}(t)\gamma(t)s(t, a, x) \tag{12}$$

where v can be trained by minimizing the quadratic objective

$$L_v(\phi) = \int_0^1 \mathbb{E}\left(v_\phi(t, a_t, x) - \partial_t I(t, a_0, a_1, x))\right)^2 dt \tag{13}$$

Given samples in the dataset D comprising tuples (x, a_1), the above objectives can be approximated via Monte-Carlo sampling. Our training algorithm is outlined in Algorithm 2.

B. Design Decisions

To apply BRIDGER in practice, we have to design several key components, specifically the source distribution π_0, the interpolant $I(t, a_0, a_1, x)$, and the noise schedule $\gamma(t)$ and the $\epsilon(t)$. In this work, we will focus on comparing specific source distributions and interpolants.

Source Distributions. As shown by Theorem 1 in Sec. III, a source distribution that closer to the target can yield better policies. Our preliminary experiments with 2D synthetic samples supports this notion (Fig. 2 and 3) — we see that

Algorithm 2 BRIDGER Training

Input: D and batch size N
  while `Not Converged` **do**
    $(x, a_1) \sim D$ and $a_0 \sim \pi_0(a|x)$ ▷ Sample data
    $t \sim \mathrm{U}(0, 1)$ ▷ Uniformly sample t
    Sample $a_t \sim I(t, a_0, a_1, x) + \gamma(t)z$
    Compute losses:
    $L_b(\theta) = \frac{1}{N}\sum\left(b_\theta(t, a_t, x) - (\partial_t I + \dot{\gamma}z)\right)^2$
    $L_s(\eta) = \frac{1}{N}\sum\left(s_\eta(t, a_t, x) + \gamma^{-1}z\right)^2$
    $L_v(\phi) = \frac{1}{N}\sum\left(v_\phi(t, a_t, x) - \partial_t I\right)^2$
    Gradient descent on θ, η and ϕ
  end while
Output: b_θ, s_η and v_ϕ

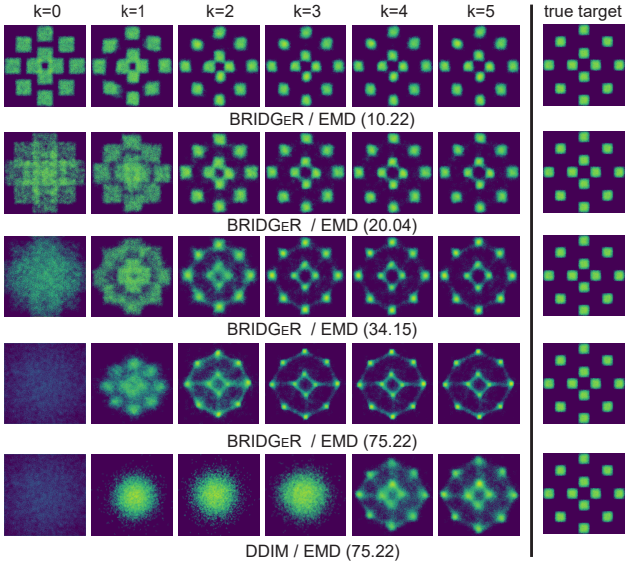

Figure 2: Intermediate distributions obtained from BRIDGER and DDIM trained on 2D synthetic data. With a source distribution that is closer to the target distribution (smaller Earth Mover's Distance (EMD) values), BRIDGER can better recover the true target distribution.

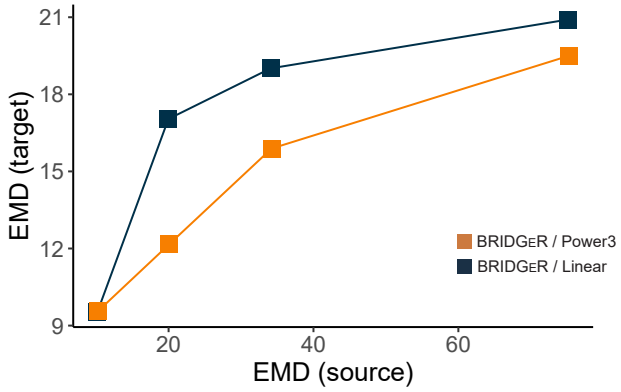

Figure 3: Earth Mover's Distance (EMD) of the generated target distributions under different source distributions and interpolant functions on our 2D synthetic dataset. Each point represents the EMD between a source/target distribution and the true target distribution.

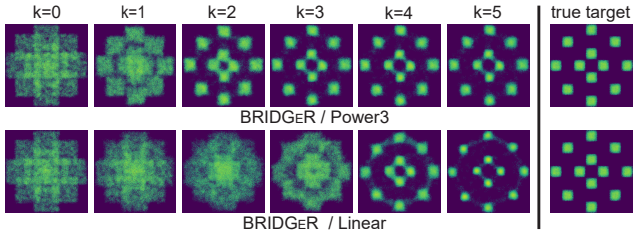

Figure 4: Intermediate distributions under different interpolant functions (trained on 2D synthetic data).

source distributions that are closer to the target (lower EMD) are more similar to the true target distribution. Along with standard Gaussians, our experiments will involve two kinds of source policies:

- *Heuristic policies*: hand-crafted policies (e.g., using rules) based on prior knowledge. Our heuristic policies are task-dependent and detailed in Appendix C.
- *Data-driven policies*: policies learned from a dataset. In our experiments, we use lightweight Conditional Variational Autoencoders (CVAEs) [45] as a representative policy. CVAEs tend to not fully capture complex target distributions, but are cheap to sample from.

