ROBOSTG: TRAINING-FREE SPATIAL-TEMPORAL GUIDANCE FOR GENERATIVE ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained generative robot policies provide expressive behavioral priors, yet their deployment-time behavior remains limited by the objectives encoded in training. The central challenge is not simply how to apply guidance at inference time, but how to convert manipulation-specific requirements into differentiable objectives that are compatible with generative action synthesis. We introduce RoboSTG, a training-free framework for reshaping frozen generative robot policies with manipulation-oriented spatial-temporal guidance. RoboSTG expresses task requirements through differentiable spatial fields and relational constraints in physical space, and lifts them to action-chunk objectives via substage-aware spatial-temporal modulation. These objectives produce gradients that directly steer iterative action generation, without policy fine-tuning, learned value models, or candidate ranking. Experiments on LIBERO and RoboTwin and four real-world manipulation settings show strong task success and robust task-aligned execution. Together, the results demonstrate that structured, temporally aligned manipulation objectives provide an effective interface for adapting pretrained generative policies at inference time.

1 INTRODUCTION

Generative models have recently emerged as a powerful paradigm for robotic manipulation, enabling robots to learn expressive and multimodal behaviors from large-scale demonstrations. Unlike conventional policies that directly regress deterministic actions, generative robot policies model distributions over action sequences and capture diverse manipulation strategies under complex visual and linguistic conditions. Recent advances in diffusion-based policies, flow-based models, and vision-language-action (VLA) models have demonstrated remarkable capabilities in learning long-horizon manipulation behaviors and generalizing across diverse tasks. These developments suggest that generative policies can serve as strong behavioral priors for embodied agents.

However, deploying generative robot policies in real-world environments requires more than reproducing behaviors observed during training. Robotic manipulation often involves task-specific requirements that are difficult to fully encode into the learned policy distribution, including precise spatial constraints, object-object interaction relationships, contact configurations, and execution-stage dependent objectives. For example, a robot may need to not only reach an object but also maintain a specific relative pose during grasping, satisfy geometric constraints during insertion, or adjust its behavior according to different manipulation stages. Such requirements may emerge after policy training and vary across tasks, creating a fundamental challenge: *how can pretrained generative policies be adapted to new manipulation requirements at inference time while preserving their learned behavioral priors?*

A natural solution is to incorporate additional task objectives during policy adaptation. Existing approaches mainly follow three directions. Training-based approaches optimize policies with additional demonstrations, reinforcement learning, or preference signals, but require costly data collection and policy updates. Trajectory optimization and sampling-based approaches evaluate or refine generated actions after prediction, which may introduce additional computation and deviate from the original generative prior. More recently, inference-time guidance methods have shown that external objectives can steer generative policies without retraining. However, existing guidance mechanisms typically rely on generic feasibility costs, learned value functions, or external evaluators. They pro-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

vide limited understanding of how task-specific manipulation knowledge should be represented and transformed into guidance signals that can directly influence action generation.

In this work, we argue that the key challenge is not merely how to add guidance to a generative policy, but how to formulate manipulation requirements into differentiable objectives that are compatible with the generative process itself. Unlike conventional trajectory evaluation, where task satisfaction is measured after generation, we aim to make manipulation objectives actionable during generation. Specifically, we introduce **RoboSTG**, a training-free **Robo**tic **S**patial-**T**emporal **G**uidance framework that reshapes generative robot policies through manipulation-oriented differentiable objectives.

The key idea of RoboSTG is to transform high-level manipulation requirements into structured differentiable objectives defined in the physical manipulation space, and then use their gradients to guide action generation. RoboSTG first represents manipulation requirements through dense spatial objectives and relational interaction constraints, capturing geometric and task-specific properties of manipulation behaviors. Since generative policies typically predict temporally structured action chunks rather than isolated poses, RoboSTG further introduces a spatial-temporal modulation operator that lifts state-level manipulation objectives into trajectory-level objectives over the predicted execution horizon. This operator allows task objectives to be aligned with different temporal phases of manipulation, enabling guidance to emphasize early approach, intermediate interaction, or final task completion according to the execution structure. The resulting objectives provide inference-time guidance signals that directly reshape the generation process while keeping the pretrained generative policy frozen.

