Motion Composition and Interpolation Using Diffusion Models

Omkar Patil, Nakul Gopalan
{opatil3, ng}@asu.edu
Arizona State University

Abstract—**Humans have the ability to perform various combinations of skills without having to relearn the overall resulting skill every single time. For example, we prefer to learn easy motions and then combine them in flexible ways to perform complicated movements in dance. Enabling robots to combine or compose skills is essential for their deployment in unstructured environments where they will be required to adapt based on their surroundings. Without such composition robots would have to learn a separate policy for each task which can be combinatorially expensive. To this end, we propose a compositional approach to blend different robot skills using diffusion models. We compose configuration-space diffusion policies for novel motion generation resulting from the corresponding skill combinations. We show that the compositional framework can be utilized to interpolate between different skills leading to greater flexibility in motion. By utilizing interpolation along with composition, we can not only constrain the motion but also generate novel trajectories. We also propose a novel metric based on Maximum Mean Discrepancy and the Forward Kinematics kernel: MMD-FK to quantitatively evaluate the composed robot motion in the task-space while remaining agnostic to the space of policy composition.**

I. Introduction

For robots to be deployed in unstructured environments and interact with humans, they have to combine different skills and motions in their movements based on the situation. For example, can a robot wiggle its end-effector while moving forward to clean a table. This requires combination of a wiggling motion and a reaching motions. This wiggling motion can be combined with different primitive motions to clean, to wash dishes, to fit a bed-sheet, to iron a cloth, etc. Can robots be expected to learn all these skills separately or can they combine known primitives to perform novel motion? Despite the ongoing work on enabling robots to learn a wide variety of skills for them to be deployed along side humans, limited work has addressed this problem of combining robot motions. In this work, we propose a compositional approach to efficiently generate robot motions composing base motion primitives zero shot. Previous works have used compositional approaches to constrain robot motion [41, 44], as it enables sampling from common high probability regions. However, we show that we can also interpolate between different motions enabling the generation of novel trajectories that the model has not learned before.

Robot learning has seen many recent advancements that allow robots to learn a wide array of skills such as flipping the pancake or spreading sauce on the pizza base [8, 47]. One approach to advance the field is to scale the size of the training data and the model enabling large models to learn varied action distributions [6, 31]. While this approach has shown considerable promise in terms of generalization, it requires large computational overheads and massive data collection, which may not be entirely possible in the case of robotics. Data collection in robotics is costly and there is a need to develop models that combine and reuse existing skills [5]. Efficiency of learning new skills and being able to reuse them will be crucial for robot learning to be successful in unstructured environments. Compositionality is one such approach that aims to maximize the reuse of policies or representations. Compositionality can be induced in various aspects of the policy learning process for sample efficiency gains. Compositional agents learn a set of base policies or representations for the given data using the objective of maximizing their composability to solve novel tasks. Functional [27], representational [2], temporal [4, 43] and policy composition [19, 33, 44] feature prominently in the machine learning and robotics literature. Policy composition enables embodied agents to reuse two or more of their learned policies together for sampling from a novel distribution. Previously, a mixture of experts [21] such as a weighted average or a product of distributions [19] representing sub-policies has been used to model skill, motion or constraint composition. Unlike other forms, in policy composition, the distributions learned for motions or constraints are probabilistically composed and the resulting sample is generated from the new probability landscape.

To that end, there is a surge of recent works that use diffusion models for policy composition, corresponding to images [11, 12] and even robot trajectories [44]. Diffusion models have shown impressive performance in robotics and learn different modes of the data distribution faithfully [8]. In this work we use diffusion models to compose learned policies zero-shot to create novel blended robot motions. We train policies in the configuration-space of manipulators to be able to learn more complex motions that are difficult to achieve with only end-effector control. Previous works have shown robot policy composition using diffusion models in the end-effector [41], tool [44] or object space [25] to constrain their trajectory or generalize it to novel objects. However to the best of our knowledge, policy composition has not been shown to generate novel trajectories in the task space. Our contributions in this work are the following:

- We compose learned configuration-space diffusion policies

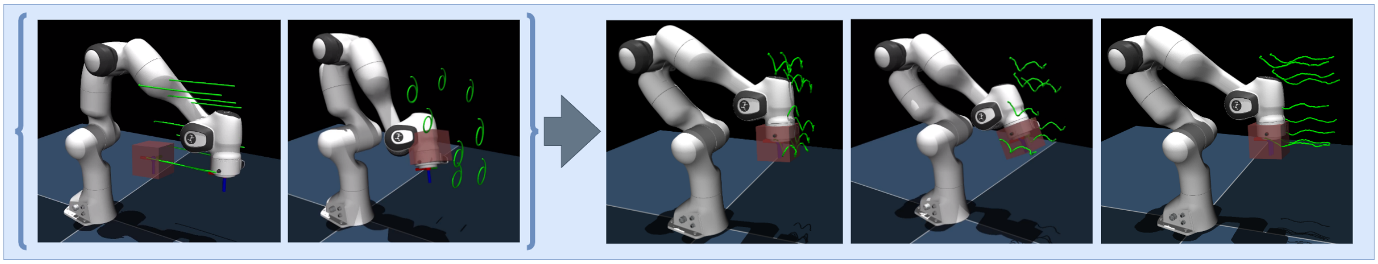

Fig. 1: Here we depict an example of policy composition in the configuration space leading to interpolation. A reach policy π_1 where the robot moves in a straight line and a policy that learns to draw a circle π_2 about an axis parallel to the line are trained independently. Their composition $\pi_1^w \pi_2^{1-w}$ results in a spiral that is visible in the task-space of the robot, without needing any new demonstrations for it. Varying w allows us to blend these motions in different proportions generating new spirals. To the best of our knowledge, ours is the first work to show how policy composition can enable novel motion generation. Moreover, we also propose a novel metric MMD-FK that evaluates the composed robot motion in task space while being agnostic to the actual space of the distribution composition. More examples of composition can be found on our project website: https://sites.google.com/asu.edu/comp-styles/.

to generate composed motions in the task-space. We believe ours is the first work to generate novel trajectories using a compositional framework. We show that weighted average of the score involved in composing diffusion models results in interpolation between the distributions.

- We propose a novel metric based on Maximum Mean Discrepancy and the Forward Kinematics kernel: MMD-FK to evaluate the composed robot motion in the task space, agnostic of the actual space of the composition of distributions.

II. Background

A. Compositionality in learning

Several works have used compositional approaches to show sample efficiency or generalization in learning. Compositional representations have been explored in natural language processing, such as word embeddings [28], with Andreas [2] suggesting a metric to measure the extent of compositionality in the generated representations. Mendez and Eaton [27] survey the functional form of composition where compositionality is induced through modular architectures serving defined purposes such as encoding the robot morphology or task. Polices can be composed temporally, by stacking them sequentially such as options in reinforcement learning [4]. Policies can also be composed probabilistically, such as a product or mixture of experts [19]. A long line of works in reinforcement learning (RL) probabilistically compose learned distributions for sample efficiency gains using entropic [15] approaches and in the model-based RL setting [16, 17]. In supervised learning, traditionally energy based models have been utilized to probabilistically compose distributions [11], since the energy functions could be added to obtain a product of corresponding distributions, from which samples could then be generated using Markov Chain Monte Carlo (MCMC). Recently, the number of works leveraging compositionality has surged owing to the discovery of the connection between energy based models and diffusion models and the ease of training the latter [12]. Liu et al. [25] and Yang et al. [45] compose learned constraints for generalizing to their novel combinations, similar to the work of Liu et al. [24] on composing learned object relations in images. Relevant to our work, Urain et al. [41] use energy based models to compose different motions for obstacle avoidance. However, the distributions are handcrafted and not learned. Wang et al. [44] compose learned distributions for the task of robotic tool use, but these policies are task-conditional and unconditional versions of the same task, and do not result in generation of novel motion.

B. Diffusion Models

Our aim is to learn the action distribution a_0^L for a fixed trajectory length L from D demonstrations. Here, we use a to denote action for all the trajectory time-steps for brevity and drop the L notation. Gaussian diffusion models [36] learn the reverse diffusion kernel $p_\theta(a_t|a_{t-1})$ for a fixed forward kernel that adds Gaussian noise at each step $q(a_t|a_{t-1}) = \mathcal{N}(a_t; \sqrt{\alpha_t}a_{t-1}, (1-\alpha_t)\mathcal{I})$, such that $q(a_T) \approx \mathcal{N}(0,\mathcal{I})$. Here $t <= T$ represents the diffusion time-step and α_t the noise schedule. To generate trajectories from the learned data distribution $p_\theta(a_0)$, we sample at time step T from $\mathcal{N}(0,\mathcal{I})$ and apply the reverse diffusion kernel $p_\theta(a_t|a_{t-1})$ at each time step. For training the model, maximization of the log-likelihood of the data distribution $log\ q(a_0)$ and reparametrization of the forward diffusion kernel yields the following loss used in practise [20]:

$$\mathcal{L}_t(\theta) = \mathbb{E}_{q(a_0)\mathcal{N}(\epsilon_0;0,\mathcal{I})}\left[\lambda_t[||\epsilon_0 - \hat{\epsilon}_\theta(a_t,o,t)||_2^2]\right] \tag{1}$$