These policy types are commonly employed in robotics and by comparing them, we aim to evaluate potential benefits that BRIDGER may offer under different use-cases.

Interpolant Function. The second major component is the interpolant function $I(t, a_0, a_1, x)$. We will use spatially linear interpolants [2], $a_t = \alpha(t)a_0 + \beta(t)a_1 + \gamma(t)z$, specifically,

- *Linear Interpolant* where $\alpha(t) = 1 - t$ and $\beta(t) = t$.
- *Power3 Interpolant* where $\alpha(t) = (1-t)^m$ and $\beta(t) = 1 - (1-t)^m$. In our experiments, we set $m = 3$, which worked well in preliminary tests.

The linear interpolant uniformly progresses samples from the source to the target distribution, which we observed makes training more stable. Conversely, the Power3 interpolant starts with larger steps that decelerate towards the end. This introduces the target pattern sooner than the linear approach, as shown in Fig. 4. Qualitatively, we find Power3 worked better in scenarios with highly multi-modal demonstrations.

Noise Schedule and Diffusion Coefficient. The noise schedule governs the variance of Gaussian latent noise across the bridge. In our experiments, we set $\gamma(t) = d\sqrt{2t(1-t)}$, with d acting as a scalar to adjust γ's magnitude. This configuration results in minimal Gaussian noise at the onset of the transition from the source distribution, increasing to a peak variance before tapering off to zero as samples approach the target distribution. Selecting a larger d facilitates exploration of low-density areas but risks introducing excessive noise, as illustrated in Figure 5. The diffusion coefficient $\epsilon(t)$ controls the level of noise in the forward SDE (Eq. 9). In our experiments, we define ϵ as $\epsilon(t) = c(1-t)$, where c is a scalar to adjust its magnitude. We set two choices for c (1 and 3) and for d (0.03 and 0.3) and reported the best results. In general, our results were relatively robust to these choices.

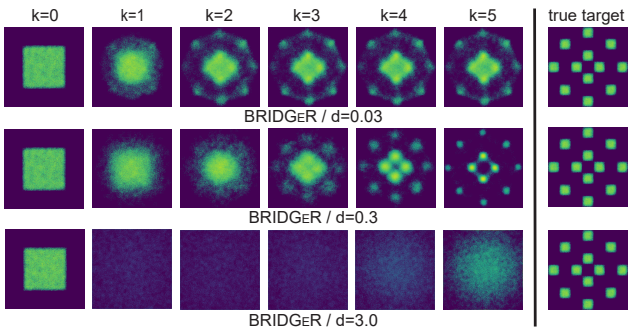

Figure 5: Intermediate distributions with varying $\gamma(t) = d\sqrt{2t(1-t)}$. When the support of the source distribution is narrower than the target distribution, selecting a small γ value ($d = 0.03$) results in samples clustering within the high-density areas of the source. Conversely, an excessively large γ ($d = 3$) results in overdispersion. However, a well-choosen γ ($d = 0.3$) facilitates coverage to ensure reasonable recovery of the target.

V. Experiments

This section describes experiments designed to evaluate the performance of BRIDGER relative to recent methods, particularly diffusion-based imitation learning. More importantly, we aimed to test our hypothesis that leveraging better source distributions within a diffusion framework leads to better policies. We further hypothesized that BRIDGER performs better than DDIM policies given a small number of diffusion steps, and limited data. Finally, we sought to examine the effect of decision decisions, particularly the interpolant choice. We first give an overview of our experimental setup (details relegated to Appendix) followed by our main results.

A. Domains

To evaluate our hypotheses above, we chose six challenging robot benchmarks in three domains [14, 35, 15, 49] (See Fig. 6). The tasks in these domains feature multi-modal demonstrations, which comprise multiple stages with high-dimensional and high-precision actions.

Franka Kitchen (State Observations). The goal in Franka Kitchen is to control a 7 DoF robot arm to solve seven subtasks by reaching a desired state configuration (Fig. 6). Franka Kitchen comprises three datasets and we chose the mixed dataset, which is considered as the most challenging; it presents various subtasks being executed with 4 target subtasks that are not completed in sequence [14]. The model is trained on three varying data sizes: small (16k sequences), medium (32k sequences), large (64k sequences).

Adroit (State Observations). The Adroit tasks require control of a 24-DoF robot hand to accomplish four tasks and is considered one of the most challenging task sets; it requires a model to generate intricate, high-dimensional, and high-precision action sequences over long time horizons. We use the dataset of human demonstrations provided in the DAPG repository [14]. Similar to Franka Kitchen, we train the model on three different data sizes; small (1.25k sequences), medium (2.5k sequences), and large (5k sequences).

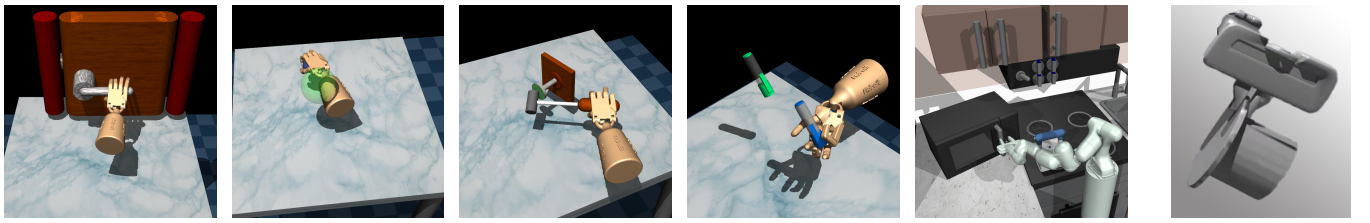

Figure 6: Experimental Domains. **(A)-(D)** Adroit tasks that involve the control of a 24-degree-of-freedom robot hand to accomplish four specific tasks: **(A)** Door: opening a door, **(B)** Relocate: moving a ball to a target position, **(C)** Hammer: driving a nail into a board, and **(D)** Pen: aligning a pen with a target orientation [14]. **(E)** Franka Kitchen includes 7 objects available for interaction and the aim is to accomplish 4 subtasks: opening the microwave, relocating the kettle, flipping the light switch, and sliding open the cabinet door, with arbitrary order. **(F)** The goal of 6-DoF Grasp-Pose generation is to generate grasp poses capable of successfully picking up an object.

