# Few-shot Imitation Learning via Compositional Diffusion

Hanming Ye, Yiding 'Vincent' Song

*CS 2821r Final Project, Fall 2025*

**Abstract**

Few-shot imitation learning requires agents to acquire a new behavior from only a handful of demonstrations. In continuous control and motion generation, this setting is challenging: behavior cloning suffers from compounding errors, while reinforcement learning is sample-inefficient. We propose a generative alternative that turns few-shot learning into inference under a compositional behavior prior. Our method jointly trains (i) an encoder that maps a trajectory demonstration to a set of $K$ latent representations and (ii) a diffusion model whose denoising field is the sum of $K$ latent-conditioned components. At test time, we infer a task representation by encoding demonstrations into latents then generating new trajectories by sampling from the latent-conditioned diffusion model. Empirically, the learned latent space supports controllable variation and faithful reconstruction in Maze2D navigation, improves qualitative few-shot motion generation on MoCap, and yields higher success on new driving scenarios compared to strong baselines. Results are best viewed on our project website.

Humans acquire new skills from very little data. A few observed demonstrations can be enough to reproduce a motion. This is the problem of few-shot learning: how can

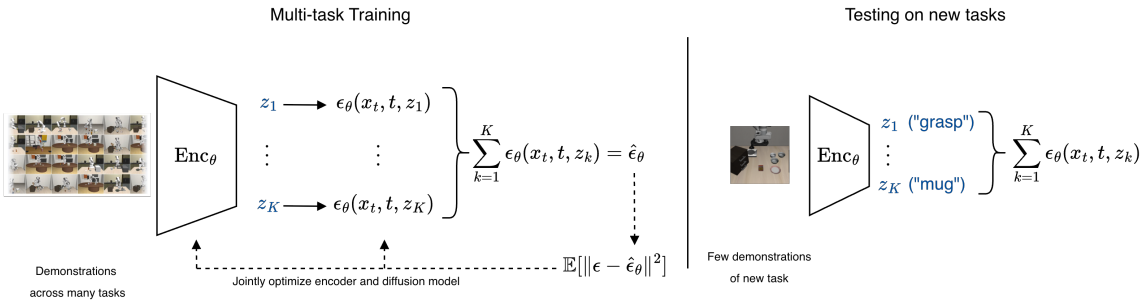


Figure 1: Illustration of our approach. During training, the model discovers concepts across a range of tasks. In particular, we jointly train 1) an encoder that maps demonstrations to latent representations, and 2) a compositional diffusion model conditioned on these representations. Then, at test-time, we infer latents from few demonstrations of a new task and use them to condition our model.

we design machine-learning models that can learn new tasks when given only a few demonstrations?

Despite extensive study, this problem remains unsolved in the robotics domain. One approach is to learn the desired policy directly from demonstrations (behavior cloning). But compounding errors and covariate shift often push the policy into unseen states at test-time [12]. . Another approach is to learn from rewards through reinforcement learning. However, it is sample-inefficient due to sparse supervisory signals and has limited ability to generalize to unseen behaviour. Instead, to enable few-shot generalization, we must leverage prior knowledge about the distribution of tasks.

To achieve this, a natural method is to make *prior knowledge* explicit. Meta-learning trains systems that "learn how to learn" by adapting quickly to new tasks from small datasets [11, 9, 12, 18]. Program-induction approaches aim to build libraries of reusable skills (eg. "grasp", "push") and compose them at test time [2, 10]. These lines of work all compress a training distribution of tasks into a representation that assists inference on unseen tasks. Yet, in continuous control, ... Symbolic representations of skills are often not flexible enough to capture the precision needed for motor control.

Inspired by successes in concept-learning in computer vision [8], we pursue a complementary route. We learn a generative model of trajectories that acts as a strong prior over behavior, then perform few-shot learning as *inference in the generative model*. Our key hypothesis is that few-shot generalization improves when the model learns a *decomposable* latent representation of behavior. We train a diffusion model which is the sum of $K$ additive components, each conditioned on a latent representation of the task Figure 1.

