FRMD: Fast Robot Motion Diffusion with Consistency-Distilled Movement Primitives for Smooth Action Generation

Xirui Shi 1,† and Jun Jin 1,2

Abstract—We consider the problem of using diffusion models to generate fast, smooth, and temporally consistent robot motions. Although diffusion models have demonstrated superior performance in robot learning due to their task scalability and multi-modal flexibility, they suffer from two fundamental limitations: (1) they often produce non-smooth, jerky motions due to their inability to capture temporally consistent movement dynamics, and (2) their iterative sampling process incurs prohibitive latency for many robotic tasks. Inspired by classic robot motion generation methods such as DMPs and ProMPs, which capture temporally and spatially consistent dynamic of trajectories using low-dimensional vectors — and by recent advances in diffusion-based image generation that use consistency models with probability flow ODEs to accelerate the denoising process, we propose Fast Robot Motion Diffusion (FRMD). FRMD uniquely integrates Movement Primitives (MPs) with Consistency Models to enable efficient, single-step trajectory generation. By leveraging probabilistic flow ODEs and consistency distillation, our method models trajectory distributions while learning a compact, time-continuous motion representation within an encoder-decoder architecture. This unified approach eliminates the slow, multi-step denoising process of conventional diffusion models, enabling efficient one-step inference and smooth robot motion generation. We extensively evaluated our FRMD on the well-recognized Meta-World and ManiSkills Benchmarks, ranging from simple to more complex manipulation tasks, comparing its performance against stateof-the-art baselines. Our results show that FRMD generates significantly faster, smoother trajectories while achieving higher success rates.

I. INTRODUCTION

Recently, diffusion models [1] have gained increasing attention in robot learning due to their scalability [2] to complex tasks and their flexibility [3] in incorporating highdimensional, multi-modal observations. Modern advancements [2, 4] in embodied artificial intelligence have shown that an "action expert" [4] using diffusion models to generate robot actions can handle various manipulation tasks and that its multi-task capabilities scale up with dataset and model size. This highlights the promise of a unified model architecture that employs diffusion models as the action decoder (motion generation) for general-purpose robotic task solvers. However, existing diffusion-based robot motion generation methods still face two fundamental challenges. First, they often produce non-smooth, jerky motions because they fail to capture temporally consistent movement dynamics [5]. Specifically, this limitation arises from the fact that these methods commonly generate raw action sequences (i.e., waypoints) without accounting for the trajectory-level temporal consistency imposed by the robot's structured dynamic constraints. Second, the iterative sampling process inherent to these diffusion models' denoising sampling process (Denoising Diffusion Probabilistic Model (DDPM) [6], Denoising Diffusion Implicit Models (DDIM) [7]) introduces significant latency when generating robot actions.

Inspired by classic dynamic-system-based robot motion generation methods, such as dynamic movement primitives (DMPs [8]) and probabilistic movement primitives (ProMPs [9]), which capture temporally and spatially consistent dynamic of trajectories using low-dimensional vectors, we propose rethinking diffusion-based robot motion generation by shifting from modeling the conditional action distribution at the raw action (waypoint) level to the trajectory level (movement primitives). This approach essentially learns the prior distribution in the trajectory parameter space. Moreover, instead of simply combing diffusion models with movement primitives (MPs) like previous methods [5, 10], to further accelerate the action denoising process, we incorporate recent advances in diffusion-based image generation [11] that employ consistency models with Probability Flow ODEs [11] and consistency distillation [12] for fast motion generation. These two effective ingredients enable our method to generate fast and smooth robot motions.

Specifically, we introduce Fast Robot Motion Diffusion (FRMD), a novel consistency-distilled movement primitives framework that integrates Consistency Models [11] and Probabilistic Dynamic Movement Primitives (ProDMPs) [8] to achieve both high inference efficiency and smooth motion generation. Instead of directly generating raw actions via iterative denoising, we employ a movement primitives framework [13, 14] to produce smooth motions with guaranteed initial conditions for planning consecutive action sequences. In this setup, the diffusion model predicts ProDMPs weight vectors to ensure structured and smooth trajectories.

Moreover, unlike recent methods [5, 10] that simply combine diffusion models with movement primitives—which incur inference latency from multi-step denoising and limit their applicability in many robotic tasks—we observe that shifting from predicting raw actions (waypoints) to predicting trajectory parameters (movement primitives) naturally enables the use of consistency models [11] for accelerated inference. We employ consistency distillation [11, 12] to train a model that directly maps noisy actions to movement primitive parameters. This single-step inference approach eliminates multi-step denoising while preserving the expressive power of Movement Primitive Diffusion (MPD) [10],

¹Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada

² Alberta Machine Intelligence Institute (Amii), Edmonton, Canada

[†]Corresponding author: dalen.shi@ualberta.ca

our teacher model. As a result, FRMD produces structured motion faster and smoother than traditional diffusion policies, broadening its applicability in many robotic tasks. Our contributions can be summarized as follows:

- We propose FRMD, a novel framework that combines Consistency Models with Movement Primitives, enabling fast inference while maintaining structured and smooth motion generation.
- We introduce a consistency-distillation strategy to eliminate iterative denoising in diffusion-based motion generation, achieving one-step inference without sacrificing motion quality. We extensively evaluate FRMD across multiple robotic manipulation tasks, demonstrating significant improvements in both inference speed and task success rates over existing diffusion-based methods.
- By bridging structured motion representation and fast generative modeling, FRMD paves the way for realtime, high-quality robotic action generation, unlocking new possibilities in robot learning.