6-DoF Grasp Pose Generation (Point-Cloud Observations). The goal here is to generate grasp poses capable of picking up an object given the object's point cloud. The target grasp-pose distribution is multi-modal [49], with high-dimensional point clouds as conditioning observations. BRIDGER is trained on Acronym dataset [50] (552 objects with 200-2000 grasps per object) after a LogMap transformation [49] of the grasp poses.

B. Compared Methods.

BRIDGER was implemented in PyTorch and trained using the Algorithm 2. Additional details regarding hyperparameters and network structure are given in Appendix D. To evaluate performance, we compared BRIDGER against strong baselines, specifically:

- *DDIM*: Diffusion policy [7, 37], a recent diffusion-based imitation-learning method which trains DDPM and applies DDIM [46] during test-time.
- *Residual Policy*: This baseline applies residual learning towards policy learning [43, 1, 22] and is a representative method not based on diffusion.
- *SE3*: A state-of-the-art score-based diffusion model designed in SE(3) [49] specifically designed for generating 6-Dof grasp poses. Note that this baseline only applies to the grasp generation domain.

We also evaluated variants of BRIDGER with different Interpolant functions (Linear and Power3), along with different source policies. As stated in Sec. IV, we investigate three different kinds of source policies:

- *Gaussian*: a standard Gaussian, similar to DDPM.
- *CVAE*: a data-driven policy. For fair comparison to the baselines, the CVAE source policy was trained using the same dataset. Qualitatively, we find the CVAE was generally not able to capture the diversity of demonstrations.
- *Heuristic*: hand-crafted policies for the Hammer, Door, and Grasp tasks (described in Appendix C). Heuristic policies were not available for the other tasks due to their complexity and the high-dimension of the action space.

We label each BRIDGER variant as BRIDGER / $[\cdots]$ / $[\cdots]$. For example, BRIDGER / CVAE / Linear represents BRIDGER with a CVAE source policy and the linear interpolant.

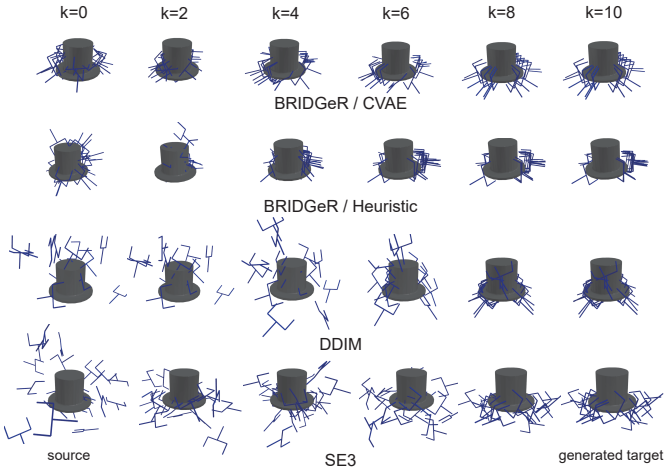

Figure 7: Fifteen sampled grasps across the diffusion steps for BRIDGER, DDIM and SE3. We visualize 15 grasps samples every two diffusion steps until $k = 10$.

C. Test Methodology.

Following prior work, we compute success measures for the different tasks. We report success rate (Adroit) and number of successful tasks (Franka Kitchen), averaged over three different seeds. Under each seed, the models are tested with 100 random initializations for each configuration (dataset sizes [small, medium, large] $\times$ the number of diffusion steps).

For the Grasp task, we evaluated the models in Nvidia Isaac Gym [27] and we report the rate of successful grasps; the robot was able to pick up the object without dropping it [49]. Each object was evaluated using 100 grasps with random initialization of the object pose. We tested our models on both on unseen-objects belonging to the object categories seen in the dataset and unseen-objects from unseen categories. In addition to success rates, we also measured the Earth Mover's Distance (EMD) between the generated grasps and the training data distribution.

D. Main Results and Discussion

In this section, we summarize our key findings, with full results and plots in Appendix F. We focus on our main hypotheses and significant observations.

Table I: Average task performance on Adroit (success rate) and Franka Kitchen (number of successful sub-tasks). Best scores in **bold**. We compare BRIDGER against state-of-the-art methods under a different number of diffusion steps when trained with the Large dataset. BRIDGER with $k = 0$ indicates the source policy. BRIDGER generally outperforms the competing methods. Results are similar for the small and medium datasets with complete results in the Appendix.

