Turning the Dial: Bridging Behavior Cloning and Reinforcement Learning via Timestep Modulation

Matthew M. Hong[1], Jesse Zhang[1], Anusha Nagabandi[2], Abhishek Gupta[1]

[1]University of Washington [2]Amazon FAR

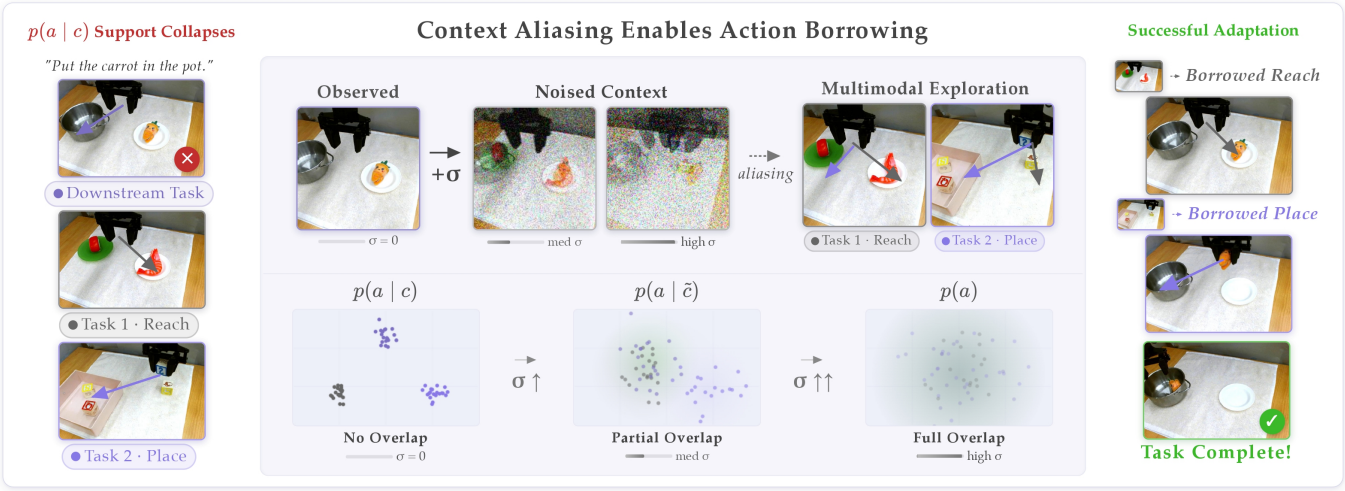

Fig. 1: TMRL bridges behavior cloning (BC) pre-training and RL fine-tuning by smoothing the conditioning of policy inputs (contexts). During pre-training, Context-Smoothed Pre-training (CSP) injects noise into contexts c, inducing a continuum from sharp imitation $p(a \mid c)$ to broader, marginal action distributions $p(a)$ via diffusion noise parameterized by σ (bottom row). This smoothing causes nearby contexts to overlap in representation space, with similar contexts (e.g., **downstream task**, **task 1** in left column) merging at lower σ than dissimilar ones (e.g., **task 2**). During RL fine-tuning, TMRL *learns* to dynamically *modulate* conditioning strength, interpolating between context-conditioned and exploratory actions for improved exploration and adaptation.

***Abstract*—Fine-tuning pre-trained robot policies with reinforcement learning (RL) is promising, but standard behavior cloning (BC) produces narrow, overconfident action distributions that generalize poorly and limit downstream RL improvement. We present a unified framework for bridging BC pre-training and RL fine-tuning. Our pre-training method, Context-Smoothed Pre-training (CSP), injects forward-diffusion noise into policy inputs, enabling a continuous spectrum between precise conditional imitation and broader action coverage. For efficient RL fine-tuning, we introduce Timestep-Modulated Reinforcement Learning (TMRL), which enables the agent to dynamically adjust conditioning strength via diffusion timestep modulation to control exploration. Across diverse settings, CSP integrates seamlessly with arbitrary policy inputs, from states to 3D pointclouds, and with image-input vision-language-action policies. TMRL with CSP significantly improves RL sample efficiency over prior approaches. Notably, TMRL enables successful real-world fine-tuning on manipulation tasks in under 1 hour. Videos and code available at https://weirdlabuw.github.io/tmrl/.**

I. Introduction

A dominant paradigm for training real-world robotic policies is to first pre-train on large-scale demonstration datasets via imitation learning, and then fine-tune the resulting policy with reinforcement learning (RL) in deployment environments [24, 25, 10, 30, 32, 15, 31, 8, 28, 27]. This RL fine-tuning stage is often critical for improving task precision and robustness [12, 15, 13]. However, because real-world robot deployment is costly and time-consuming, sample efficiency during RL remains a major bottleneck. While much prior work focuses on improving the efficiency of the fine-tuning algorithm itself, comparatively little attention has been given to ensuring that the pre-trained policy provides an *effective initialization for downstream RL*. In this work, we study how to design a pre-training procedure that better prepares policies for RL fine-tuning.