First, we train an encoder–diffusion pair end-to-end on offline demonstrations. Given a clean trajectory, an encoder produces a set of latent representations $z = (z_1, \ldots, z_K)$. We train a diffusion model to predict the noising term $\epsilon$ while conditioning on these latents. After training, we perform few-shot adaptation by encoding a small set of demonstrations for a new task and using the latent-conditioned diffusion model to generate new trajectories consistent with the demonstrations. Because the model is compositional at the level of score/energy contributions, it also supports inference-time composition [16, 7, 20, 8].

We make two main contributions: **(1)** We train an amortized encoder jointly with a compositional diffusion model, so few-shot task inference reduces to a forward pass on a handful of demonstrations rather than per-task optimization or fine-tuning, **(2)** We demonstrate the resulting compositional prior on long-horizon navigation, few-shot MoCap motion generation, and few-shot driving scenario synthesis, showing that the learned latent space supports controllable variation, faithful reconstruction of demonstrated behavior, and superior generalization to new concepts compared to baselines.

# 1 Related Work

**Meta-learning.** Meta-learning optimizes systems for rapid learning on new tasks [11]. In reinforcement learning, meta-RL methods such as $RL^2$ embed the learning

algorithm in recurrent dynamics [9], while context-based approaches (e.g., PEARL) infer latent task variables from experience for efficient off-policy adaptation [18]. In imitation learning, meta-imitation methods learn to infer task intent from one or a few demonstrations, including visual settings [12]. Our work shares the goal of fast task inference, but differs in mechanism: instead of adapting a policy/value function through an inner loop, we learn a generative prior over trajectories and perform task inference by encoding demonstrations into latent representations used for generation.

**Generative modeling for decision making.** Several recent works treat decision making as conditional sequence modeling. Decision Transformer frames offline RL as autoregressive modeling conditioned on return [3]. Conditional diffusion models have also been applied to decision-making by conditioning on returns, constraints, or skills, demonstrating flexible test-time control [1]. Diffuser plans by iteratively denoising trajectories and interprets guidance and inpainting as coherent planning operators [15]. Diffusion Policy uses diffusion to model action distributions for visuomotor control, improving robustness under multimodality and high dimensionality [4]. Our work builds directly on this line: we use diffusion as a trajectory prior, but introduce an explicit decomposition into latent primitives to support few-shot task inference and compositional recombination.

**Task concept learning from demonstrations.** A common strategy for few-shot behavior learning is to introduce an explicit *task concept* variable that summarizes the intent shared across demonstrations (e.g., goal attributes, relations, motion styles), then use that variable to generate or control behavior. Concretely, concept-learning pipelines typically (i) learn a conditional generative model of trajectories from a large pretraining set of behaviors paired with task representations, and (ii) infer a concept for a novel task from a few unlabeled demonstrations, enabling generation in new initial states and compositions with known concepts [17]. FTL-IGM exemplifies this approach by formalizing few-shot task learning as concept inference under a pretrained conditional generative model and demonstrating concept composition across domains such as navigation and motion capture [17]. In parallel, unsupervised concept discovery in vision has shown that representing data as multiple compositional components (rather than a single global code) can yield interpretable and recombinable factors; COMET, for example, discovers concepts as separate energy functions and recomposes them to generate new scenes [8].

**Guidance, steering, and composition in diffusion models.** Diffusion models admit a score-based interpretation in which denoising updates approximate gradients of a sequence of log-densities [13, 19]. This perspective motivates *guidance*: steering sampling by adding auxiliary gradients (e.g., from a classifier or condition likelihood) to bias generated samples toward desired attributes [5, 14]. More generally, diffusion models are composable: if multiple constraints correspond to additive terms in log-density, then combining behaviors can be approximated by summing score- or energy-like contributions during sampling [16]. In robotics, this has inspired methods that modify sampling rather than retraining policies, including policy composition and constraint integration

via diffusion [20], human-in-the-loop inference-time steering for diffusion policies [21], and guidance signals derived from external models such as learned dynamics or value functions [6, 22].

