Lightweight Adaptation of Pretrained Robot Manipulation Systems: Two Approaches

Adam Lalani

Brown University, Department of Computer Science

Advisor: Professor Chen Sun

Reader: Professor Hui Wang

April 2026

Abstract

Modern robot manipulation increasingly relies on large pretrained models: video diffusion planners that generate action-guiding trajectories, and vision-language-action models that map observations to motor commands. These models are capable but expensive to retrain, motivating the question: how much can we improve them with minimal or zero modification to their weights?

This thesis documents two systematic attempts to answer that question. The first applies bidirectional sampling to AVDC, a video diffusion planner, augmenting inference with CLIP-based goal image conditioning on MetaWorld [19] tasks. The approach improved performance in the no-replan setting but produced mixed results overall, limited by AVDC's lack of classifier-free guidance. A subsequent pivot to VPP on CALVIN found that fine-tuning on bidirectional data degraded VPP's strong baseline due to distribution shift. The second attempt trains small sets of learnable parameters (soft prompt vectors, 65K; and attention prefix pairs, 4M) via reinforcement learning on a frozen OpenVLA-7B, using binary task-success rewards on LIBERO-Goal. Neither method improved over the vanilla baseline; high initial-state variance dominated the reward signal and made short RL runs inherently noisy.

Both attempts reveal a common ceiling: interventions that leave the backbone frozen are bounded by what that backbone already knows.

Contents

1 Introduction

1.1 Motivation

Large pretrained models have transformed robot manipulation. Video diffusion models learn to generate plausible action-guiding video trajectories from demonstrations, enabling zero-shot generalization to new tasks. Vision-language-action models (VLAs) map raw visual observations and natural language instructions directly to motor commands, achieving strong performance across diverse manipulation benchmarks without task-specific engineering.

The cost of these capabilities is scale. A modern VLA like OpenVLA [4] contains 7 billion parameters; retraining it from scratch requires thousands of GPU-hours and large curated datasets. Fine-tuning even a fraction of these parameters demands careful regularization to avoid catastrophic forgetting. For researchers and practitioners who want to adapt a pretrained model to a new task or improve its behavior in a specific scenario, full retraining is often simply not feasible.

This creates a natural question: can we meaningfully improve pretrained robot manipulation models at inference time, or with a minimal number of learned parameters, without touching the backbone weights? Two complementary points on the "adaptation budget" axis define the scope of this thesis:

- **Zero-parameter adaptation**: modify only the inference procedure, leaving all model weights unchanged.
- **Minimal-parameter adaptation**: learn a small set of parameters via reinforcement learning to steer a frozen 7B model's behavior. Specifically, soft prompt tuning (input embedding vectors, 65K parameters) and attention prefix tuning (K,V injection into all transformer layers, 4M parameters).

Both are attractive for the same reason: they preserve the pretrained model's knowledge while attempting to redirect its outputs. Both, as we will show, are harder than they appear.

1.2 Overview of Attempts

Attempt 1: Bidirectional Sampling on AVDC addresses the open-loop planning gap in AVDC [5], a video diffusion-based manipulation planner. AVDC generates action trajectories conditioned only on a start frame; without replanning, its performance degrades substantially relative to a full closed-loop version (e.g., 28% vs. 56% on button-press). We hypothesized that anchoring the backward end of the diffusion trajectory to a goal image (via bidirectional DDIM sampling and CLIP goal conditioning) could reduce this gap without any model modification. The approach improved button-press in the no-replan setting but hurt door-open across all variants. We hypothesize that the naive bidirectional nudge degrades performance on tasks requiring non-linear trajectories, where a goal-anchored backward pass conflicts with the curved path the robot must actually take. A deeper incompatibility also applies: AVDC's pixel-level conditioning is not designed for temporally reversed sequences, and CFG++ [20] (which would allow proper goal weighting) is unavailable in AVDC. A subsequent pivot to VPP [24] on CALVIN [12] attempted to address the CFG++ limitation via a more compatible backbone, but fine-tuning SVD with ViBiD on CALVIN data caused performance regression, likely due to distribution shift between the fine-tuned policy and VPP's expected input representation.

Attempt 2: RL Prompting for OpenVLA explores two parameter-efficient interventions on a frozen OpenVLA-7B: soft prompt tuning (input embedding vectors, 65K parameters) and attention prefix tuning (K,V injection into all transformer layers, 4M parameters), both optimized

via GRPO [14] with binary task-success rewards on LIBERO-Goal [11]. Short RL runs with both approaches proved noisy and did not improve over the vanilla baseline. A key methodological finding is that initial-state variance (0–80% SR range on the training task) dominates the reward signal, which explains the difficulty and has implications for future RL experiments on LIBERO.

1.3 Contributions

- **Bidirectional Sampling NR sim+CLIP improves over AVDC NR on 3 of 7 tasks**: In the no-replan setting, goal conditioning yields gains on button-press (+10.7pp), faucet-open (+12.0pp), and faucet-close (+6.7pp), with regressions on the remaining four. Gains concentrate on tasks with clear spatial endpoints; regressions occur on tasks requiring precise approach trajectories.
- **Bidirectional Sampling Full matches or exceeds AVDC Full on 5 of 7 tasks**: In the closed-loop setting, Bidirectional Sampling (sim+CLIP) outperforms AVDC Full on door-open (+1.3pp), door-close (+2.7pp), faucet-close (+4.0pp), handle-press (+6.6pp), and assembly (+2.7pp), with regressions only on button-press (−1.3pp) and faucet-open (−2.7pp). The gains consistently exceed the regressions in magnitude, establishing that goal conditioning with CLIP provides reliable benefit in the closed-loop setting across diverse manipulation tasks.
- **Backward pass and CLIP are jointly necessary: neither works independently**: backward-pass-only (λ sweep) and CLIP-forward-only both fail independently; neither gives gain on button-press alone. Only backward+CLIP together yields +10.7pp, establishing that the two are synergistic, not redundant.
- **ViBiD fine-tuning on VPP regresses by 17%**: Fine-tuning SVD with ViBiD on CALVIN data degrades `avg_seq_len` from 4.284 to 3.558. The regression is attributed to latent distribution shift: VPP's policy head expects low-t latents from a standard forward SVD pass; ViBiD's composite denoising trajectory produces latents with different statistics, an incompatibility that fine-tuning the SVD backbone alone cannot resolve.
- **Short GRPO runs fail to improve frozen OpenVLA; initial-state variance dominates the reward signal**: Both soft prompt tuning (65K params) and attention prefix tuning (4M params), trained for ∼50 GRPO steps, produced noisy training and did not improve over the vanilla baseline. Per-state SR on task 3 spans 0–80% (50 states × 10 trials), establishing that reward contamination rather than prompt quality drives most of the observed GRPO gradient, and that runs of this length are insufficient for zero-initialized modules to converge against a high-variance reward signal.

2 Background

2.1 Video Diffusion for Robot Planning

Diffusion models learn to reverse a noise process, progressively denoising samples from a learned data distribution. When applied to video, they can model the distribution of plausible future frames conditioned on observed context. AVDC (Actionless Video Diffusion for Control) [5] exploits this to build a manipulation planner: a video diffusion model generates a sequence of future frames starting from the current observation, then optical flow between consecutive frames is converted to robot actions via inverse kinematics. The model operates entirely without explicit action labels in training; it learns to generate plausible manipulation videos from demonstration footage alone.

