Refining Compositional Diffusion for Reliable Long-Horizon Planning

Kyowoon Lee[1] **Yunhao Luo**[2] **Anh Tong**[3] **Jaesik Choi**[4,5]

[1] Ajou University [2] University of Michigan [3] Korea University
[4] KAIST [5] INEEJI

Abstract

Compositional diffusion planning generates long-horizon trajectories by stitching together overlapping short-horizon segments through score composition. However, when local plan distributions are multimodal, existing compositional methods suffer from mode-averaging, where averaging incompatible local modes leads to plans that are neither locally feasible nor globally coherent. We propose *Refining Compositional Diffusion* (RCD), a training-free guidance method that steers compositional sampling toward high-density, globally coherent plans. RCD leverages the *self-reconstruction error* of a pretrained diffusion model as a proxy for the log-density of composed plans, combined with an *overlap consistency* term that enforces consistency at segment boundaries. We show that the combined guidance concentrates sampling on high-density plans that mitigate mode-averaging. Experiments on challenging long-horizon tasks from OGBench, including locomotion, object manipulation, and pixel-based observations, demonstrate that RCD improves compositional planning, with the largest gains on long-horizon tasks where mode-averaging is severe. Project website at https://refining-compositional-diffusion.github.io/.

1 Introduction

Planning plays a fundamental role in sequential decision-making, enabling agents to reason about future consequences before committing to actions. Classical approaches such as Model Predictive Control [72] and Monte Carlo tree search [67, 68, 39] achieve strong performance when accurate dynamics models are available, yet constructing such models for high-dimensional, continuous systems remains a significant challenge [31]. Diffusion probabilistic models have recently emerged as a compelling alternative for trajectory planning, learning to sample directly from the distribution of feasible plans and incorporating task-specific guidance through auxiliary gradient signals [31, 3, 45]. However, the effectiveness of a diffusion planner is inherently constrained by the horizon of available training data. Collecting long-horizon demonstrations covering all possible start-to-goal combinations is prohibitively expensive, motivating *compositional* approaches that assemble long plans from short, reusable segments [51, 81, 49, 52].

Compositional approaches address this limitation by decomposing a long trajectory into overlapping local segments, each modeled by a single short-horizon diffusion model, and composing their score functions at inference time to sample from an approximation of the joint distribution [17, 18, 81]. This strategy is data efficient and modular, allowing a model trained on short data to plan over arbitrarily long horizons. GSC [51] pioneered this for trajectory planning by chaining short-horizon diffusion trajectories via score composition. In practice, this composition averages the noise predictions of adjacent segments at every denoising step on their shared overlap regions. CompDiffuser [49] additionally introduced a bidirectional conditioning mechanism that propagates information between adjacent chunks during denoising. However, when the local plan distribution is *multimodal*, this

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

averaging-based composition suffers from *mode-averaging* [52]. Adjacent segments independently commit to incompatible modes, and their averaged scores produce composed trajectories that lie in low-density regions of the true distribution. While CDGS [52] mitigates this issue through population-based search and DDIM-inversion-based pruning [69], it incurs substantial inference-time overhead because many candidate trajectories must be repeatedly resampled, ranked, and filtered during the denoising process. More importantly, it improves final outcomes primarily through post hoc candidate selection, rather than by directly steering the denoising process toward more coherent plans.

In this paper, we propose **R**efining **C**ompositional **D**iffusion (**RCD**), a training-free guidance method that steers compositional sampling toward high-density regions of the local plan distributions while maintaining consistency across overlapping boundaries. We first show that the *self-reconstruction error*, the discrepancy between a predicted clean sample and its reconstruction after partial renoising, provides a differentiable proxy for the local log-density. High-density samples reconstruct faithfully, whereas off-manifold trajectories, including mode-averaging artifacts, exhibit large reconstruction error. We then introduce an *overlap consistency* term that penalizes score disagreement between adjacent segments at shared boundaries. Computed from pretrained local models alone, the combined signal directly steers denoising toward a tilted distribution that concentrates on high-density, globally coherent plans, mitigating mode-averaging without the population-based resampling and ranking overhead of prior search-based approaches, making RCD a plug-and-play component universally applicable to diffusion models.

Our main contributions are as follows: **(1)** We analyze the self-reconstruction error as a density proxy for composed trajectories and show its formal connection to the diffusion Evidence Lower Bound (ELBO), complemented by an overlap consistency term that measures score disagreement at segment boundaries. **(2)** We propose *Refining Compositional Diffusion* (RCD), a training-free guidance method that combines these signals to steer compositional sampling toward high-density, globally coherent plans using only a pretrained local diffusion model. **(3)** We evaluate RCD on OGBench long-horizon tasks spanning locomotion, object manipulation, and pixel-based observations, where it consistently improves success rates over prior compositional methods while running an order of magnitude faster than search-based alternatives, all without additional training or architectural changes to the local diffusion model.

2 Background

2.1 Planning with Diffusion Models

Diffusion probabilistic models [29, 71] learn a data distribution by defining a forward process that gradually adds Gaussian noise and a reverse process that recovers the clean data. Given a clean trajectory $\boldsymbol{\tau}^{(0)} \sim p(\boldsymbol{\tau})$, the forward process produces a sequence of increasingly noisy versions $\boldsymbol{\tau}^{(1)}, \dots, \boldsymbol{\tau}^{(T)}$ via

$$\boldsymbol{\tau}^{(t)} = \sqrt{\alpha_t}\,\boldsymbol{\tau}^{(0)} + \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

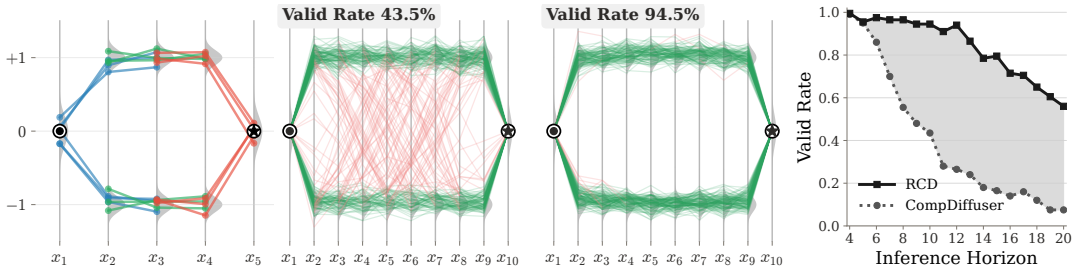
where $\alpha_1 > \alpha_2 > \cdots > \alpha_T \approx 0$ is a monotonically decreasing noise schedule. The reverse process recovers $\boldsymbol{\tau}^{(0)}$ by learning a noise-prediction network $\boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^{(t)}, t)$ trained to minimize $\mathbb{E}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^{(t)}, t)\|^2]$. Sampling proceeds iteratively from $\boldsymbol{\tau}^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\boldsymbol{\tau}^{(t-1)} = \boldsymbol{\mu}_\theta(\boldsymbol{\tau}^{(t)}, t) + \sigma_t\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $\boldsymbol{\mu}_\theta$ is the predicted mean derived from $\boldsymbol{\epsilon}_\theta$ and σ_t is the reverse-process standard deviation. Diffuser [31] applies this framework to trajectory planning by training a diffusion model over trajectories $\boldsymbol{\tau} = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \dots)$ from offline data. Task-specific objectives such as goal-reaching or return maximization are incorporated via gradient-based guidance at inference time [13].