# 2 Method

We assume access to an offline dataset of demonstrations drawn from a distribution over behaviors. Each demonstration is a fixed-horizon trajectory segment

$$x_0 = (x_0^{(1)}, \dots, x_0^{(H)}) \in \mathbb{R}^{H \times d},$$

where $H$ is the segment length and $d$ is the per-timestep feature dimension. In navigation tasks, $x_0^{(t)}$ typically concatenates state and action. In motion generation (e.g., MoCap), $x_0^{(t)}$ may contain only states (joint positions), with no actions. At test time, we receive a small set of demonstrations $\{x_0^{(i)}\}_{i=1}^n$ for a novel task and wish to generate new trajectories consistent with the demonstrated behavior (which may have an implicit goal like "pick up the coffee mug").

## 2.1 Diffusion preliminaries for trajectories

We model trajectories with a denoising diffusion probabilistic model (DDPM) [13]. Let $T$ be the number of diffusion steps and let $\{\beta_t\}_{t=1}^T$ be a variance schedule. Define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. The forward process adds Gaussian noise to data (i.e. trajectories) via

$$q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\, x_{t-1},\, (1-\alpha_t)I)\,,$$

which implies the closed form

$$q(x_t \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t)I\big)\,, \quad x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ \epsilon \sim \mathcal{N}(0, I).$$

A neural denoiser $\epsilon_\theta$ predicts the noise from $(x_t, t)$, trained with the standard noise-prediction objective [13]:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{x_0,\, t,\, \epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right].$$

Following prior work, we interpret trajectory generation and constraint satisfaction as sampling under this learned prior, optionally with inpainting-style constraints [15].

## 2.2 Latent primitives via an encoder

Our goal is to endow the trajectory prior with a *decomposable* latent representation that supports few-shot adaptation and recombination. Given a clean trajectory segment $x_0$, we infer a set of $K$ latent primitives:

$$z = (z_1, \dots, z_K) = \text{Enc}_\phi(x_0), \qquad z_k \in \mathbb{R}^{d_z}.$$

The encoder $\text{Enc}_\phi$ is a lightweight temporal network (e.g., temporal CNN) that maps $x_0 \in \mathbb{R}^{H \times d}$ to $K$ vectors. We train $\phi$ jointly with the diffusion model using the same diffusion objective (below), so primitive inference is *amortized* rather than performed by per-task optimization as in inverse generative modeling approaches [17].

### 2.3 Compositional latent-conditioned denoiser

We condition the denoiser on the inferred primitives and enforce a compositional (additive) structure:

$$\epsilon_\theta(x_t, t; z) = \sum_{k=1}^{K} \hat{\epsilon}_{\theta,k}(x_t, t; z_k). \tag{1}$$

This mirrors the way diffusion models can support composition through additive score/energy contributions [16, 7] while learning the components end-to-end within a single model.

### 2.4 Training objective

We train the encoder and denoiser jointly with the standard diffusion loss, conditioned on primitives inferred from the *clean* trajectory:

$$z = \mathrm{Enc}_\phi(x_0), \tag{2}$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \tag{3}$$

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{x_0, t, \epsilon} \left[ \| \epsilon - \epsilon_\theta(x_t, t; \mathrm{Enc}_\phi(x_0)) \|_2^2 \right]. \tag{4}$$

We backpropagate through both $\epsilon_\theta$ and $\mathrm{Enc}_\phi$, so the encoder learns to produce primitive latents that make denoising easier under the additive decomposition.