		BRIDGER			Residual Policy		DDIM
		CVAE	Heuristic	Gaussian	CVAE	Heuristic	
Door	$k=0$	0.21 ± 0.04	$\mathbf{0.22 \pm 0.06}$	0.00 ± 0.00	0.21 ± 0.04	$\mathbf{0.22 \pm 0.06}$	0.00 ± 0.00
	$k=5$	$\mathbf{0.60 \pm 0.15}$	0.00 ± 0.00	0.08 ± 0.06	0.04 ± 0.06	0.00 ± 0.00	0.02 ± 0.02
	$k=20$	$\mathbf{0.52 \pm 0.23}$	0.45 ± 0.04	0.10 ± 0.07	0.04 ± 0.06	0.00 ± 0.00	0.38 ± 0.11
	$k=80$	0.50 ± 0.14	$\mathbf{0.63 \pm 0.08}$	0.12 ± 0.09	0.04 ± 0.06	0.00 ± 0.00	0.05 ± 0.02
Relocate	$k=0$	$\mathbf{0.31 \pm 0.15}$	-	0.00 ± 0.00	$\mathbf{0.31 \pm 0.15}$	-	0.00 ± 0.00
	$k=5$	$\mathbf{0.75 \pm 0.11}$	-	0.61 ± 0.05	0.3 ± 0.04	-	0.17 ± 0.04
	$k=20$	0.7 ± 0.11	-	$\mathbf{0.72 \pm 0.09}$	0.34 ± 0.04	-	0.37 ± 0.04
	$k=80$	0.79 ± 0.08	-	$\mathbf{0.81 \pm 0.04}$	0.34 ± 0.04	-	0.26 ± 0.03
Hammer	$k=0$	$\mathbf{0.16 \pm 0.03}$	0.11 ± 0.08	0.00 ± 0.00	$\mathbf{0.16 \pm 0.03}$	0.11 ± 0.08	0.00 ± 0.00
	$k=5$	$\mathbf{0.44 \pm 0.11}$	0.2 ± 0.24	0.16 ± 0.04	0.13 ± 0.06	0.24 ± 0.07	0.01 ± 0.01
	$k=20$	0.63 ± 0.08	$\mathbf{0.74 \pm 0.07}$	0.35 ± 0.02	0.13 ± 0.06	0.24 ± 0.07	0.17 ± 0.12
	$k=80$	$\mathbf{0.72 \pm 0.09}$	0.47 ± 0.25	0.43 ± 0.09	0.13 ± 0.06	0.24 ± 0.07	0.03 ± 0.04
Pen	$k=0$	$\mathbf{0.29 \pm 0.05}$	-	0.00 ± 0.00	$\mathbf{0.29 \pm 0.05}$	-	0.00 ± 0.00
	$k=5$	0.45 ± 0.04	-	0.43 ± 0.02	0.33 ± 0.12	-	$\mathbf{0.54 \pm 0.03}$
	$k=20$	0.49 ± 0.09	-	0.53 ± 0.02	0.33 ± 0.12	-	$\mathbf{0.51 \pm 0.03}$
	$k=80$	$\mathbf{0.55 \pm 0.06}$	-	$\mathbf{0.55 \pm 0.03}$	0.33 ± 0.12	-	0.52 ± 0.05
Franka Kitchen	$k=0$	$\mathbf{1.53 \pm 0.09}$	-	0.00 ± 0.00	$\mathbf{1.53 \pm 0.09}$	-	0.00 ± 0.00
	$k=5$	$\mathbf{1.96 \pm 0.03}$	-	1.18 ± 0.02	1.55 ± 0.10	-	1.84 ± 0.06
	$k=20$	$\mathbf{2.09 \pm 0.04}$	-	1.54 ± 0.03	1.55 ± 0.10	-	1.93 ± 0.07
	$k=80$	$\mathbf{2.16 \pm 0.03}$	-	1.70 ± 0.05	1.55 ± 0.10	-	1.92 ± 0.02

Table II: Success rate (averaged over 100 grasps on ten test objects). BRIDGER significantly outperforms DDIM and Residual Policy across the number of diffusion steps. Compared to SE3, BRIDGER achieve higher success rate when the number of diffusion steps is small. We show up to $k = 160$ steps to be consistent with prior reported results [49]. Best scores in **bold**.

		BRIDGER			Residual Policy		DDIM	SE3
		CVAE	Heuristic	Gaussian	CVAE	Heuristic		
Seen Categories	$k=0$	$\mathbf{0.26 \pm 0.02}$	0.06 ± 0.00	0.00 ± 0.00	$\mathbf{0.26 \pm 0.02}$	0.06 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	$k=5$	$\mathbf{0.73 \pm 0.15}$	0.64 ± 0.17	0.56 ± 0.17	0.09 ± 0.15	0.01 ± 0.03	0.52 ± 0.24	0.38 ± 0.26
	$k=20$	$\mathbf{0.93 \pm 0.08}$	0.91 ± 0.07	0.83 ± 0.28	0.09 ± 0.15	0.01 ± 0.03	0.64 ± 0.21	0.78 ± 0.19
	$k=160$	0.88 ± 0.10	$\mathbf{0.91 \pm 0.08}$	0.90 ± 0.06	0.09 ± 0.15	0.01 ± 0.03	0.64 ± 0.26	$\mathbf{0.91 \pm 0.08}$
Unseen Categories	$k=0$	$\mathbf{0.23 \pm 0.04}$	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.23 \pm 0.04}$	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	$k=5$	$\mathbf{0.48 \pm 0.12}$	0.43 ± 0.19	0.45 ± 0.14	0.10 ± 0.16	0.12 ± 0.28	0.33 ± 0.20	0.20 ± 0.10
	$k=20$	$\mathbf{0.67 \pm 0.21}$	$\mathbf{0.67 \pm 0.26}$	0.65 ± 0.25	0.10 ± 0.16	0.12 ± 0.28	0.41 ± 0.23	0.55 ± 0.24
	$k=160$	$\mathbf{0.71 \pm 0.24}$	0.66 ± 0.23	0.64 ± 0.23	0.10 ± 0.16	0.12 ± 0.28	0.35 ± 0.19	0.66 ± 0.25