We evaluate FRMD on a diverse set of robotic manipulation tasks, ranging from simple to complex, using the MetaWorld [15] and ManiSkill [16] benchmarks following [17]. We compare FRMD with state-of-the-art diffusion policies, including the vanilla Diffusion Policy (DP) [1] and a recent method that combines diffusion models with movement primitives (MPD) [10]. The results show that FRMD achieves the highest success rate (64.8%) while operating 10× faster than MPD (the teacher model) and 7× faster than DP, enabling the generation of fast and smooth robot motions.

II. RELATED WORK

Our work draws inspiration from a diverse range of topics, including recent advances in diffusion models for robot learning, consistency models, and classic dynamic-system-based methods for robot motion generation.

Diffusion Models for Robot Learning: Recent research in robot learning has increasingly adopted diffusion models as the action decoder to generate smooth and diverse trajectories from noisy inputs. Diffusion models, which were first introduced in computer vision for denoising tasks [6], have been adapted for robotics by leveraging U-Net based architectures to decode actions from noisy state-action pairs. Early approaches demonstrate that by treating planning as a generative process, the diffusion model can iteratively refine noisy trajectories into feasible control commands [1, 18]. More recently, the π_0 work [4] has further established diffusion models as a robust action decoder, showing that they can effectively map high-dimensional latent representations into precise robotic actions, thereby broadening the applicability of diffusion-based methods in complex control scenarios. Despite its success, existing Diffusion Policy approaches often focus solely on end-to-end trajectory generation without explicitly incorporating structured priors about motion, such as movement primitives. This limits their capability for fast and smooth robot motion generation.

Consistency Models for Fast Diffusion Consistency models were originally introduced in the image generation domain to accelerate the sampling speed of diffusion models while maintaining high generation quality [11]. The core principle is to enforce self-consistency along the Probability Flow Ordinary Differential Equation (PF-ODE) trajectory so that any intermediate noisy sample can be directly mapped to the final clean output in a single inference step, as opposed to the multiple denoising steps required in conventional diffusion models. Recent work on Latent Consistency Models (LCMs) has further demonstrated significant speed improvements in text-to-image generation, especially when integrated with diverse conditional control mechanisms [19– 21]. Although consistency models have advanced vision and language synthesis, their use as fast action decoders in robotic motion remains underexplored. To bridge this gap, our work integrates MPs with consistency models to achieve fast, structured, and temporally consistent trajectory generation, thereby extending consistency distillation to robotic motion planning and accelerating robot action inference without compromising success rates.

Movement Primitives in Robotics: MPs constitute a fundamental framework in robotics, offering a compact, structured representation for encoding and generating smooth motion trajectories while ensuring temporal coherence. Among the most established formulations, DMPs [14, 22] encode motions as a combination of attractor dynamics and forcing functions, which guarantees spatial and temporal invariance and allows for the generalization of learned trajectories to new goals, yet their reliance on numerical integration limits full trajectory modeling and handling of stochasticity. ProMPs [9] overcome these issues by representing motions as Gaussian distributions that capture temporal correlations across multiple degrees of freedom, making them effective for learning-from-demonstration. More recently, ProDMPs [23] eliminate costly integration via precomputed basis functions, though scaling to high-dimensional motion remains challenging. In contrast to diffusion-based methods that better manage high-dimensional observations, traditional MPs often struggle with scalability in modern robot learning.

Diffusion Models using Trajectory Parametrization Part of our proposed diffusion framework that applies diffusion to the parametrization weights of trajectories (movement primitives) rather than raw actions or waypoints is also inspired by recent works [5, 10], which have shown that coupling diffusion models with movement primitives yields gentle motions for deformable object manipulation and smoother trajectories for general robotic tasks. However, rather than merely combining diffusion models with movement primitives—which still incur inference latency from multi-step denoising—we propose to learn the trajectory-level distribution as the problem to optimize a consistency function that directly maps noise samples to trajectory parameters using consistency distillation. This approach enables fast and smooth action generation.