Standard behavior cloning (BC) trains a policy to directly imitate demonstrator actions [20, 1]. When demonstrations densely cover a context c (e.g., observations and task instructions), BC can effectively model the conditional action distribution $p(a \mid c)$ [5]. In sparse or unseen regions, however, BC overfits to observed data: the conditional support collapses, assigning near-zero probability to potentially optimal actions. Consequently, online rollouts provide little reward signal, and RL may fail entirely to improve behavior.

Recent work attempts to mitigate this issue by widening the learned action distribution by adding isotropic Gaussian noise to action targets during BC pre-training [24]. However, this approach has two key limitations. First, the optimal trade-off between action coverage and sample efficiency varies significantly across tasks and is rarely known *a priori*. Second, injecting noise directly into the action space often produces temporally incoherent behavior, such as a robot arm "dithering" across adjacent timesteps. Rather than heuristically broadening the action distribution, we propose a pre-training

objective that enables adaptive broadening while preserving action coherence.

Our key insight is to train policies that can *interpolate* between the conditional action distribution $p(a \mid c)$ and the marginal action distribution $p(a)$. While $p(a \mid c)$ provides precise behavior in familiar contexts, the marginal distribution $p(a)$ ensures broad action coverage across the entire dataset. By enabling smooth interpolation between these extremes, a policy can balance precision and exploration depending on the difficulty of or how unseen the downstream task is.

We instantiate our insight through *Context-Smoothed Pre-training (CSP)*, a pre-training framework that widens the support of BC policies by injecting noise into its contexts via a *forward diffusion process*. As the context noise increases, the learned policy transitions smoothly from the conditional distribution $p(a \mid c)$ toward the marginal distribution $p(a)$. At low noise levels, the policy represents a mixture of behaviors associated with nearby contexts to c [9], thereby producing a wider yet coherent distribution of action sequences for c. At maximum noise, the context becomes fully uninformative and the policy recovers $p(a)$, guaranteeing coverage over the full training distribution. Importantly, because CSP trains across all diffusion noise levels, the tradeoff between action coverage and conditioning-following can be selected at inference time by choosing the appropriate diffusion timestep.

Of course, the optimal amount of context conditioning may vary across timesteps within a trajectory. Therefore, we also introduce *Timestep-Modulated Reinforcement Learning* (TMRL), which learns to dynamically adjust the diffusion timestep during deployment. This mechanism provides an RL agent with an explicit control variable that modulates conditioning strength, allowing it to interpolate between conditional and marginal behaviors for more effective exploration. In effect, the policy learns when to rely on precise imitation and when to broaden its action support. In practice, TMRL enables steering/fine-tuning of arbitrary policies, from state-input diffusion policies to vision-language-action (VLA) models.

We evaluate across a range of simulated and real-world robotic tasks. First, we show that context-smoothed policies alone substantially improve action coverage and achieve stronger zero-shot performance on unseen tasks over standard BC and prior pre-training approaches. TMRL then converts this coverage advantage into significantly better RL sample efficiency on manipulation and navigation tasks in simulation, outperforming state-of-the-art steering methods even when the base policy contains sparse behavioral coverage. We further show that context-smoothing extends naturally to VLA policies and 3D-input policies, where noising VLM embeddings and point-clouds respectively enables broader exploration across contact-rich and dexterous manipulation tasks. Finally, we demonstrate that our approach scales to the real world, enabling rapid RL of manipulation behaviors within *an hour* of total experiment time, while baseline methods achieve near-zero success rates.

II. Timestep Modulated RL on Context- Smoothed Policiesing

A. Problem setting

Pre-training. We assume access to an offline dataset of near-optimal trajectories $\mathcal{D} = \{\tau_j\}$, where $\tau_j = \{(c_i, a_i)\}_{i=1}^{T_j}$ contains context–action pairs. The context c abstracts policy inputs such as states, images, point clouds, or language.

We learn a policy via a supervised objective:

$$\min_{\theta} \; \mathbb{E}_{(c,a)\sim\mathcal{D}}\big[\ell(\theta; c, a)\big], \tag{1}$$

which subsumes behavior cloning with $\ell(\theta; c, a) = -\log p_\theta(a \mid c)$, as well as alternatives such as denoising or score-matching objectives.

We focus on generative control policies (GCPs) [4, 29, 17], where $p_\theta(a \mid c)$ is parameterized by a generative model (e.g., diffusion or flow). For simplicity, we use (c, a) notation, though this extends to action chunks and context histories. Our approach does not depend on this specific policy class.

RL Fine-tuning. Given a pre-trained policy, we seek to efficiently adapt it to new contexts via RL. In GCPs, inference is controlled by a latent variable z, yielding $p_\theta(a \mid c, z)$ (e.g., diffusion noise initialization) [25]. We therefore optimize a high-level policy $\pi_{\text{HL}}(z \mid c)$:

$$\max_{\pi_{\text{HL}}} \mathbb{E}\Big[\sum_{t\geq 0} \gamma^t r_t\Big], \tag{2}$$
$$\text{s.t.} \quad z_t \sim \pi_{\text{HL}}(\cdot \mid c_t), \quad a_t^{1:H} \sim p_\theta(\cdot \mid c_t, z_t).$$

Action Coverage. A key challenge in fine-tuning is distribution shift: $p_\theta(\cdot \mid c, z)$ can collapse in novel contexts, limiting exploration. Following Wagenmaker et al. [24], a policy has demonstrator coverage $\kappa > 0$ if

$$p_\theta(a \mid c) \geq \kappa \cdot p^\beta(a \mid c), \tag{3}$$

where p^β is the demonstrator policy. Sufficient coverage is necessary for RL, as missing support ($\kappa \to 0$) prevents sampling reward-relevant actions.