### 2.5 Few-shot task inference

At test time, we infer a task-level latent representation from a few demonstrations and then sample trajectories from the latent-conditioned diffusion model. Given few-shot demonstrations $\{x_0^{(i)}\}_{i=1}^n$, we compute per-demo representations

$$z^{(i)} = \mathrm{Enc}_\phi(x_0^{(i)}),$$

and aggregate them into a task latent (primitive-wise mean):

$$\bar{z}_k = \frac{1}{n} \sum_{i=1}^{n} z_k^{(i)}, \qquad \bar{z} = (\bar{z}_1, \ldots, \bar{z}_K).$$

We then generate trajectories by sampling the reverse diffusion process using $\epsilon_\theta(\cdot; \bar{z})$.

## 3 Results

We evaluate our compositional diffusion model on three domains that stress few-shot generalization from demonstrations: (i) long-horizon navigation in Maze2D, (ii) few-shot motion generation on MoCap, and (iii) few-shot driving scenario generation. Across domains, the training protocol is the same: we train an encoder jointly with a diffusion model on trajectories from a set of test tasks. At test time, we encode a few demonstrations into latents and sample trajectories by denoising with those latents. **Results are best viewed on our website.**

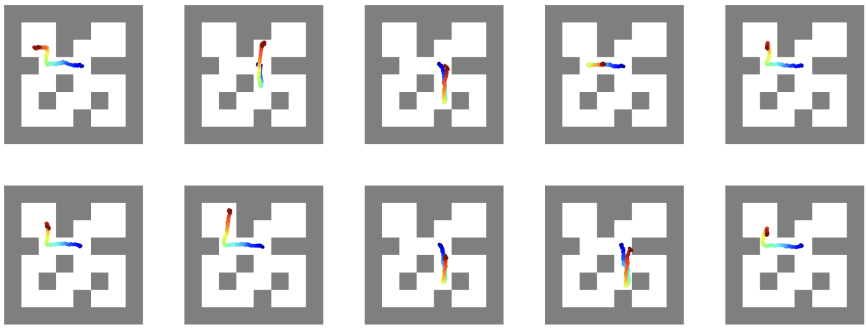

Figure 2: **Trajectories generated with random latents**. We use a model where the number of latent dimensions $k$ and the number of primitives $n$ are set to 1. For each trial, we sample a latent $z \sim \mathcal{N}(0, 1)$ and plot the trajectory generated by $\epsilon(\mathbf{s}_0, \text{Enc}(\mathbf{z}))$. The trajectories vary, while successfully navigating the maze. This demonstrates that the model learns to represent end state locations in its latent space.

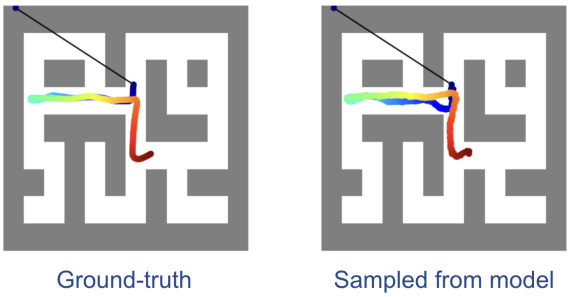


Figure 3: Reconstruction of a given trajectory. The generated sample follows the same path as the demonstration, despite not being a straight-line path between the start and end position. This shows that our model learns to capture detailed information about demonstrations in its latent representation.

## 3.1 Maze2D: latent space supports diverse solutions and trajectory reconstruction

We train on Maze2D trajectory data and generate rollouts conditioned on a fixed initial state.

First, we probe whether the latent space parameterizes meaningful variations by sampling random latents and generating trajectories from the same start state. Even with a single primitive and a one-dimensional latent, the model produces diverse paths that still successfully navigate the maze, suggesting that the latent controls high-level aspects of the solution such as the endpoint region rather than only local jitter (Figure 2).