Here λ_t is a function of α_t, and the network ϵ_θ is conditioned on observation o. We train our model to predict the noise ϵ_0 added to action a_0 for generating the noisy action a_t taken as the input to the network. Tweedie's formula [13] can be used to show that ϵ_0, and consequently ϵ_θ are proportional to the score of the diffused data distribution $q(a_t) = \int q(a_t|a_0)q(a_0)da_0$ [26].

$$\frac{-1}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}_\theta(a_t,t) \approx \frac{-1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_0 = \nabla_a log\ q(a_t) \tag{2}$$

C. Energy Based Models (EBMs)

EBMs are a class of probabilistic models of the form $p_\theta(a) = \frac{e^{f_\theta(a)}}{Z}$ where $Z(\theta) = \int e^{f_\theta(a)}da$ is the normalizing constant. Denoising score matching (DSM) [42] used to train EBMs [38] minimizes the Fisher divergence between the model $p_\theta(a)$ and the Gaussian-smoothed data distribution $q(\tilde{a}) = \int q(a)\mathcal{N}(\tilde{a}; a, \sigma_t^2 I)da$ at various noise scales σ_t.

$$\mathcal{J}_{\sigma_t}(\theta) = \mathbb{E}_{q(a,\tilde{a})}\left[\frac{1}{2}||\nabla_{\tilde{a}} log\ q(\tilde{a}|a) - \nabla_{\tilde{a}} log\ p_\theta(\tilde{a})||_2^2\right] \quad (3)$$

This circumvents the normalizing constant by evaluating the gradient of the log-probability of the model $\nabla_a p_\theta(\tilde{a}) = \nabla_a f_\theta(\tilde{a})$ at different noise scales. Equation 3 simplifies to [42]:

$$\mathcal{J}_{\sigma_t}(\theta) = \mathbb{E}_{q(a)\mathcal{N}(\epsilon;0,\mathcal{I})}\left[||\frac{\epsilon}{\sigma_t} + \nabla_a f_\theta(a + \sigma_t\epsilon)||_2^2\right] \quad (4)$$

Once trained, MCMC methods such as Langevin [3] can be used to sample from EBMs since they only depend on the score of the data distribution. This approach is also known in the literature as score-based modeling [37].

D. Policy Composition and Sampling

Song et al. [39] show that score-based and denoising diffusion models can be considered as discretizations of a family of stochastic differential equations (SDE) that slowly add noise to the data distribution. For the generation process, the time-reversal of this SDE was given by Anderson [1] $da = \left[f(a,t) - g(t)^2\nabla_a log\ q_t(a)\right]dt + g(t)d\bar{w}$, where $\bar{w}$ is a standard Wiener process for reverse time, $g(t)$ is the scalar diffusion coefficient and $f(\cdot, t)$ is the drift coefficient. The discretization of the reverse SDE equation leads to the ancestral sampling [39] method proposed by Ho et al. [20] $a_{t-1} \sim \mathcal{N}\left(a_t; \frac{1}{\sqrt{\alpha_t}}\left[a_t + (1-\alpha_t)\nabla_{a_t} log\ q(a_t)\right], \sqrt{1-\alpha_t}\mathcal{I}\right)$. Equation 3 is then used to obtain an estimate of the score of the perturbed data distribution $\nabla_a log\ q_t(a)$, where the transition kernel $q(\tilde{a}|a)$ varies between approaches. Diffusion models use a forward transition kernel of $\mathcal{N}(\tilde{a}; \sqrt{\bar{\alpha}_t}a, (1-\bar{\alpha}_t)I)$ while score-based model typically use $\mathcal{N}(\tilde{a}; a, \sigma_t^2 I)$, where α_t and σ_t are respective noise scales.

Du et al. [12] suggest an equivalence between the loss functions of diffusion in Equation 1 and score-based models in Equation 4 to present the result in Equation 5. The same can be verified through the substitution of the diffusion transition kernel $q(a_t|a)$ in Equation 3 and parametrizing the score with the gradient of the energy function $\nabla_a f_\theta(a + \sigma_t\epsilon)$, only to obtain Equation 4.

$$\frac{-1}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}_\theta(a_t, t) \approx \nabla_a log\ q_t(a_t) \approx \nabla_a f_\theta(a + \sigma_t\epsilon) \quad (5)$$

To sample from the product distribution, we need the score of the composition at each noise scale of the ancestral sampling chain. Our product distribution can be expressed as $p^{comp}(a_0) = p_\theta^1(a_0) * p_\theta^2(a_0) \propto e^{f_\theta^1(a_0)+f_\theta^2(a_0)}$, where a_0 has been specifically written to reflect that the distributions are composed in the data space. Then the score of the composed distribution at diffusion time t can be written as:

$$\nabla_{a_t} log\ q^{comp}(a_t) = \nabla_{a_t} log\left(\int\left[\prod q^i(a_0)\right]q(a_t|a_0)da_0\right) \quad (6)$$