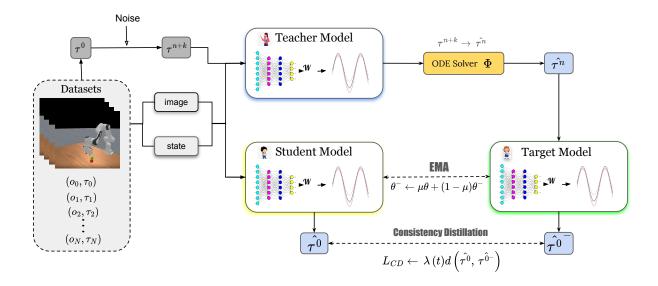


Fig. 1: Overview of FRMD Training Framework. Given observations o_i , raw action sequence τ^0 and initial state (y_0, y_0) from the robot datasets, we first perform a forward diffusion to introduce noise over n+k steps. The resulting noisy sequence τ^{n+k} is then fed into both the student model and the teacher model to predict the action sequence τ^0 and τ^n . The target model uses the teacher network's k-step estimation results to predict the action sequence. The student model, trained via consistency distillation and its weights are updated through an Exponential Moving Average (EMA).

III. METHODOLOGY

In this section, we detail our method – Fast Robot Motion Diffusion (FRMD) – and the necessary background. An overview of the Consistency Distillation process is illustrated in Figure 1.

A. Problem Formulation

Our objective is to directly map observations to structured action sequences while ensuring both efficiency and high quality. Formally, we define the motion trajectory as a sequence of robot actions: $\tau = \{a_i\}_{i=0}^n, a_i \in \mathbb{R}^k$ where each action a_i lies in a k-dimensional action space \mathbb{R}^k , determined by the control mode and the robot's degrees of freedom (DoFs). The goal is to predict the next n time steps of the trajectory based on current and past observations. We seek to learn a policy π from an expert demonstration dataset, mapping the robot's observations $o \in \mathcal{O}$ to an action sequence: $\pi: \mathcal{O} \to \mathcal{A}$. The observation o comprises RGB images from onboard cameras and robot proprioceptive states, capturing both environmental context and robot dynamics.

B. Preliminaries

1) Movement Primitives: MPs provide a framework for representing complex motor skills through simple and parameterizable models. ProDMPs offer a unifying framework that overcomes weaknesses and combines the strengths of ProMPs and DMPs. ProDMPs eliminate the need for costly numerical integration associated with DMPs by utilizing precomputed position and velocity basis functions of the fundamental ODE that are valid for all trajectories.

Unlike traditional DMPs, which require solving differential equations for trajectory generation, ProDMPs formulate motion as a weighted combination of precomputed basis functions. In ProDMPs, the positions y of a trajectory are formulated as:

$$y = c_1 y_1 + c_2 y_2 + [y_2 p_2 - y_1 p_1 \quad y_2 q_2 - y_1 q_1] \begin{bmatrix} w \\ g \end{bmatrix}$$
(1)
= $c_1 y_1 + c_2 y_2 + \mathbf{\Phi}^{\top} \mathbf{w}$

where y_1 and y_2 are the two linearly independent complementary functions of the ProDMP's homogeneous ODE. The constants c_1 and c_2 are determined by solving a boundary condition problem where we use the current position and velocity, ensuring smooth transitions.

Similar to position y, the velocity \dot{y} can be formulated as:

$$\dot{y} = c_1 \dot{y}_1 + c_2 \dot{y}_2 + \dot{\mathbf{\Phi}}^\top \mathbf{w} \tag{2}$$

where y_1 , y_2 are time derivatives version of y_1 and y_2 , respectively. The basis functions for position and velocity, Φ and $\dot{\Phi}$ are predefined and used for motion representation.

The weights w are N+1-dim vectors containing the DMP's original weight vector with the goal attractor to which the ODE converges. ProDMPs facilitate planning smooth trajectories with guaranteed boundary conditions while minimizing computational demands.

2) Consistency Models: The Consistency Model introduces an efficient generative model designed for effective single-step or few-step inference generation while maintaining a comparable performance. Consistency models are built upon the PF-ODE, which describes the evolution of data

distribution over time. Given an data x_t at time t, the PF-ODE is defined as:

$$dx = -\dot{\sigma}(t)\sigma(t)\nabla_x \log p(x_t)dt \tag{3}$$

where x_t is the noisy observation at time t, $\sigma(t)$ is the noise schedule and $\nabla_{x_t} \log p(x_t)$ is the score function, guiding the denoising process.

The objective of consistency model is to learn the solution function $f(\cdot, \cdot)$ of this PF-ODE. Given a solution trajectory $\{\mathbf{x}_t\}_{t\in[0,T]}$ of the PF-ODE, the consistency function f is defined as $f:(\mathbf{x}_t,t)\mapsto\mathbf{x}_0$ that directly maps any noisy input x_t at time t to the original clean data x_0 , enforcing the self-consistency property:

$$f(\mathbf{x}_t, t) = f(\mathbf{x}_{t'}, t'), \quad \forall t, t' \in [\epsilon, T]$$
 (4)

As shown in Equation 4, the implication of the self-consistency property is that for any input pairs (\mathbf{x}_t,t) on the same PF-ODE trajectory, their outputs $f(\mathbf{x}_t,t)$ remain consistent. All consistency models have to meet the boundary condition $f(\mathbf{x}_0,t_0)=\mathbf{x}_0$. The boundary condition ensures that the model does not converge to a meaningless solution like $f_{\theta}(\mathbf{x},t)\equiv 0$.