Second, we test whether the encoder captures demonstration-specific information beyond the start and end states. Given a demonstration trajectory, we encode it into latents and sample from the model conditioned on the same start state. The generated trajectory follows the same nontrivial route as the demonstration (rather than collapsing to a straight-line shortcut), indicating that the latent representation captures detailed path structure (Figure **??**).

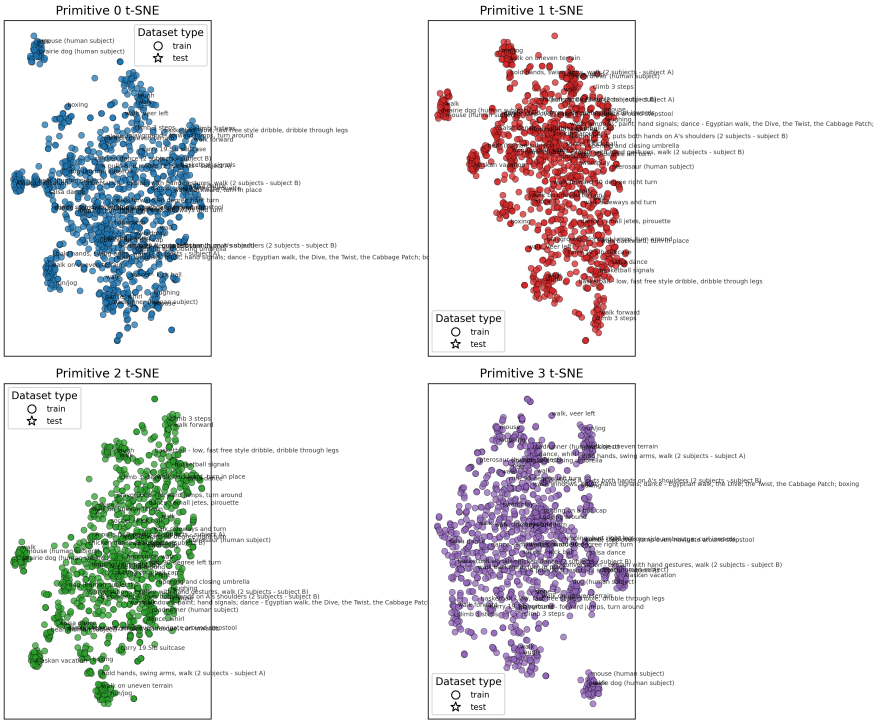


Figure 4: T-sne plot of the $k$-th latent representation from $\text{Enc}_\theta$ for every task in the MOCAP dataset, for $k = 1, \ldots, 4$. An interactive plot (where text is visible) can be found on our website.

## 3.2 MoCap: few-shot motion generation from demonstrations

We study few-shot motion generation on the MoCap benchmark, where trajectories are sequences of human poses (no actions). At test time, the model receives only a few demonstrations of a held-out action and must generate new motion sequences consistent with the demonstrated concept. We aggregate the latent primitives inferred from the demonstrations into a task representation and then sample novel sequences conditioned on this task latent, clamping the initial pose when required.

Qualitatively, our generations more faithfully express the target action than baselines that (i) regress directly to a policy (BC), (ii) attempt in-context adaptation, (iii) condition on language, (iv) use a VAE-style latent variable model, or (v) use the FTL-IGM concept-learning baseline. In particular, on 3-shot "Jumping Jacks" and "Breast Stroke", our samples exhibit clearer action-specific structure and more consistent limb coordination (see website).

To inspect the structure of the learned primitives, we embed the encoder outputs with t-SNE. Each primitive latent forms a structured, semantically meaningful organization over the dataset, and held-out tasks appear in coherent regions of the learned space, consistent with the idea that different primitives capture different factors of motion (Figure 4).