Formally, RoboSTG views the pretrained generative policy as a behavioral prior and introduces manipulation objectives through an energy-based distribution reshaping perspective:

$$p^*(A|o,c) \propto p_\theta(A|o)\exp(\lambda J_m(A,o,c)),$$

where J_m denotes the manipulation-oriented differentiable objective. The gradient of this objective modifies the generation trajectory toward task-satisfying behaviors without requiring additional policy training, reward model learning, or large-scale candidate sampling. Figure 1 illustrates this distribution-reshaping view.

Our contributions are summarized as follows:

- We introduce **RoboSTG**, a training-free spatial-temporal guidance framework for generative robotic manipulation policies, which transforms manipulation requirements into differentiable objectives and reshapes action generation during inference.
- We propose **manipulation-oriented differentiable objectives** that unify dense spatial fields and relational interaction constraints, enabling task-specific guidance beyond generic feasibility objectives.
- We introduce a **spatial-temporal modulation operator** that converts state-level manipulation objectives into trajectory-level guidance over predicted action chunks, aligning task objectives with the temporal structure of robot execution.
- We demonstrate that RoboSTG can adapt pretrained generative robot policies to diverse manipulation requirements across simulation benchmarks without additional policy training.

2 Related Work

2.1 Generative Robot Policies

Generative robot policies model action-sequence distributions rather than deterministic actions. Diffusion Policy Chi et al. (2023) learns multimodal action chunks; related work studies goal-conditioned imitation Reuss et al. (2023), offline RL Wang et al. (2023), compact 3D representations Ze et al. (2024), and geometric equivariance Yang et al. (2024). ACT Zhao et al. (2023) is a representative non-diffusion action-chunking policy. These methods provide strong priors, but deployment-time adaptation remains limited by their training objectives and data.

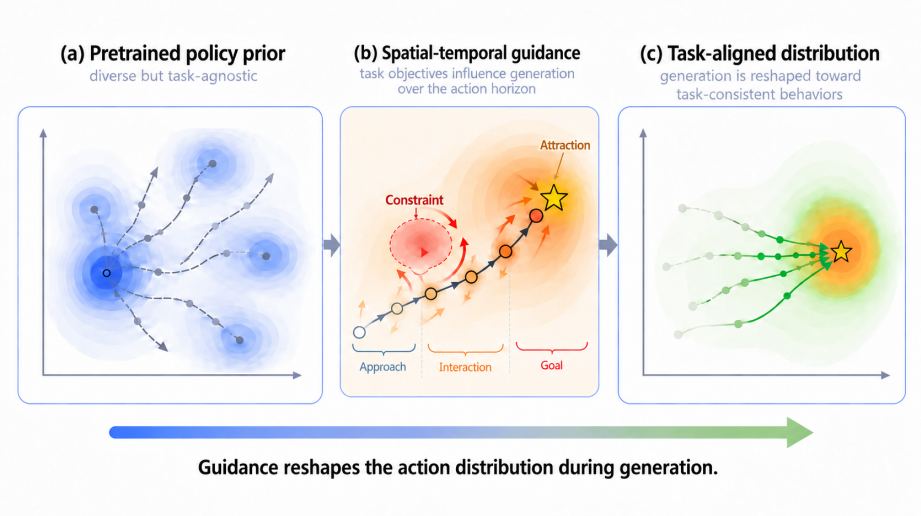


Figure 1: **RoboSTG paradigm.** Spatial, relational, and stage-aware objectives reshape a frozen policy prior during generation.

2.2 Inference-Time Guidance for Generative Models

Classifier and classifier-free guidance steer diffusion sampling without modifying the base model Dhariwal & Nichol (2021); Ho & Salimans (2022). Energy composition Liu et al. (2022), Universal Guidance Bansal et al. (2023), and Diffusion Posterior Sampling Chung et al. (2023) extend this idea to multiple or arbitrary differentiable objectives. Manipulation additionally requires physical relations and execution-time structure.