Standard BC often yields poor coverage in low-density regions, producing ineffective initializations. Our approach addresses this via context-smoothed pre-training, enabling the RL agent to adaptively modulate action coverage across contexts.

B. Context-smoothing

Intuitively, context-smoothed policies make a simple change to standard policy pre-training: they inject noise into the context c while being pre-trained on the offline dataset $\mathcal{D}$. We show that doing so enables policies to expand their action coverage during evaluation and RL fine-tuning.

To define this formally - let $q_\sigma(\tilde{c} \mid c)$ be a corruption kernel that injects noise into the context, with noise scale $\sigma \geq 0$. We define the *context-smoothed* controllable policy as the mixture

$$\begin{aligned} p_{\theta,\sigma}(a \mid c, z) \;&\coloneqq\; \mathbb{E}_{\tilde{c}\sim q_\sigma(\cdot|c)}\Big[p_\theta(a \mid \tilde{c}, z)\Big] \\ &=\; \int p_\theta(a \mid \tilde{c}, z)\, q_\sigma(\tilde{c} \mid c)\, d\tilde{c}. \end{aligned} \tag{4}$$

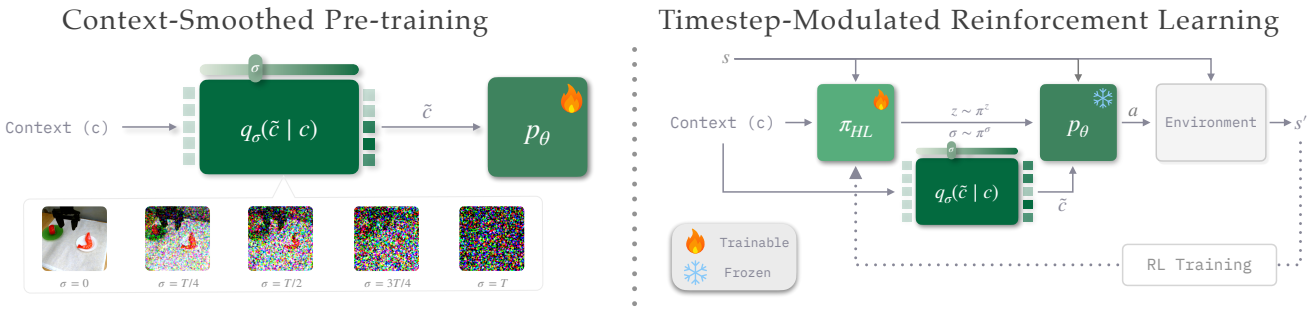

Fig. 3: **Timestep-modulated exploration via context smoothing.** (*Left*) During pre-training, a controllable policy p_θ is trained across all noise levels σ by corrupting the context c via the kernel $q_\sigma(\tilde{c} \mid c)$, producing a policy that can be queried at ny conditioning strength during inference. (*Right*) During RL fine-tuning, TMRL exposes p_θ with a *context-noise dial* σ as an explicit control variable for the high-level policy π_{HL}.

We refer to $p_{\theta,\sigma}$ as a *context-smoothed policy*. Given a context c, action inference with a context-smoothed policy amounts to corrupting the context with $\tilde{c} \sim q_\sigma(\tilde{c} \mid c)$, and then sampling actions with the corrupted context from $p_\theta(a \mid \tilde{c}, z)$.

Intuition behind context smoothed policies: For $\sigma \downarrow 0$, $q_\sigma(\tilde{c} \mid c)$ concentrates near c, so $p_{\theta,\sigma}(\cdot \mid c, z)$ approaches the original conditional controller. As σ increases, the corrupted context $\tilde{c}$ becomes less informative, and the induced action distribution becomes a broader mixture over behaviors associated with *nearby / aliased* contexts. This controlled interpolation between the conditional distribution $p(a|c)$ and the marginal $p(a)$ creates *structured action coverage expansion*: instead of random action noise, the policy can borrow coherent action chunks from related contexts present in the dataset.

a) Implementation of corruption kernel $q_\sigma(\tilde{c} \mid c)$: We define the corruption kernel q_σ via an iterative forward-noising process over contexts, much like the forward process of a diffusion model. Let $c_0 \equiv c$ and choose a variance schedule $\{\beta_t\}_{t=1}^{T_c}$ with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{i=1}^{t} \alpha_i$. The forward process implies the closed-form marginal

$$q(c_t \mid c_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, c_0,\ (1 - \bar{\alpha}_t) I\big). \tag{5}$$

We take the corruption kernel at noise level t_c to be

$$q_\sigma(\tilde{c} \mid c) \equiv q_{t_c}(\tilde{c} \mid c) = \mathcal{N}\big(\sqrt{\bar{\alpha}_{t_c}}\, c,\ (1 - \bar{\alpha}_{t_c}) I\big), \tag{6}$$

Since the forward corruption process is entirely determined by how many time-steps it is run forward, the noise-level σ for $q_\sigma(\tilde{c} \mid c)$ can be parameterized entirely by controlling the **diffusion timestep** $t_c \in \{0, \dots, T_c\}$. We use σ to refer to corruption kernels q_σ generally and t_c for our specific instantiation with diffusion noise.