Consistency Distillation is a method widely used for training consistency model by distilling knowledge from pretrained diffusion models (teacher model). The consistency loss is defined as:

$$\mathcal{L}(\theta, \theta^-; \phi) = \mathbb{E}\left[d\left(f_{\theta}(\mathbf{x}_{t_{n+1}}, t_{n+1}), f_{\theta^-}(\hat{\mathbf{x}}_{t_n}^{\phi}, t_n)\right)\right]$$
(5)

where $d(\cdot,\cdot)$ is a metric function chosen for measuring the distance between two samples. $\phi(\cdot,\cdot)$ is the update function of ODE solver applied to the PF-ODE. $f_{\theta}(\cdot,\cdot)$ and $f_{\theta^-}(\cdot,\cdot)$ are referred to as 'online network' and 'target network' according to . When the optimization of the online network converges, the target network will eventually match the online network since θ^- is a running average of θ . The estimated consistency model can become arbitrarily accurate as long as the step size of the ODE solver is sufficiently small and the consistency distillation loss reaches zero.

C. FRMD: Fast Robot Motion Diffusion

The overvew of FRMD is presented in Figure 1. In the pre-training phase, we train the teacher model follow the pipeline proposed in [10]. Then FRMD implements a consistency distillation method to distill the knowledge from the teacher diffusion-based policy. We adopt a consistency function to predict the action sample, opposite to the noise prediction that is commonly employed in image generation. This modification results in a faster convergence to the low-dimensional robot action manifold. In the inference phase, FRMD is able to decode high-quality action within one inference.

1) Teacher Model Set-up: For Teacher Model, we follow the pipeline proposed in [10]. The teacher model consists of a trainable model E_{θ} that outputs a weight vector w. Combined with initial values y_0 , y_0 for position and velocity, w is decoded into an action sequence τ using ProDMP decoder

P with predefined parameters Φ as mentioned in III-B.1. Assume that the noise action is $\tilde{\tau}$, the teacher model denoise pipeline can be represented as:

$$F_{\theta}(\tilde{\tau}, o, t) = P_{\Phi}(y_0, \dot{y_0}, E_{\theta}(\tilde{\tau}, o, t)) \tag{6}$$

For the diffusion part, we adopt score-based generative model which is also built upon PF-ODE, here we rewrite the PF-ODE in Equation 3 as follows:

$$d\tau = -\dot{\sigma}(t)\sigma(t)\nabla_{\tau}\log p(\tau|o,\sigma(t))dt \tag{7}$$

where $\sigma(t)$ represents the noise scheduler and the score function $\nabla_{\tau} \log p(\tau|o,\sigma(t))$ can be seen as the gradient of the probability of action sequences τ conditioned by observations o and $\sigma(t)$. According to Equation. 7, the score function can be approximated as:

$$\nabla_{\tau} \log p(\tau|o, \sigma(t)) \doteq \frac{F_{\theta}(\tau, o, \sigma(t)) - \tau}{\sigma(t)^2}$$
 (8)

During training, we adopt score matching method to minimize the loss function:

$$\mathbb{E}\left[\left\|\frac{F_{\theta}(\tilde{\tau}, o, \sigma(t)) - \tilde{\tau}}{\sigma(t)^{2}} - \nabla_{\tilde{\tau}} \log q(\tilde{\tau}|\tau)\right\|^{2}\right]$$
(9)

where $\tilde{\tau} = \tau + \epsilon$ with Gaussian noise $\epsilon \sim \mathcal{N}(0, t^2 I)$.

During inference, new actions are generated by gradually denoising samples of a unit Gaussian by solving the PF-ODE using DPM-Solver [24] that are specifically designed for fast inference(~10 steps) in ODE-based diffusion.

2) Student Model Consistency Distillation: To train student model, we adopt the method of consistency distillation, as depicted in Figure 1. Given a ground-truth expert action sequence τ_0 , we first obtain the noisy action τ^{n+k} by conducting a forward operation with n+k steps to add noise on τ^0 , where k is the skipping interval in sampling of teacher model. Subsequently, the noisy action τ^{n+k} is feedforwarded to teacher model and student model, respectively. The student model directly predict clean action sequence $\hat{\tau}^0$. The teacher model output action sequence $\hat{\tau}^n$ via k-step estimation:

$$\hat{\tau^n} \leftarrow \tau^{n+k} + (t_n - t_{n+k})\phi(\tau^{n+k}, t_{n+k}) \tag{10}$$

In order to make sure the self-consistency property (Equation 4) holds, we also design a target model cloned from the student model, and we expect the outputs of the student model and the target model to be aligned. Specifically, the target network generates $\tau^{\hat{0}-}$ utilizing $\hat{\tau}^n$ and we expect that $\tau^{\hat{0}-}=\hat{\tau}^0$.