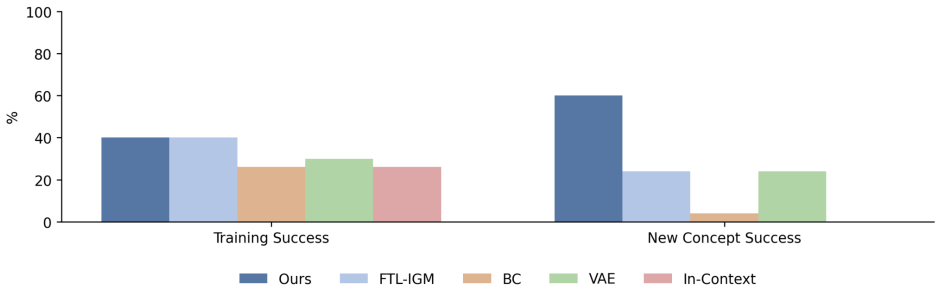


Figure 5: Success rates on training tasks (left) and testing tasks (right).

### 3.3 Driving: few-shot scenario generation and improved new-task success

We evaluate few-shot generalization on driving scenarios. The model is trained on a set of training scenarios (e.g., highway, merge, intersection) and tested on a novel scenario in a one-shot setting. The model generates plausible rollouts on training tasks (see website) and can synthesize behavior for the novel scenario (roundabout) from a single demonstration.

Quantitatively, our method matches the strongest baselines on training-task success, while substantially improving success on new concepts. In our experiments, the gap is most pronounced in new-task generalization: our method achieves 60% new-concept success versus 24% for FTL-IGM, while both achieve 40% on training tasks (Figure 5).

## References

[1] Anurag Ajay et al. "Is Conditional Generative Modeling all you need for Decision-Making?" In: *arXiv preprint arXiv:2211.15657* (2022). DOI: 10.48550/arXiv.2211.15657. URL: https://arxiv.org/abs/2211.15657.

[2] Ferran Alet, Tomás Lozano-Pérez, and Leslie P. Kaelbling. "Modular meta-learning". In: *Proceedings of the 35th International Conference on Machine Learning*. arXiv:1806.10166. 2018. DOI: 10.48550/arXiv.1806.10166. URL: https://arxiv.org/abs/1806.10166.

[3] Lili Chen et al. "Decision Transformer: Reinforcement Learning via Sequence Modeling". In: *arXiv preprint arXiv:2106.01345* (2021). DOI: 10.48550/arXiv.2106.01345. URL: https://arxiv.org/abs/2106.01345.

[4] Cheng Chi et al. "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion". In: *arXiv preprint arXiv:2303.04137* (2023). DOI: 10.48550/arXiv.2303.04137. URL: https://arxiv.org/abs/2303.04137.

[5] Prafulla Dhariwal and Alex Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. 2021. URL: https://arxiv.org/abs/2105.05233.

[6] Maximilian Du and Shuran Song. "DynaGuide: Steering Diffusion Policies with Active Dynamic Guidance". In: *arXiv preprint arXiv:2506.13922* (2025). DOI: 10.48550/arXiv.2506.13922. URL: https://arxiv.org/abs/2506.13922.

[7] Yilun Du et al. "Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC". In: *Proceedings of the 40th International Conference on Machine Learning*. PMLR 202. 2023. DOI: 10.48550/arXiv.2302.11552. URL: https://arxiv.org/abs/2302.11552.

[8] Yilun Du et al. "Unsupervised Learning of Compositional Energy Concepts". In: *Advances in Neural Information Processing Systems*. 2021. DOI: 10.48550/arXiv.2111.03042. URL: https://arxiv.org/abs/2111.03042.

[9] Yan Duan et al. "$RL^2$: Fast Reinforcement Learning via Slow Reinforcement Learning". In: *arXiv preprint arXiv:1611.02779* (2016). DOI: 10.48550/arXiv.1611.02779. URL: https://arxiv.org/abs/1611.02779.