C. Pre-training a context-smoothed policy

Training a context-smoothed policy requires learning action predictors conditional on corrupted context - $p_\theta(a \mid \tilde{c}, z)$. Towards realizability, we train a policy class that can be queried at arbitrary context-noise levels by providing σ (or t_c) as an explicit input to the policy - $p_\theta(a \mid \tilde{c}, z, \sigma)$. Concretely, we introduce a policy - $p_\theta(a \mid \tilde{c}, z, \sigma)$, and train it so that, for each training pair $(c, a) \sim \mathcal{D}$, we maximize action likelihood

Algorithm 1 Context-smoothed pre-training.

Input: dataset $\mathcal{D} = \{(c, a^{1:H})\}$, corruption kernel q_σ, noise-level sampler $\sigma \sim \mathcal{S}$
Initialize: parameters θ of $p_\theta(a^{1:H} \mid c, z, \sigma)$ (and any latent prior over z if used)
while not converged **do**
  Sample $(c, a^{1:H}) \sim \mathcal{D}$
  Sample $\sigma \sim \mathcal{S}$ (diffusion timestep t_c in our instantiation)
  Sample $\tilde{c} \sim q_\sigma(\cdot \mid c)$
  Update θ to minimize $\ell(\theta;\ a^{1:H}, \tilde{c}, \sigma)$
end while

noising the context with a small modification to Equation (1):

$$\min_\theta\ \mathbb{E}_{(c,a)\sim\mathcal{D}}\ \mathbb{E}_{\substack{\sigma\sim\mathcal{S} \\ \tilde{c}\sim q_\sigma(\cdot|c)}} \Big[\ell\big(\theta;\ a, \tilde{c}, \sigma\big)\Big], \tag{7}$$

where $\mathcal{S}$ is a distribution over noise levels and ℓ is any supervised learning objective suitable for the chosen controllable policy class. In our specific instantiation using GCPs as p_θ, ℓ is the denoising score-matching or flow-matching loss [23, 7, 22] and σ corresponds to the diffusion noise timestep used to corrupt $\tilde{c}$, t_c. Notably, this is *not* a much more complicated procedure than standard imitation learning; it merely introduces *controlled context smoothing* during pre-training, similar to data augmentation, and conditioning p_θ on the amount of smoothing σ. As we show next, this context-smoothed pre-trained policy can then be used for efficient RL fine-tuning. See Algorithm 1 for pseudocode.

D. RL fine-tuning with timestep-modulated context smoothing

Compared to standard pre-training, training a context-smoothed policy $p_\theta(a^{1:H} \mid c, z, \sigma)$ naturally expands the exploration space during RL fine-tuning. For example, in *steering algorithms* [25], which RL fine-tune by training a high-level RL policy to select the latent noise z to initialize action selection from a GCP (see Section II-A for a refresher), the high-level policy can now select both the latent z and the context-noise level σ:

$$(z_t, \sigma_t) \sim \pi_{\text{HL}}(\cdot \mid s_t),\ \tilde{c}_t \sim q_{\sigma_t}(\cdot \mid c(s_t)),\ a_t \sim p_\theta(\cdot \mid \tilde{c}_t, z_t, \sigma_t). \tag{8}$$

With a frozen context smoothed policy, we optimize $\pi_{\text{HL}}(z, \sigma \mid s)$ using any RL method (e.g., off-policy actor-critic [6]) to maximize the expected return in Equation (2).

Algorithm 2 Timestep-Modulated RL (TMRL).

Input: pretrained context-smoothed controller $p_\theta(\cdot \mid c, z, \sigma)$; initialize replay buffer $\mathcal{B}$
Initialize high-level actor $\pi_{\text{HL}}(z, \sigma \mid s)$ and critic(s)
for each training iteration **do**
  Observe s_t and set $c_t \leftarrow c(s_t)$
  Sample $(z_t, \sigma_t) \sim \pi_{\text{HL}}(\cdot \mid s_t)$, $\tilde{c}_t \leftarrow c_t$ smoothed with σ_t
  Sample and execute action chunk $a_t^{1:H} \sim p_\theta(\cdot \mid \tilde{c}_t, z_t, \sigma_t)$
  Store transitions in $\mathcal{B}$ and update π_{HL} and critics
end for

Intuitively, σ provides an exploration–exploitation, action coverage (Eq. (3)) dial: large σ aliases contexts (broader borrowing, better exploration, more coverage), while small σ sharpens conditioning (better exploitation, lower action coverage) once progress is found. See Algorithm 2 for pseudocode.

