DiSCo: Distilled Steering via Consolidation for Robot Diffusion Policies

Sakthivel Sivaraman
Department of Computer Science
Stanford University
sakthivels@nvidia.com

William Liu
Department of Computer Science
Stanford University
wzliu@stanford.edu

Jerry Gu
Department of Computer Science
Stanford University
jerrygu@stanford.edu

Abstract

Diffusion policies are a leading approach to visuomotor robot control, and DynaGuide has shown that a separately trained dynamics model can steer a frozen diffusion policy toward desired outcomes during action denoising. This steering is useful but transient: the policy does not retain the guided skill, and deployment must repeatedly run the dynamics-guidance loop. We present DiSCo (Distilled Steering via Consolidation), a training-time procedure that converts DynaGuide behavior into durable unguided policy skills. The key empirical lesson is that the teacher must optimize completion, not only the first correct object motion. With full no-early-stop DynaGuide rollouts and diffusion-policy behavior cloning, unguided CALVIN students reach 91.8%, 86.8%, and 57.2% strict success on `switch_on`, `drawer_close`, and `door_left` over four seeds, and a merged hard-three policy reaches 81/90/49%. However, naive sequential consolidation still forgets: final continual checkpoints average 1.5% strict success unguided and 8.3% with DynaGuide reapplied. These results show that inference-time guidance can be amortized into policy weights, while continual retention and explicit goal conditioning remain the central open problems.

1 Introduction

Diffusion policies Chi et al. (2023) model robot action chunks as samples from a conditional denoising process. This gives stable, multimodal visuomotor policies, but it does not by itself solve test-time steering: once a policy is trained, changing the desired behavior usually requires either retraining on a new objective distribution or modifying inference.

DynaGuide Du and Song (2025) takes the second route. It keeps the base diffusion policy fixed and uses a separately trained latent dynamics model to guide denoising toward desired future examples. This is attractive because guidance can be changed at inference time without retraining the policy. The limitation is that the guided behavior is stateless. If DynaGuide repeatedly shows the robot how to perform a task, the base policy still does not learn that task; the same dynamics model and guidance loop must be run again at deployment.

We ask whether inference-time guidance can be turned into a durable skill. Our project went through three stages. First, we reproduced DynaGuide and confirmed that guidance improves method selection even when strict task completion is not reached. Second, we tried direct on-policy/noise-matching distillation inspired by SDFT Shenfeld et al. (2026); this transferred "what to start doing" but not

reliably "how to finish." Third, we changed the teacher-data construction: instead of stopping at first motion, we ran DynaGuide through full episodes, selected completion-quality trajectories, and trained an unguided diffusion policy by behavior cloning those guided action chunks. This final recipe is the source of our strongest results.

Our contributions are:

- We formulate DiSCo, a practical pipeline for consolidating DynaGuide into an unguided diffusion policy.
- We show that completion-quality teacher trajectories, not first-motion-only guidance, are the critical bottleneck for strict CALVIN success.
- We provide replicated single-task results and a preliminary multitask result showing that guided behavior can be absorbed into policy weights.
- We characterize unsuccessful directions: naive sequential continual learning, post-hoc DynaGuide on continual checkpoints, and simple task/guidance-demo conditioning remain insufficient.

2 Related Work

Diffusion policies. Diffusion Policy Chi et al. (2023) represents visuomotor control as conditional denoising over action sequences. We use this policy class throughout: both the DynaGuide teacher and the DiSCo student operate over diffusion action chunks.

Inference-time guidance. DynaGuide Du and Song (2025) guides a frozen diffusion policy with a latent dynamics model and a set of desired future examples. The original method is valuable because it separates the guidance signal from the base policy and can steer toward objectives that were not explicitly provided as training conditions. Our work asks whether the resulting behavior can be made persistent by training the policy on what the guided teacher actually does.