2.3 Guided Generation for Robotic Manipulation

Diffuser and Decision Diffuser generate trajectories under task conditions Janner et al. (2022); Ajay et al. (2023); later methods incorporate motion costs, geometry, compositional constraints, or safety Carvalho et al. (2024); Urain et al. (2023); Yang et al. (2023); Xiao et al. (2025). Policy steering includes value guidance, score composition, and human interaction Nakamoto et al. (2024); Wang et al. (2024; 2025), while other work uses language, object-centric, or kinematic constraints Hao et al. (2025); Li et al. (2025); Lee et al. (2025). We instead jointly structure spatial and relational objectives over the action horizon.

3 Method

3.1 Overview: Inference-Time Guidance for Generative Robot Policies

Generative robot policies provide expressive behavioral priors by modeling distributions over temporally structured action sequences. However, the learned distribution is inherently coupled with the demonstrations and task objectives available during training, which limits adaptation to new manipulation requirements at deployment time. We propose **RoboSTG**, a training-free inference-time guidance framework that reshapes pretrained generative policies using manipulation-oriented differentiable objectives.

Given an observation o and a pretrained generative policy $p_\theta(A|o)$, where $A = \{a_h\}_{h=1}^{H}$ denotes an action chunk of horizon H, RoboSTG defines a task-aligned target distribution as

$$p^*(A|o,c) \propto p_\theta(A|o)\exp(\lambda J(A,o,c))\,, \tag{1}$$

where c represents deployment-time task requirements, J is a manipulation-oriented objective, and λ controls guidance strength. This formulation retains the pretrained policy as a behavioral prior while increasing the probability of action sequences that better satisfy task-specific requirements.

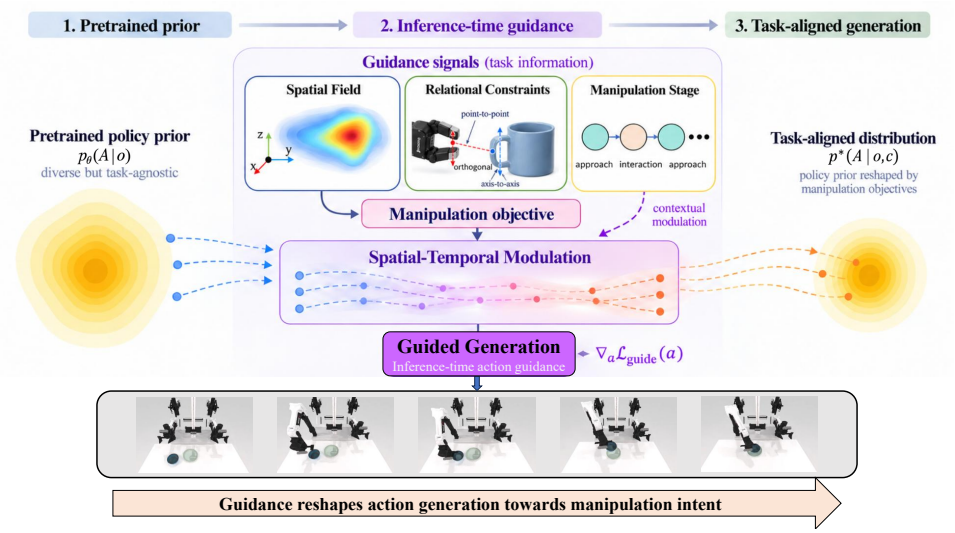


Figure 2: **RoboSTG overview.** Spatial fields and relational constraints are stage-modulated over the action horizon and injected as gradients during generation.

The main challenge is therefore how to construct J such that it captures physical manipulation requirements while remaining compatible with gradient-based generation. RoboSTG addresses this problem in three steps. First, spatial and relational manipulation requirements are formulated as differentiable objectives in physical space. Second, a *Spatial-Temporal Modulation Operator* lifts these objectives from individual configurations to the predicted action horizon. Finally, gradients of the resulting trajectory-level objective are injected into the generative process, reshaping action generation without updating policy parameters. Figure 2 summarizes this route from task specification to guided action generation.