With a more informative source policy, BRIDGER outperforms the baselines, especially with a small of diffusion steps. We observed that BRIDGER achieves the best success rate across the diffusion steps for many of the tasks. The differences in success rates for Adroit and Franka Kitchen tasks (Table I) are significant when the number of diffusion steps was small $k = 5$; the exception was the Adroit pen task where DDIM performs slightly better. The grasp generation results in Table II further supports this finding, where we see BRIDGER's average success rates surpassing the state-of-the-art SE3 model with $k = 5$ and $k = 20$. For higher diffusion steps, SE3 catches up and the methods appear comparable in terms of success rate. However, SE3 achieves poorer EMD scores compared to BRIDGER (Fig. 9). Fig. 7 shows sample grasps at different diffusion steps; we see that using an informative source policy enables BRIDGER to more quickly converge to a reasonable set of grasps.

Interestingly, we observed that BRIDGER consistently achieved better scores than DDIM and the Residual Policy regardless of the dataset size (See Fig. 8 for the Door task, with plots for other tasks in the Appendix). Potentially, larger datasets may negate the benefit afforded by the source policy since more data could enable DDIM to better generalize.

BRIDGER achieves better performance when using better source policies. Fig. 10 illustrates the difference in success rates between the source and final action distributions. Overall, starting from a source policy with higher success rates tended to result in a better action distributions. This difference can persists even with $k = 80$ steps, with potentially diminishing marginal improvement as illustrated in the Relocate and Franka Kitchen tasks. As suggested by our theoretical results, this could be due to variations in the gradual improvements over the diffusion steps.

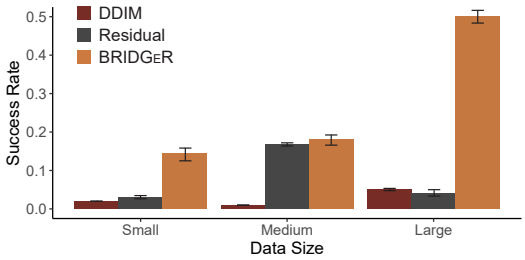

Figure 8: Average success rate under different training dataset on Adroit Door task ($k = 80$ for DDIM and BRIDGER). BRIDGER consistently surpasses baselines across different training data size.

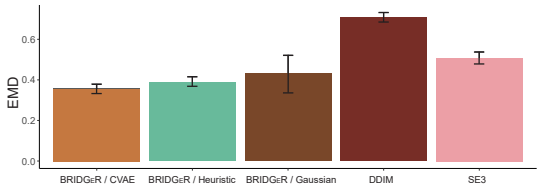

Figure 9: Average Earth Mover's Distance between generated grasps pose and target grasp poses on Seen Categories in Grasp Generation task ($k = 160$ for SE3, DDIM and BRIDGER). Lower EMDs indicate BRIDGER better mimics the dataset distribution.

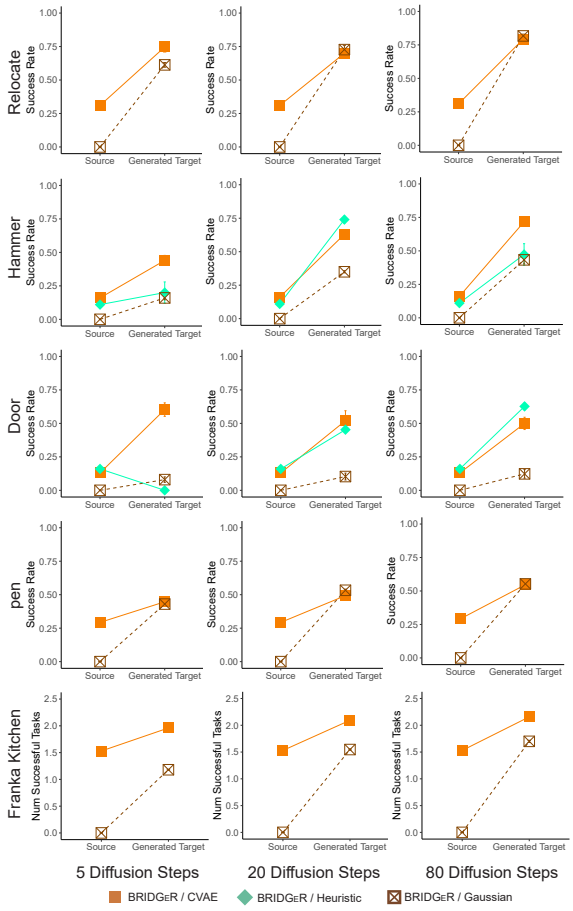

Figure 10: Task performance using different source policies. Each colored line shows the performance of a source policy before ("Source") and after diffusion ("Generated Target").

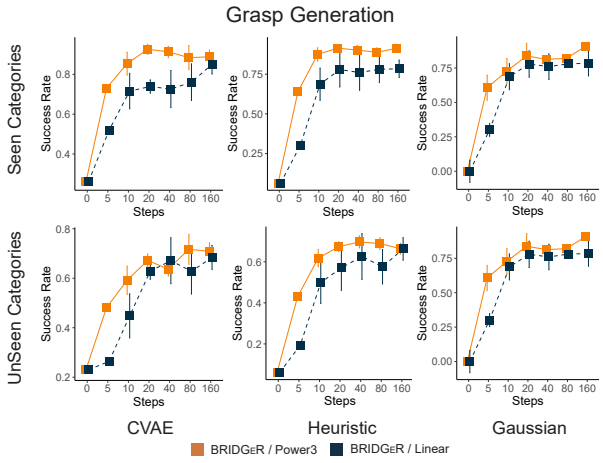

Figure 11: The success rate of BRIDGER with the Power3 and Linear interpolants for grasp generation. In general, Power3 achieves better scores when starting from the same source policy, especially when the number of diffusion steps is small. The error bars represent standard deviations over 10 test objects.