In the student and target model, clean action sequence $\hat{\tau}^{0-}$ and $\hat{\tau}^0$ are obtained by consistency function. We parameterize the consistency function using skip connections as mentioned in [11]:

$$f_{\theta}(\tau, o, t) = c_{\text{skip}}(t)\tau + c_{\text{out}}(t)F_{\theta}(\tau, o, t) \tag{11}$$

where the $F_{\theta}(\cdot,\cdot,\cdot)$ is the same structure as teacher model proposed in Equation. 6. Meanwhile, to strength the boundary condition mentioned in III-B.2, we set $c_{\text{skip}}(t) =$

 $\gamma_d^2/(\beta^2 t^2 + \gamma_d^2)$ and $c_{\text{out}}(t) = \beta t/\sqrt{\beta^2 t^2 + \gamma_d^2}$ where β denotes scaling value while γ_d is a balance value. Combined with Equation. 5 and Equation. 11, the consistency distillation loss can be expressed as:

$$\mathcal{L}_{CD} = \mathbb{E}\left[\lambda(t_n)d(f_{\theta}(\tau^{n+k}, o, t_{n+k}), f_{\theta^-}(\hat{\tau}^n, o, t_n))\right]$$
(12)

where $\lambda(\cdot) \in \mathbb{R}^+$ is a positive weighting function, $\hat{\tau}^n$ is given by Equation 10, $d(\cdot, \cdot)$ is a metric function that satisfies $\forall \mathbf{x}, \mathbf{y} : d(\mathbf{x}, \mathbf{y}) \geq 0$ and $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$. θ^- is updated with the exponential moving average (EMA) of the parameter θ , defined as:

$$\theta^- \leftarrow \text{stopgrad}(\mu \theta^- + (1 - \mu)\theta),$$
 (13)

where stopgrad(\cdot) denotes the stop-gradient operation and μ satisfies $0 \le \mu < 1$.

To summarize, we propose Student Model Consistency Distillation as detailed in Algorithm1.

Algorithm 1 Student Model Consistency Distillation

- 1: Initialization: Dataset \mathcal{D} , initial parameter θ , learning rate η , ODE solver $\phi(\cdot,\cdot)$, $d(\cdot,\cdot)$, $\lambda(\cdot)$ and μ
- 2: repeat
- Sample $o, \tau^{\mathbf{0}} \sim \mathcal{D}$ and $n \sim \mathcal{U}\left[1, N-k\right]$ Sample $\tau^{n+k} \sim \mathcal{N}(\tau; t_{n+k}^2 \mathbf{I})$ 3:
- 4:
- Teacher Model k-step Denoise: $\hat{\tau}^n \leftarrow \tau^{n+k} + (t_n - t_{n+k}) \phi(\tau^{n+k}, t_{n+k})$
- Compute loss:

$$\mathcal{L}_{CD} \leftarrow \lambda(t_n) d(f_{\theta}(\tau^{n+k}, o, t_{n+k}), f_{\theta^-}(\hat{\tau}^n, o, t_n))$$

- Update student model:
 - $\theta \leftarrow \theta \eta \nabla_{\theta} \mathcal{L}_{CD}(\theta, \theta^{-}; \phi)$
- Update target model using EMA: $\theta^- \leftarrow \text{stopgrad}(\mu\theta^- + (1-\mu)\theta)$
- 9: until convergence

IV. EVALUATIONS

In the evaluation part, we aim to answer the following questions:

- (1) Can our proposed FRMD generate smooth and temporally consistent trajectories across different environments and various maniplation tasks ranging from simple to complex?
- (2) Can our proposed FRMD achieve one-step inference without sacrificing motion quality in these benchmark tasks?
- (3) How much additional performance gain can we achieve when comparing FRMD with other SOTA baseline methods?
- (4) How will different backbone network architecture affect the final performance of our proposed FRMD?

A. Environments, Tasks and Datasets

We conduct our experiments in the well-recognized Meta-World [15] and Maniskills [16] benchmarks. Following [17], we use a total of 12 tasks ranging from simple tasks like pick-cube to more challenging ones such as dexterous manipulation. Like [17], we divided the 12 tasks into three categories, including 4 easy tasks, 5 medium tasks and 3 hard tasks based on their difficulty levels. For datasets collection, we collect demonstrations by running expert policies in Metaworld and replaying trajectory method in Maniskills. Each demonstration is a sequence pair (τ_i, o_i) over one full task execution with N time steps. During preprocessing, the demonstration datasets are split into multiple action and observation sequences of lengths n and m, respectively. So that the model will predict next n-step actions based on observations from the previous m time steps. We obtain a number of 100 expert demonstrations for training in each benchmark for fair comparisons.

B. Evaluation Metrics

To answer the (1)-(3) questions proposed in the beginning of section IV, we report three key metrics to assess the performance of our method: (1) Task Metrics, which measure the success rate on benchmark tasks, (2) Time Metrics, which evaluate the efficiency of inference, and (3) Motion Smoothness Metrics, which assess the quality and smoothness of generated motions.