AVDC uses DDIM sampling [16], a deterministic denoising schedule parameterized by $\eta = 0$. DDIM enables a key property for our approach: the denoising trajectory is a fixed function of the initial noise, making it amenable to structured modification. AVDC conditions video generation on a single start frame, iteratively replanning from updated observations in closed-loop mode.

The gap between open-loop and closed-loop AVDC is substantial. On button-press, AVDC No-Replan achieves 28.0% and AVDC Full (with replanning) achieves 56.0%, a 28pp gap. On door-open, the gap is 42.7% vs. 66.7%. This open-loop degradation motivates exploring whether goal conditioning can reduce trajectory drift in the no-replan setting, without a replanning module.

VPP (Video Prediction Policy) [24] is a more recent video diffusion policy built on Stable Video Diffusion (SVD) [1], which supports classifier-free guidance (CFG) [3] natively. VPP is evaluated on CALVIN [12] ABC→D, a long-horizon manipulation benchmark where the metric is average sequence length (`avg_seq_len`) over 1000 test sequences, each consisting of up to 5 chained tasks. A strong VPP baseline achieves `avg_seq_len` 4.284, meaning the policy completes on average 4.284 consecutive tasks before failing.

2.2 Vision-Language-Action Models

OpenVLA [4] is a 7B-parameter vision-language-action model built on LLaMA-2 [23] with a visual encoder. It tokenizes robot actions into 256-bin discretized values across 7 degrees of freedom, then predicts action tokens autoregressively given an image observation and a language instruction. The autoregressive structure is crucial for Attempt 2: it provides exact log-probabilities for any action sequence, which GRPO requires to compute policy gradients.

LIBERO [11] is a benchmark for lifelong robot learning consisting of multiple task suites. We use LIBERO-Goal, a 10-task suite of tabletop manipulation tasks (picking, placing, opening drawers, etc.) with diverse initial object configurations. The suite exposes 50 distinct initial states per task, drawn from a fixed distribution.

2.3 Bidirectional Diffusion Sampling

ViBiDSampler [15] proposes a general framework for bidirectional video diffusion: instead of denoising only from a start condition, the algorithm anchors the trajectory at both ends (start and goal) and blends the resulting denoising paths. The full algorithm requires CFG++ [20] for score mixing and DDS [21] (Dual Diffusion Solver) for trajectory alignment, neither of which is available in AVDC. We implement a simplified version: a forward pass conditioned on the start frame, a backward pass on a temporally reversed sequence conditioned on the goal frame, and a weighted blend of the two denoising trajectories.

The blend parameter $\lambda \in [0, 1]$ controls the relative weight of the forward and backward passes. At $\lambda = 1$, the algorithm reduces to standard AVDC. As λ decreases, the backward (goal-conditioned) pass exerts more influence. A renoise scale ρ adds a stochastic bridge between passes. We additionally inject a CLIP [13] visual embedding of the goal image into the backward pass as a semantic conditioning signal.

2.4 Reinforcement Learning for Frozen Models

DSRL [17] demonstrates that RL can steer a frozen diffusion policy by optimizing over its input space (initial noise vectors), without modifying any weights. Crucially, DSRL requires only black-box

access to the policy; it does not backpropagate through the model, instead using RL to search over the noise distribution that seeds the denoising process.

GRPO (Group-Relative Policy Optimization) [14] is a variant of policy gradient that eliminates the value network by normalizing advantages within a group of rollouts from the same state. For binary rewards (success/failure), GRPO reduces to a reweighted log-likelihood loss: successful trajectories are reinforced, failed ones are suppressed, with weights determined by group-relative advantage. GRPO has been applied to mathematical reasoning and, more recently, to VLA fine-tuning [8, 10].

RLPrompt [2] provides an earlier precedent for RL-trained prompts in frozen language models, demonstrating that discrete or continuous prompt optimization can steer model behavior without fine-tuning. Our approach adapts this idea to a continuous embedding space in a VLA setting.

Soft prompt tuning [6] prepends a small number of learnable embedding vectors to the input token sequence of a frozen model. Rather than modifying any weights, these vectors are optimized to shift the model's activations toward a desired behavior. We use 16 vectors in the input embedding space of OpenVLA (65,536 total parameters).

Attention prefix tuning [7] extends this idea into the attention layers: instead of prepending tokens to the input, learnable key-value pairs are injected directly into the attention computation of every transformer layer. This provides a richer intervention surface (4M parameters across 32 layers) and a shorter path between the learned parameters and the model's internal representations, potentially making optimization easier.

3 Attempt 1: Bidirectional Video Diffusion

3.1 Problem Setup

AVDC's open-loop degradation is significant and consistent across three MetaWorld [19] tasks: button-press drops from 56.0% (Full) to 28.0% (No-Replan), door-open from 66.7% to 42.7%, and assembly from 4.0% to 2.7%. The replanning module in AVDC Full re-conditions the video diffusion model on updated observations at each step, correcting for early errors. Without this, the model must commit to a single generated trajectory from the start frame alone.

Our hypothesis is that this open-loop degradation is partly attributable to trajectory drift: without an endpoint constraint, the diffusion model has no information about where the trajectory should end. A goal image (showing the desired final state of the scene) could provide this endpoint constraint and reduce drift, without requiring a replanning module.

We implement this via bidirectional DDIM: a forward pass (start-conditioned) is blended with a backward pass (goal-conditioned, temporally flipped), with a CLIP visual embedding of the goal image injected as an additional conditioning signal in the backward pass. No model weights are modified.

3.2 Method

3.2.1 Bidirectional DDIM Sampling

At each denoising step t, the standard AVDC forward pass computes a predicted clean frame sequence $\hat{x}_0^{fwd}$ conditioned on the start observation x_0^{start}. Our backward pass computes $\hat{x}_0^{bwd}$ on a temporally reversed sequence conditioned on a goal frame x_0^{goal}. The blend is:

$$\hat{x}_0 = \lambda \cdot \hat{x}_0^{fwd} + (1 - \lambda) \cdot \hat{x}_0^{bwd} \tag{1}$$

followed by re-noising with scale $\rho \in [0, 1]$ to introduce a stochastic bridge:

$$x'_t = \sqrt{\bar{\alpha}_t}\, \hat{x}_0 + \rho\sqrt{1 - \bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{2}$$

The parameter $\lambda \in [0, 1]$ controls the balance between forward and backward passes; $\lambda = 1$ recovers standard AVDC. The DDIM update rule is then applied to the blended prediction:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, \hat{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1}}\, \epsilon_\theta(x'_t, t) \tag{3}$$

where $\bar{\alpha}_t$ are the DDIM noise schedule coefficients and ϵ_θ is the AVDC denoising network.