3.2 Manipulation-Oriented Differentiable Objectives

Manipulation requirements are naturally defined in physical state space, whereas generative policies operate in action space. We therefore connect the two through a differentiable action-to-trajectory mapping

$$X_{1:H} = \Phi(s_t, A), \qquad X_{1:H} = \{x_h\}_{h=1}^{H}, \tag{2}$$

where s_t denotes the current state and x_h is the predicted physical configuration induced by the generated actions up to horizon step h. Depending on the action representation, Φ can correspond to pose integration, robot kinematics, or another differentiable transformation from actions to task-relevant configurations.

We represent manipulation requirements using two complementary objective families. The **spatial objective** describes where desirable configurations lie in the workspace, while the **relational objective** describes how task-relevant entities should interact.

Spatial objective. Given a predicted configuration x_h, we define

$$J_s(x_h) = \Phi_s(x_h, o, c), \tag{3}$$

where Φ_s is a continuous task-dependent spatial field. It can encode attraction toward task-relevant regions, avoidance of constrained regions, desired workspace configurations, or geometric feasibility. Unlike a binary task-success indicator, the spatial field provides dense feedback throughout the predicted trajectory and therefore yields informative gradients before a target configuration is reached.

Relational objective. Many manipulation requirements cannot be characterized by absolute positions alone. Grasping depends on the relative configuration between gripper and object, while insertion or placement may require precise object-object alignment. We therefore define

$$J_r(x_h) = \Phi_r(x_h, o, c), \tag{4}$$

where Φ_r measures task-relevant geometric relations such as relative position, orientation, keypoint alignment, or interaction configuration.

The per-step manipulation objective combines these two sources of task information:

$$J_m(x_h) = \alpha J_s(x_h) + \beta J_r(x_h), \tag{5}$$

where α and β control their relative contributions. This decomposition separates task specification from the architecture of the pretrained policy: the policy provides a learned behavioral prior, whereas J_s and J_r specify the manipulation requirements to be imposed at inference time.

3.3 Differentiable Field and Relation Construction

We instantiate the spatial and relational objectives as continuous differentiable functions so that they can directly influence generated actions.

Spatial field. Let $p_k(x_h) \in \mathbb{R}^3$ denote a task-relevant point induced by configuration x_h, such as the end-effector, object center, or an interaction keypoint. Given a scalar workspace field F, the spatial objective is obtained by querying the field at these points,

$$J_s(x_h) = \frac{1}{K} \sum_{k=1}^{K} F(p_k(x_h)) \,. \tag{6}$$

The field can combine attractive regions associated with task targets and repulsive or low-valued regions associated with constraints. When represented on a discrete workspace grid, we use differentiable interpolation such that both the field value and its spatial gradient can be propagated to the predicted trajectory.

This construction converts geometric task preference into a smooth optimization landscape. For example, positions closer to a desired interaction region may receive higher values, while proximity to constrained regions decreases the objective. Importantly, RoboSTG does not rely on a particular field generator; any field that provides differentiable task preference can be incorporated into the same guidance formulation.

Relational constraints. Relational requirements are expressed through continuous residuals between task-relevant entities. Let $\xi_u(x_h)$ and $\xi_v(x_h)$ denote geometric representations such as object poses, gripper poses, keypoints, or interaction axes. A relation is represented by a residual $r_j(x_h) = g_j(\xi_u, \xi_v) - \rho_j^*$, where g_j measures the current relation and ρ_j^* denotes its desired value. We convert these residuals into smooth scores and aggregate them as

$$J_r(x_h) = -\sum_{j=1}^{N_r} \omega_j \rho(r_j(x_h)) \,, \tag{7}$$

where $\rho(\cdot)$ is a differentiable penalty and ω_j controls the importance of each relation.

This generic form supports distance constraints, keypoint alignment, relative pose matching, orientation consistency, and other interaction requirements. In contrast to discrete relational predicates, continuous residuals provide gradients that indicate how the predicted manipulation configuration should change to better satisfy the desired interaction.

Figure 3 illustrates representative objective constructions. The dense field supplies a workspace-wide score landscape, while task entities and their desired relations determine the local geometric residuals evaluated along a predicted trajectory.

3.4 Substage-Aware Spatial-Temporal Modulation

The previous objectives evaluate individual predicted configurations. However, generative robot policies typically output temporally structured action chunks, and manipulation requirements are not equally important at every horizon step. For instance, approach motions may emphasize target reaching and collision avoidance, whereas interaction phases may require more accurate relative alignment.