- 1) Task Metrics: We evaluate 10 episodes every 5,000 training steps and compute the average success rates(SR) throughout the training process and until each training converges. For success rate, the higher is the better.
- 2) Time Metrics: In the time assessment phase, we measure the average runtime per step. To mitigate performance fluctuations, each main experiment is conducted using three different random seeds (0, 1, and 2). For inference time, lower values indicate better performance.
- 3) Motion Quality Metrics: Defining Motion Quality Metrics is challenging since directly measuring the model's inference consistency involves complex statistical methods [25] [26]. We adopt a geometric approach [27] to visualize generated trajectories and assess their smoothness by computing the curvature k_i at each data point. A transition is classified as non-smooth if $k_i > k_{max} = 1$. Finally, we visualize these non-smooth transitions in our visualizations and compare our method with baselines.

C. Baselines

Our work primarily focuses on two key contributions: accelerating inference speed and improving the quality of robotic action generation. To evaluate our approach, we compare it against state-of-the-art baselines, including DP and MPD. Notably, MPD serves as the teacher model in our consistency distillation framework. We make comparison between our method and teacher model to see if our model could surpasses MPD by achieving significantly faster inference while maintaining or even enhancing action quality.

D. Implementation Details

1) Dataset: We predict action sequences of length n =12, conditioned on the previous m=3 observations. As the result, the demonstrations are split into multiple action and observation sequences of lengths 12 and 3, respectively.

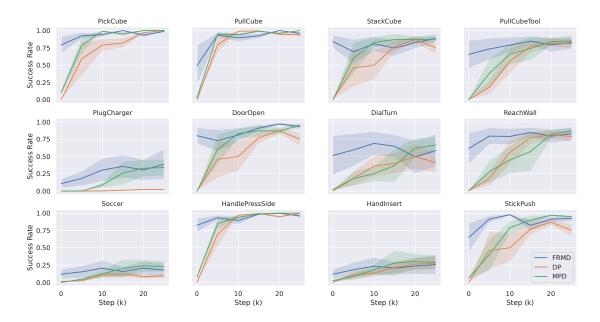


Fig. 2: Learning Curve comparison of different methods across various robotic tasks. We compare FRMD (ours), Diffusion Policy (DP) and Movement Primitives Diffusion (MPD) across 12 different tasks from the MetaWorld and ManiSkill benchmarks. The success rate is computed by evaluating 10 episodes with random seeds in each environment at every 5k training steps for each method, until convergence. The mean success rate is plotted as a solid line, and the variance is shown as shaded areas. Results show that our method consistently achieves higher success rates with significantly smaller inference latency (10x faster than MPD [10], 7x faster than DP [1], as shown in Table 1) compared to baselines. Note that the intermediate success rate of our method in the initial training steps is due to model distillation using the teach model.

- 2) Observation Encoder: The encoder maps the raw image sequence into a latent embedding o_t and is trained end-to-end with our method and baselines. We used a standard ResNet18 (without pretraining and with output size 128) as the encoder. As decribed in [1] [28], we also made some modifications: Replace the global average pooling with a spatial softmax pooling and replace BatchNorm with GroupNorm.
- 3) Network Architectures: For Diffusion Policy, we use CNN layer with sizes of (256, 512, 1024). MPD and FRMD share an optimal transformer architecture, as proposed in [1], with 6 layers, 4 heads, dropout probability of 0.3 and embedding size of 256.
- 4) Training: An AdamW optimizer with a batch size of 128, a learning rate of 1e-4 and a weight decay of 1e-6 is employed for 30k steps training. We also adopt a cosine decay learning rate scheduler and 500 iterations of linear warm-up. The EMA rate is set to $\mu = 0.95$. We implement our model in PyTorch, and train the model on one NVIDIA RTX 4090 GPU. All compared models are evaluated on the same device to achieve fairness.

E. Results and Comparisons

We compare FRMD with existing state-of-the-art methods on Maniskills and Metaworld to demonstrate the performence of our method. As described in IV-B, we make the comparison on three parts: Success Rate, Inference time

TABLE I: Success rate and inference time across all tasks based on our models and baselines. The best results for each category are in bold font and the second best ones are underlined for an easier comparison.

Methods	Easy(4)	Medium(5) Hard(3)		Average						
Success Rate (%)										
DP	99.3 ±0.1	41.0 ± 3.2	10.1 ± 1.4	50.1						
MPD	98.9 ± 0.3	$\underline{64.8} \pm 2.6$	28.6 ± 2.9	64.1						
FRMD	99.2 ± 0.1	66.3 ±1.2	29.0 ±2.3	64.8						
Inference Time (ms)										
DP	119.8 ± 1.4	121.3 ± 2.3	118.2 ± 3.6	<u>119.7</u>						
MPD	162.7 ± 3.5	173.2 ± 3.6	169.9 ± 1.2	168.6						
FRMD	15.2 ± 0.8	18.6 ± 3.4	17.9 ±1.1	17.2						

and Motion quality. According to these result, we have the following analysis concerning the three questions proposed in the start of Section IV.