2.2 Compositional Diffusion via Factor Graphs

Consider a global plan $\boldsymbol{\tau} = (x_1, \dots, x_N) \in \mathbb{R}^{N \times D}$ decomposed into M overlapping local segments (factors) y_j, where each variable $x_i \in \mathbb{R}^D$ represents a state or state-action pair and adjacent segments y_j and y_{j+1} share boundary variables in the overlap $y_j \cap y_{j+1}$. Following the Bethe



(a) Training Segments (b) CompDiffuser (c) RCD **(ours)** (d) Valid Rate vs. Horizon

Figure 1: **Toy illustration of the mode-averaging problem. (a)** Training data consists of overlapping l=3 segments, each shown in a distinct color. They are anchored at a fixed start ◉ and goal ⊛, and pass through a bimodal distribution with modes at $+1$ and -1. **(b)** CompDiffuser averages incompatible modes in overlap regions, producing many invalid (red) trajectories off both modes. **(c)** RCD guides the denoising toward high-density modes, yielding mostly valid (green) trajectories. **(d)** Valid rate drops sharply with horizon for CompDiffuser, while RCD maintains high validity.

approximation [78, 52], the global distribution factorizes as:

$$p(\boldsymbol{\tau}) = \frac{\prod_{j=1}^{M} p(y_j)}{\prod_{i=1}^{N} p(x_i)^{d_i-1}}, \tag{3}$$

where d_i is the degree of variable x_i in the factor graph ($d_i = 2$ for overlap variables, $d_i = 1$ otherwise). Taking log-gradients yields the compositional score:

$$\nabla \log p(\boldsymbol{\tau}) = \sum_{j=1}^{M} \nabla \log p(y_j) + \sum_{i=1}^{N} (1 - d_i) \nabla \log p(x_i). \tag{4}$$

In practice, each $p(y_j)$ is represented by a shared diffusion model conditioned on the noisy states of its neighbors, $\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t \mid y_{j-1}^{(t)}, y_{j+1}^{(t)})$ [49]. For notational simplicity, we write $\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)$ throughout, with the neighbor conditioning left implicit. The global score is computed by averaging noise predictions in overlapping regions [51, 81, 52]. Bethe marginal consistency requires $p_{y_j}(x_i) = p_{y_{j+1}}(x_i)$ for all shared variables $x_i \in y_j \cap y_{j+1}$. When local distributions are multimodal, score averaging blends incompatible modes, pushing samples toward low-density inter-modal regions. This problem worsens with the number of composed segments.

2.3 Tweedie's Formula for Denoising

When samples are perturbed by Gaussian noise $\tilde{\boldsymbol{\tau}} \sim \mathcal{N}(\boldsymbol{\tau}, \sigma^2 \mathbf{I})$, Tweedie's formula [64] provides the Bayes-optimal denoised estimate $\mathbb{E}[\boldsymbol{\tau}|\tilde{\boldsymbol{\tau}}] = \tilde{\boldsymbol{\tau}} + \sigma^2 \nabla_{\tilde{\boldsymbol{\tau}}} \log p(\tilde{\boldsymbol{\tau}})$. Applying this to the diffusion forward process in Equation (1), we obtain the posterior mean [11]:

$$\mathbb{E}[\boldsymbol{\tau}^{(0)}|\boldsymbol{\tau}^{(t)}] = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{\tau}^{(t)} + (1 - \alpha_t) \nabla_{\boldsymbol{\tau}^{(t)}} \log p(\boldsymbol{\tau}^{(t)}) \right) \approx \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{\tau}^{(t)} - \sqrt{1 - \alpha_t}\, \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^{(t)}, t) \right), \tag{5}$$

where the score function $\mathbf{s}_\theta$ is approximated by $\nabla_{\boldsymbol{\tau}^{(t)}} \log p(\boldsymbol{\tau}^{(t)}) \approx -\boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^{(t)}, t)/\sqrt{1 - \alpha_t}$. The right-hand side of Equation (5) defines the *Tweedie denoised estimate* $\hat{\boldsymbol{\tau}}_0^{(t)}$, which serves as a one-step prediction of the clean trajectory from a noisy sample at timestep t.

3 Refining Compositional Diffusion

3.1 Mode-Averaging Problem in Compositional Diffusion

Compositional score averaging as described in Section 2.2 provides a reasonable approximation when the Bethe-style factorization is accurate and the composed score field faithfully represents the intended global distribution. In practice, however, both are only approximate, and the resulting artifacts are especially severe when $p(y)$ is multimodal, where incompatible local modes can be

averaged together. Consider two adjacent segments y_k and y_{k+1} sharing overlap variables. If $p(y)$ is bimodal, for instance the overlap variables can take values near $+1$ or -1 with equal probability, then y_k and y_{k+1} may independently commit to different modes. When the scores from the two segments point in opposite directions, their average nearly cancels, leaving the overlap variables stranded in a low-density region between modes (near 0). The composed trajectory thus contains transitions that are individually implausible under any local model, leading to plan failure (Figure 1b). The probability of mode mismatch grows with the number of segments M. Under idealized independent mode selection, the probability that all $M-1$ adjacent pairs agree on the same mode decays exponentially with M. In practice, iterative denoising introduces correlations that slow the decay, but the qualitative trend persists and the valid rate drops sharply as M grows (Figure 1d). Figure 2 illustrates how RCD addresses this problem.

3.2 Self-Reconstruction Error as a Density Proxy

The core observation of RCD is that a pretrained diffusion model already provides an intrinsic density signal. Samples in high-density regions are faithfully reconstructed through a noise-denoise cycle, while mode-averaged samples in low-density regions are not. This *self-reconstruction error* serves as a training-free density proxy for compositional guidance. Given a candidate clean trajectory $\hat{\boldsymbol{\tau}}_0$ obtained via the Tweedie estimate in Equation (5), we assess its quality by measuring how well it can be *self-reconstructed* through the local models.

Definition 1 (Self-Reconstruction Error)**.** *Given a candidate trajectory $\hat{\boldsymbol{\tau}}_0$, a shared diffusion model $\boldsymbol{\epsilon}_\theta$, and a probe noise level $s \in \{1, \dots, T\}$, the self-reconstruction error is*

$$\mathcal{E}_{\text{recon}}(\hat{\boldsymbol{\tau}}_0;\, s) = \mathbb{E}_{\boldsymbol{\epsilon}}\big[\, \|\hat{\boldsymbol{\tau}}_0 - \hat{\boldsymbol{\tau}}_0^{\text{rec}}(\hat{\boldsymbol{\tau}}_0, \boldsymbol{\epsilon}, s)\|^2 \,\big], \tag{6}$$

where $\hat{\boldsymbol{\tau}}_0^{\text{rec}}(\hat{\boldsymbol{\tau}}_0, \boldsymbol{\epsilon}, s)$ is the composed Tweedie reconstruction obtained from the noised sample $\hat{\boldsymbol{\tau}}_s = \sqrt{\alpha_s}\,\hat{\boldsymbol{\tau}}_0 + \sqrt{1-\alpha_s}\,\boldsymbol{\epsilon}$: each local segment $\hat{y}_{j,s} = \hat{\boldsymbol{\tau}}_s[j]$ is independently denoised via $\hat{y}_{j,0} = (\hat{y}_{j,s} - \sqrt{1-\alpha_s}\,\boldsymbol{\epsilon}_\theta(\hat{y}_{j,s}, s))/\sqrt{\alpha_s}$, and the per-segment estimates are merged by the Bethe composition rule in Equation (3)*.*

The probe timestep s controls the perturbation magnitude. In practice, we approximate the expectation with a single Monte Carlo sample. Intuitively, when $\hat{\boldsymbol{\tau}}_0$ lies near a mode of $p(\boldsymbol{\tau})$, the score network $\boldsymbol{\epsilon}_\theta$ accurately predicts the injected noise at level s, yielding a small reconstruction error. Conversely, when $\hat{\boldsymbol{\tau}}_0$ falls between modes, as occurs under mode-averaging, the score prediction is biased toward a neighboring mode rather than recovering the original inter-modal input, resulting in large reconstruction error. The following proposition formalizes this connection.