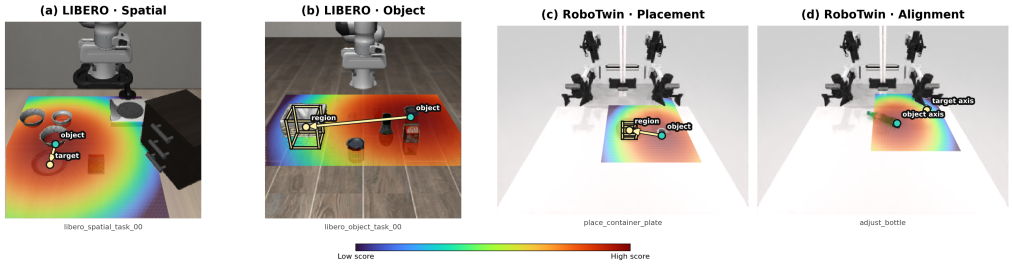


Figure 3: **Spatial and relational objectives.** Heat maps show field scores; overlays denote task entities and desired relations in LIBERO and RoboTwin.

RoboSTG therefore introduces a **Spatial-Temporal Modulation Operator** that lifts configuration-level objectives into an objective over the full action chunk:

$$J(A) = \mathcal{M}_\kappa[J_m](A) = \sum_{h=1}^{H} w_h J_m(x_h). \tag{8}$$

Here, $w_h \geq 0$ denotes the temporal modulation weight and $\sum_h w_h = 1$. The operator can be viewed as a discrete trajectory functional, or equivalently as an approximation to $\int \kappa(\tau) J_m(x(\tau)) d\tau$, where κ is a temporal modulation kernel over the physical execution horizon.

The role of the temporal kernel is to specify *when* different parts of the predicted trajectory should contribute most strongly to guidance. We consider increasing, decreasing, and stage-focused kernels, implemented using normalized exponential weights. These respectively emphasize later execution, near-term execution, or an intermediate interaction region of the action horizon.

We further allow manipulation substages to modulate the composition of task objectives. Let z_h denote the execution substage at step h. Instead of treating substage as an additional objective, we use it as context to adjust the relative importance of existing objectives:

$$J_m(x_h, z_h) = \alpha(z_h) J_s(x_h) + \beta(z_h) J_r(x_h), \tag{9}$$

with the temporal weights optionally conditioned on z_h as well. This yields a compact interpretation: spatial objectives specify *where* the robot should move, relational objectives specify *how* entities should interact, and substage-aware modulation specifies *when* each requirement should dominate.

Compared with independently evaluating predicted actions, Eq. 8 preserves the temporal coupling of an action chunk. Because an early action affects multiple downstream configurations, optimizing the trajectory-level functional naturally propagates future manipulation requirements to preceding actions.

3.5 Distribution Reshaping through Guided Generation

Given the modulated trajectory objective, RoboSTG computes the action-space guidance

$$g_A = \nabla_A J(A, o, c). \tag{10}$$

Through the differentiable mapping in Eq. 2, gradients of spatial and relational objectives are propagated from physical configuration space back to the generated action sequence.

The distributional interpretation of this guidance follows directly from Eq. 1. The target distribution is also the solution to the KL-regularized objective

$$\max_q \ \mathbb{E}_{A\sim q}[J(A)] - \frac{1}{\lambda} D_{\mathrm{KL}}(q(A) \| p_\theta(A|o)) \,. \tag{11}$$

Thus, RoboSTG can be interpreted as balancing two competing terms: improving task consistency through $J(A)$ while remaining close to the pretrained behavioral prior. The corresponding log-density gradient satisfies $\nabla_A \log p^* = \nabla_A \log p_\theta + \lambda \nabla_A J$, which motivates using $\nabla_A J$ as an additive task-dependent direction during iterative generation.