III. Theoretical Results: What does context smoothing buy us?

In this section, we provide theoretical evidence and empirical analysis to better understand the gains from timestep-modulated, context-smoothed policies for RL fine-tuning.

a) Empirical Analysis: We first empirically validate that increasing σ (or t_c) yields smooth, meaningful behavioral changes. We show in Figure 4 that as corruption noise increases, the distribution of actions at a given context broadens, thereby increasing overlap with nearby contexts. Taken to the extreme, this shows interpolation between the conditional $p(a \mid c)$ and marginal $p(a)$ action distribution.

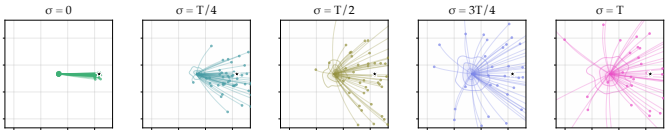

Fig. 4: **Empirical validation of context smoothing.** We train a context-smoothed diffusion policy $p(x, y \mid \tilde{\theta}, \sigma)$ with Equation (7) to produce 2D points on a unit circle conditioned on $c = \theta$. Here, $\sigma \in \{0, 1, ..., T\}$ representing diffusion scheduler timesteps. The above plot shows $p(x, y \mid \tilde{\theta} = 0, \sigma)$ at various levels of σ noise. As context noise controlled by σ increases ($\rightarrow T$), the induced action distribution becomes broader and overlaps more across nearby contexts; as noise decreases, conditioning sharpens and outputs center around unit circle points for $\theta = 0$.

b) Theoretical Analysis: We formalize how Gaussian context smoothing mitigates the coverage collapse of standard BC. Informally, we show: **(i)** increasing noise increases overlap between action distributions across contexts, and **(ii)** nearby contexts overlap more than distant ones. Together, these imply that if a context c' provides better action coverage than c, increasing noise σ increases overlap with $p(\cdot \mid c')$, improving coverage at c (cf. Equation (3)).

c) Setup (Gaussian context smoothing): Given $p(\cdot \mid c)$ with $c \in \mathbb{R}^d$, define the smoothed policy

$$p_\sigma(\cdot \mid c) := \mathbb{E}_{w \sim \mathcal{N}(0, I_d)} \big[p(\cdot \mid c + \sigma w) \big]. \tag{9}$$

Let $\text{TV}(\cdot, \cdot)$ denote total variation distance and $\text{Ov}(P, Q) := 1 - \text{TV}(P, Q)$.

Theorem 1 (Smoothing increases overlap and induces Lipschitzness)**.** *For any $c, c' \in \mathbb{R}^d$,*

$$\text{TV}\big(p_\sigma(\cdot \mid c), p_\sigma(\cdot \mid c')\big) \leq \frac{\mathbb{E}\|w\|}{\sigma} \|c - c'\|, \tag{10}$$

equivalently,

$$\text{Ov}\big(p_\sigma(\cdot \mid c), p_\sigma(\cdot \mid c')\big) \geq 1 - \frac{\mathbb{E}\|w\|}{\sigma} \|c - c'\|. \tag{11}$$

Interpretation.

- **More noise $\Rightarrow$ broader coverage.** For fixed (c, c'), increasing σ decreases Equation (10) and increases Equation (11), forcing overlap with higher-coverage contexts, thereby improving coverage at novel states.
- **Closer contexts $\Rightarrow$ structured coverage.** The bound scales with $\|c - c'\|$, so overlap is larger for nearby contexts. Coverage expansion is thus structured, borrowing actions from semantically related states.

Proof sketch. Smoothing averages $p(\cdot \mid c)$ over Gaussian perturbations, yielding a Lipschitz dependence on c via a Stein/score identity: $\big|p_\sigma(A \mid c) - p_\sigma(A \mid c')\big| \leq \frac{\mathbb{E}\|w\|}{\sigma} \|c - c'\|$ Taking the supremum over measurable A gives Equation (10); overlap follows directly.

Corollary 1.1 (Smoothing increases coverage beyond BC)**.** *Fix c, c' with $\Delta = c - c'$, and assume local identifiability:*

$$\text{TV}\big(p(\cdot \mid c), p(\cdot \mid c')\big) \;\geq\; m\|\Delta\|. \tag{12}$$

If

$$\sigma \;\geq\; \frac{\mathbb{E}\|w\|}{m}, \tag{13}$$

then

$$\text{Ov}\big(p_\sigma(\cdot \mid c), p_\sigma(\cdot \mid c')\big) \geq \text{Ov}\big(p(\cdot \mid c), p(\cdot \mid c')\big). \tag{14}$$

Interpretation. BC can induce narrow, disjoint action supports for identifiable contexts. Once σ exceeds Equation (13), smoothing dominates, increasing overlap and expanding action coverage relative to the base policy, mitigating zero-probability actions during RL.

Proof sketch. Combine Theorem 1 with Equation (12): $\text{TV}(p_\sigma) \leq \frac{\mathbb{E}\|w\|}{\sigma}\|\Delta\| \leq m\|\Delta\| \leq \text{TV}(p)$ which implies Equation (14).