Proposition 1 (Reconstruction Error as Density Proxy)**.** *Consider the self-reconstruction error $\mathcal{E}_{\text{recon}}(\hat{\boldsymbol{\tau}}_0; s)$ with squared-error distance. Assume that each local denoiser $\boldsymbol{\epsilon}_\theta$ is trained via the standard DDPM objective. Then the weighted sum over probe timesteps satisfies*

$$\sum_{s=1}^{T} \frac{\alpha_s}{1-\alpha_s}\, \mathcal{E}_{\text{recon}}(\hat{\boldsymbol{\tau}}_0;\, s) = \sum_{s=1}^{T} \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\, \big\|\boldsymbol{\epsilon} - \bar{\boldsymbol{\epsilon}}_\theta\big(\sqrt{\alpha_s}\,\hat{\boldsymbol{\tau}}_0 + \sqrt{1-\alpha_s}\,\boldsymbol{\epsilon},\, s\big)\big\|_2^2 \,\Big] \geq -\log p_\theta(\hat{\boldsymbol{\tau}}_0), \tag{7}$$

where $\bar{\boldsymbol{\epsilon}}_\theta$ is the composed noise prediction and $p_\theta(\hat{\boldsymbol{\tau}}_0)$ is the marginal likelihood.

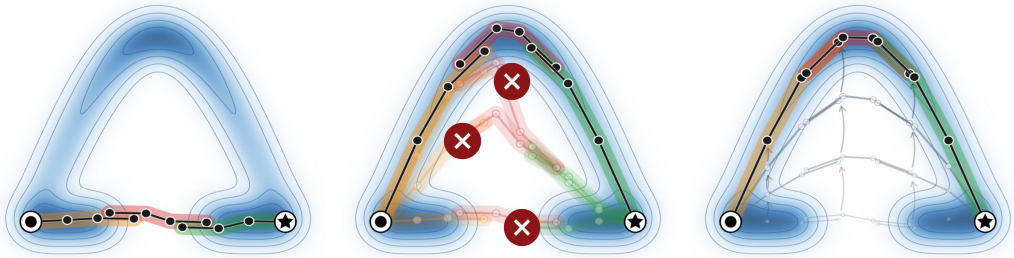
The proof can be found in Appendix D.1. Since $\mathcal{E}_{\text{recon}}$ at a single probe timestep s is one non-negative term of this upper bound on $-\log p_\theta(\hat{\boldsymbol{\tau}}_0)$, minimizing it is directionally aligned with maximizing the composed log-density, providing a principled justification for using $\mathcal{E}_{\text{recon}}$ as a guidance objective.

3.3 Overlap Consistency

The self-reconstruction error captures overall fidelity but may miss *local inconsistencies* at segment boundaries. Two adjacent segments may each reconstruct well individually while predicting different values for their shared variables $y_k \cap y_{k+1}$. We introduce the overlap consistency term to detect and penalize such disagreements.

Definition 2 (Overlap Consistency)**.** *Let $\hat{y}_{1,0}, \dots, \hat{y}_{M,0}$ be the per-segment Tweedie estimates from the reconstruction step of Definition 1. The overlap consistency is*

$$\mathcal{E}_{\text{ov}}(\hat{y}_{1:M}) := \frac{1}{M-1} \sum_{k=1}^{M-1} \big\|\hat{y}_{k,0}\big|_{y_k \cap y_{k+1}} - \hat{y}_{k+1,0}\big|_{y_k \cap y_{k+1}}\big\|_2^2, \tag{8}$$



(a) CompDiffuser (Luo et al., 2025) (b) CDGS (Mishra et al., 2026) (c) RCD **(ours)**

Figure 2: **Overview of RCD, contrasted with CompDiffuser and CDGS.** Trajectories are planned from start ◉ to goal ⊛ over a transition distribution ●. **(a)** CompDiffuser composes overlapping segments via bidirectional conditioning and autoregressive denoising, but produces mode-averaged trajectories in low-density regions when local distributions are multimodal. **(b)** CDGS mitigates this through population-based search and pruning at substantial inference cost. **(c)** RCD iteratively refines each denoising step toward high-density, globally coherent plans using the self-reconstruction error and overlap consistency (Section 3.4).

where $\hat{y}_{j,0}\big|_{y_k \cap y_{k+1}}$ *denotes the Tweedie estimate of segment* j *restricted to the variables shared between segments* k *and* $k{+}1$.

The overlap consistency is zero when all segments agree perfectly in their shared regions, a necessary condition for the Bethe approximation to be self-consistent, and grows when independent denoising pushes adjacent segments toward incompatible modes.

Proposition 2 (Overlap Consistency Measures Score Disagreement)**.** *Let* $\hat{y}_{k,0}\big|_{y_k \cap y_{k+1}}$ *and* $\hat{y}_{k+1,0}\big|_{y_k \cap y_{k+1}}$ *be the per-segment Tweedie estimates restricted to the overlap region* $y_k \cap y_{k+1}$. *Since both segments receive the same noisy input in the overlap (extracted from* $\hat{\boldsymbol{\tau}}_s$*), the overlap mismatch decomposes as*

$$\left\|\hat{y}_{k,0}\big|_{y_k \cap y_{k+1}} - \hat{y}_{k+1,0}\big|_{y_k \cap y_{k+1}}\right\|^2 = \frac{(1-\alpha_s)^2}{\alpha_s} \left\|\mathbf{s}_\theta^{(k)} - \mathbf{s}_\theta^{(k+1)}\right\|^2\big|_{y_k \cap y_{k+1}}, \tag{9}$$

where $\mathbf{s}_\theta^{(j)} := -\boldsymbol{\epsilon}_\theta^{(j)} / \sqrt{1-\alpha_s}$ *is the score from segment* j.

The proof can be found in Appendix D.2. Proposition 2 reveals that the overlap consistency measures the *score disagreement* between adjacent segments. Bethe marginal consistency requires that both segments produce identical scores in their shared region, so $\mathbf{s}_\theta^{(k)} \neq \mathbf{s}_\theta^{(k+1)}$ signals a violation. When they disagree, the overlap variables are pulled in different directions, a hallmark of mode-averaging. Minimizing $\mathcal{E}_{\text{ov}}$ thus encourages the marginal consistency implicit in the Bethe approximation, providing complementary information to the global reconstruction error.

3.4 RCD Guidance for Compositional Diffusion

Combining the self-reconstruction error and overlap consistency, we define the RCD guidance objective:

$$\mathcal{E}_{\text{RCD}}(\hat{\boldsymbol{\tau}}_0;\, s) := \mathcal{E}_{\text{recon}}(\hat{\boldsymbol{\tau}}_0;\, s) + \lambda_{\text{ov}}\, \mathcal{E}_{\text{ov}}(\hat{y}_{1:M}), \tag{10}$$

where $\lambda_{\text{ov}} \geq 0$ weights the overlap consistency. Following the training-free guidance framework (Appendix C), RCD treats $\mathcal{E}_{\text{RCD}}$ as the loss function and applies its gradient to steer the reverse process. At each denoising step t, we first obtain the Tweedie estimate $\hat{\boldsymbol{\tau}}_0^{(t)}$ from the current noisy sample $\boldsymbol{\tau}^{(t)}$ via Equation (5), then compute the guidance gradient with respect to $\boldsymbol{\tau}^{(t)}$ through the Tweedie estimate:

$$\mathbf{g}^{(t)} = \nabla_{\boldsymbol{\tau}^{(t)}} \mathcal{E}_{\text{RCD}}\big(\hat{\boldsymbol{\tau}}_0^{(t)};\, s\big). \tag{11}$$

The gradient is normalized and applied as a correction to the standard reverse step [66]:

$$\boldsymbol{\tau}^{(t-1)} = \boldsymbol{\mu}_\theta(\boldsymbol{\tau}^{(t)}, t) + \sigma_t\, \mathbf{z} - w\, \sigma_t^2\, \tilde{\mathbf{g}}^{(t)}, \quad \tilde{\mathbf{g}}^{(t)} = \frac{\mathbf{g}^{(t)}}{\|\mathbf{g}^{(t)}\|_\infty + \delta}, \tag{12}$$

where $w > 0$ is the guidance weight and $\delta > 0$ ensures numerical stability. The resulting guided reverse process targets a tilted distribution $\tilde{p}(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) \cdot \exp(-w \cdot \mathcal{E}_{\mathrm{RCD}}(\boldsymbol{\tau}))$ that concentrates on high-density, globally coherent plans, with the guidance weight w controlling the sharpness of this concentration. The full procedure is summarized as pseudo-code in Algorithm 1, with further practical details provided in Appendix E.