Algorithm 1 Core RoboSTG inference-time guidance

Require: Frozen generator f_θ, observation o_t, state s_t, context c_t, weights $w_{1:H}$, scale η
Initialize generative state A^S
for $k = S, \dots, 1$ **do**
  $\hat{A}_t^{0,k} \leftarrow \mathrm{PredictClean}(f_\theta, A^k, o_t, k)$
  $X_{1:H} \leftarrow \Phi(s_t, \hat{A}_t^{0,k})$
  $\bar{w}_h \leftarrow w_h / (\sum_{j=1}^{H} w_j)$
  $G_t \leftarrow \sum_{h=1}^{H} \bar{w}_h \left[F_t(x_h; c_t) - F_t(x_0; c_t)\right]$
  $g_t^k \leftarrow \nabla_{\hat{A}_t^{0,k}} G_t$
  $\tilde{A}_t^{0,k} \leftarrow \hat{A}_t^{0,k} + \eta\, g_t^k / (\|g_t^k\|_2 + \epsilon)$
  $A^{k-1} \leftarrow \mathrm{SamplerStep}(A^k, \tilde{A}_t^{0,k}, k)$
end for
return executable actions from $\tilde{A}_t^{0,1}$

In practice, RoboSTG evaluates the manipulation objective on the action sequence predicted by the generative process and repeatedly injects its gradient during inference. For policies that expose an estimated clean action chunk during iterative sampling, the objective is evaluated in this clean-action space, where actions can be directly mapped to physically meaningful trajectories.

RoboSTG requires neither policy retraining nor an additional learned value model. It also avoids generating and ranking multiple action candidates. Instead, manipulation requirements continuously influence the generation trajectory itself. Consequently, the pretrained model remains frozen while its original action distribution is reshaped toward task-aligned regions:

$$p_\theta(A|o) \;\longrightarrow\; p^*(A|o, c). \tag{12}$$

This formulation provides a unified route from manipulation requirements to generative action guidance: spatial and relational objectives define task preference in physical space, spatial-temporal modulation transforms them into a trajectory-level objective, and its gradient reshapes the pretrained generation process at inference time.

Algorithm 1 summarizes the core computation performed at each policy query.

4 Experiments

We evaluate whether RoboSTG improves a frozen policy under paired rollouts and whether its guidance follows the intended approach-to-interaction transition.

4.1 Experimental Setup

We use frozen X-VLA Zheng et al. (2025) on LIBERO Liu et al. (2023) and RoboTwin Mu et al. (2025). LIBERO uses ten Object-suite tasks with 50 episodes per task and method; RoboTwin uses 50 paired episodes per method. Each pair shares the initial state and generation noise, and only guidance is enabled or disabled. Both benchmarks use 30-step chunks, ten flow steps, and guidance scale 0.05. We report success and mean executed actions, treating differences descriptively because the baselines are near saturation.

4.2 Paired Evaluation Results

We use task success rate as the primary metric on LIBERO and RoboTwin and report mean executed actions as an execution-efficiency indicator. All comparisons use matched initial states and generation noise, allowing changes in completion and trajectory length to be attributed to inference-time guidance under the fixed evaluation protocol. Because the pretrained policy is already strong, these metrics are interpreted descriptively rather than as a broad statistical superiority claim.

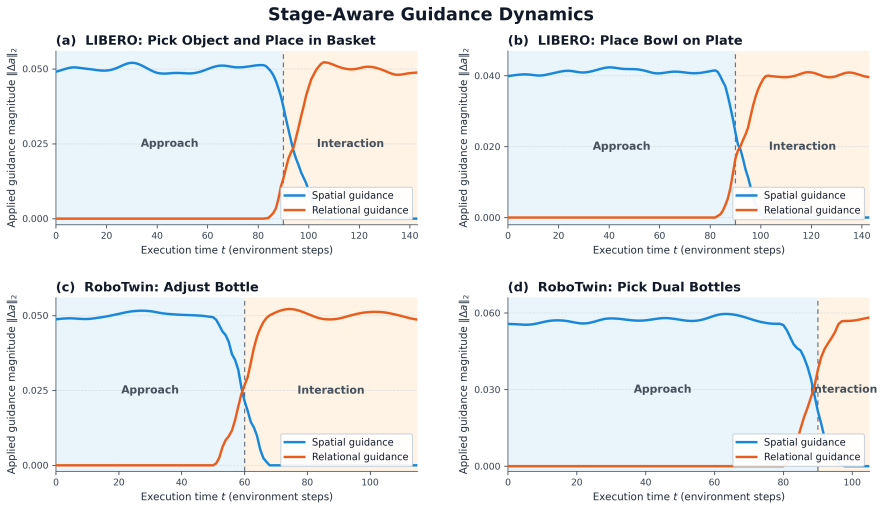


Figure 4: **Stage-aware guidance dynamics.** Guidance shifts from spatial approach objectives (blue) to relational interaction objectives (orange).