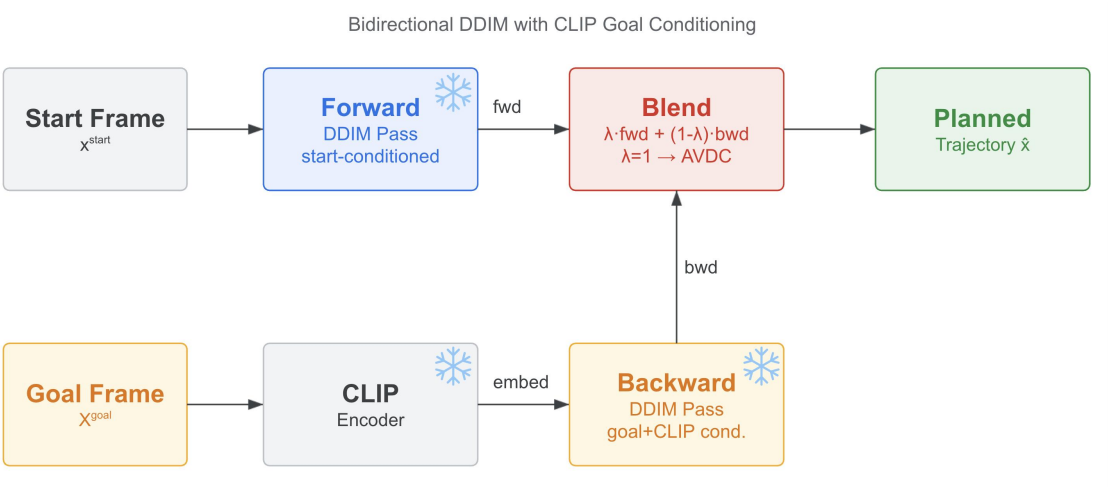

Figure 1: Architecture of bidirectional DDIM sampling. The forward pass conditions on the start frame; the backward pass conditions on the goal frame with CLIP; outputs are blended with weight λ.

3.2.2 CLIP Goal Image Conditioning

The backward pass alone provides only pixel-level information about the goal frame, which may conflict with the model's learned pixel statistics for reversed sequences. To add semantic information, we extract the CLS token of a CLIP ViT-B/32 [13] vision encoder (a 512-dimensional vector) from the goal image and concatenate it with the text embedding in the backward pass conditioning:

$$c_{bwd} = [c_{\text{text}};\ c_{\text{CLIP}}], \quad c_{\text{CLIP}} = \text{CLIP}_{\text{CLS}}(x_0^{goal}) \in \mathbb{R}^{512} \tag{4}$$

This requires no architectural modification to AVDC; CLIP runs as a preprocessing step.

3.2.3 Goal Frame Generation

Two sources of goal frames are evaluated:

- **Sim**: a scripted policy (separate from AVDC) executes the task in simulation, and the final observation frame is extracted. This requires simulator access but produces ground-truth goal images.
- **GPT-4o (gpt-image-1)** [18]: given the current observation and task description text, GPT-4o generates a plausible goal image. This removes the scripted policy dependency and represents a more deployable setting.

3.3 Experimental Setup

Experiments run on three MetaWorld [19] tasks: button-press-v2, door-open-v2, and assembly-v2. We use the AVDC milestone-24 checkpoint with no fine-tuning. Each condition is evaluated over 75 trials (25 seeds $\times$ 3 camera poses), and the per-camera success rates are averaged. The AVDC No-Replan and AVDC Full conditions serve as lower and upper baselines respectively.

Hyperparameter selection for the blend weight λ and renoise scale ρ used a preliminary sweep over 35 conditions on button-press with 25 trials each ($\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\} \times \rho \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$ plus additional CLIP variants), selecting the configuration that maximized NR success rate for each goal source.

3.4 Results

3.4.1 Main Results

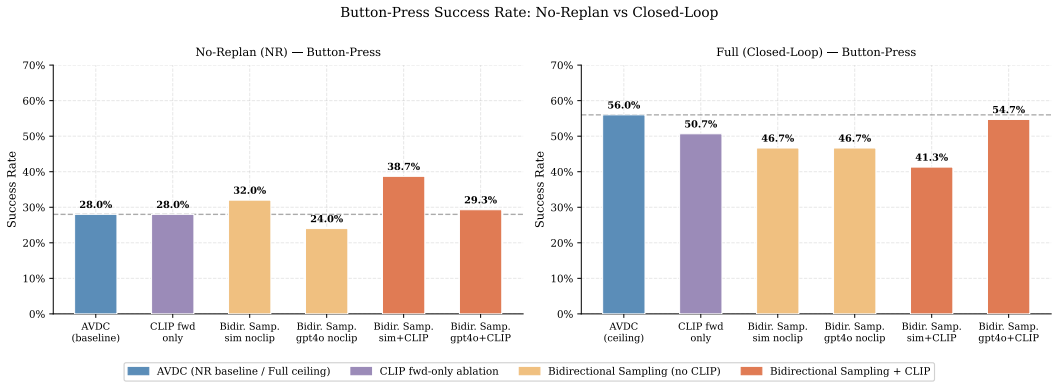

Figure 2: Button-press success rates across all conditions. Bidirectional Sampling NR sim+CLIP achieves +10.7pp over AVDC NR; in the Full setting, the method reaches 54.7%, within 1.3pp of the AVDC Full ceiling.

As shown in Tables 1 and 2, on **button-press**, Bidirectional Sampling NR sim+CLIP achieves 38.7% vs. AVDC NR's 28.0% (+10.7pp); Bidirectional Sampling Full gpt4o+CLIP (opt (λ, ρ)) reaches 54.7%, within 1.3pp of the AVDC Full ceiling (56.0%). On **door-open**, all NR variants hurt relative to AVDC NR (42.7%), with the best reaching 37.3%; Bidirectional Sampling Full sim+CLIP (opt (λ, ρ)) reaches 68.0%, surpassing AVDC Full (66.7%). On **assembly**, all variants remain near zero across both settings.

Table 1: No-replan (NR) success rates (75 trials, averaged over 3 camera poses). Rows are grouped by intervention type: baseline, ablation, backward-only (no CLIP), and backward+CLIP. Bold marks the best result per task across NR variants.

Group	Variant	Button-Press	Door-Open	Assembly
Baseline	AVDC NR	28.0%	**42.7%**	**2.7%**
Ablation	CLIP forward only sim	28.0%	34.7%	0.0%
No CLIP	Bidirectional Sampling NR sim noclip	32.0%	34.7%	0.0%
	Bidirectional Sampling NR gpt4o noclip	24.0%	37.3%	0.0%
+CLIP	Bidirectional Sampling NR gpt4o+CLIP	29.3%	37.3%	0.0%
	Bidirectional Sampling NR sim+CLIP	**38.7%**	30.7%	0.0%

Table 2: Full (closed-loop) success rates (75 trials, averaged over 3 camera poses). *Noclip* rows use $\lambda = 0.5$, no CLIP conditioning. *Initial* rows use $\lambda = 0.5$ with CLIP; *opt* rows use the best (λ, ρ) from the hyperparameter sweep. Bold marks the best result per task.

Condition	Goal	CLIP	Button-Press	Door-Open	Assembly
AVDC Full (ceiling)			**56.0%**	66.7%	4.0%
Noclip	sim	no	46.7%	41.3%	1.3%
	gpt4o	no	46.7%	44.0%	6.7%
Initial	sim	+CLIP	46.7%	41.3%	2.7%
	gpt4o	+CLIP	48.0%	37.3%	6.7%
Opt (λ, ρ)	sim	+CLIP	41.3%	**68.0%**	5.3%
	gpt4o	+CLIP	54.7%	58.7%	**6.7%**

3.4.2 Blend Weight Analysis

To understand what drives the button-press gain, we swept λ on button-press in the no-replan, sim-goal, no-CLIP setting (25 trials each):

Table 3: Button-press NR success rate vs. blend weight λ (sim goal, no CLIP, $\rho = 1.0$, 25 trials each).

λ	0.1	0.3	0.5	0.7	0.9	1.0 (AVDC NR)
NR sim noclip SR	2.7%	8.0%	13.3%	25.3%	26.7%	28.0%

The finding is unambiguous: without CLIP conditioning, Bidirectional Sampling NR is strictly worse than AVDC NR at every blend weight. The backward pass, operating on a pixel-reversed sequence, actively degrades performance relative to the forward-only baseline. Performance increases monotonically with λ, with the best result simply being standard AVDC ($\lambda = 1.0$). The +10.7pp gain from Bidirectional Sampling NR sim+CLIP is therefore not attributable to bidirectionality; it is attributable to CLIP goal conditioning delivered through the backward pass as a vehicle.