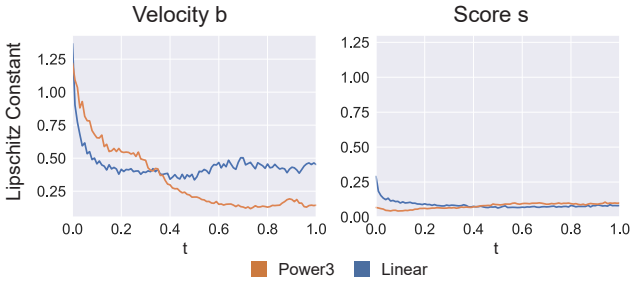

Figure 12: Approximate Lipschitz constants of the velocity b and score s for the Grasp task.

The Power3 interpolant is more appropriate than the Linear interpolant when behavior distributions exhibit high multi-modality. We find that both interpolants perform comparably in tasks like Adroit, where robot behaviors are intricate, but largely uni-modal. In contrast, the Power3 interpolant significantly outperforms the Linear interpolant in Grasp Generation (Fig. 11) where the distribution of end-effector poses is highly multi-modal (Fig. 6.F). This efficiency potentially stems from Power3's rapid convergence to a variety of high-density target areas, followed by fine-scale adaptation. This notion is supported by Fig. 12, which illustrates the smoothness of the velocity and score functions over time (as measured by approximate Lipschitz constants [55]). We observe the Lipschitz constants for Power3 to be larger at the start of the process (a "rougher" function) but gradually falls below that of Linear. This suggests that the function is making more rapid changes at the outset and smaller adaptations towards the middle and end of the diffusion process.

VI. Real World Robot Experiments

In this section, we present findings from experiments aimed at evaluating BRIDGER in real-world domains with noisy and high-dimensional observations (point-cloud and image). As in the previous section, we convey our main results and

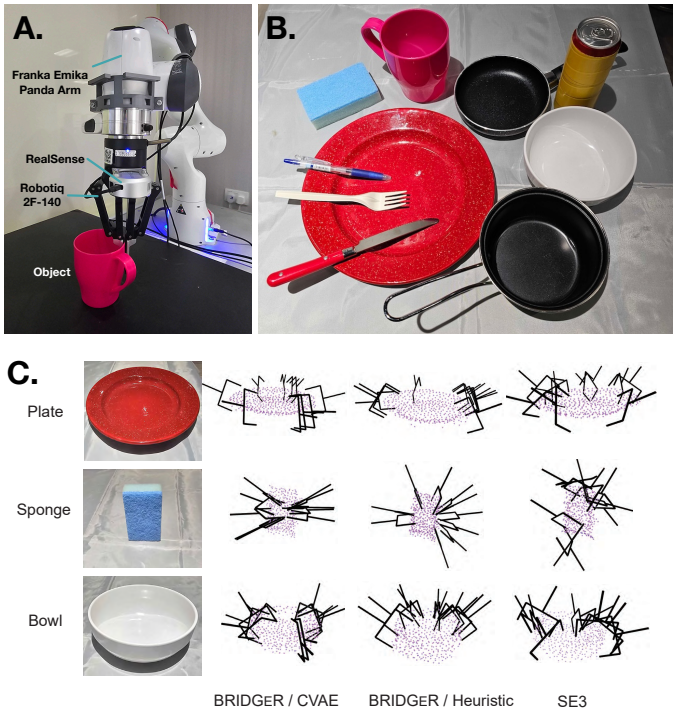

Figure 13: (**A**) Real-world Grasping using a Panda arm with a two-finger gripper. Observations were point clouds obtained from the RealSense Camera on the robot (**B**) Test objects used in our experiments (unseen during training). (**C**) Grasp samples from the competing models on three objects (20 diffusion steps).

Table III: Real-World Grasping Success rate (averaged over 10 grasps on 10 test objects). Best scores shown in **bold**.

	BRIDGER		SE3
	CVAE	Heuristic	
$k=0$	0.07 ± 0.12	$\mathbf{0.02 \pm 0.00}$	0.00 ± 0.00
$k=5$	0.59 ± 0.17	$\mathbf{0.66 \pm 0.13}$	0.05 ± 0.07
$k=20$	0.71 ± 0.18	$\mathbf{0.79 \pm 0.18}$	0.56 ± 0.20
$k=160$	0.75 ± 0.18	$\mathbf{0.81 \pm 0.24}$	0.73 ± 0.21

refer readers to the appendix for further details. Given our simulation results, we hypothesized that BRIDGER would outperform existing state-of-the-art diffusion-based methods under a computation budget and conducted experiments using two tasks:

6-DoF Grasping (Point Cloud Observations) where a Franka-Emika Panda arm has to grasp and lift objects (Fig. 13). The experiment involved 10 everyday objects with 10 grasp trials per object. Objects were perceived using a RealSense camera that provided point cloud observations. We used the models obtained from our simulation experiments and compared BRIDGER (with the Power3 interpolant) against the SE3 diffusion model. We used each method to generate a set of end-effector grasp poses, and the MoveIt motion planner to plan and execute a collision-free path to grasp and lift the objects. A trial was considered successful if the robot managed to lift the object without it falling out of the robot's grasp.