Table 1: **Real-robot success over 30 trials per task.**

Method	Screwdriver → mug	Cube → bowl	Aggregate
Pretrained prior	63.3% (19/30)	73.3% (22/30)	68.3% (41/60)
+RoboSTG	**76.7% (23/30)**	**83.3% (25/30)**	**80.0% (48/60)**
Absolute gain	+13.3 pp	+10.0 pp	+11.7 pp

4.3 Guidance Behavior Across Manipulation Stages

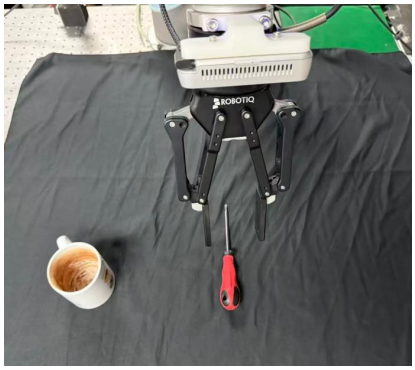
We inspect guidance finiteness, nonzero activation, spatial-versus-relational objective usage, and the timing of stage transitions. These diagnostics verify that guided rollouts execute both objective families rather than silently reducing to the pretrained prior.

Figure 4 visualizes representative temporal profiles from four tasks. Spatial guidance dominates while the robot approaches the task-relevant object or region, then decreases around the detected stage boundary as relational guidance increases for grasping, placement, or alignment. The transition is smooth because the objective weights are modulated over the action horizon rather than switched independently for every predicted action. This visualization is a mechanism diagnostic, not a causal ablation of either objective family.

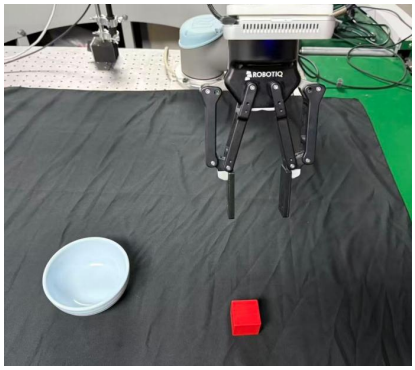
4.4 Real-Robot Evaluation

We further evaluate RoboSTG on a physical UR5 manipulator using an Intel RealSense camera for visual observation and a Robotiq gripper for object interaction. The two tasks are to pick a screwdriver and place it into a mug, and to pick a red cube and place it into a bowl (Figure 5). We conduct 30 trials for each task and method while keeping the pretrained policy frozen. As reported in Table 1, guidance improves success by 13.3 and 10.0 percentage points, respectively, with an aggregate increase from 68.3% to 80.0% over 60 trials per method.

Limitations. The study covers one policy and a limited task set, and assumes task geometry plus a differentiable action-to-trajectory mapping; broader policies, environments, and automatic grounding remain future work.



(a) Pick the screwdriver, place into the mug



(b) Pick the red cube, place into the bowl

Figure 5: **UR5 tasks.** Place a screwdriver into a mug (left) and a red cube into a bowl (right), using RealSense vision and a Robotiq gripper.

5 Conclusion

We introduced RoboSTG, a training-free framework that converts manipulation requirements into differentiable spatial and relational objectives, aligns those objectives with the execution structure of predicted action chunks, and uses their gradients to reshape a frozen generative robot policy at inference time. Paired experiments show a modest aggregate success increase and shorter executions on LIBERO-Object, while RoboTwin retains saturated success with a small reduction in episode length; two physical UR5 tasks further show gains of 13.3 and 10.0 percentage points. Mechanism diagnostics confirm that guidance shifts from spatial approach behavior to relational interaction behavior across task stages. Together, these results support spatial-temporal objective design as a practical interface for adapting generative policy priors, while also highlighting the need for broader evaluation and perceptual task grounding.