Continual self-distillation. SDFT Shenfeld et al. (2026) argues that on-policy self-distillation can reduce forgetting because the student trains on states it actually visits. This motivated our Phase 1.1 noise-matching experiments. In our robotics setting, however, on-policy matching alone was not enough; the stronger result came from improving teacher completion quality and using standard diffusion BC.

CALVIN. We evaluate on CALVIN Mees et al. (2022), a simulated benchmark for long-horizon language-conditioned robot manipulation. We use individual articulated-object skills from the environment rather than the full language-conditioned sequence benchmark.

3 Method

3.1 DynaGuide Teacher

Let s_t be the current observation and a_t be a candidate action chunk sampled during diffusion denoising. DynaGuide uses a learned latent dynamics model $h_\phi(s_t, a_t)$ to predict a future scene embedding and compares it to desired future embeddings $\{\hat{z}_i\}_{i=1}^N$. The guidance score is

$$G(s_t, a_t) = \log \sum_i \exp\left(-\frac{\|h_\phi(s_t, a_t) - \hat{z}_i\|_2}{\alpha}\right),$$

and the denoising process is nudged in the direction of $\nabla_{a_t} G(s_t, a_t)$. In our experiments, the base policy, DINOv2 visual encoder Oquab et al. (2024), and latent dynamics model are frozen.

3.2 Completion-Quality Trajectory Distillation

Our initial attempt matched the student's noise prediction to the guided teacher on visited states. This produced high target-first-motion on hard tasks, but strict success stayed near zero. The failure mode was that the teacher signal often specified the correct *prefix* of the method but did not provide enough terminal progress.

The final recipe therefore treats DynaGuide as a trajectory generator. For each task, we run DynaGuide to strict success or horizon, keep full observation/action trajectories, and select completion-quality

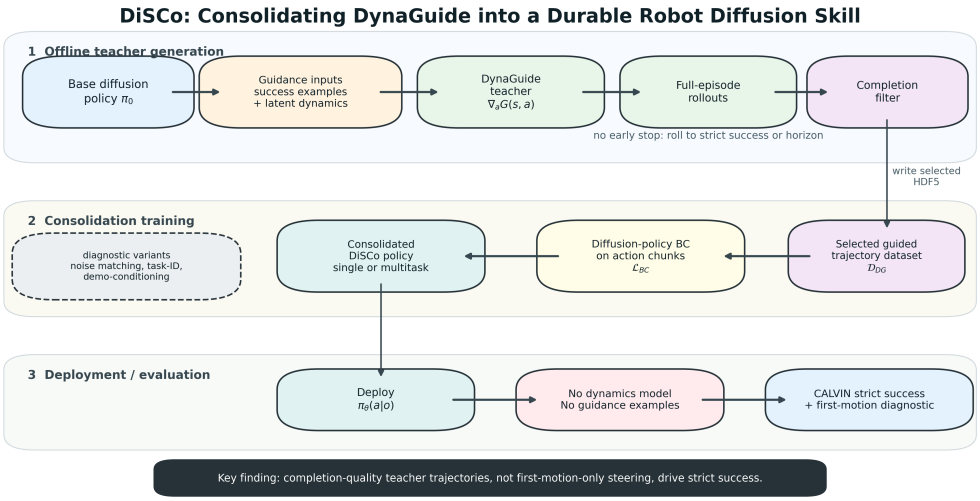

Figure 1: DiSCo pipeline. A frozen base diffusion policy is steered by DynaGuide during teacher collection. The key change is to collect full completion-quality trajectories rather than stop at first correct object motion. A student diffusion policy is then trained on the guided action chunks and deployed without the dynamics model or guidance loop.

examples. Given a guided action chunk $a_{t:t+H}^{DynaGuide}$, the student is trained with the standard diffusion-policy BC objective:

$$\mathcal{L}_{\mathrm{BC}} = \mathbb{E}_{t,k,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(s_t, \sqrt{\bar{\alpha}_k}a_{t:t+H}^{\mathrm{DG}} + \sqrt{1-\bar{\alpha}_k}\epsilon, k\right)\right\|_2^2\right].$$

After training, the student is evaluated with no dynamics model, no success examples, and no test-time guidance.