[10] Kevin Ellis et al. "DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning". In: *arXiv preprint arXiv:2006.08381* (2020). DOI: 10.48550/arXiv.2006.08381. URL: https://arxiv.org/abs/2006.08381.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv preprint arXiv:1703.03400* (2017). DOI: 10.48550/arXiv.1703.03400. URL: https://arxiv.org/abs/1703.03400.

[12] Chelsea Finn et al. "One-Shot Visual Imitation Learning via Meta-Learning". In: *Proceedings of the 34th International Conference on Machine Learning*. arXiv:1709.04905. 2017. DOI: 10.48550/arXiv.1709.04905. URL: https://arxiv.org/abs/1709.04905.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *arXiv preprint arXiv:2006.11239* (2020). DOI: 10.48550/arXiv.2006.11239. URL: https://arxiv.org/abs/2006.11239.

[14] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *arXiv preprint arXiv:2207.12598* (2022). DOI: 10.48550/arXiv.2207.12598. URL: https://arxiv.org/abs/2207.12598.

[15] Michael Janner et al. "Planning with Diffusion for Flexible Behavior Synthesis". In: *Proceedings of the 39th International Conference on Machine Learning*. 2022. DOI: 10.48550/arXiv.2205.09991. URL: https://arxiv.org/abs/2205.09991.

[16] Nan Liu et al. "Compositional Visual Generation with Composable Diffusion Models". In: *European Conference on Computer Vision*. 2022. URL: https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136770426.pdf.

[17] Aviv Netanyahu et al. "Few-Shot Task Learning through Inverse Generative Modeling". In: *arXiv preprint arXiv:2411.04987* (2024). DOI: 10.48550/arXiv.2411.04987. URL: https://arxiv.org/abs/2411.04987.

[18] Kate Rakelly et al. "Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables". In: *arXiv preprint arXiv:1903.08254* (2019). DOI: 10.48550/arXiv.1903.08254. URL: https://arxiv.org/abs/1903.08254.

[19] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *arXiv preprint arXiv:2011.13456* (2020). DOI: 10.48550/arXiv.2011.13456. URL: https://arxiv.org/abs/2011.13456.

[20] Lirui Wang et al. "PoCo: Policy Composition from and for Heterogeneous Robot Learning". In: *arXiv preprint arXiv:2402.02511* (2024). DOI: 10.48550/arXiv.2402.02511. URL: https://arxiv.org/abs/2402.02511.

[21] Yanwei Wang et al. "Inference-Time Policy Steering through Human Interactions". In: *arXiv preprint arXiv:2411.16627* (2024). DOI: 10.48550/arXiv.2411.16627. URL: https://arxiv.org/abs/2411.16627.

[22] Hanming Ye. *Steering Diffusion Policies with Value-Guided Denoising*. OpenReview. 2025. URL: https://openreview.net/forum?id=wrcTncImde.

# 4 Appendix

**Parameterization.** We use a shared temporal U-Net backbone $f_{\theta,\text{back}}$ (as in trajectory diffusion for planning [15]) to produce features:

$$h = f_{\theta,\text{back}}(x_t, t) \in \mathbb{R}^{C\times H}.$$

For each primitive $z_k$, we compute FiLM-style modulation parameters

$$(\gamma_k, \delta_k) = g_\theta(z_k), \qquad \gamma_k, \delta_k \in \mathbb{R}^C,$$

broadcast them over time, and modulate backbone features:

$$h_k = (1 + \gamma_k) \odot h + \delta_k.$$

A shared projection head $p_\theta$ maps $h_k$ to a per-primitive noise prediction:

$$\hat{\epsilon}_{\theta,k}(x_t, t; z_k) = p_\theta(h_k) \in \mathbb{R}^{H\times d}.$$

We sum across $k$ to obtain $\epsilon_\theta(x_t, t; z)$.

This design reuses the standard diffusion backbone and introduces compositionality by (i) sharing the feature extractor and (ii) injecting each primitive through a simple modulation-plus-projection pathway.