4 Experiments

In this section, we present the effectiveness of RCD on long-horizon planning tasks from OGBench [58]. Specifically, we demonstrate **(1)** that RCD produces more physically feasible plans than existing compositional methods, **(2)** that it further enhances planning performance on locomotion, object manipulation, and pixel-based observation tasks, and **(3)** that the two guidance components are complementary. Additional details regarding our experimental setup and implementation are provided in Appendix F, and further results including additional ablation studies in Appendix G.

Datasets and Environments. We evaluate on OGBench [58], a benchmark for offline goal-conditioned reinforcement learning that provides diverse long-horizon tasks with varying difficulty levels. Our evaluation covers three broad categories: **(1) Locomotion:** `PointMaze`, `AntMaze`, and `HumanoidMaze` environments in `Stitch` datasets across `Medium`, `Large`, and `Giant` maze sizes. **(2) Object Manipulation:** `Cube` manipulation tasks (`Single`, `Double`, `Triple`, `Quadruple`) and `AntSoccer` (`Arena`, `Medium`). **(3) Pixel-based Observations:** `AntMaze-Medium-Stitch` and `AntMaze-Large-Stitch` with visual observations, where planning operates in a learned latent space. The `Stitch` datasets are constructed from short, unconnected trajectory segments that do not individually span start-to-goal pairs, making long-horizon planning critically dependent on compositional stitching. Further details on environments and evaluation protocol are provided in Appendix F.

Baselines. We compare with offline goal-conditioned RL methods including goal-conditioned behavioral cloning (GCBC) [50], goal-conditioned implicit V-learning (GCIVL) and Q-learning (GCIQL) [35], Quasimetric RL (QRL) [73], Contrastive RL (CRL) [19], and Hierarchical implicit Q-learning (HIQL) [56]. We also include diffusion-based generative planning baselines: Generative Skill Chaining (GSC) [51], CompDiffuser (CD) [49], and CDGS [52]. For CD, CDGS, and RCD, we use the same pretrained local diffusion model for fair comparison.

4.1 Plan Feasibility Analysis

We first assess whether RCD improves the physical feasibility of composed plans. Figure 3 shows 20 sampled plans from CompDiffuser, CDGS, and RCD on each of the 5 test-time tasks in `AntMaze-Giant-Stitch`, together with the *valid plan rate*: the fraction of plans that do not pass through maze walls. CompDiffuser produces many infeasible plans, with valid rates ranging from 15% to 60% across tasks, as mode-averaging at corridor junctions pushes plans into physically impossible regions. CDGS improves validity through population-based pruning, but remains inconsistent across tasks since pruning only filters already-corrupted trajectories post hoc, and tends to collapse plan diversity toward a single mode. RCD achieves the highest valid rates across all 5 tasks by directly steering denoising toward high-density regions through the self-reconstruction error and overlap consistency, while preserving diverse plans across different feasible modes. We observe similar trends on `PointMaze-Giant-Stitch`; see Appendix G.1 for additional visualizations.

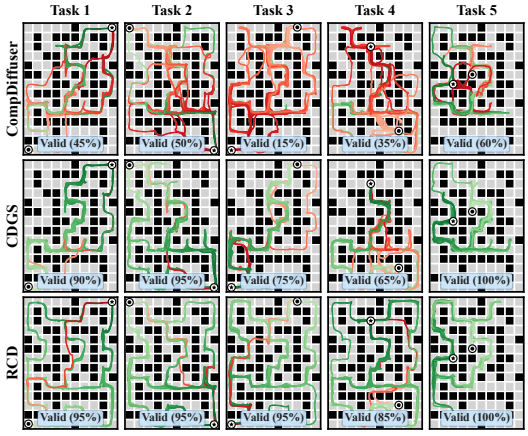


Figure 3: **Plan quality comparisons of CompDiffuser, CDGS, and RCD.** Each column shows 20 sampled plans in the `AntMaze-Giant-Stitch` environment for 5 test-time tasks defined in OGBench. Plans that violate environment constraints (wall penetration) are shown in red; feasible plans in green.

Table 1: **Quantitative results on OGBench locomotion tasks.** Each environment provides 5 goal-conditioned tasks; we evaluate 20 episodes per task and report mean ± std across 5 seeds.

Env	Type	Size	GCBC	GCIVL	GCIQL	QRL	CRL	HIQL	GSC	CD	CDGS	RCD
pointmaze	stitch	Medium	23±18	70±14	21±9	80±12	0±1	74±6	**100**±0	**100**±0	**100**±0	**100**±0
		Large	7±5	12±6	31±2	84±15	0±0	13±6	**100**±0	**100**±0	**100**±0	**100**±0
		Giant	0±0	0±0	0±0	50±8	0±0	0±0	29±3	69±3	74±3	**100**±0
antmaze	stitch	Medium	45±11	44±6	29±6	59±7	53±6	94±1	**97**±2	91±2	93±3	**97**±2
		Large	3±3	18±2	7±2	18±2	11±2	67±5	66±2	89±2	63±4	**91**±2
		Giant	0±0	0±0	0±0	0±0	0±0	2±2	20±1	67±3	83±3	**89**±2
humanoid maze	stitch	Medium	29±5	12±2	12±3	18±2	36±2	88±2	92±1	92±2	90±2	**93**±2
		Large	6±3	1±1	0±0	3±1	4±1	28±3	70±3	74±3	63±4	**79**±3
		Giant	0±0	0±0	0±0	0±0	0±0	3±2	5±1	44±4	35±4	**62**±3

4.2 Enhancing Long-Horizon Planning Performance

Locomotion. Table 1 reports success rates on `PointMaze-Stitch`, `AntMaze-Stitch`, and `HumanoidMaze-Stitch` tasks, where all methods plan in the 2D x-y position space and execute actions via a learned inverse dynamics model [49]. Offline GCRL algorithms struggle on `Giant` mazes, where the required planning horizon far exceeds the length of any single training trajectory. Compositional diffusion-based planners overcome this limitation by assembling long plans from short segments. RCD achieves the best performance across nearly all tasks. As indicated by the enhanced plan feasibility in Section 4.1, RCD translates improved plan quality into higher success rates, particularly on the most challenging `Giant` tasks where mode-averaging is most severe.

Object Manipulation in High-Dimensional State Spaces. Table 3 reports success rates on `Cube` and `AntSoccer` tasks, which involve planning in high-dimensional state spaces. For `Cube`, the planner operates on the full state including the 6-DoF robot arm joint positions and all cube poses, and actions are generated by a DQL-based value-learning policy [75]; for `AntSoccer`, we use the 17D planner that includes the x-y positions of the ant and ball along with all 13 joint positions of the ant [49], with actions produced by a learned inverse dynamics model. `Cube` tasks require composing dexterous pick-and-place sequences, ranging from `Single` cube to `Quadruple` with increasing horizon requirements. `AntSoccer` requires coordinating ant locomotion with ball dribbling to reach a goal location. Among offline GCRL methods, GCIQL performs best on `Cube`, while most baselines struggle on `AntSoccer` due to the long planning horizon. RCD achieves the best or competitive performance across all tasks, demonstrating that the RCD guidance generalizes effectively to high-dimensional planning spaces.

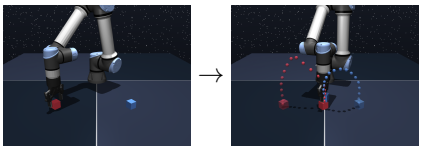
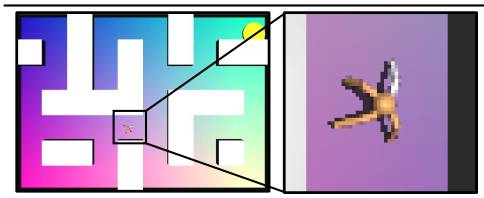


Figure 4: **A cube manipulation sequence executed by RCD.**

Visual AntMaze. To evaluate RCD beyond state-based settings, we test on `AntMaze-Stitch` tasks with pixel observations, where the agent observes 64×64 RGB images and planning operates in a 16-dimensional latent space learned by a variational autoencoder. Actions are produced by an inverse dynamics model that maps consecutive latent states to actions. Table 2 reports results for GSC, CD, CDGS, and RCD. RCD achieves improved performance on both `Medium` and `Large` mazes, confirming that the self-reconstruction error provides a meaningful density signal even in a compressed latent space where the original pixel-level structure is abstracted away.