3.3 Multitask and Continual Consolidation

For multitask BC, we merge the selected guided trajectory datasets from multiple tasks and train a single diffusion policy on the union. For continual consolidation, we fine-tune a policy sequentially across tasks and evaluate the final checkpoint on all tasks. We also evaluate final continual checkpoints with DynaGuide reapplied at inference to test whether a weak sequential policy can still be steered.

4 Experimental Setup

Tasks. We evaluate four CALVIN skills: `button_on`, `switch_on`, `drawer_close`, and `door_left`. Most final single-task and multitask evaluations use 100 rollouts per cell. The continual+DynaGuide matrix uses 24 task orderings, 4 evaluation tasks, and 50 rollouts per cell, for 4,800 rollouts.

Metrics. Strict success is the main metric: it checks whether the simulator state satisfies the task-completion threshold. Target-first-motion is a diagnostic metric: it checks whether the first articulated object that moves matches the intended skill. Early in the project this diagnostic was useful because policies often reached for the correct object but failed to complete the motion. For the final no-early-stop BC results, strict success is primary.

Baselines and variants. We compare the base policy, the base policy with online DynaGuide, distilled students without test-time guidance, merged multitask BC students, sequential continual students, and sequential students with DynaGuide reapplied at inference. We also ran task-ID and guidance-demo conditioning controls to test whether explicit conditioning solves the multitask/continual ambiguity.

Table 1: Single-task consolidation. Base and base+DynaGuide columns report strict success / target-first-motion from the method-prefix baseline sweep. The final student column reports strict success without test-time guidance. Hard-task student values are four-seed means; the `button_on` value is the best completed Phase 1 distilled checkpoint.

Task	Base	Base + DynaGuide	Student, no DynaGuide
`button_on`	.33 / .31	.72 / .70	.94
`switch_on`	.00 / .20	.00 / .65	$.918 \pm .028$
`drawer_close`	.00 / .17	.00 / .70	$.868 \pm .054$
`door_left`	.00 / .16	.00 / .75	$.572 \pm .064$

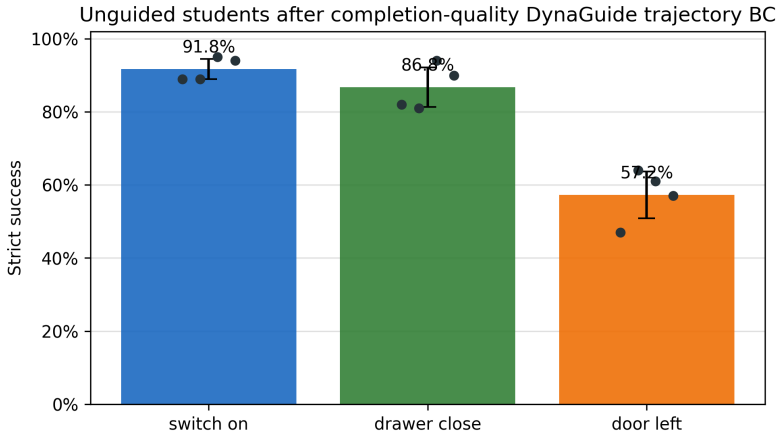

Figure 2: Four-seed replication for the hard-task students trained from completion-quality no-early-stop DynaGuide trajectories. Dots are individual seeds; bars show mean and standard deviation.

5 Results

5.1 Single-Task Consolidation

Table 1 shows the main result. The standard online DynaGuide setting improves target-first-motion on the hard tasks, but strict completion is still poor. Once DynaGuide is used as a full-episode teacher and we clone completion-quality trajectories, the unguided student reaches high strict success on `switch_on` and `drawer_close`, and moderate success on `door_left`. The result is replicated across seeds in Figure 2. This supports the central claim that inference-time guidance can be amortized into policy weights, but only if the teacher data contains terminal progress.