A long line of works instead add the individual scores of the distributions being composed $\sum_i\left(\nabla_{a_t} log\left[\int q^i(a_0)q(a_t|a_0)da_0\right]\right)$, since Equation 6 is not tractable. Du et al. [12] bring this out as the reason for inferior quality of samples from composed image distributions and suggest Annealed MCMC samplers instead of ancestral sampling that does not result in the correct sequence of marginals expected by the reverse diffusion process. Unadjusted Langevin Dynamics (ULA) [35], also used by score-based models [39], samples from the reverse transition kernel $\mathcal{N}\left(a_t; a_{t-1} + \frac{\sigma_L^2}{2}\nabla_{a_t} log\ p(a_t), \sigma_L^2\mathcal{I}\right)$, where $\frac{\sigma_L^2}{2}$ corresponds to the step size. Metropolis corrections lead to better sample quality but require explicit parametrization of the energy function, leading to more computational overhead [12].

III. Methodology

A. Blending Motions Using Diffusion Models

Diffusion models have shown impressive results for compositional generation [11, 12]. As elaborated in Section II-D, diffusion and score-based models can be understood as discretizations of forward and reverse SDEs, where the score of the noisy data distribution can be estimated from Equation 3. Varying forward transition kernels $q(a_t|a_0)$ give rise to different marginals $q(a_t)$ at each time-step for diffusion and score-based models. The score of the marginal of the composed distribution can be written as Equation 6. Substituting the forward transition kernel for diffusion models $q(a_t|a_0)$ in Equation 6, we get:

$$\nabla_{a_t} log\left(\int\left[\prod q^i(a_0)\right]\Phi\left(\frac{a_t - \sqrt{\bar{\alpha}_t}a_0}{1-\bar{\alpha}_t}\right)da_0\right) \quad (7)$$

Where Φ is the standard normal distribution. Substituting $\sqrt{\bar{\alpha}_t}a_0$ as a_0', we get the following equation which resembles Gaussian convolution corresponding to the forward transition kernel for score-based models.

$$\nabla_{a_t} log\left(\int\left[\prod q^i(\frac{a_0'}{\sqrt{\bar{\alpha}_t}})\right]\Phi\left(\frac{a_t - a_0'}{1-\bar{\alpha}_t}\right)da_0'\right) \quad (8)$$

For brevity, we have eliminated the constant terms in Equations 7 and 8 due to the action of log. This resembles the score of the composed distribution corresponding to the forward transition kernel of score-based models, with an extra factor of $\frac{1}{\sqrt{\bar{\alpha}_t}}$ multiplied to the input of each distribution being composed. Here, we have split the effect of the mean and variance of the forward diffusion transition kernel to suggest that the individual distributions being composed are not invariant across time-steps.

For blending between distributions at each time-step, a weighted average of the scores returned by the individual policies can be utilized. This is equivalent to sampling from $\prod_i^N p_i^{w_i}$ where $\sum_i^N w_i = 1$, where p indicates the distribution at a specific timestep. We can write the averaged score estimate as

$$\frac{1}{N}\sum_i^N\left(\nabla_{a_t} log\ \left[\int q^i(\frac{a_0'}{\sqrt{\bar{\alpha}_t}})\Phi\left(\frac{a_t - a_0'}{1-\bar{\alpha}_t}\right)da_0'\right]\right) \quad (9)$$

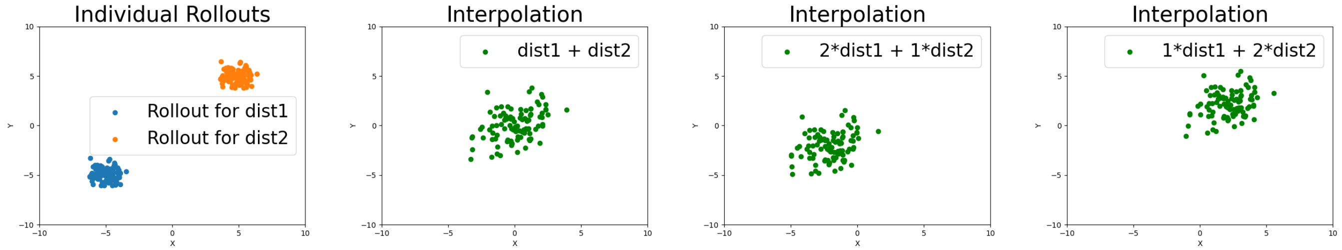

Fig. 2: This panel of figures shows the result of 100 samples from A: the individual distributions. B: Derived from the approximated score giving equal weighting to both. C: Giving 2:1 weighting to the distribution 1 and 2 respectively D: Giving 1:2 weighting to the distributions. The scores are normalized with respect to their weights. These images clearly show that diffusion models can have an interpolating effect between two distributions based on the weighting of the scores.