Takeaway for RL Fine-tuning. Context smoothing provably increases overlap: if $\exists\, c'$ with better coverage, increasing σ raises the coverage parameter κ at c (Eq. (3)). Thus, σ acts as a *coverage dial*: large σ approaches the marginal $p(a)$ for broad coverage, while small σ preserves precise conditioning.

Summary. We train a context-smoothed policy $p_\theta(a \mid \tilde{c}, z, \sigma)$ via forward diffusion, enabling queries at any $\sigma \in [0, T_c]$. During RL fine-tuning, a high-level policy $\pi_{\text{HL}}(z, \sigma \mid s)$ modulates σ to interpolate between precise execution and broad coverage, borrowing coherent behaviors from nearby contexts for efficient adaptation.

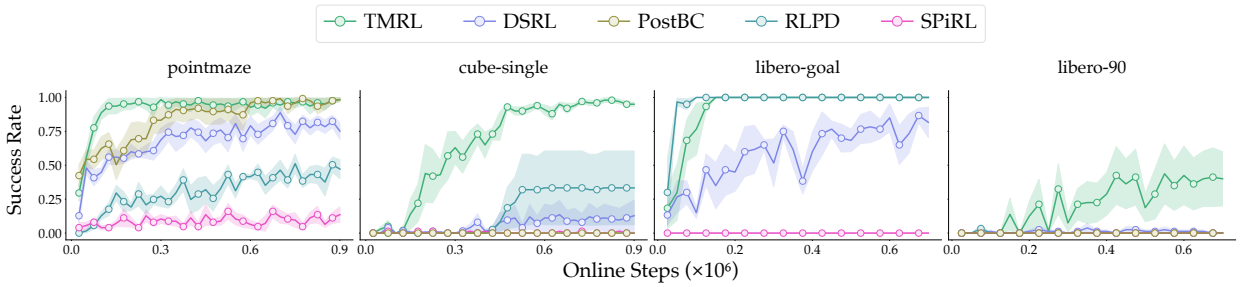

Fig. 5: **RL Success Rates for simulation tasks.** TMRL attains near 100% success rate in both OGBench tasks, outperforming the best baselines by 14% in `pointmaze-giant` and 200% in `cube-single` at final performance. In `libero-goal`, TMRL and RLPD [2] all reach 100% success. However, for the longer-horizon `libero-90` task, only TMRL explores sufficiently to achieve non-trivial success rates.

IV. Experiments

Our experiments study the following research questions:

(Q1) Does context-smoothed pre-training produce better action coverage over other pre-training approaches?

(Q2) Does TMRL effectively RL fine-tune policies trained across a variety of policy conditioning variables?

(Q3) Does TMRL enable real-world RL on VLA policies?

A. *(Q1) Comparing Pre-Training Action Coverage*

We first compare the effect of various pre-training procedures on action coverage by measuring success rates on unseen tasks. We compare context-smoothed pre-training (CSP) against standard BC, i.e., training $p(a \mid c)$ with Eq. (1) and PostBC [24]. Overall, **CSP outperforms both baselines in zero-shot success rate of unseen tasks**.

We demonstrate this result on two tasks from OGBench [18]: navigation (`pointmaze-giant`) and manipulation (`cube-single`). For our navigation task, we train policies on the `pointmaze-large-navigate` dataset and evaluate on the larger `pointmaze-giant` environment. The downstream environment has a larger state space, necessitating a broader action distribution. For our manipulation task, `cube-single`, we use the `cube-single-play` dataset and evaluate adaptation for initial positions beyond the dataset.

In Figure 6, we plot "success @ K," measuring the fraction of out-of-distribution initial states for which at least one of the K base policy rollouts succeeds. This success rate represents a direct empirical measurement of *demonstrator action coverage* from Eq. (3). On both OGBench tasks, CSP achieves higher success rates at every K with a fixed smoothing σ, with the gap most pronounced on `cube` where BC and PostBC have zero success at all K. Next, we show that CSP's broader action coverage translates to better RL performance with TMRL.

B. *(Q2) RL Fine-tuning Across Varied Policy Conditioning*

We compare TMRL against prior approaches for RL with prior data or combining pre-training and RL fine-tuning:

- RLPD [2]: an off-policy algorithm that learns online while incorporating offline data as an additional buffer. It relies on Gaussian action noise for expanding action coverage.
- SPiRL [19]: a hierarchical RL algorithm that trains an RL policy over pre-trained skills learned from offline data. It uses skill sampling for expanding coverage.

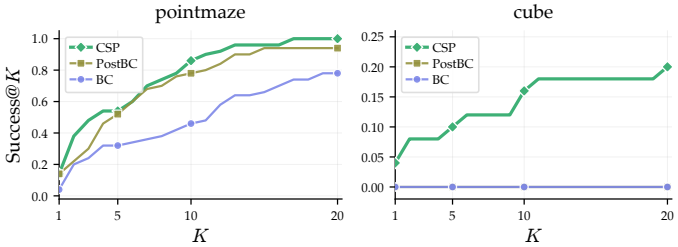

Fig. 6: **Context-smoothed pre-training enables better action coverage than BC and PostBC before any RL fine-tuning.** Success@K measures the fraction of out-of-distribution initial states where at least one of K base-policy rollouts succeeds. CSP is better at all K on both tasks.