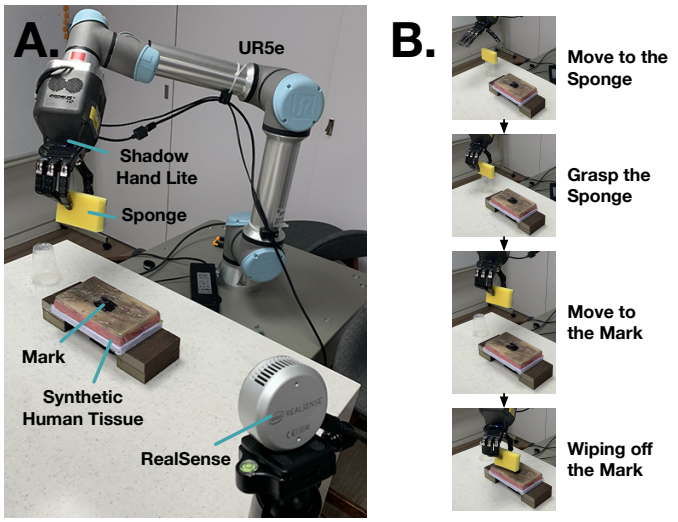

Figure 14: (**A**) Real-world Synthetic Wound Cleaning using a UR5e with Shadow Dextrous Hand Lite. (**B**) Demonstrations consisted of moving the hand to the sponge from an initial position, grasping it, then manipulating the sponge to wipe off the mark. Initial positions were randomized in a region roughly 20-30cm above the sponge and the mark was in one of 9 possible positions. The robot had to learn an action policy conditioned upon RGB images from the RealSense Camera and its joint angles (both arm and hand).

Table IV: Normalized cleaned area (averaged over 9 positions) for the Cleaning Task. Best scores shown in **bold**.

	BRIDGER		DDIM
	CVAE	Heuristic	
$k=0$	0.00 ± 0.00	$\mathbf{0.14 \pm 0.31}$	0.00 ± 0.00
$k=5$	0.46 ± 0.42	$\mathbf{0.50 \pm 0.35}$	0.19 ± 0.25
$k=20$	0.47 ± 0.39	$\mathbf{0.51 \pm 0.33}$	0.27 ± 0.33
$k=80$	$\mathbf{0.32 \pm 0.31}$	0.31 ± 0.29	0.16 ± 0.15

Cleaning (Image and State Observations). Inspired by assistive tasks in healthcare such as wound cleaning and bed-bathing, we used a Shadow Dexterous Hand Lite mounted on a UR-5e arm to perform a synthetic wound cleaning task. The robot had to grasp a sponge and wipe marks off a surgical practice model that mimics human tissue (Fig. 14). This task is challenging as it involves complex high-dimensional action sequences (22 dimensions per time-step, 48 time-steps per prediction), multi-modal demonstrations (e.g., wiping off a mark either from left to right, or from top to bottom), and manipulating a deformable sponge. The mark can be at one of nine possible locations and the robot has to learn appropriate behavior conditioned on visual observations (a RGB image from a RealSense Camera) and its current joint angles. We compared BRIDGER against Diffusion Policy [7] (DDIM) using a normalized cleaned area score, which represents how much of the mark was wiped off. We apply receding-horizon control; the models predict 48 action steps, of which 16 steps of actions are executed on the robot without re-planning. The models were trained using 60 demonstrations provided via kinesthetic teaching and replay.

Table V: Average time to successfully grasp the sponge and roughness of generated action sequences.

	BRIDGER		DDIM
	CVAE	Heuristic	
Time (Seconds)	14.25 ± 3.09	$\mathbf{13.00 \pm 1.87}$	36.25 ± 11.17
Roughness	0.12 ± 0.01	$\mathbf{0.08 \pm 0.01}$	0.34 ± 0.02

A. Results

BRIDGER outperforms the baselines, with larger gaps when the number of diffusion steps is small. For the grasping task, BRIDGER achieves significantly higher success rates compared to SE3 (Table. III). Qualitatively, we observed the grasps generated by BRIDGER to be more accurately positioned (samples shown in Fig. 13), which led to more stable grasps.

For the Cleaning task, Table. IV shows that BRIDGER was better at wiping off the marks compared to the DDIM diffusion policy. Interestingly, performance for all models fell at the largest number of diffusion steps ($k = 80$); a potential cause is that errors accumulate during diffusion given the high-dimensional actions. Nevertheless, BRIDGER attains significantly higher scores. Qualitatively, we observed BRIDGER generates smoother action trajectories (please see the accompanying supplementary videos for examples). At $k = 5$, the DDIM model produced jerkier behavior, with random movements in the arm and fingers. In contrast, BRIDGER's trajectories better mimicked the demonstrations, which led to faster completion of the task. Table V summarizes this observation quantitatively; we see BRIDGER was quicker to grasp the sponge and had lower trajectory roughness (the average norm of the second derivative of the action trajectories).

VII. Conclusions and Future Work

In this work, we investigate the potential of integrating informative source distributions into diffusion-style imitation learning. We provide theoretical results that support this idea and propose BRIDGER, a stochastic interpolant method for imitation learning. Our experiments results show that BRIDGER outperforms strong baselines, including state-of-the-art diffusion policies on various benchmark tasks and real-world robot experiments. We provide additional analyses on our experimental results to elucidate the effect of various design decisions within BRIDGER.