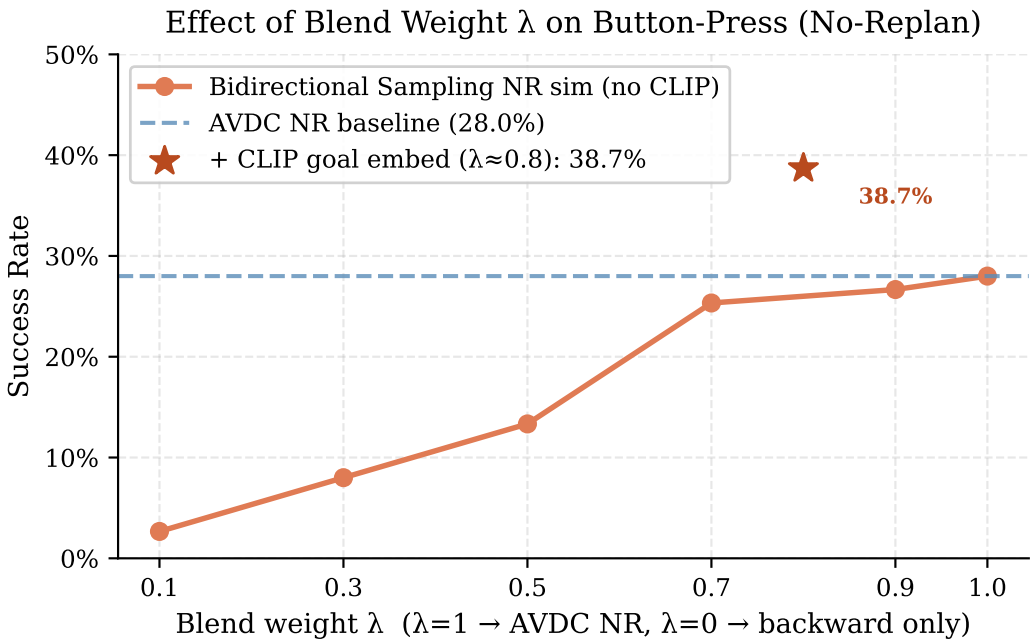

Figure 3: λ sweep on button-press (NR, sim goal, no CLIP, $\rho = 1.0$). Performance increases monotonically with λ; the backward pass alone is strictly harmful at every blend weight. The star marks Bidirectional Sampling NR sim+CLIP at its optimal λ.

3.4.3 Extended Task Results

We additionally evaluated four extended MetaWorld tasks (door-close, faucet-close, faucet-open, handle-press) to assess breadth.

Table 4: Extended task results (AVDC NR/Full vs. Bidirectional Sampling NR/Full). All Bidirectional Sampling variants use simulator-extracted goal frames (sim) with CLIP conditioning.

Variant	Goal	Door-Close	Faucet-Close	Faucet-Open	Handle-Press
No-Replan (NR)					
AVDC	—	34.7%	36.0%	10.7%	54.7%
Bidirectional Sampling	sim	32.0%	42.7%	22.7%	42.7%
Full (Closed-Loop)					
AVDC	—	92.0%	54.7%	**34.7%**	78.7%
Bidirectional Sampling	sim	**94.7%**	**58.7%**	32.0%	**85.3%**

Bidirectional Sampling NR shows gains on faucet-close (+6.7pp) and faucet-open (+12.0pp), regression on handle-press (−12.0pp), and near-parity on door-close (−2.7pp). No consistent direction emerges. In the Full setting, Bidirectional Sampling outperforms AVDC on three of four tasks: door-close (+2.7pp), faucet-close (+4.0pp), and handle-press (+6.6pp), with a small regression on faucet-open (−2.7pp).

3.5 The VPP Pivot

3.5.1 Motivation

The λ sweep and per-task analysis point to a fundamental limitation of applying Bidirectional Sampling to AVDC. AVDC conditions video generation on raw pixel observations and lacks classifier-free guidance (CFG) [3]. The full ViBiDSampler algorithm requires CFG++ [20] for proper score mixing between forward and backward passes, and DDS [21] for trajectory alignment. Without CFG, we can only implement a simplified variant: a forward DDIM pass conditioned on the start frame, a separate backward DDIM pass on a temporally reversed sequence conditioned on the goal frame, and a weighted blend of the two resulting trajectories. There is no proper score mixing or trajectory alignment; just a linear interpolation between two independently denoised outputs. The backward pass operates on reversed pixel sequences that are out-of-distribution for a model trained only on forward-time observations.

VPP addresses both issues: it is built on SVD, which supports CFG natively, and it is a stronger baseline on CALVIN ABC→D, a long-horizon benchmark better suited to evaluating trajectory-level improvements. We pivoted to VPP in the second half of Attempt 1.

3.5.2 Results

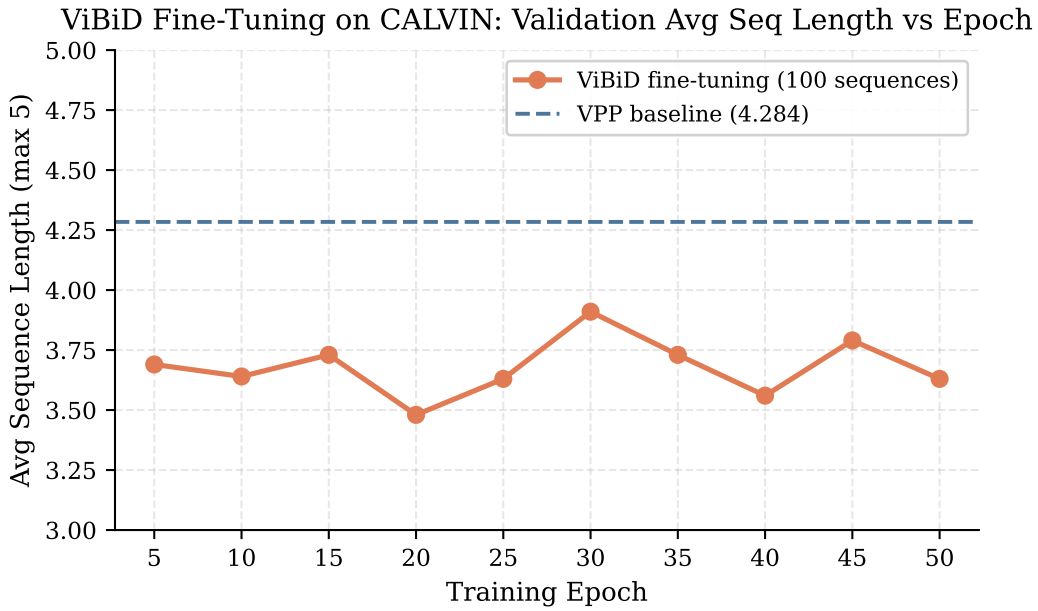

Figure 4: ViBiD fine-tuning validation `avg_seq_len` vs. epoch. Peak at epoch 30 (3.910) remains below the pretrained VPP baseline (4.284).