- DSRL [25]: a steering algorithm that first trains a diffusion policy over action sequences with standard BC, then performs off-policy RL over the policy's noise space.
- PostBC [24]: pre-trains with additive Gaussian action noise to expand action coverage and then fine-tunes with DSRL.

Now, we examine how TMRL enables effective RL fine-tuning on difficult, unseen tasks. We consider downstream tasks that are OOD for the base policy, i.e., the training data contains no demonstrations for these downstream tasks.

State-Based Conditioning. As seen in the two left plots in Figure 5, we observe clear differences in how methods handle out-of-distribution generalization for the aforementioned OG bench domains from Section IV-A. RLPD achieves limited success, as it explores purely through additive Gaussian noise.

In contrast, methods that exploit action priors from the pre-training data are able to perform better; DSRL reaches around 90% success rates in `pointmaze-giant`; however, it achieves near-0 success rates on the higher-dimensional action space task `cube-single` as it lacks any mechanism to go beyond the exploration of the base policy. PostBC performs very similarly to DSRL—while it expands action coverage during pre-training, the coverage comes from single-step Gaussian noise, causing some dithering behavior, and, unlike TMRL, cannot be adaptively controlled during RL fine-tuning. TMRL consistently outperforms these baselines, achieving an overall 101% improvement over the best-performing baseline across both tasks by *learning* to systematically expand coverage of context-smoothed policies. Moreover, while several baselines exhibit high variance across seeds, TMRL is notably

more stable while still performing well.

Image-Based Tasks + VLM Embedding Conditioning: We next consider a set of image-based tasks with a large pre-trained VLA policy, π_0 [3], which uses VLM embeddings to condition a flow-based action expert. We evaluate on the LIBERO benchmark [14]. To enable TMRL we fine-tune a context-smoothed π_0 model on the `Libero-{Spatial, Object, Goal, 10}` datasets by adding noise to the VLM embeddings context c before they are input to the action expert head. We then evaluate adaptation on two unseen tasks: a position-perturbed version of a `Libero-Goal` task [33] and a task from the `Libero-90` task suite. Due to time constraints, we exclude PostBC from this comparison.

Figure 5 shows that TMRL leverages action chunks from different tasks in the dataset to solve the `libero-goal-swap` task more efficiently than DSRL. RLPD also learns the task because it is very short-horizon. In `libero-90`, we observe that only TMRL is able to consistently solve the task; TMRL leverages action sequences from other tasks, meanwhile DSRL overfits to picking up the same object each time, resulting in an overly narrow action distribution that is insufficient to solve the task.

Pointcloud Conditioning: Next, we look at a simulated dexterous manipulation grasping setting with a LEAP hand [21] on a Franka in IsaacLab [16]. We are interested in evaluating TMRL's RL fine-tuning ability on a *pointcloud*-input policy; in particular, we examine whether context-smoothing on pointclouds enables TMRL to adapt to grasping new objects. We pre-train context-smoothed policies on three can-shaped objects and evaluate on a new marker-shaped object. While grasp strategies do not immediately transfer, we find that noised pointclouds enable TMRL to *share grasping strategies* across different objects, enabling broader exploration and, consequently, faster learning and 2.5× higher final success rates than DSRL in Figure 7.

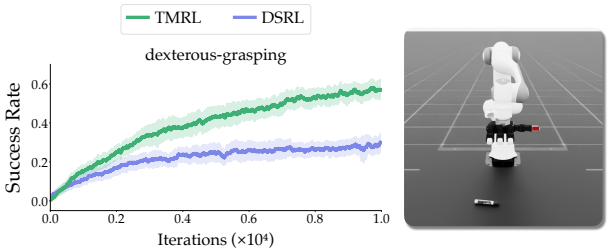

Fig. 7: TMRL enables efficient RL over 3D policies for dexterous grasping.

C. *(Q3): Does TMRL enable real-world RL on VLAs?*

Next, we demonstrate that TMRL enables effective adaptation of VLA policies in the real world. We evaluate on two platforms: a WidowX 250 6-DoF robot arm using the BridgeData-v2 [26] setup and dataset, and a Franka Panda 7-DoF robot arm using the DROID [11] dataset. For each, we pre-train a context-smoothed policy by fine-tuning π_0 [3] on the respective dataset using VLM embeddings as context c and the context noising objective in Equation (7).

We evaluate TMRL against DSRL on three tasks, `sausage-in-pot`, `shrimp-in-white-drawer`, and `press-button`, with evaluation curves shown in Figure 8. While the pre-trained policy is unable to solve the task successfully (often reaching for the wrong object), fine-tuning with TMRL improves success rates to near-perfect levels. This is in stark contrast to DSRL [25], which performs poorly because it cannot perform tasks beyond the coverage of the base policy. We omit a PostBC comparison here because the simulation performance in Figure 5 is similar to that of DSRL and it requires 2 additional steps: 1-step Gaussian ensemble policy pre-training and data labeling, then π_0 VLA re-training.