Limitations and Future Work. Here, we have shown that leveraging prior knowledge (in the form of source policies) improves learned diffusion policy performance. BRIDGER opens up avenues to explore different forms of prior knowledge, e.g., policies designed for other tasks (transfer learning) and those constructed using foundation models. Here, we explored a salient but limited set of design considerations; future work can examine more elaborate interpolant functions, along with more in-depth analysis of noise schedules and diffusion coefficients. Finally, we plan to extend BRIDGER incorporate considerations such as safety and user preferences.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its Medium Sized Center for Advanced Robotics Technology Innovation.

References

[1] Minttu Alakuijala, Gabriel Dulac-Arnold, Julien Mairal, Jean Ponce, and Cordelia Schmid. Residual reinforcement learning from demonstrations. *arXiv preprint arXiv:2106.08050*, 2021.

[2] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[3] Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. Refining deep generative models via discriminator gradient flow. In *International Conference on Learning Representations*, 2021.

[4] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. in 2022 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2022.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[6] Lili Chen, Shikhar Bahl, and Deepak Pathak. Playfusion: Skill acquisition via diffusion from language-annotated play. In *Conference on Robot Learning*, pages 2012–2029. PMLR, 2023.

[7] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[8] Gaurav Datta, Ryan Hoque, Anrui Gu, Eugen Solowjow, and Ken Goldberg. Iifl: Implicit interactive fleet learning from heterogeneous human supervisors. In *Conference on Robot Learning*, pages 2340–2356. PMLR, 2023.

[9] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

[10] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *Advances in neural information processing systems*, 30, 2017.

[11] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

[12] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong,

Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[13] Peter Florence, Lucas Manuelli, and Russ Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019.

[14] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[15] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pages 1025–1037. PMLR, 2020.

[16] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[18] Xixi Hu, Bo Liu, Xingchao Liu, et al. Rf-policy: Rectified flows are computation-adaptive decision makers. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[19] Ding Huang, Jian Huang, Ting Li, and Guohao Shen. Conditional stochastic interpolation for generative learning. *arXiv preprint arXiv:2312.05579*, 2023.

[20] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50 (2):1–35, 2017.

[21] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Strictly batch imitation learning by energy-based distribution matching. *Advances in Neural Information Processing Systems*, 33:7354–7365, 2020.

[22] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Jiayi Li, Tao Lu, Xiaoge Cao, Yinghao Cai, and Shuo Wang. Meta-imitation learning by watching video demonstrations. In *International Conference on Learning Representations*, 2021.

[25] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I^2"sb": Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[27] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[28] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pages 1678–1690. PMLR, 2022.

[29] Ajay Mandlekar, Caelan Reed Garrett, Danfei Xu, and Dieter Fox. Human-in-the-loop task and motion planning for imitation learning. In *Conference on Robot Learning*, pages 3030–3060. PMLR, 2023.

[30] Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pages 2905–2925. PMLR, 2023.

[31] Eley Ng, Ziang Liu, and Monroe Kennedy. Diffusion co-policy for synergistic human-robot collaborative tasks. *IEEE Robotics and Automation Letters*, 2023.

[32] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023.

[33] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[34] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3758–3765. IEEE, 2018.

[35] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *Robotics: Science and Systems XIV*, 2018.

[36] Moritz Reuss and Rudolf Lioutikov. Multimodal diffusion transformer for learning from play. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[37] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[38] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction

to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[39] Charles Schaff and Matthew R Walter. Residual policy learning for shared autonomy. *arXiv preprint arXiv:2004.05097*, 2020.

[40] Paul Maria Scheikl, Nicolas Schreiber, Christoph Haas, Niklas Freymuth, Gerhard Neumann, Rudolf Lioutikov, and Franziska Mathis-Ullrich. Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects. *arXiv preprint arXiv:2312.10008*, 2023.

[41] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[42] Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *Conference on robot learning*, pages 906–915. PMLR, 2018.

[43] Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

[44] Sumeet Singh, Stephen Tu, and Vikas Sindhwani. Revisiting energy based models as policies: Ranking noise contrastive estimation and interpolating energy models. *arXiv preprint arXiv:2309.05803*, 2023.

[45] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[47] Tasbolat Taunyazov, Heng Zhang, John Patrick Eala, Na Zhao, and Harold Soh. Refining 6-dof grasps with context-specific classifiers. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6861–6867. IEEE, 2023.

[48] Sam Toyer, Rohin Shah, Andrew Critch, and Stuart Russell. The magical benchmark for robust imitation. *Advances in Neural Information Processing Systems*, 33: 18284–18295, 2020.

[49] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023.

[50] Nikolaus Vahrenkamp, Martin Do, Tamim Asfour, and Rüdiger Dillmann. Integrated grasp and motion planning. In *2010 IEEE International Conference on Robotics and Automation*, pages 2883–2888. IEEE, 2010.

[51] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[52] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial action maps for mobile manipulation. *arXiv preprint arXiv:2004.09141*, 2020.

[53] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning*, pages 2323–2339. PMLR, 2023.

[54] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, pages 3536–3555. PMLR, 2023.

[55] Zhantao Yang, Ruili Feng, Han Zhang, Yujun Shen, Kai Zhu, Lianghua Huang, Yifei Zhang, Yu Liu, Deli Zhao, Jingren Zhou, et al. Eliminating lipschitz singularities in diffusion models. *arXiv preprint arXiv:2306.11251*, 2023.

[56] 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.

[57] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018.

[58] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.

[59] Boyuan Zheng, Sunny Verma, Jianlong Zhou, Ivor W Tsang, and Fang Chen. Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–16, 2022.

[60] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.