Table 2: **Quantitative results on OGBench visual antmaze tasks.**

Env	Size	GSC	CD	CDGS	RCD
visual antmaze	Medium	40±4	55±3	26±4	**63**±3
	Large	8±3	15±3	11±3	**18**±3



Visual AntMaze Pixels

4.3 Additional Results

We present an ablation on the two RCD guidance components, an analysis of replanning, and a planning-time comparison with baselines below. Additional ablations on the guidance weight, the

Table 3: **Quantitative results on OGBench object manipulation tasks.** Each environment provides 5 goal-conditioned tasks; we evaluate 20 episodes per task and report mean ± std across 5 seeds.

Env	Type	Size	GCBC	GCIVL	GCIQL	QRL	CRL	HIQL	GSC	CD	CDGS	RCD
cube	play	Single	6±2	53±4	68±6	5±1	19±2	15±3	**100**±0	**100**±0	**100**±0	**100**±0
		Double	1±1	36±3	40±5	1±0	10±2	6±2	70±4	75±3	78±4	**85**±3
		Triple	1±1	1±0	3±1	0±0	4±1	3±1	45±4	51±4	**60**±4	**60**±3
		Quadruple	0±0	0±0	0±0	0±0	0±0	0±0	30±4	48±4	51±4	**55**±3
antsoccer	stitch	Arena	24±8	21±3	2±0	1±1	1±0	15±1	65±3	69±3	61±3	**72**±3
		Medium	2±1	1±0	0±0	0±0	0±0	4±1	12±2	17±3	11±2	**20**±2

overlap consistency weight, and the probe level are provided in Appendix G.2, confirming that RCD is robust to the choice of these hyperparameters.

Ablation on Guidance Components. RCD combines two guidance signals: the self-reconstruction error ($\mathcal{E}_{\text{recon}}$) and the overlap consistency ($\mathcal{E}_{\text{ov}}$). Table 4 isolates their contributions on `PointMaze-Giant-Stitch` and `AntMaze-Giant-Stitch`. The self-reconstruction error alone already provides a large improvement over the unguided baseline by steering samples toward high-density regions. Adding the overlap consistency further improves performance, particularly on `AntMaze-Giant-Stitch`, confirming that the two signals are complementary. $\mathcal{E}_{\text{recon}}$ addresses global density while $\mathcal{E}_{\text{ov}}$ targets local boundary agreement.

Table 4: **Ablation on guidance components.** Success rates with different combinations of $\mathcal{E}_{\text{recon}}$ and $\mathcal{E}_{\text{ov}}$.

$\mathcal{E}_{\text{recon}}$	$\mathcal{E}_{\text{ov}}$	pointmaze Giant	antmaze Giant
✗	✓	59±3	56±3
✓	✗	95±2	79±3
✓	✓	**100**±0	**89**±2

Ablation Study on Replanning. Replanning periodically regenerates a plan from the current state during execution to mitigate execution drift (see Appendix F.4 for details). Table 5 compares CD and RCD with and without replanning on `PointMaze-Stitch` and `AntMaze-Stitch` tasks. Replanning improves both methods, with the largest gains on `Giant` mazes where long plans are most susceptible to drift. Notably, RCD without replanning already outperforms CD with replanning on the most challenging `Giant` tasks, highlighting the importance of refining plans during generation where mode-averaging is most severe.

Table 5: **Ablation study on replanning.** Success rates on `pointmaze` and `antmaze stitch`, comparing CD and RCD with (✓) and without (✗) replanning.

Env	Size	CD		RCD	
		✗	✓	✗	✓
pointmaze	Medium	100±0	100±0	100±0	100±0
	Large	100±0	100±0	100±0	100±0
	Giant	55±5	69±3	98±1	**100**±0
antmaze	Medium	90±3	91±2	91±2	**97**±2
	Large	73±3	89±2	76±3	**91**±2
	Giant	29±4	67±3	69±3	**89**±2
Average		74.5	86.0	89.0	**96.2**

Planning-Time Comparison. Table 6 compares the success rate and planning time of RCD with CompDiffuser (CD) and CDGS on `PointMaze-Giant-Stitch` and `AntMaze-Giant-Stitch`, measured on a single NVIDIA H100 GPU. RCD applies its guidance on top of the parallel compositional score averaging (Equation (4)). Despite introducing this guidance step, its planning time remains below CD, which adopts an autoregressive variant for bidirectional conditioning, and is an order of magnitude below CDGS (population-based search). Intuitively, RCD is fast because all M segments are evaluated in a single batched call at each denoising step, whereas CD must sweep segments sequentially to propagate bidirectional conditioning and CDGS runs multiple sequential resampling and pruning passes per step.

Table 6: **Planning-time comparison with baselines.** Success rates and planning times (sec.).

Method	pointmaze-Giant		antmaze-Giant	
	Succ.	Time (s)	Succ.	Time (s)
CD	69±3	8.0±0.5	67±3	8.9±0.7
CDGS	74±3	44.7±3.9	83±3	56.7±7.4
RCD	**100**±0	**4.9**±0.1	**89**±2	**4.6**±0.3

5 Related Work

Planning with Diffusion Models. Diffusion probabilistic models [29, 71] have emerged as a powerful framework for trajectory planning in reinforcement learning. Diffuser [31] pioneered this direction by training an unconditional diffusion model on offline trajectories and guiding it toward high-return regions via a learned value function. Decision Diffuser [3] introduced classifier-free guidance, conditioning directly on reward or constraint signals, while AdaptDiffuser [45] progressively fine-tuned the model with synthetic high-quality data. Subsequent work extended diffusion planning to hierarchical settings [8, 43], multi-agent coordination [85], latent trajectory spaces [76], and inference-time tree search over diffusion rollouts [79, 80]. Monolithic diffusion planners, however, are limited by the training data horizon, motivating *compositional* approaches that stitch short segments at inference time. GSC [51] chains skills via score composition, CompDiffuser [49] introduces bidirectional neighbor conditioning for smoother stitching, and CDGS [52] combines compositional diffusion with population-based guided search to mitigate mode-averaging. A complementary line of work improves the reliability of plans produced by a diffusion planner. LoMAP [37] projects intermediate diffusion samples onto a local approximation of the data manifold to suppress off-manifold artifacts, while RGG [40] detects infeasible plans via a separately trained, time-dependent classifier and resamples them, which requires additional training data. RCD shares the goal of refining diffusion-generated plans but targets the compositional setting, where mode-averaging is the dominant failure mode. Unlike RGG, which trains a time-dependent classifier on additional labeled data, RCD is fully training-free and uses only the pretrained local denoiser as an intrinsic density proxy.

Trajectory Stitching. The ability to compose novel long-horizon behaviors from short trajectory fragments is a long-standing challenge in offline RL. One broad category of approaches relies on offline data augmentation, where short or reward-suboptimal trajectories are stitched or synthesized and a downstream policy is retrained on the enriched data, spanning generative augmentation [41, 30, 36, 38, 9], model-based stitching [6, 28, 84], and clustering-based augmentation [25]. RCD instead follows the *inference-time* compositional diffusion line of CompDiffuser [49] and CDGS [52], which compose local diffusion models directly at sampling time and avoid regenerating the dataset.