As suggested by [12], for sampling from composed distributions, ULA along with ancestral sampling is used to ameliorate the issue of mismatch between the sequence of marginals resulting from the approximation and true composition. Annealed MCMC sampling at different noise scales helps the sampling process converge to a sequence of distributions that is different from the one implied by Equation 8. However, we utilize this to our advantage by using the resulting sequence of marginals to blend distributions. The score approximated by Equation 9 composes perturbed (and temperature scaled) distributions, rather than perturbing the composed distribution, at each time-step. Similar to the effect of introducing perturbations at different scales observed by [39], composing perturbed distributions populates the low density regions with a score estimate directed towards the mean of the reverse diffusion transition kernel $q(a_{t-1}|a_t, a_0)$. Equation 9 can be equivalently written as Equation 10, where μ_i are the means of the individual reverse transition kernels.

$$\frac{1}{1-\alpha_t}\left(\frac{\sqrt{\bar{\alpha}_t}}{N}\sum_{i=1}^{N}\mu_i(a_t,t) - a_t\right) \tag{10}$$

We hypothesize that the distribution converges to regions where the net weighted scores cancel to zero. This suggests that composing diffusion models using this commonly used approximation has an interpolating-effect between the individual distributions through the weighted averaging of scores. This interpolation tends to actual composition of temperature scaled distributions as the noise tends to zero at the end of the sampling process.

To verify our claims, we train two diffusion models on 2D data samples from Gaussian's with means at $(5, 5)$ and $(-5, -5)$ and variance 1. The data distribution has limited overlap. The results on composing them using reverse diffusion sampling are shown in figure 2. On adjusting the weight applied to the score of each model, the samples derived from the approximated score of the composition shift towards the respective distributions, implying that we can blend them in relative proportions of their scores.

B. Composition Metric

We learn several base policies in the configuration-space using demonstrations generated in simulation. These base policies are then composed to generate novel motion. The demonstrations for the base policies are generated based on trajectories in the task-space, following which differential inverse kinematics is used to obtain configuration-space trajectories. This is essential as then the effect of composition can also be evaluated in the task space, rather than the configuration-space. Notably, the work by De Bortoli [10] bounds the convergence of diffusion models when the data distribution is supported by a low-dimensional manifold.

To the best of our knowledge, there is no existing approach or dataset in robotics that enables quantitative evaluation of samples from the composed policy. Wang et al. [44] and Urain et al. [41] use success rate as a metric to evaluate policy composition. However, this is not indicative of the quality of the composed motion. Since we want to evaluate a composed probability distribution using samples, we require a metric that measures the distance between the composed and the individual distributions, usually in the absence of the ground truth composition. Several integral probability metrics have been proposed in the image generation literature such a FID [18] and Maximum Mean Discrepancy (MMD) [14] to quantitatively evaluate the generated samples with respect to the data distribution. However, evaluating the quality of the composed motion in the task space is a complex problem as the space of policy composition may be different. Since the effect of motion composition is visible in the task space, the metric for evaluating the composed motion should also operate in the task-space, while our policies compose in the configuration-space. To that end, we propose a novel approach to measure the quality of composition for robot motion called MMD-FK. Our metric for m and n samples from the two distributions respectively can be expressed as:

$$\begin{aligned} \hat{dist}^2_{MMD-FK}(X,Y) = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}K_{FK}(x_i,x_j) + \\ \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n}K_{FK}(y_i,y_j) \\ -\frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}K_{FK}(x_i,y_j) \end{aligned} \tag{11}$$

It leverages MMD for it's kernel support that enables measurement of the distance between two distributions in terms of the distance between their feature means in a latent space. Lastly, for the kernel that will take configuration state as the input to evaluate task-space distances, we use the positive-definite Forward Kinematics kernel as suggested in Das and Yip [9]. Here K_{FK} is the positive-definite Forward Kinematics kernel in Equation 12. Equation 12 sums over the m control points defined on the robot, typically associated with each link in the kinematic chain. For our purposes, one control point on each kinematic chain allows us to capture the movements of the links of the robot in the task-space. In Equation 12, K_{RQ} is a second-order rational quadratic kernel $K_{RQ}(x, x') = \left(1 + \frac{\gamma}{2}||x - x'||^2\right)^{-2}$, with the width of the kernel being $\gamma > 0$.

$$K_{FK}(x, x') = \frac{1}{M}\sum_{m=1}^{M} K_{RQ}(FK_m(x), FK_m(x')) \qquad (12)$$

IV. Experimental Results

A. Data Generation and Model Architecture

We generate data using damped-least squares based differential inverse kinematics [7] for Franka Research-3 robot [1] in Mujoco [40]. We generate 250 demonstrations for training each of the base policies. We generate demonstrations for motor primitives [29] of arm extension (reach task), shaking, oscillating and circling about the x, y and z directions/axis. Unlike previous works that compose broad-based action distributions [44, 41], we show our results on narrow distributions to illustrate novel motion generation more clearly. We also train policies with multiple primitives to validate our compositional framework, as explained in the subsequent sections. We generate demonstrations for different states in the task-space of the robot and record the joint states at the control frequency.