We fine-tuned VPP/SVD on bidirectional CALVIN data, conditioning the backward pass on final-state frames extracted from demonstrations. The VPP baseline achieves `avg_seq_len` = **4.284** over 1000 evaluation sequences (85.7% of tasks completed across all sequences, out of a maximum chain length of 5). The best ViBiD fine-tuning checkpoint achieves `avg_seq_len` = **3.558**, a regression of −0.726 (−17%) relative to the baseline. The validation curve peaks at epoch 30 (`avg_seq_len` = 3.910 on 100 sequences), still below the baseline. Fine-tuning SVD with ViBiD on CALVIN data does not improve over the pretrained VPP.

3.6 Discussion

3.6.1 What CLIP Conditioning Is Really Doing

Three ablations together isolate what drives the gains.

In the **NR setting**, the λ sweep (Section 3.4.2) shows the backward pass alone is never beneficial: bidirectionality without CLIP strictly hurts at every blend weight. The converse ablation (CLIP injected into the forward pass only, no backward pass; Table 1, "CLIP forward only") achieves 28.0% on button-press, identical to the AVDC NR baseline. Neither component alone works. The +10.7pp gain requires their combination: CLIP provides the goal signal, and the backward pass provides the structural slot to deliver it.

In the **Full setting**, the role of CLIP depends on the task. On button-press, replanning alone (noclip Full, Table 2) already closes most of the gap to AVDC Full, and CLIP provides incremental benefit on top. On door-open, replanning alone leaves performance flat (41.3–44.0%), while CLIP is decisive: Bidirectional Sampling Full sim+CLIP (opt λ) reaches 68.0%, a +26.7pp gain over the sim noclip Full baseline (41.3%). CLIP is not redundant when replanning is available; its importance is task-dependent.

3.6.2 Why NR Helps on Button-Press but Hurts on Door-Open

Button-press has an unambiguous spatial goal: a button that needs to be pressed downward. The CLIP embedding of a "button pressed" goal image encodes this goal unambiguously, and the AVDC model's pixel-level forward pass only needs a small nudge to reach it from a nearby initial configuration.

Door-open is structurally different. The task requires grasping a door handle and pulling, a directional motion that depends on the approach trajectory, not just the final state. A goal image showing an open door provides an endpoint constraint, but the backward pass conditioned on this image may produce a conflicting spatial signal: the reversed sequence suggests the robot should move away from the handle rather than toward it. Additionally, per-camera SR on door-open with Bidirectional Sampling NR sim+CLIP varies dramatically (52%, 8%, 32% across the three cameras), suggesting that camera–goal image alignment interacts badly with the backward pass dynamics.

3.6.3 Why VPP Did Not Help

The VPP fine-tuning regression (`avg_seq_len` 3.558 vs. 4.284 baseline) suggests that training VPP on bidirectional CALVIN data interferes with the representation that makes VPP strong on CALVIN. VPP's baseline is already very strong: 85.7% of tasks completed across all test sequences (`avg_seq_len` 4.284 out of 5 maximum), and the fine-tuning objective, which requires the model to simultaneously predict forward and backward trajectories, may disrupt the learned task-chaining dynamics.

There is also a deeper architectural incompatibility. Unlike AVDC, which generates a complete video rollout, VPP does not run SVD to full denoising. Instead, VPP samples the SVD backbone at a single low noise timestep t and passes that partially-denoised latent directly to its action-prediction head. The policy head was trained to receive latents from exactly this distribution: low-t samples from a standard forward SVD pass. ViBiD requires a complete forward-and-backward denoising trajectory; the resulting composite latent has different statistical properties from what the policy head expects, introducing distribution shift at the head regardless of whether the video content looks

plausible. Fine-tuning SVD with ViBiD on CALVIN data does not resolve this: the fine-tuning objective trains the SVD backbone on bidirectional sequences, but does not retrain the policy head on the resulting latent distribution, leaving the mismatch intact.

4 Attempt 2: RL-Trained Soft Prompts for Frozen OpenVLA

4.1 Problem Setup

OpenVLA-7B fine-tuned on LIBERO-Goal achieves 79.0% overall success rate on LIBERO-Goal (500 trials, 10 tasks). The per-task distribution is wide: task 7 (turn on the stove) achieves 100%, while task 0 (open the middle drawer) achieves only 38% and task 3 (open the top drawer and put the bowl inside) achieves 54% in a standard 50-trial evaluation (45.6% in a rigorous 500-trial diagnostic described in Section 4.3.2). Our goal is to improve performance on specific tasks via a trained prompt that steers the frozen model's behavior, without modifying any backbone weights.

The approach is motivated by DSRL [17], which shows that RL over the input space of a frozen diffusion policy can improve task success. We adapt this idea to the VLA setting: instead of optimizing over noise vectors, we optimize a continuous embedding injected into the language token stream.

4.2 Method

4.2.1 SoftPromptModule

We define a `SoftPromptModule` consisting of $K = 16$ learnable embedding vectors, each of dimension 4096 (matching OpenVLA's hidden dimension). The total parameter count is $16 \times 4096 = 65{,}536$, approximately 0.001% of the model's 7B parameters. All other model parameters are frozen.

The prompt embeddings are injected by adding them element-wise to the first K language token embeddings in each forward pass (add-to-first-K injection):

$$e'_k = e_k + \phi_k, \quad k = 1, \ldots, K \tag{5}$$

where e_k are the original token embeddings and $\phi_k \in \mathbb{R}^{4096}$ are the learnable prompt parameters. This design choice is critical: an alternative approach of prepending the embeddings as additional tokens would shift all subsequent token positions, disrupting the rotary positional embeddings (RoPE) [22] that OpenVLA uses for sequence ordering. Add-to-first-K avoids this by leaving positional encodings unchanged, preserving vanilla performance (54% standard 50-trial eval) at initialization with zero-initialized embeddings.

4.2.2 GRPO Training

We use GRPO [14] to optimize the prompt embeddings via binary task-success rewards, following the VLA fine-tuning methodology of [10]. At each training step, we collect $n_{\text{rollouts}} = 8$ episodes from the current policy (model + prompt) on task 3, compute the binary reward (1 if task succeeded, 0 otherwise), normalize advantages within the group, and update the prompt embeddings to reinforce successful rollout log-probabilities and suppress failed ones.

The group-normalized advantage for rollout i within a batch of N rollouts is:

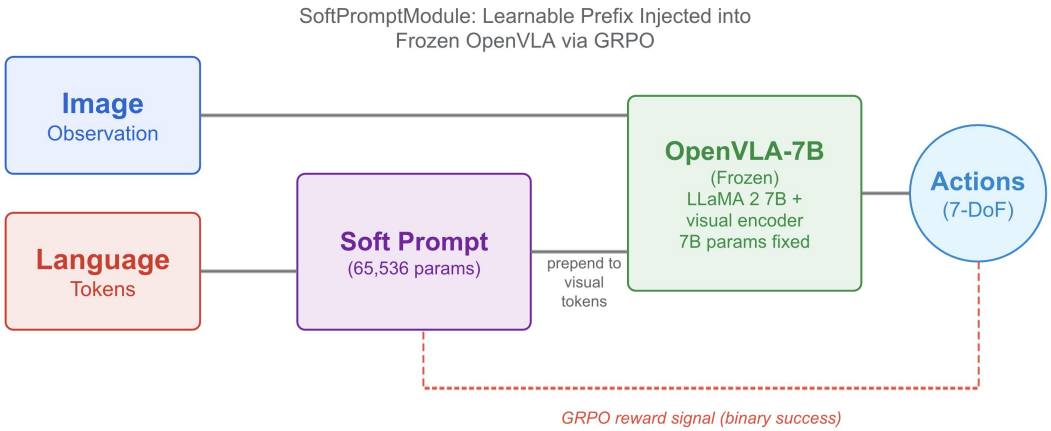

Figure 5: `SoftPromptModule` injection into OpenVLA. Sixteen learnable embedding vectors are added element-wise to the first 16 language token embeddings, leaving all model weights and positional encodings unchanged.

$$\hat{A}_i = \frac{r_i - \mu_G}{\sigma_G}, \quad \mu_G = \frac{1}{N}\sum_{j=1}^{N} r_j, \quad \sigma_G = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(r_j - \mu_G)^2} \tag{6}$$

where $r_i \in \{0, 1\}$ is the binary task-success reward. The training objective is:

$$\mathcal{L}(\phi) = -\sum_{i=1}^{N} \hat{A}_i \cdot \log p_\theta(a_i \mid o_i, e_\phi) \tag{7}$$

where e_ϕ are the trainable prompt embeddings and p_θ is the frozen OpenVLA.