6 Conclusion

We presented Refining Compositional Diffusion (RCD), a training-free guidance method that addresses the mode-averaging problem in compositional diffusion planning. By leveraging the self-reconstruction error of a pretrained diffusion model as a density proxy and combining it with an overlap consistency term that penalizes score disagreement at segment boundaries, RCD steers compositional sampling toward high-density, globally coherent plans without additional training. We established theoretical connections between the reconstruction error and the composed model's evidence lower bound, and showed that the RCD-guided reverse process targets a tilted distribution concentrating on globally coherent plans. Empirically, RCD achieves consistent improvements over existing compositional methods across OGBench tasks spanning locomotion, object manipulation, and pixel-based observations, with particularly pronounced gains on the most challenging long-horizon tasks where mode-averaging is most severe.

Limitations. The current formulation operates within the Bethe factor graph framework of [78, 52], which assumes chain-structured overlapping segments. Extending RCD guidance to other compositional structures, such as hierarchical or temporal abstractions, or factor graphs with loops, is an interesting future direction. Another practical aspect, shared with prior compositional planners such as CompDiffuser [49] and CDGS [52], is that the number of local segments used to determine the total horizon is pre-specified at inference time. Adaptively choosing it is another promising direction, for instance via tree-search-style horizon expansion [79] or by monitoring the self-reconstruction error along candidate horizons.

Impact Statement. This paper advances compositional diffusion planning by introducing a training-free guidance method that refines long-horizon plans toward high-density, globally coherent trajectories. While we do not identify direct negative societal impacts stemming from this research, practitioners deploying diffusion-based planners in real-world decision-making systems are encouraged to carefully assess safety and reliability prior to deployment, particularly in settings where unsafe actions could carry significant consequences.

Acknowledgments and Disclosure of Funding

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (grant Nos. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST); RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University); RS-2022-II220984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation; RS-2024-00457882, AI Research Hub Project; IITP-2026-RS-2023-00255968, Artificial Intelligence Convergence Innovation Human Resources Development), the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2025-23525509), and the InnoCORE program of the Ministry of Science and ICT (N10250156).

The authors declare no competing financial interests.

References

[1] Agrawalla, B., Nauman, M., Agrawal, K., and Kumar, A. floq: Training critics via flow-matching for scaling compute in value-based rl. In *International Conference on Learning Representations (ICLR)*, 2026.

[2] Ahn, H., Choi, H., Han, J., and Moon, T. Option-aware temporally abstracted value for offline goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[3] Ajay, A., Du, Y., Gupta, A., Tenenbaum, J. B., Jaakkola, T. S., and Agrawal, P. Is conditional generative modeling all you need for decision making? In *International Conference on Learning Representations (ICLR)*, 2023.

[4] Baek, S., Park, T., Park, J., Oh, S., and Kim, Y. Graph-assisted stitching for offline hierarchical reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2025.

[5] Cai, Y., Wu, Y., Li, K., Zhou, Y., Zheng, B., and Liu, H. Flooddiffusion: Tailored diffusion forcing for streaming motion generation. *arXiv preprint arXiv:2512.03520*, 2025.

[6] Char, I., Mehta, V., Villaflor, A., Dolan, J. M., and Schneider, J. Bats: Best action trajectory stitching. *arXiv preprint arXiv:2204.12026*, 2022.

[7] Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[8] Chen, C., Deng, F., Kawaguchi, K., Gulcehre, C., and Ahn, S. Simple hierarchical planning with diffusion. *arXiv preprint arXiv:2401.02644*, 2024.

[9] Chen, C., Hamed, H., Baek, D., Kang, T., Bengio, Y., and Ahn, S. Extendable long-horizon planning via hierarchical multiscale diffusion. *arXiv e-prints*, pp. arXiv–2503, 2025.

[10] Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[11] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

[12] Clark, Q. and Shkurti, F. What do you need for diverse trajectory composition in diffusion planning? *arXiv preprint arXiv:2505.18083*, 2025.

[13] Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[14] Dong, J., Huang, W., Zhang, J., Chen, Z., Yuan, X., Gu, Q., Jiang, Z., and Ye, N. Proximal action replacement for behavior cloning actor-critic in offline reinforcement learning. *arXiv preprint arXiv:2602.07441*, 2026.

[15] Dong, Z., Hao, J., Yuan, Y., Ni, F., Wang, Y., Li, P., and Zheng, Y. Diffuserlite: Towards real-time diffusion planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[16] Dong, Z., Yuan, Y., Hao, J., Ni, F., Ma, Y., Li, P., and Zheng, Y. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

[17] Du, Y., Li, S., and Mordatch, I. Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[18] Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning (ICML)*, 2023.

[19] Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R. R. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[20] Farebrother, J., Pirotta, M., Tirinzoni, A., Munos, R., Lazaric, A., and Touati, A. Temporal difference flows. In *International Conference on Machine Learning (ICML)*, 2025.

[21] Feng, F., Ge, S., Fu, M., Li, Z., Zheng, Y., Tang, Z., Hu, Y., Huang, B., and Zhang, K. Ada-diffuser: Latent-aware adaptive diffusion for decision-making. In *International Conference on Learning Representations (ICLR)*, 2026.

[22] Feng, L., Gu, P., An, B., and Pan, G. Resisting stochastic risks in diffusion planners with the trajectory aggregation tree. In *International Conference on Machine Learning (ICML)*, 2024.

[23] Frans, K., Park, S., Abbeel, P., and Levine, S. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

[24] Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C., Eysenbach, B., and Levine, S. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019.

[25] Ghugare, R., Geist, M., Berseth, G., and Eysenbach, B. Closing the gap between td learning and supervised learning–a generalisation point of view. In *International Conference on Learning Representations (ICLR)*, 2024.

[26] Haramati, D., Qi, C., Daniel, T., Zhang, A., Tamar, A., and Konidaris, G. Hierarchical entity-centric reinforcement learning with factored subgoal diffusion. In *International Conference on Learning Representations (ICLR)*, 2026.

[27] He, Y., Murata, N., Lai, C.-H., Takida, Y., Uesaka, T., Kim, D., Liao, W.-H., Mitsufuji, Y., Kolter, J. Z., Salakhutdinov, R., et al. Manifold preserving guided diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

[28] Hepburn, C. A. and Montana, G. Model-based trajectory stitching for improved offline reinforcement learning. *arXiv preprint arXiv:2211.11603*, 2022.

[29] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[30] Jackson, M. T., Matthews, M. T., Lu, C., Ellis, B., Whiteson, S., and Foerster, J. Policy-guided diffusion. *arXiv preprint arXiv:2404.06356*, 2024.

[31] Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022.

[32] Jeon, H., Min, C., and Park, J. Tree-guided diffusion planner. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[33] Ki, D., Oh, J., Shim, S.-W., and Lee, B.-J. Prior-guided diffusion planning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[34] Kim, C., Lee, H., Seo, Y., Lee, K., and Zhu, Y. DEAS: DEtached value learning with action sequence for scalable offline RL. In *International Conference on Learning Representations (ICLR)*, 2026.

[35] Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[36] Lee, J., Yun, S., Yun, T., and Park, J. Gta: Generative trajectory augmentation with guidance for offline reinforcement learning. *arXiv preprint arXiv:2405.16907*, 2024.

[37] Lee, K. and Choi, J. Local manifold approximation and projection for manifold-aware diffusion planning. In *International Conference on Machine Learning (ICML)*, 2025.

[38] Lee, K. and Choi, J. State-covering trajectory stitching for diffusion planners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[39] Lee, K., Kim, S.-A., Choi, J., and Lee, S.-W. Deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling. In *International Conference on Machine Learning (ICML)*, 2018.

[40] Lee, K., Kim, S., and Choi, J. Refining diffusion planner for reliable behavior synthesis by automatic detection of infeasible plans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[41] Li, G., Shan, Y., Zhu, Z., Long, T., and Zhang, W. Diffstitch: Boosting offline reinforcement learning with diffusion-based trajectory stitching. In *International Conference on Machine Learning (ICML)*, 2024.

[42] Li, Q., Park, S., and Levine, S. Decoupled q-chunking. In *International Conference on Learning Representations (ICLR)*, 2026.