Our policies are trained on the smaller variant of DiT [32], conditioned on the initial state of the robot in configuration space. The model $\hat{\epsilon}_\theta(a_t, o, t)$ learns to predict the noise that was added to the input a_t, conditioned on the diffusion timestep t and the observation o using AdaLN [34]. We train all the models to predict the whole trajectory of a set length L, conditioned on the initial state of the robot.

B. Results

We present results for 3 cases- composition of distributions corresponding to multiple primitives, blending single-primitive distributions and interpolating between multi-primitive distributions. We graphically show samples of the rollout from the individual and composed polices and also evaluate the MMD-FK distance between them and the ground-truth, wherever available.

1) Composition of Policies with Common Primitives: We train policies with multiple primitives to test if our compositional approach is able to sample from regions of high probability in both the distributions. We train a policy to reach towards the $+X$ or $+Y$ direction and another to reach towards the $+X$ or the $-Y$ direction. We expect the composed policy to sample from the modes corresponding to reaching towards the $+X$ direction, as can be seen in figure 3. Composing policies to sample from the common regions of high probability was also shown for the reach and obstacle avoidance task by [41], however using analytically crafted policies. The MMD-FK metrics obtained for this specific case are provided in table I.

TABLE I: Composition: MMD-FK values between samples from the composed and the individual distributions. We also show the value between the composed distribution and the expected ground truth +X. Values are shown both for reverse diffusion and ULA.

	+X/+Y	+X/-Y	+X
Reverse Diffusion	0.055	0.056	0.0578
ULA	0.0518	0.0529	0.061

2) Blending Single-Primitive Distributions: We show interpolation between policies by leveraging the approximation used for composing diffusion models III-A. We compose a policy learned to follow a circular trajectory in the YZ plane and a reach policy moving along the X axis to create a spiral, as shown in figure 1. We further can modulate the interpolation by re-weighting the addition of scores corresponding to the individual policies. The MMD-FK distance for the interpolation is as given in table II. For comparison, the distance obtained using the metric between policy rollouts and ground-truth demonstrations for the reach-X and circle-YZ motions are 0.01 and 0.017.

TABLE II: Interpolation: MMD-FK values between samples from the interpolated and the individual distributions. Values are shown both for reverse diffusion and ULA. We measure the distance between the spiral and the individual distributions of circle and reach.

	Reach	Circle
Reverse Diffusion	0.1662	0.1393
ULA	0.1674	0.1394

3) Interpolating Between Multi-Primitive Distributions: Interpolation between two single-primitives distributions may also be achieved by directly sampling from the modes of the individual policies and then adding the configuration-space values together. However, this brute-force way does not interpolate faithfully for multi-modal distributions as is usually the case for real world policies. We show that it is possible to use our compositional framework to interpolate between the nearest modes of two policies in the presence of other modes. We show two test cases where we isolate modes from distributions and generate novel motions by interpolating from them. We isolate modes in policies trained to reach +X or +Y, and +X or -Y using the directional similarity of the trajectories of circles, as shown in figure 4 and 5.

[1]https://franka.de/research

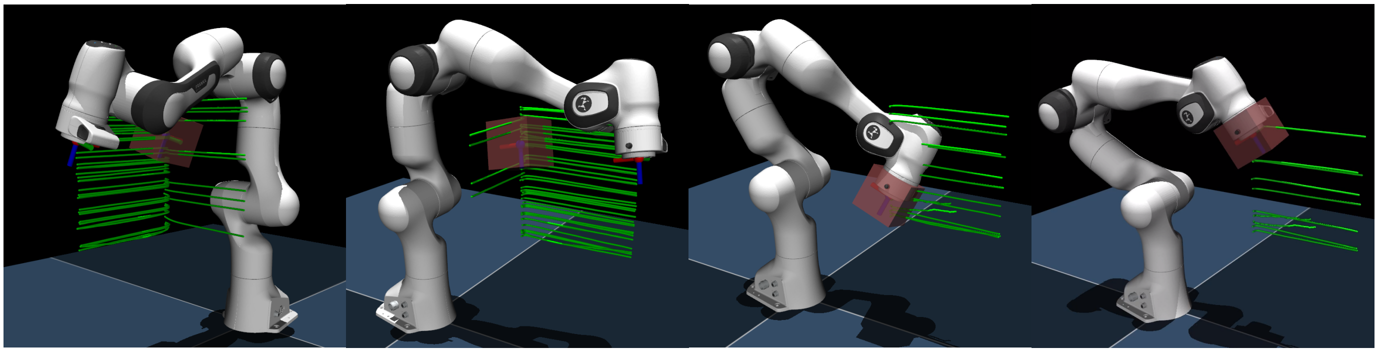

Fig. 3: This panel of figures shows the policy rollout for A: Policy trained on +X and +Y data. B: Policy trained on +X and -Y data. C: Composed policy from A and B sampled using reverse diffusion. D: Composed policy sampled using ULA. The composed samples are primarily generated from +X as expected.