5.2 Multitask Consolidation

Merged multitask BC is stronger than expected. A single hard-three policy reaches 81% on `switch_on`, 90% on `drawer_close`, and 49% on `door_left`. Pairwise policies also show non-trivial transfer to held-out tasks, for example `drawer_close+door_left` training reaches 76% on `switch_on`. We do not treat this as a final claim of robust goal-conditioned control because the policy is not explicitly conditioned on a task instruction and may exploit task or initialization biases. It is nevertheless evidence that a single diffusion policy can store multiple guided behaviors when trained on full guided trajectories.

5.3 Continual Learning and Conditioning

The continual result is the main negative result. Across 24 sequential task orderings, final checkpoints average 1.5% strict success and 18.1% target-first-motion without guidance. Reapplying DynaGuide at inference improves target-first-motion to 44.3% and strict success to 8.3%, but strict success is almost entirely from `button_on`. For the hard tasks, DynaGuide often makes the final checkpoint

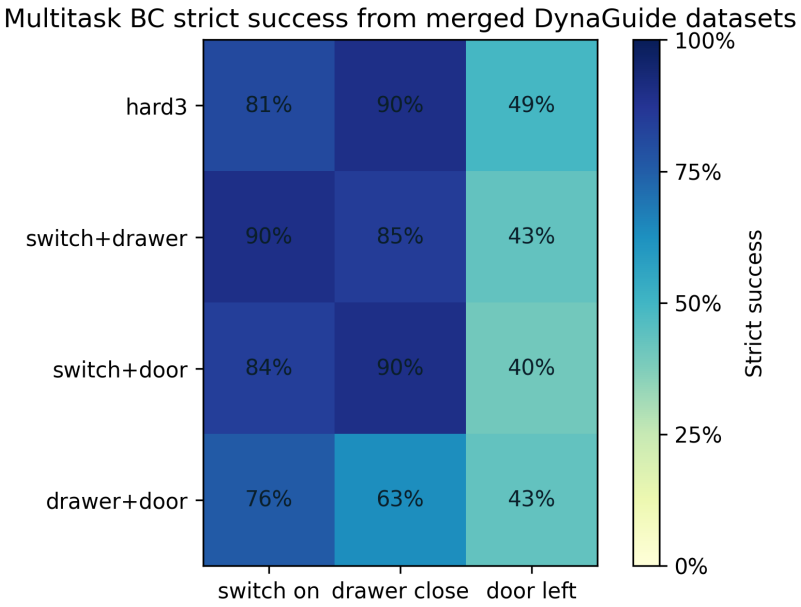

Figure 3: Strict success for merged multitask BC policies trained from seed-1 completion-quality DynaGuide datasets. The `hard3` policy is trained on all three hard tasks; pairwise policies are trained on two-task subsets and evaluated on all three tasks.