1) Success Rate: Figure. 2 presents the learning curve of our method and baselines. The results show that our method accelerates the diffusion process without any performance drop. Specifically, with only one-step inference, our FRMD can approximate or even surpass the state-of-the-art model.

The detailed results are presented on the Table I, that FRMD achieves the highest overall success rate of 64.8%, outperforming both MPD (64.1%) and DP (50.1%). For easy tasks, FRMD (99.2%) performs on par with DP (99.3%) and slightly better than MPD (98.9%), indicating that all methods perform well when task complexity is low. As the task difficulty increases, FRMD consistently maintains its advantage, achieving 66.3% success on medium tasks, confirming that our method effectively distills MPD's structured motion representation into a more efficient form. On hard tasks, FRMD continues to outperform MPD and significantly surpasses DP, demonstrating its superior ability to handle complex robotic motion generation. These results provide strong evidence that while FRMD is built upon MPD as its teacher model, it ultimately surpasses MPD in performance, successfully retaining the structured motion quality while enhancing policy expressiveness and robustness.

- 2) Inference Time: The inference time results highlight the efficiency advantage of FRMD, which achieves real-time action generation with an average inference time of 17.2ms, making it 10× faster than MPD (168.6ms) and 7× faster than DP (119.7ms). Across different task complexities, FRMD maintains consistently low latency, achieving 15.2ms on easy tasks, 18.6ms on medium tasks, and 17.9ms on hard tasks, demonstrating its ability to efficiently generate high-quality actions in a single step. These results confirm that FRMD's consistency-distilled model significantly reduces computational overhead while preserving high motion quality, making it well-suited for real-time robotic control applications.
- 3) Motion Quality: To further evaluate the motion quality of different methods, we visualize the end-effector trajectory during execution. To ensure fairness, we fix the environment's initial conditions before conducting experiments, maintaining the same initial and goal states. The visualization results for the PlugCharger-v1 task are shown in Figure 3. As described in Section IV-B, a transition is classified as non-smooth if the curvature satisfies $k_i > 1$. We record the number of non-smooth transitions as $N = num\{k_i > 1\}$ in the generated trajectory. The results show that $N_{\rm DP} = 82$ and $N_{\rm FRMD} = 21$. Given that the maximum episode step is fixed across all experiments, this indicates that our method (FRMD) produces significantly smoother trajectories compared to DP.

Based on the above results analysis, accordingly, we conclude the answers to Question(1)-(3) raised before:

- (1) Our proposed FRMD effectively generates smooth and temporally consistent trajectories across different environments, significantly reducing non-smooth transitions compared to DP.
- (2) Our FRMD achieves one-step inference with an average latency of 17.2ms, which is 10× faster than MPD, without compromising motion quality.
- (3) Compared to SOTA baselines, our FRMD achieves the highest success rate and demonstrates superior efficiency and performance.

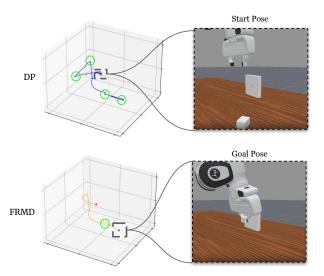


Fig. 3: Trajectories generated in PlugCharger-v1 task. The top plot shows the trajectory generated by DP, while the bottom plot presents the trajectory produced by our method (FRMD). The green circles highlight regions where the data point transitions exhibits significant non-smoothness which is computed by comparing its curvature k_i with a threshold $k_{max}=1$, as defined in Section IV-B . In comparison, our method results in a significantly smoother trajectory with fewer oscillations, demonstrating improved motion stability, especially near the start and goal point.

F. Ablation Study

We also conducted ablation studies to further evaluate the impact of different design choices in our method. Specifically, we aim to answer Question (4) raised at the beginning of this section: How will different backbone network architecture affect the final performance of our proposed FRMD?

We investigate the impact of different backbone architectures on the performance of FRMD. Our approach adopts a transformer-based model [2] as the backbone for the teacher model (MPD [10]). Specifically, the denoiser network is implemented as a transformer with 6 layers, 4 attention heads, and a hidden size of 256, enabling it to effectively capture temporal dependencies in robotic trajectories. To evaluate the effectiveness of this architecture, we compare it against two alternative designs: (1) a lightweight transformer with fewer parameters and (2) a simple feedforward MLP network composed of fully connected layers. To ensure a fair comparison, all other components (e.g., observation encoders) remain unchanged across experiments.

The comparison results are presented in Table II. The original version of the Transformer achieves the highest success rates but at the cost of slightly higher inference time. The lightweight Transformer (Transformer* in Table II) achieves lower inference time while maintaining strong performance across tasks. MLP is computationally efficient but significantly underperforms in complex tasks. These results validate that Transformer-based architectures are well-suited for our method, enabling the generation of fast and

temporally consistent robot motions.