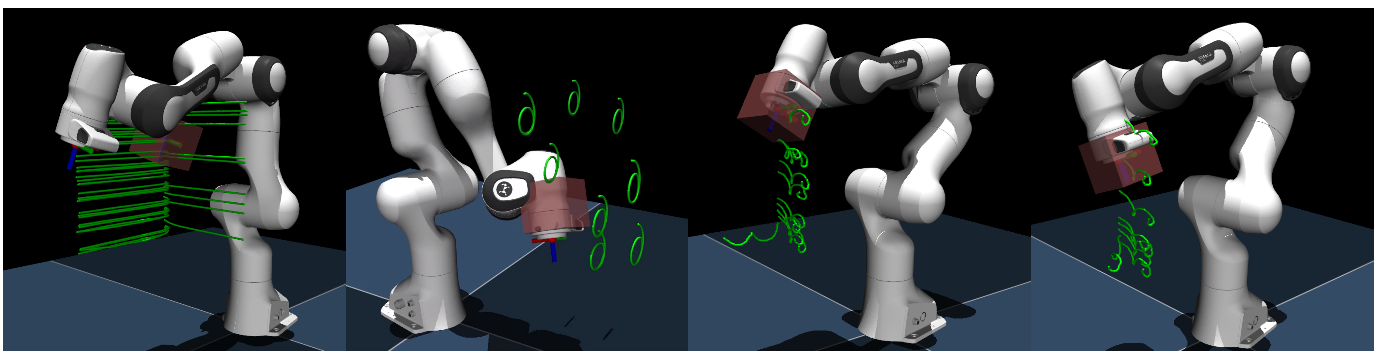

Fig. 4: This panel of figures shows the policy rollout for A: Policy trained on +X and +Y data. B: Policy trained on circle data C: Composed policy from A and B sampled using reverse diffusion. D: Composed policy sampled using ULA. The expected outcome of the composition is an interpolation between the circle and the reach towards +Y axis due to the directional similarity. We are able to isolate the mode from the policy and generate a novel motion by interpolating between circle and reach towards +Y. We see better sample quality from ULA over reverse diffusion, which supports the claims by [12].

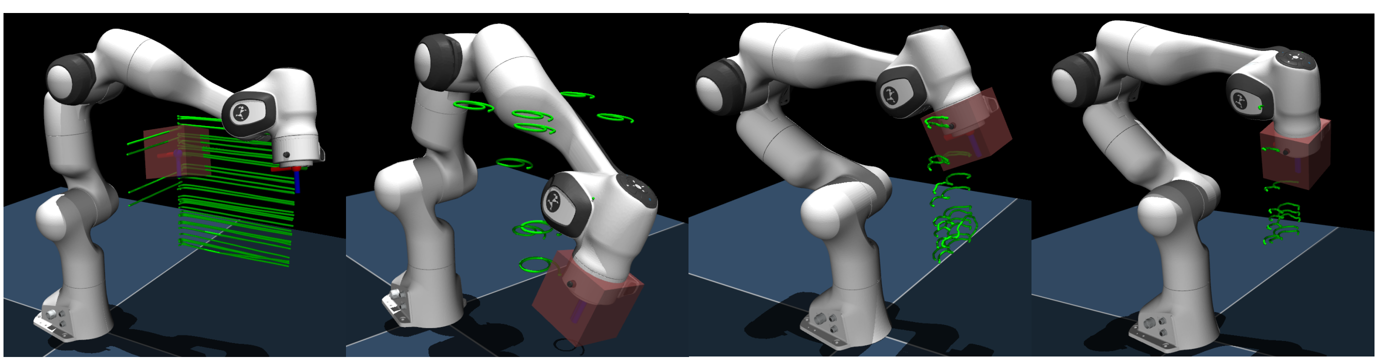

Fig. 5: This panel of figures shows the policy rollout for A: Policy trained on +X and -Y data. B: Policy trained on circle data in the XY plane C: Composed policy from A and B sampled using reverse diffusion. D: Composed policy sampled using ULA. The expected outcome of the composition is an interpolation between the circle and the reach towards -Y axis due to the directional similarity. We are able to isolate the -Y mode from the policy and generate a novel motion by interpolating between circle and reach towards -Y.