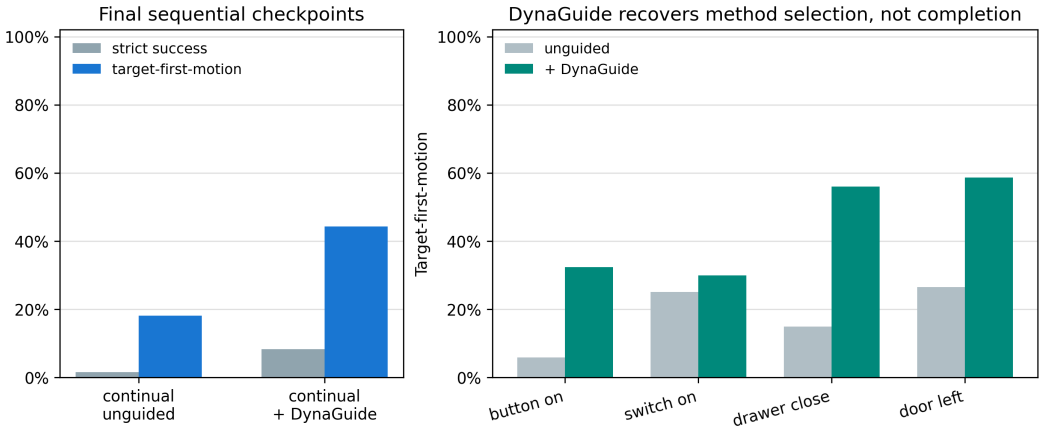

Figure 4: Continual-learning diagnostic. Left: final sequential checkpoints are weak unguided and remain low in strict success even when DynaGuide is reapplied. Right: DynaGuide recovers method selection on several tasks, especially `drawer_close` and `door_left`, but this does not convert to strict completion.

move the intended object first, yet the rollout still does not finish. This shows that the sequential policy has not retained a completion-capable action manifold for DynaGuide to steer.

Simple conditioning did not solve the issue. The best task-ID or guidance-demo conditioned Phase 2 models reached roughly 0.50 target-first-motion but at most 0.065 strict success, and shuffled conditions still produced nonzero behavior. These controls suggest that a weak condition head can become a task-ID shortcut or partial steering signal, but not a reliable goal-conditioned policy.

6 Discussion

What worked. The successful recipe is not simply "run more DynaGuide." The important change was using DynaGuide as a completion-quality trajectory teacher. Earlier policies learned method prefixes because the guidance signal made the first object motion correct. The final BC students learned completion because the training targets included the full action sequence that actually crossed the strict simulator threshold. This explains why target-first-motion became less informative for the final no-early-stop experiments: a trajectory can complete the task even if the first-motion heuristic misses the relevant behavior.

What did not work. Direct on-policy noise matching, completion-weighted variants, and simple task/guidance-demo conditioning did not produce strong hard-task strict success. Continual sequential fine-tuning also failed: even with DynaGuide at inference, final checkpoints mostly recover intent rather than terminal progress. The likely cause is ambiguity and forgetting. Without explicit goal conditioning or replay, the policy updates for a new task overwrite the action regions needed to finish previous tasks.

Implications for a publishable next step. The strongest next experiment is a clean continual benchmark using the successful teacher data: compare naive sequential fine-tuning, balanced replay, adapters, and frozen-backbone task heads under matched train data. A stronger research direction is completion-aware or world-model-guided teacher repair: use a learned progress model or JEPA-style latent world model to rank action chunks by predicted terminal progress, then distill only trajectories that both choose the right method and finish the task.

7 Conclusion

We found that DynaGuide guidance can be consolidated into unguided robot diffusion policies, but the quality of the teacher trajectories is decisive. Full no-early-stop DynaGuide trajectories distilled by diffusion BC yield strong replicated strict success on `switch_on` and `drawer_close`, moderate success on `door_left`, and promising preliminary multitask results. At the same time, naive sequential learning and simple goal-conditioning controls do not solve durable multi-skill retention. The project therefore supports a clear claim and a clear limitation: guidance can be amortized into policy weights, but continual consolidation requires explicit anti-forgetting and completion-aware supervision.

8 Team Contributions

- **William Liu:** Environment setup and codebase integration; simulator setup; rollout-collection pipeline coupling the CALVIN environment; primary ablations; transferability to other tasks.
- **Sakthivel Sivaraman:** Distillation training loop implementation; continual-learning scheduler; catastrophic-forgetting analysis; result compilation and final report integration.
- **Jerry Gu:** DynaGuide reproduction on `switch_on`; guided-rollout BC baseline; inference-efficiency comparison; ablations. Experimenting with alternative latent represenation spaces (DINOv3).