6 Ethics Statement

The reported experiments use simulated benchmarks and controlled physical robot trials; they do not involve human participants, personal data, or surveillance data. Inference-time guidance can improve task compliance, but it is not a safety guarantee: gradients from misspecified objectives or incorrect geometry may produce unsafe actions on physical hardware. Any real-world deployment should therefore include calibrated sensing, collision checking, force and workspace limits, emergency stopping, and human oversight appropriate to the robot and environment.

7 Reproducibility Statement

We use frozen checkpoints and paired simulation protocols that match initial states and action-generation noise across the prior and guided variants; each physical task uses 30 trials per method on the same UR5, RealSense, and Robotiq setup. The supplementary material includes implementation code, resolved configurations, task and seed manifests, raw episode records, guidance diagnostics, and validation scripts. The main experimental settings are stated in Section 4.1.

8 AI Use Statement

Generative AI tools were used to assist with writing refinement and code modification. The authors reviewed and verified the scientific claims, mathematical content, experimental results, citations, and final manuscript, and remain responsible for all submitted content.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.15657.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 843–852, 2023. URL https://arxiv.org/abs/2302.07121.

Joao Carvalho, Anh Le, Pawel Kicki, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and adapting robot motion planning with diffusion models. *arXiv preprint arXiv:2412.19948*, 2024. URL https://arxiv.org/abs/2412.19948.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2303.04137.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.14687.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021. URL https://arxiv.org/abs/2105.05233.

Ce Hao, Kelvin Lin, Zhiwei Xue, Siyuan Luo, and Harold Soh. DISCO: Language-guided manipulation with diffusion policies and constrained inpainting. *IEEE Robotics and Automation Letters*, 10(10):9726–9733, 2025. doi: 10.1109/LRA.2025.3595034. URL https://arxiv.org/abs/2406.09767.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915, 2022. URL https://arxiv.org/abs/2205.09991.

Kin Man Lee, Sean Ye, Qingyu Xiao, Zixuan Wu, Zulfiqar Zaidi, David B. D'Ambrosio, Pannag R. Sanketi, and Matthew Gombolay. Learning diverse robot striking motions with diffusion models and kinematically constrained gradient guidance. In *IEEE International Conference on Robotics and Automation*, pp. 8273–8280, 2025. URL https://arxiv.org/abs/2409.15528.

Hang Li, Qian Feng, Zhi Zheng, Jianxiang Feng, Zhaopeng Chen, and Alois Knoll. Language-guided object-centric diffusion policy for generalizable and collision-aware manipulation. In *IEEE International Conference on Robotics and Automation*, pp. 12834–12841, 2025. URL https://arxiv.org/abs/2407.00451.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.03310.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439, 2022. URL https://arxiv.org/abs/2206.01714.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2409.02920.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2410.13816.

Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2304.02532.

Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. SE(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *IEEE International Conference on Robotics and Automation*, pp. 5923–5930, 2023. URL https://arxiv.org/abs/2209.03855.

Lirui Wang, Jialiang Zhao, Yilun Du, Edward H. Adelson, and Russ Tedrake. PoCo: Policy composition from and for heterogeneous robot learning. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2402.02511.

Yanwei Wang, Lirui Wang, Yilun Du, Balakumar Sundaralingam, Xuning Yang, Yu-Wei Chao, Claudia Perez-D'Arpino, Dieter Fox, and Julie Shah. Inference-time policy steering through human interactions. In *IEEE International Conference on Robotics and Automation*, pp. 15626–15633, 2025. URL https://arxiv.org/abs/2411.16627.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2208.06193.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. SafeDiffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2306.00148.

Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. EquiBot: SIM(3)-equivariant diffusion policy for generalizable and data efficient learning. In *Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2407.01479.

Zhutian Yang, Jiayuan Mao, Yilun Du, Jiajun Wu, Joshua B. Tenenbaum, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Compositional diffusion-based continuous constraint solvers. In *Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2309.00966.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.03954.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2304.13705.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. URL https://arxiv.org/abs/2510.10274.