[43] Li, W., Wang, X., Jin, B., and Zha, H. Hierarchical diffusion for offline decision making. In *International Conference on Machine Learning (ICML)*, 2023.

[44] Liang, J., Christopher, J. K., Koenig, S., and Fioretto, F. Simultaneous multi-robot motion planning with projected diffusion models. In *International Conference on Machine Learning (ICML)*, 2025.

[45] Liang, Z., Mu, Y., Ding, M., Ni, F., Tomizuka, M., and Luo, P. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. In *International Conference on Machine Learning (ICML)*, 2023.

[46] Lu, H., Han, D., Shen, Y., and Li, D. What makes a good diffusion planner for decision making? In *International Conference on Learning Representations (ICLR)*, 2025.

[47] Lu, Y., Han, D., Wang, Y., and Li, D. Improving diffusion planners by self-supervised action gating with energies. *arXiv preprint arXiv:2603.02650*, 2026.

[48] Luo, Y., Sun, C., Tenenbaum, J. B., and Du, Y. Potential based diffusion motion planning. In *International Conference on Machine Learning (ICML)*, 2024.

[49] Luo, Y., Mishra, U. A., Du, Y., and Xu, D. Generative trajectory stitching through diffusion composition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[50] Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. Learning latent plans from play. In *Conference on robot learning*, 2020.

[51] Mishra, U. A., Xue, S., Chen, Y., and Xu, D. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pp. 2905–2925. PMLR, 2023.

[52] Mishra, U. A., He, D., Chen, Y., and Xu, D. Compositional diffusion with guided search for long-horizon planning. In *International Conference on Learning Representations (ICLR)*, 2026.

[53] Nandiraju, G., Ju, Y., Xu, C., and Wang, H. Hdflow: Hierarchical diffusion-flow planning for long-horizon robotic assembly. In *NeurIPS 2025 Workshop on Embodied World Models for Decision Making*, 2025.

[54] Opryshko, E., Quan, J., Voelcker, C., Du, Y., and Gilitschenski, I. Test-time graph search for goal-conditioned reinforcement learning. *arXiv preprint arXiv:2510.07257*, 2025.

[55] Park, K., Park, S., Lee, Y., and Levine, S. Scalable offline model-based RL with action chunks. In *International Conference on Learning Representations (ICLR)*, 2026.

[56] Park, S., Ghosh, D., Eysenbach, B., and Levine, S. Offline goal-conditioned rl with latent states as actions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[57] Park, S., Kreiman, T., and Levine, S. Foundation policies with hilbert representations. In *International Conference on Machine Learning (ICML)*, 2024.

[58] Park, S., Frans, K., Eysenbach, B., and Levine, S. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

[59] Park, S., Frans, K., Mann, D., Eysenbach, B., Kumar, A., and Levine, S. Horizon reduction makes rl scalable. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[60] Park, S., Li, Q., and Levine, S. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025.

[61] Park, S., Mann, D., and Levine, S. Dual goal representations. In *International Conference on Learning Representations (ICLR)*, 2026.

[62] Park, S., Oberai, A., Atreya, P., and Levine, S. Transitive rl: Value learning via divide and conquer. In *International Conference on Learning Representations (ICLR)*, 2026.

[63] Ren, Y., Gao, W., Ying, L., Rotskoff, G. M., and Han, J. Driftlite: Lightweight drift control for inference-time scaling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2026.

[64] Robbins, H. E. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pp. 388–394. Springer, 1992.

[65] Shaoul, Y., Mishani, I., Vats, S., Li, J., and Likhachev, M. Multi-robot motion planning with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

[66] Shen, Y., Jiang, X., Yang, Y., Wang, Y., Han, D., and Li, D. Understanding and improving training-free loss-based diffusion guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[67] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[68] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[69] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[70] Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.-Y., Kautz, J., Chen, Y., and Vahdat, A. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning (ICML)*, 2023.

[71] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

[72] Tassa, Y., Erez, T., and Todorov, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4906–4913, 2012.

[73] Wang, T., Torralba, A., Isola, P., and Zhang, A. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning (ICML)*, 2023.

[74] Wang, Y., Wang, L., Du, Y., Sundaralingam, B., Yang, X., Chao, Y.-W., Pérez-D'Arpino, C., Fox, D., and Shah, J. Inference-time policy steering through human interactions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[75] Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[76] Xie, A., Rybkin, O., Sadigh, D., and Finn, C. Latent diffusion planning for imitation learning. In *International Conference on Machine Learning (ICML)*, 2025.

[77] Yang, L., Ding, S., Cai, Y., Yu, J., Wang, J., and Shi, Y. Guidance with spherical gaussian constraint for conditional diffusion. In *International Conference on Machine Learning (ICML)*, 2024.

[78] Yedidia, J. S., Freeman, W. T., and Weiss, Y. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, 51(7): 2282–2312, 2005.

[79] Yoon, J., Cho, H., Baek, D., Bengio, Y., and Ahn, S. Monte carlo tree diffusion for system 2 planning. In *International Conference on Machine Learning (ICML)*, 2025.

[80] Yoon, J., Cho, H., Bengio, Y., and Ahn, S. Fast monte carlo tree diffusion: 100x speedup via parallel sparse planning. *arXiv preprint arXiv:2506.09498*, 2025.

[81] Zhang, Q., Song, J., Huang, X., Chen, Y., and Liu, M.-Y. Diffcollage: Parallel generation of large content with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10188–10198. IEEE, 2023.

[82] Zhang, Y., Luo, Y., Mishra, U. A., Shin, W. C., Chen, Y., and Xu, D. Compositional visual planning via inference-time diffusion scaling. In *International Conference on Learning Representations (ICLR)*, 2026.

[83] Zheng, Y., Liang, R., Zheng, K., Zheng, J., Mao, L., Li, J., Gu, W., Ai, R., Li, S. E., Zhan, X., et al. Diffusion-based planning for autonomous driving with flexible guidance. In *International Conference on Learning Representations (ICLR)*, 2025.

[84] Zhou, Z., Zhu, C., Zhou, R., Cui, Q., Gupta, A., and Du, S. S. Free from bellman completeness: Trajectory stitching via model-based return-conditioned supervised learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[85] Zhu, Z., Liu, M., Mao, L., Kang, B., Xu, M., Yu, Y., Ermon, S., and Zhang, W. Madiff: Offline multi-agent learning with diffusion models. *arXiv preprint arXiv:2305.17330*, 2023.

Appendix

Table of Contents

Appendix A. Notation 16
Appendix B. Extended Related Work 16
Appendix C. Training-free Diffusion Guidance 18
Appendix D. Proofs 18
D.1. Proof of Proposition 1 18
D.2. Proof of Proposition 2 18
D.3. Proof of Proposition 3 19
D.4. Decomposition into Local Reconstruction Errors 19
Appendix E. Algorithm 20
Appendix F. Implementation Details 21
F.1. Environments and Datasets 21
F.2. Diffusion Model Architecture and Training 21
F.3. Hyperparameters 22
F.4. Evaluation Protocol 22
F.5. Compute Resources 23
F.6. Low-Level Controllers 23
Appendix G. Additional Results 23
G.1. Additional Plan Quality Visualizations 23
G.2. Additional Ablation Studies 23
G.3. Reconstruction Error and Plan Feasibility 25
G.4. RCD with Unconditional Score Composition 25
G.5. Composition with Out-of-Distribution Constraints 26
G.6. Failure Analysis 26
Appendix H. Baseline Result Sources 27