TABLE III: Interpolation: MMD-FK values between samples from the interpolated and the individual distributions. Values are shown both for reverse diffusion and ULA. We measure the distance between the spring and the circle in YZ plane and multi-modal policy of reaching +X or +Y.

	Reach	Circle
Reverse Diffusion	0.1644	0.2297
ULA	0.1718	0.2387

V. Discussion and Limitations

There are many applications of blending between motions such as inducing various styles in robot movement. There is a long line of works on generating content conforming to specific styles such as anime, sketch or comic for images [46] and polite, toxic or Shakespearean for text [22]. This is achieved through style transfer from a given sample [46], or through text conditioned generation [30, 23]. Generating the same content in different styles from a model meets the specific demands of the consumers of the content. However, this requirement is not limited to texts or images but also applies to the way humans, and consequently robots behave. Our method can help induce styles in different motions, thus filling the gap in robotics.

Our work has several limitations which form the possible avenues for future research.

- In this work we have combined primitive motions. In future, we would like to extend this work to more complex skills and possibly visuo-motor policies.
- We would like to explore if the interpolated motion can be used as a prior for few-shot learning of a novel skill.
- We observe better quality of trajectories when sampled using ULA for some of the compositions. However, this is not reflected in our metric possibly due to the overpowering effect of jitters.

VI. Conclusion

In conclusion, we propose a novel approach to blend motions using diffusion models by utilizing a weighted average of scores, an approximation employed to sample from composed distributions. This can be leveraged in robotics to generate novel motion by blending specific modes of distributions. We generate data for chosen motor primitives in the configuration space and show that our method can faithfully blend the nearest modes of two distributions. Finally, we also propose a novel metric to evaluate composed robot motions in the task-space, irrespective of the actual space of composition. We have several avenues to improve our work, such as application to real world domains and developing a better calibrated metric to discriminate the quality of the composed trajectory.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12 (3):313–326, 1982.

[2] Jacob Andreas. Measuring compositionality in representation learning. *arXiv preprint arXiv:1902.07181*, 2019.

[3] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50:5–43, 2003.

[4] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture, 2016.

[5] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *arXiv preprint arXiv:2309.01918*, 2023.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[7] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.

[8] 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.

[9] Nikhil Das and Michael C Yip. Forward kinematics kernel for improved proxy collision checking. *IEEE Robotics and Automation Letters*, 5(2):2349–2356, 2020.

[10] Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

[11] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems*, 2020.

[12] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc, 2023.

[13] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106 (496):1602–1614, 2011.

[14] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[15] Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, and Sergey Levine. Composable deep reinforcement learning for robotic manipulation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6244–6251. IEEE, 2018.

[16] Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

[17] Zhanpeng He, Ryan Julian, Eric Heiden, Hejia Zhang, Stefan Schaal, Joseph J Lim, Gaurav Sukhatme, and

Karol Hausman. Zero-shot skill composition and simulation-to-real transfer by learning task representations. *arXiv preprint arXiv:1810.02422*, 2018.
[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
[19] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
[21] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
[22] Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205, 04 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00426. URL https://doi.org/10.1162/coli_a_00426.
[23] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18062–18071, June 2022.
[24] Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to compose visual relations. *Advances in Neural Information Processing Systems*, 34: 23166–23178, 2021.
[25] Weiyu Liu, Jiayuan Mao, Joy Hsu, Tucker Hermans, Animesh Garg, and Jiajun Wu. Composable part-based manipulation. *arXiv preprint arXiv:2405.05876*, 2024.
[26] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.
[27] Jorge A. Mendez and Eric Eaton. How to reuse and compose knowledge for a lifetime of tasks: A survey on continual learning and functional composition, 2023.
[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.
[29] Ferdinando A Mussa-Ivaldi. Modular features of motor control and learning. *Current opinion in neurobiology*, 9 (6):713–717, 1999.
[30] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
[31] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.
[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
[33] Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems*, 32, 2019.
[34] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
[35] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996.
[36] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.
[37] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.
[38] Yang Song and Diederik P. Kingma. How to train your energy-based models, 2021.
[39] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
[40] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.
[41] Julen Urain, Anqi Li, Puze Liu, Carlo D'Eramo, and Jan Peters. Composable energy policies for reactive motion generation and reinforcement learning. *The International Journal of Robotics Research*, 42(10):827–858, 2023.
[42] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674, 2011.
[43] Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery, 2024.
[44] Lirui Wang, Jialiang Zhao, Yilun Du, Edward H Adelson, and Russ Tedrake. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024.
[45] Zhutian Yang, Jiayuan Mao, Yilun Du, Jiajun Wu, Joshua B Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Compositional diffusion-based continuous constraint solvers. *arXiv preprint arXiv:2309.00966*, 2023.
[46] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023.

[47] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.