4.2.3 Attention Prefix Module

As a second injection method, we implement attention prefix tuning [7] adapted for OpenVLA's LLaMA backbone. Rather than modifying the input embedding sequence, we prepend trainable key-value pairs directly into the attention computation of every transformer layer. Concretely, for each of the 32 LLaMA decoder layers, we maintain a learned prefix of shape ($n_{kv} = 32$, $n_{prefix} = 16$, $d_h = 128$), yielding 4,194,304 total trainable parameters, 64× more than the soft prompt.

The modified attention computation at each layer ℓ is:

$$\text{Attn}^\ell(Q, K, V) = \text{softmax}\left(\frac{Q\,[K_\phi^\ell;\, K]^\top}{\sqrt{d_h}}\right)[V_\phi^\ell;\, V] \tag{8}$$

where $K_\phi^\ell, V_\phi^\ell \in \mathbb{R}^{n_{kv} \times n_{prefix} \times d_h}$ are the learned prefix pairs for layer ℓ, $[\,;\,]$ denotes concatenation along the sequence dimension, and $d_h = 128$ is the per-head dimension.

Injection is implemented by patching `F.scaled_dot_product_attention` globally via a thread-local layer index set by pre-hooks on each decoder layer. This avoids replacing `self_attn.forward` directly (which is incompatible with OpenVLA's `accelerate` device mapping) and preserves the

model's native SDPA implementation. The prefix K vectors are initialized from a single forward-pass activation capture rather than zero, ensuring prefix tokens compete fairly in the attention softmax at initialization (mean $|K| \approx 1.03$); prefix V vectors are initialized to zero so the module is transparent at step 0 and vanilla SR is preserved.

4.3 Experiments

4.3.1 Vanilla Baseline

The vanilla OpenVLA-7B model fine-tuned on LIBERO-Goal achieves 79.0% overall (395/500 trials across 10 tasks, 50 trials per task). Per-task results are shown in Table 5.

Table 5: Vanilla baseline per-task success rates on LIBERO-Goal (50 trials each).

Task ID	Task	SR
7	turn on the stove	100%
4	put the bowl on top of the cabinet	98%
1	put the bowl on the stove	94%
2	put the wine bottle on top of the cabinet	92%
8	put the bowl on the plate	84%
5	push the plate to the front of the stove	82%
6	put the cream cheese in the bowl	80%
9	put the wine bottle on the rack	68%
3	open the top drawer and put the bowl inside	54%
0	open the middle drawer of the cabinet	38%
Overall		**79.0%**

Task 3 (54% standard eval; 45.6% over 500 trials) is our RL training target: low enough to have room for improvement, high enough that the model has a functional policy to build on (see Section 4.3.2 for the state variance characterization).

4.3.2 State Variance Diagnostic

Before interpreting any RL training results, we characterize the noise floor of the task-3 reward signal. We run the vanilla model on all 50 initial states of task 3, 10 trials per state, and record per-state success rates.

The results are striking: per-state SR ranges from 0% (states 7, 20, 28: the robot systematically fails regardless of policy) to 80% (states 0, 11, 41, 45). All failed episodes timeout at exactly 410 steps (400 max steps + 10 stabilization steps), indicating the robot reaches a stuck configuration rather than failing early. With 8 rollouts per GRPO step, the sampled initial states may have SR distributions spanning 60–80pp by chance alone. Under these conditions, the GRPO advantage estimate primarily reflects which initial states were sampled, not whether the prompt is improving.

This diagnostic is an important methodological result independent of the RL outcomes: LIBERO-Goal task 3, while appearing tractable at 54% in a standard 50-trial evaluation, has a true vanilla SR of 45.6% over 500 trials, and a reward landscape dominated by initial-state sampling noise at small batch sizes.

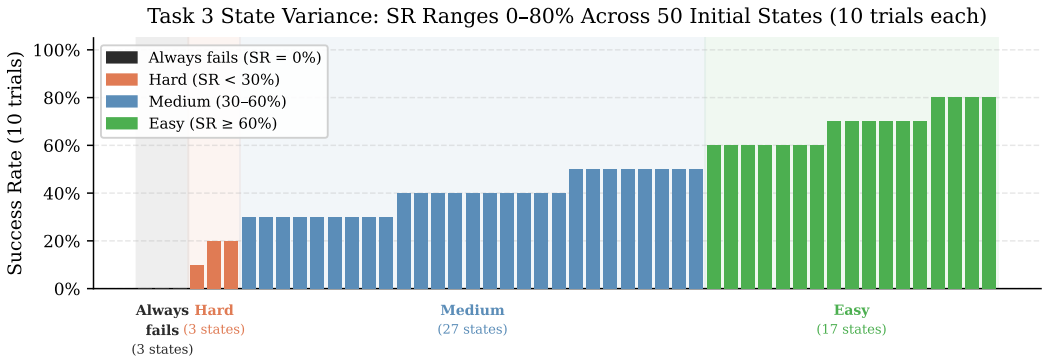

Figure 6: Per-state success rate for task 3 (50 initial states, 10 trials each). SR ranges from 0% to 80%, establishing that reward variance is dominated by initial-state sampling at small batch sizes.

4.3.3 Standard GRPO

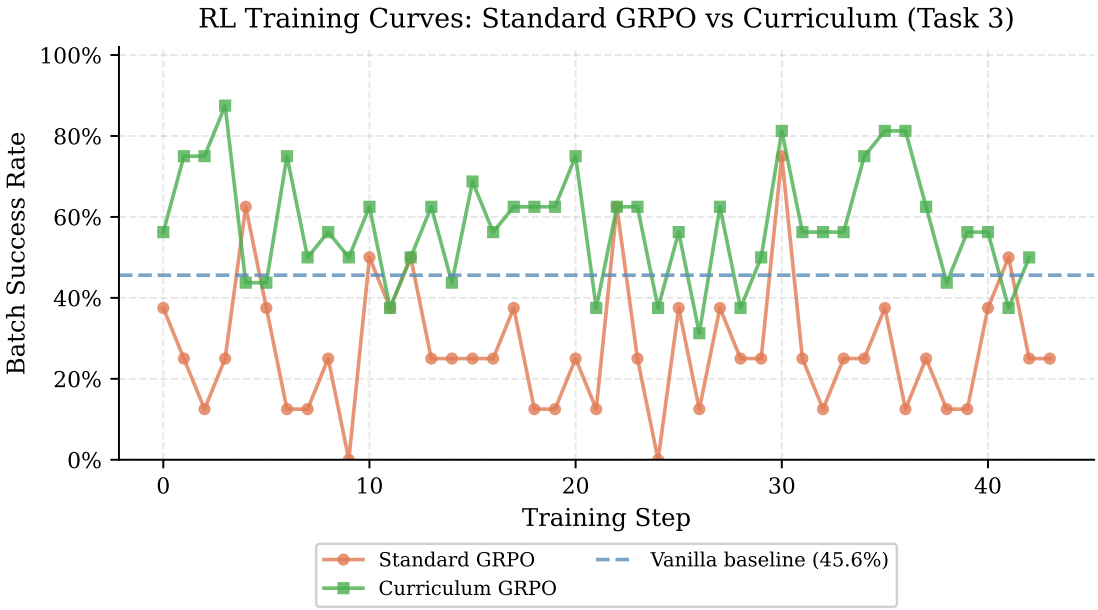

Figure 7: Training SR vs. step for standard GRPO (orange) and curriculum RL (green). Standard GRPO oscillates with no trend; curriculum RL shows qualitatively different, more stable dynamics.