Changes from Proposal. The proposal focused on on-policy self-distillation of DynaGuide noise predictions. Our experiments showed that this was not the bottleneck: hard tasks needed completion-quality teacher trajectories. We therefore shifted from noise-matching as the main result to full-trajectory DynaGuide distillation, while retaining the continual-learning question as the main unresolved challenge.

A Additional Experiments

Representation backbone. We tested a DINOv3 ViT-S/16 dynamics-model variant with the same broad training recipe and swept guidance temperature/distance choices. The best `switch_on` setting reached 54% in the older reproduction setup, below the DINOv2-based 68% reproduction. We therefore kept the original DINOv2 backbone for final experiments.

Phase 1.1 on-policy noise matching. The SDFT-inspired variant trained the student on states it visited and matched the teacher's guided noise targets. It produced high target-first-motion on hard tasks, often above 80%, but strict completion remained near zero. This motivated the switch to full-trajectory BC.

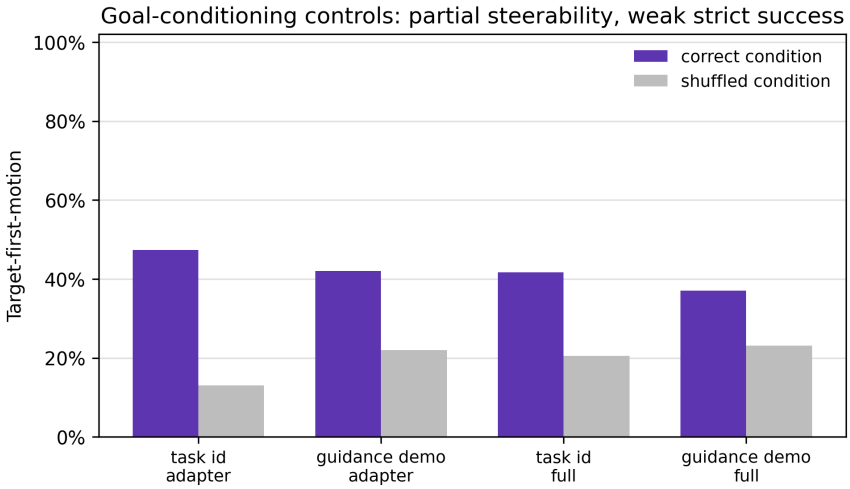

Figure 5: Conditioning controls. Correct task-ID or guidance-demo conditions improve target-first-motion relative to shuffled conditions, but strict success remains weak, so this is not a sufficient solution.

Phase 2 conditioning controls.

JEPA/progress-model direction. We ran lightweight progress-model and JEPA-ranker probes. These were useful diagnostics, but did not yet produce a main-result-quality teacher. The most promising use is not to replace the successful BC pipeline, but to rank or filter candidate guided rollouts by predicted terminal progress before distillation.

B Implementation Details

The base policy is a 1-D temporal U-Net with ϵ-prediction, 7-D actions, and 16-step action chunks, denoised with DDIM. The DynaGuide teacher uses a Transformer dynamics model over frozen DINOv2-S/14 patch tokens Oquab et al. (2024) that predicts a future scene embedding. Guidance is computed as $\nabla_a \log \sum_i \exp(-\|h_\phi(s,a) - \hat{z}_i\|/\alpha)$ over success-demonstration embeddings $\hat{z}_i$, with temperature $\alpha = 30$, guidance scale 1.5, and $M = 4$ stochastic sampling steps in the original reproduction setting. Later teacher collection jobs used no first-motion early stopping and selected trajectories by strict completion or progress before BC training.

References

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*. arXiv:2303.04137

Maximilian Du and Shuran Song. 2025. DynaGuide: Steering Diffusion Policies with Active Dynamic Guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. 2022. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation Letters* (2022). arXiv:2112.03227

Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research (TMLR)* (2024).

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-Distillation Enables Continual Learning. *arXiv preprint arXiv:2601.19897* (2026).