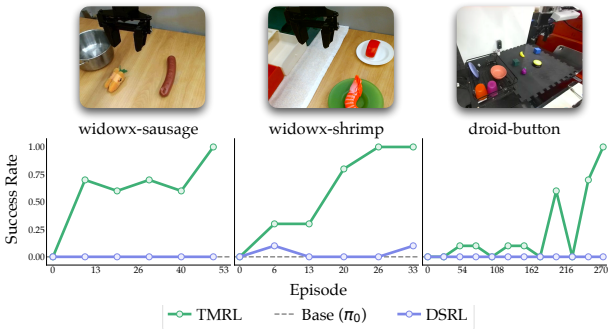

Fig. 8: TMRL enables steering of π_0 [3] across three real-world tasks, while DSRL [25] fails to learn any task.

D. *Analysis and Ablations.*

Finally, we conduct additional analysis experiments to better understand the behavior of TMRL.

Action Coverage Visualization: We then visualize the exploration behavior of TMRL as compared to DSRL on top of a standard BC policy in Fig 9. The exploration distribution of DSRL is relatively narrow, staying close to the coverage of the base policy. In contrast, the exploration behavior of TMRL is considerably broader, showing a diversity of strategies beyond the narrow base policy distribution. TMRL does not simply perform random actions, but rather aliases coherent actions from nearby states, for more informed exploration.

Fig. 9: Comparison of exploration behaviors on the task, "pick up the butter and put it in the basket" for RLPD (left), DSRL (center), and TMRL (right).TMRL leverages action sequences from smoothed contexts, resulting in broadened, yet coherent exploration.

References

[1] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, 1995.

[2] Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1577–1594. PMLR, 23–29 Jul 2023.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

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

[5] Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, 2021.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[8] Jiaheng Hu, Rose Hendrix, Ali Farhadi, Aniruddha Kembhavi, Roberto Martin-Martin, Peter Stone, Kuo-Hao Zeng, and Kiana Ehsani. Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning. *arXiv preprint: arXiv:2409.16578*, 2024.

[9] Kaizhe Hu, Zihang Rui, Yao He, Yuyao Liu, Pu Hua, and Huazhe Xu. Stem-OB: Generalizable visual imitation learning with stem-like convergent observation through diffusion inversion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xaYlO03tIk.

[10] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: A vla that learns from experience. *arXiv:2511.14759*, 2025.

[11] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[12] Kun Lei, Huanyu Li, Dongjie Yu, Zhenyu Wei, Lingxiao Guo, Zhennan Jiang, Ziyu Wang, Shiyu Liang, and Huazhe Xu. Rl-100: Performant robotic manipulation with real-world reinforcement learning. *arXiv preprint arXiv: 2510.14830*, 2026.

[13] Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, Wanli Peng, Jingchao Qiao, Zeyu Ren, Haixin Shi, Zhi Su, Jiawen Tian, Yuyang Xiao, Shenyu Zhang, Liwei Zheng, Hang Li, and Yonghui Wu. Gr-

rl: Going dexterous and precise for long-horizon robotic manipulation. *arXiv preprint: arXiv:2512.01801*, 2025.

[14] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[15] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024.

[16] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter, Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

[17] Chaoyi Pan, Giri Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, and Max Simchowitz. Much ado about noising: Dispelling the myths of generative robotic control, 2025. URL https://arxiv.org/abs/2512.01809.

[18] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

[19] Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning (CoRL)*, 2020.

[20] Dean Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D.S. Touretzky, editor, *Proceedings of (NeurIPS) Neural Information Processing Systems*, pages 305 – 313. Morgan Kaufmann, December 1989.

[21] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023.

[22] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[23] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

[24] Andrew Wagenmaker, Perry Dong, Raymond Tsao, Chelsea Finn, and Sergey Levine. Posterior behavioral cloning: Pretraining bc policies for efficient rl finetuning. *arXiv preprint arXiv:2512.16911*, 2025.

[25] Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Conference on Robot Learning*, 2025.

[26] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1723–1736. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

[27] Jingyun Yang, Max Sobol Mark, Brandon Vu, Archit Sharma, Jeannette Bohg, and Chelsea Finn. Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[28] Patrick Yin, Tyler Westenbroek, Simran Bagaria, Kevin Huang, Ching-An Cheng, Andrey Kolobov, and Abhishek Gupta. Rapidly adapting policies to the real-world via simulation-guided fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2025.

[29] Fan Zhang and Michael Gienger. Affordance-based robot manipulation with flow matching, 2025. URL https://arxiv.org/abs/2409.01083.

[30] Jesse Zhang, Jiahui Zhang, Karl Pertsch, Ziyi Liu, Xiang Ren, Minsuk Chang, Shao-Hua Sun, and Joseph J Lim. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. In *Conference on Robot Learning (CoRL)*, 2023.

[31] Jesse Zhang, Minho Heo, Zuxin Liu, Erdem Biyik, Joseph J Lim, Yao Liu, and Rasool Fakoor. EXTRACT: Efficient policy learning by extracting transferrable robot skills from offline data. In *Conference on Robot Learning*, 2024.

[32] Jesse Zhang, Karl Pertsch, Jiahui Zhang, and Joseph J. Lim. Sprint: Scalable policy pre-training via language instruction relabeling. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[33] Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *[arXiv preprint arXiv:2510.03827]*, 2025.