TABLE II: Ablation study on different architectures. Here Transformer* refers to a lightweight transformer with the same architecture as the Transformer baseline but with fewer parameters. The best results are highlighted in bold.

Architecture	Easy(4) Time SR		Medium(5)		Hard(3)	
	Time	SR	Time	SR	Time	SR
Transformer	15.2	99.2	18.6	66.3	17.9	29.0
Transformer*	16.2	98.5	15.8	68.5	15.9	26.2
MLP	9.9	74.9	10.3	54.2	8.9	9.2

V. CONCLUSIONS AND LIMITATIONS

In this work, we propose FRMD, a novel consistency-distilled movement primitives model that integrates Consistency Models with Probabilistic Dynamic Movement Primitives (ProDMPs [9]) for efficient, smooth robotic motion generation. Using Movement Primitive Diffusion (MPD [10]) as the teacher model, our approach learns to produce structured motion trajectories while dramatically improving inference speed via consistency distillation. Unlike conventional diffusion policies that require multiple denoising steps, FRMD achieves real-time, single-step inference without sacrificing trajectory quality. In future work, we plan to extend FRMD to complex, high-dimensional tasks and integrate task-specific cost functions into the consistency training framework to further evaluate its applicability in more robotic tasks.

Due to time limits, we did not test our method on large-scale task pre-training using large datasets and large parameterized networks. It's worth noting that our primary contribution — the design of the action decoder — is inherently modular and can be seamlessly integrated into large-scale diffusion-based VLA (vision-language-action) models [2, 4]. By employing our method as the action decoder ("action expert") and pairing it with a powerful encoders, such as the one used in the π_0 [4] model, our framework holds significant potential for improved performance and scalability across diverse robotic tasks.

REFERENCES

- [1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.
- [2] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine, "The ingredients for robotic diffusion transformers," arXiv preprint arXiv:2410.10088, 2024.
- [3] Z. Dong, Y. Yuan, J. Hao, F. Ni, Y. Mu, Y. Zheng, Y. Hu, T. Lv, C. Fan, and Z. Hu, "Aligndiff: Aligning diverse human preferences via behavior-customisable diffusion model," arXiv preprint arXiv:2310.02054, 2023.
- [4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π₀: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.
- [5] J. Carvalho, A. Le, P. Kicki, D. Koert, and J. Peters, "Motion planning diffusion: Learning and adapting robot motion planning with diffusion models," arXiv preprint arXiv:2412.19948, 2024.

- [6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in Advances in Neural Information Processing Systems, 2020.
- [7] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," arXiv preprint arXiv:2010.02502, 2020.
- [8] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.
- [9] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 115–122.
- [10] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, "Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects," *IEEE Robotics and Automation Letters*, 2024.
- [11] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," 2023.
- [12] Y. Song and P. Dhariwal, "Improved techniques for training consistency models," arXiv preprint arXiv:2310.14189, 2023.
- [13] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics Research. The Eleventh International* Symposium: With 303 Figures. Springer, 2005, pp. 561–572.
- [14] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023.
- [15] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.
- [16] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," arXiv preprint arXiv:2107.14483, 2021.
- [17] G. Lu, Z. Gao, T. Chen, W. Dai, Z. Wang, and Y. Tang, "Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation," arXiv preprint arXiv:2406.01586, 2024.
- [18] M. Janner, Q. Yang, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Learning Representations*, 2022.
- [19] J. Kim, K. Lee, and J. Park, "Latent consistency models for accelerated text-to-image generation," arXiv preprint arXiv:2301.00000, 2023.
- [20] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang, "Motionlcm: Real-time controllable motion generation via latent consistency model," in *European Conference on Computer Vision*. Springer, 2024, pp. 390–408.
- [21] J. Chen, Y. Wu, S. Luo, E. Xie, S. Paul, P. Luo, H. Zhao, and Z. Li, "Pixart-{\delta}: Fast and controllable image generation with latent consistency models," arXiv preprint arXiv:2401.05252, 2024.
- [22] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [23] J. Doe and J. Smith, "Probabilistic dynamic movement primitives for efficient robotic trajectory generation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4567–4573.
- [24] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5775–5787, 2022.
- [25] J. Jin, L. Petrich, M. Dehghan, and M. Jagersand, "A geometric perspective on visual imitation learning," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 5194–5200.
- [26] B. Flury and T. Tarpey, "Self-consistency: A fundamental concept in statistics," *Statistical Science*, vol. 11, no. 3, pp. 229–243, 1996.
- [27] S. Guillén Ruiz, L. V. Calderita, A. Hidalgo-Paniagua, and J. P. Bandera Rubio, "Measuring smoothness as a factor for efficient and socially accepted robot motion," *Sensors*, vol. 20, no. 23, p. 6822, 2020.
- [28] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in Proceedings of Robotics: Science and Systems (RSS), 2023.