We run standard GRPO for 43 steps on task 3 with zero-initialized prompt embeddings. SR oscillates between 12.5% and 62.5% with no consistent upward trend. Gradient norms are high (3.5–16.8), indicating large, potentially destabilizing updates to the prompt embeddings. No reliable learning signal emerges.

4.3.4 Curriculum RL

To reduce reward noise, we identify 17 "easy" initial states of task 3 where vanilla SR $\geq 60\%$ (from the state variance diagnostic) and restrict GRPO rollouts to this curriculum subset. We also switch from the `SoftPromptModule` (input-space injection) to an attention prefix module: 16 trainable

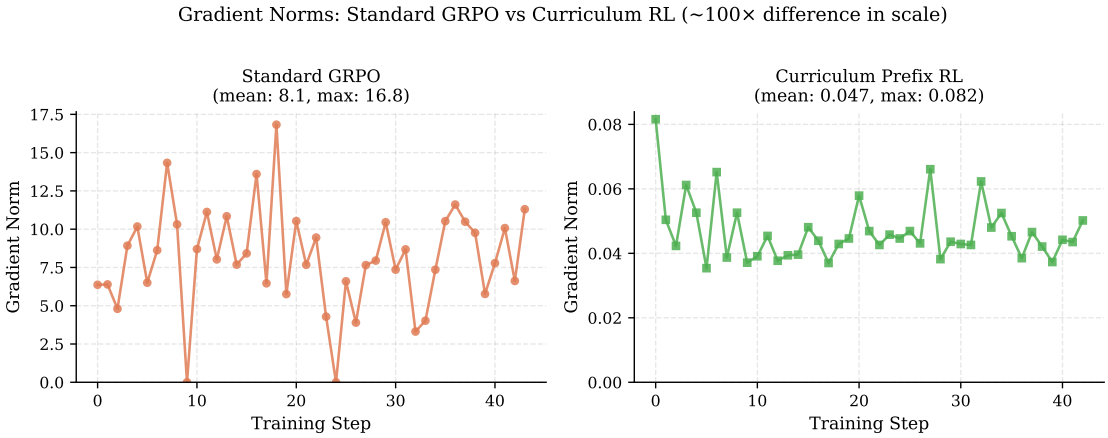

Figure 8: Gradient norm vs. step for standard GRPO (left) and curriculum RL (right). Standard GRPO norms reach 3.5–16.8; curriculum RL norms are 0.03–0.08, approximately 100× smaller.

K,V pairs injected into all 32 transformer layers via an SDPA patch, for 4,194,304 total trainable parameters.

The curriculum run (`rl_prefix_curriculum`, 43 steps) shows qualitatively different optimization dynamics from standard GRPO:

- Gradient norms: 0.03–0.08, approximately 100× **smaller** than standard GRPO
- No `group_skipped` steps (no degenerate batches)

A 50-trial evaluation on task 3 (same seed and state ordering as the vanilla baseline) yields the following results:

Table 6: Curriculum checkpoint evaluation, task 3, 50 trials.

Checkpoint	Eval SR (50 trials)	95% Wilson CI
`best.pt`	**38%**	[25.9%, 51.8%]
`latest.pt`	**36%**	[24.1%, 49.9%]
Vanilla baseline	**54%**	[40.4%, 67.2%]

Both checkpoints underperform the vanilla baseline. The two checkpoints are directionally similar (38% vs. 36%), indicating this is not a simple step-count artifact.

4.3.5 BC Warm-Start RL

As an additional baseline, we initialize the `SoftPromptModule`'s RL training from a BC (behavioral cloning) checkpoint, a `SoftPromptModule` fine-tuned on task-3 demonstration data via supervised learning, which achieves 48% SR on the standard 50-trial evaluation (24/50 trials). In the first RL training step, temperature-sampled rollouts yield SR as high as 87.5% (14/16 rollouts, a noisy single-step estimate at temperature 1.3), but performance degrades and oscillates between 12.5%–87.5% over 63 steps without consolidating. The pattern mirrors standard GRPO: strong initialization does not protect against reward-noise-driven instability.

4.4 Discussion

4.4.1 Why Short RL Runs Fail Here

Three factors combine to make short GRPO runs ineffective regardless of the injection method:

1. **State variance dominates reward**: with 8 rollouts per step from 50 possible initial states, the variance in sampled SR due to initial state alone (0–80% range) vastly exceeds any signal attributable to the prompt. The GRPO gradient primarily tells the prompt to mimic behavior from lucky rollouts, not behavior that generalizes.
2. **Overfitting to sampled states**: optimizing on a small batch per step from a high-variance pool means the prompt memorizes which behaviors work on the specific states sampled, not what makes the task structurally succeed.
3. **Insufficient training duration**: both experiments were terminated early (the curriculum prefix run when the GPU node became unresponsive at step 43, and the BC warm-start run after 63 steps due to time constraints), which may be insufficient for zero-initialized modules of this size to converge.

4.4.2 Soft Prompting vs. Attention Prefix

The two injection methods show meaningfully different optimization dynamics. Soft prompting (65K params, input embedding space) produced high gradient norms (3.5–16.8) and noisy oscillating training curves. Attention prefix tuning (4M params, K,V injection across all layers) produced gradient norms $\approx 100\times$ smaller with no degenerate batches, a qualitatively more stable regime.

Despite the difference in dynamics, the 50-trial evaluation of the attention prefix run shows similar degradation versus vanilla as the soft prompt run. Neither improved over the baseline. It is plausible that the attention prefix approach with a BC warm-start and longer training could produce better results; the soft prompt BC warm-start experiment showed that strong initialization is achievable, even if RL failed to consolidate from it.

4.4.3 Credit Assignment and the Frozen Backbone

DSRL's success with RL over diffusion inputs relies on a short credit assignment path: the noise vector is directly transformed into the output trajectory via the fixed denoising process, with one forward pass between input and reward. In our setting, the prompt injection passes through 32 frozen transformer layers before producing action tokens. The credit assignment path is longer, and the frozen landscape is non-convex; small prompt changes may cause large, unpredictable changes in action distributions due to attention interactions across layers.

The curriculum's more stable gradient dynamics suggest that restricting training to easier states and using a higher-dimensional injection (attention prefix vs. input embeddings) can partially reduce noise. But the fundamental difficulty remains: steering a frozen 7B-parameter model through a 65K–4M parameter intervention is a hard optimization problem with a reward signal dominated by environmental noise, and neither variant improved over vanilla in evaluation.

Table 7: Summary comparison of the two adaptation approaches.

	Bidirectional Sampling	**RL Soft Prompts**
Parameters changed	0 (inference-time)	65K–4M (trained)
Base model	AVDC (video diffusion)	OpenVLA 7B (VLA)
Benchmark	MetaWorld (3 tasks, 75 trials)	LIBERO-Goal (10 tasks, 20–500 trials)
Best result	+10.7pp button-press NR (sim+CLIP)	−7.6pp task 3 (curriculum RL, 38% vs. 45.6% 500-trial diagnostic)
Core failure mode	Backward pass OOD; no CFG in AVDC	State variance contaminates reward; both GRPO variants degrade performance

5 Cross-Cutting Discussion

5.1 Comparing the Two Approaches

The two approaches fail for different reasons that share a common structure: both hit a ceiling imposed by the frozen backbone's internal representations. Bidirectional Sampling asks AVDC to denoise reversed sequences it was never trained on; RL prompts ask OpenVLA to change behavior through a bottleneck of frozen attention layers.

5.2 The Common Ceiling: Frozen Backbone Limits

Both approaches are bounded by what the frozen backbone already knows how to represent. This is not a flaw of the specific methods; it is a structural property of any lightweight adaptation strategy.

For Bidirectional Sampling, the backward pass asks AVDC to denoise a temporally reversed sequence, a distribution that AVDC never encountered in training. The model produces plausible-looking frame sequences for reversed time, but the resulting action vectors, computed via optical flow from these frames, are not reliably goal-directed. CLIP conditioning partially patches this by injecting a semantic signal that doesn't depend on the reversed pixel distribution, which is why it helps and the backward pass alone does not.

For RL prompts, 65K–4M trainable parameters can adjust the input distribution to the frozen model but cannot reshape the model's internal representation of what task success looks like. The frozen attention layers process any prompt through fixed weight matrices; the prompt can shift activations in the token-embedding space but cannot change how those activations are transformed into action predictions. This explains the ceiling on curriculum RL: the prompt can steer the model toward states it already handles well, but cannot teach it new manipulation capabilities.

6 Conclusion

This thesis documents two attempts to improve pretrained robot manipulation systems with minimal modification to their weights.

Attempt 1 establishes that the utility of CLIP conditioning is task-dependent. On button-press,

the NR setting yields +10.7pp (sim+CLIP, 28.0% → 38.7%); in the Full setting, replanning alone (AVDC Full) achieves 56.0%; Bidirectional Sampling Full at best reaches 54.7%, a marginal 1.3pp below the closed-loop ceiling. On door-open, all NR variants degrade performance (−5.4pp best case); but in the Full setting CLIP is decisive, recovering from a flat noclip baseline (41.3–44.0%) to surpass the AVDC Full ceiling (68.0% vs. AVDC Full's 66.7%). The core NR result is the dual ablation: neither the backward pass alone nor CLIP alone yields gain; both are required, establishing them as synergistic rather than redundant.

Attempt 2 explores two parameter-efficient RL approaches for adapting a frozen OpenVLA: soft prompt tuning and attention prefix tuning, both optimized via short GRPO runs on LIBERO-Goal. Neither improved over the vanilla baseline. The primary finding is methodological: initial-state variance (0–80% SR range on the training task) overwhelms the reward signal at practical batch sizes, making short RL runs inherently noisy regardless of the injection method. Longer runs with BC warm-starting and variance reduction were not completed due to compute and time constraints.

The most direct next steps, given more compute and time, are:

- **Longer RL runs with BC warm-starting**: the curriculum GRPO run terminated at 43 steps with no BC initialization for the attention prefix. A proper run with BC warm-start followed by hundreds of GRPO steps and per-state reward normalization would give the method a fair test.
- **Direct CLIP conditioning on AVDC**: we tested CLIP injected into the forward pass only (28.0% on button-press, identical to baseline), confirming that goal conditioning requires the backward pass structure to take effect. A natural extension is fine-tuning AVDC on goal-conditioned data to give the model native support for goal images in the forward pass.
- **Retrain VPP policy head on Bidirectional Sampling latents**: the architectural mismatch between Bidirectional Sampling's full denoising trajectory and VPP's low-t latent distribution could be resolved by retraining only the policy head on Bidirectional Sampling latents, leaving the SVD backbone frozen.
- **ReFT (Representation Finetuning)** [9]: rather than prepending soft prompts to the input or injecting prefix key-value pairs into attention, ReFT directly intervenes on hidden representations at specific layers via low-rank linear transformations. This is a strictly richer intervention surface than either approach tested here, and may overcome the bottleneck imposed by frozen attention weights that limits both the soft prompt and attention prefix approaches.

Acknowledgements

I would like to extend a special thank you to my advisor, Professor Chen Sun, for his guidance and support throughout this project. I would also like to thank Professor Hui Wang for serving as my second reader. Lastly, I would like to thank my family for their encouragement and patience.

References

[1] Blattmann, A., et al. (2023). Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. *arXiv preprint arXiv:2311.15127*.

[2] Deng, M., et al. (2022). RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. *arXiv preprint arXiv:2205.12548*.

[3] Ho, J., & Salimans, T. (2022). Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*.

[4] Kim, M. J., et al. (2024). OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246*.

[5] Ko, P.-C., et al. (2023). Learning to Act from Actionless Videos through Dense Correspondences. *arXiv preprint arXiv:2310.08576*.

[6] Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. *arXiv preprint arXiv:2104.08691*.

[7] Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv preprint arXiv:2101.00190*.

[8] Li, H., et al. (2025). SimpleVLA-RL: Scaling VLA Training via Reinforcement Learning. *arXiv preprint arXiv:2509.09674*. ICLR 2026.

[9] Wu, Z., et al. (2024). ReFT: Representation Finetuning for Language Models. *arXiv preprint arXiv:2404.03592*.

[10] Zang, H., et al. (2025). RLinf-VLA: A Unified and Efficient Framework for Reinforcement Learning of Vision-Language-Action Models. *arXiv preprint arXiv:2510.06710*.

[11] Liu, B., et al. (2023). LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*.

[12] Mees, O., et al. (2022). CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation Letters*, 7(3), 7327–7334.

[13] Radford, A., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*.

[14] Shao, Z., et al. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*. (Introduces GRPO.)

[15] Yang, S., Kwon, T., & Ye, J. C. (2024). ViBiDSampler: Enhancing Video Interpolation Using Bidirectional Diffusion Sampler. *arXiv preprint arXiv:2410.05651*.

[16] Song, J., Meng, C., & Ermon, S. (2020). Denoising Diffusion Implicit Models. *arXiv preprint arXiv:2010.02502*.

[17] Wagenmaker, A., et al. (2025). Steering Your Diffusion Policy with Latent Space Reinforcement Learning. *Conference on Robot Learning (CoRL)*. arXiv:2506.15799.

[18] OpenAI. (2025). gpt-image-1. *OpenAI Product Release*. https://openai.com/index/introducing-4o-image-generation/.

[19] Yu, T., et al. (2020). Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *Conference on Robot Learning (CoRL)*.

[20] Chung, H., et al. (2024). CFG++: Manifold-constrained Classifier Free Guidance for Diffusion Models. *arXiv preprint arXiv:2406.08070*.

[21] Chung, H., Lee, S., & Ye, J. C. (2023). Decomposed Diffusion Sampler for Accelerating Large-Scale Inverse Problems. *arXiv preprint arXiv:2303.05754*.

[22] Su, J., et al. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*.

[23] Touvron, H., et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

[24] Hu, Y., et al. (2024). Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations. *arXiv preprint arXiv:2412.14803*. ICML 2025.