A Notation

Table 7: **Table of notation.**

Notation	Description
Factor Graph & Plan Structure	
$\boldsymbol{\tau} = (x_1, \ldots, x_N) \in \mathbb{R}^{N \times D}$	Global plan consisting of N variable nodes
$x_i \in \mathbb{R}^D$	Variable node i (state or state-action pair)
D	Per-variable dimension (state or state-action)
y_j	Factor node j (local segment, contiguous subsequence of $\boldsymbol{\tau}$)
M	Number of factors (local segments)
N	Number of variable nodes
$y_j \cap y_{j+1}$	Overlap region between adjacent segments j and $j{+}1$
d_i	Degree of variable x_i ($d_i = 2$ for overlap, $d_i = 1$ otherwise)
Diffusion Process	
$\boldsymbol{\tau}^{(t)}$, $y_j^{(t)}$	Noisy sample at diffusion timestep t
$\boldsymbol{\tau}^{(0)}$, $y_j^{(0)}$	Clean sample
α_t	Cumulative noise schedule (DDPM)
$\boldsymbol{\epsilon}$	Gaussian noise
T	Total number of diffusion timesteps
$\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)$	Noise prediction network; neighbor conditioning implicit (see Section 2.2)
$\mathbf{s}_\theta(\cdot, t)$	Score function $\mathbf{s}_\theta := -\boldsymbol{\epsilon}_\theta / \sqrt{1-\alpha_t}$ (see Section 2.3)
$\bar{\boldsymbol{\epsilon}}_\theta$	Composed noise prediction (averaging in overlaps)
$\hat{\boldsymbol{\tau}}_0^{(t)}$, $\hat{y}_{j,0}^{(t)}$	Tweedie denoised estimate
$\hat{\boldsymbol{\tau}}_0^{\mathrm{rec}}$	Composed Tweedie reconstruction
$\boldsymbol{\mu}_\theta$, σ_t	Predicted mean and standard deviation of reverse process
RCD-Specific	
$\mathcal{E}_{\mathrm{recon}}(\hat{\boldsymbol{\tau}}_0;\, s)$	Self-reconstruction error at probe level s
$\mathcal{E}_{\mathrm{ov}}(\hat{y}_{1:M})$	Overlap consistency
$\mathcal{E}_{\mathrm{RCD}}$	Combined RCD guidance objective
s	Probe timestep for self-reconstruction
w	Guidance weight
λ_{ov}	Weight for overlap consistency
$\tilde{p}(\boldsymbol{\tau})$	RCD-guided (tilted) distribution
δ	Numerical stability constant for gradient normalization

B Extended Related Work

Diffusion and Flow Models for Offline Decision Making. Beyond trajectory planning, diffusion and flow models have been widely adopted as policy and value classes in offline RL and imitation learning. Diffusion Policy [10] and Latent Diffusion Planning [76] train diffusion- (or latent-diffusion-) based visuomotor policies, and Ada-Diffuser [21] extends this line with a causal diffusion framework that identifies latent dynamics from short temporal observation blocks and applies to both planning and policy learning. Diffusion Q-Learning [75] and more recent flow-matching-based variants such as FQL [60], DEAS [34], MAC [55], and DQC [42] combine diffusion/flow policy classes with Q-learning. floq [1] instead applies flow matching on the critic side. On a different axis, training-time data augmentations such as Proximal Action Replacement [14] progressively replace low-value actions in the offline buffer with high-value ones from a stable actor to lift the performance ceiling of behavior-cloning-regularized actor-critic methods. A complementary line treats guidance itself as a policy-improvement step [23, 30], including Prior-Guided Diffusion Planning [33], which replaces the standard Gaussian noise prior of a behavior-cloned diffusion planner with a state-conditioned learnable prior concentrated on high-value trajectories. Sequence-level diffusion models have also been extended along the time axis: Diffusion Forcing [7] unifies next-token prediction with full-

sequence denoising, FloodDiffusion [5] streams motion generation online, and DiffuserLite [15] employs a coarse-to-fine planning refinement process to accelerate inference. Structured variants of diffusion planners include Hierarchical Diffusion for Offline Decision Making [43], which uses a two-level trajectory-level diffusion with classifier-free guidance, and Potential-Based Diffusion Motion Planning [48], which parametrizes motion-planning potentials with diffusion models and composes constraints by summing the learned potentials, and HDFlow [53], which couples a high-level diffusion planner over latent subgoals with a low-level rectified-flow trajectory generator for long-horizon robotic assembly. Diffusion planners have further been deployed across domains: Diffusion-Based Planning for Autonomous Driving [83] brings related ideas to driving-policy generation, and multi-robot motion planning methods [65, 44] compose single-robot diffusion planners under collision and kinematic constraints. An empirical study of design principles by Lu et al. [46] systematically evaluates over $6{,}000$ diffusion-planner configurations for offline RL, Clark & Shkurti [12] identifies shift equivariance and local receptive fields as the architectural ingredients that enable diffusion planners to compose diverse trajectories from short BC-trained segments, and CleanDiffuser [16] provides a modular library that unifies many of the above components. A broader set of inference-time methods such as Inference-Time Policy Steering [74] and DriftLite [63] modify a pretrained diffusion model at test time without additional training, but target settings orthogonal to compositional long-horizon planning.

Offline Goal-Conditioned RL and Horizon Reduction. Offline goal-conditioned RL [24, 73, 19, 35, 56, 58] studies how to train a policy that can reach any target state from any starting state using only a fixed, pre-collected dataset. Representation-learning approaches such as HILP [57], Dual Goal Representations [61], Transitive RL [62], and Temporal Difference Flows [20] learn geometry-aware state embeddings tailored for long-horizon value estimation or temporal-distance prediction. Horizon-reduction strategies explicitly shrink the effective planning horizon and shift reasoning to a hierarchical or graph-based representation: SHARSA [59], option-aware temporal abstraction [2], Graph-Assisted Stitching [4], hierarchical RL with factored subgoal diffusion [26], and Test-Time Graph Search [54], which builds a weighted graph over offline dataset states and assembles a subgoal sequence at inference time for a frozen goal-conditioned policy, all trade long reasoning chains for shorter, compositional segments. Along the data-centric axis, trajectory-stitching methods synthesize longer trajectories *offline*, either via model-based return-conditioned supervised learning [84] or generative augmentation [41, 30, 36, 38, 9], after which a downstream policy is retrained on the enriched dataset.

Inference-Time Search and Plan Refinement for Diffusion Planners. A recent line of work explores explicit search or population-based selection over diffusion-generated plans. Monte Carlo Tree Diffusion [79] and its faster variant [80] perform MCTS over denoising trajectories, and Tree-Guided Diffusion Planner [32] balances exploration and exploitation through a bi-level tree-search sampling procedure. Within the compositional diffusion setting, CDGS [52] augments segment-level score composition with population-based guided search and plan pruning. Plan-refinement methods instead detect or project infeasible trajectories: RGG [40] trains a time-dependent classifier to flag infeasible plans, LoMAP [37] projects intermediate samples onto a local approximation of the data manifold, TAT [22] aggregates multiple diffusion-generated trajectories into a dynamic tree and filters unreliable predictions via weighted majority voting over shared states, SAGE [47] re-ranks sampled plans at inference time using a JEPA-based latent prediction error as a feasibility energy combined with value estimates, and AdaptDiffuser [45] generates synthetic trajectories via reward-gradient guidance, filters them with a discriminator, and fine-tunes the diffusion planner on the filtered high-quality samples in a self-evolving loop. In contrast, RCD uses the pretrained diffusion model's own self-reconstruction error as an intrinsic density proxy, directly at the composed-trajectory level where mode-averaging originates, and requires no auxiliary classifier, manifold estimator, or search tree. Compositional Visual Planning [82] addresses long-horizon video planning by enforcing boundary residual constraints across overlapping chunks via message passing. RCD instead introduces a self-reconstruction density signal with a formal ELBO connection that explicitly targets mode-averaging, and applies gradient guidance on a single composed trajectory rather than residual-based message passing.

C Training-free Diffusion Guidance

Training-free guidance methods steer diffusion sampling toward a target condition y without retraining the model. Starting from classifier guidance [13], which modifies the score via $\nabla_{\boldsymbol{\tau}^{(t)}} \log p(\boldsymbol{\tau}^{(t)}|y) = \nabla_{\boldsymbol{\tau}^{(t)}} \log p(\boldsymbol{\tau}^{(t)}) + \nabla_{\boldsymbol{\tau}^{(t)}} \log p(y|\boldsymbol{\tau}^{(t)})$, recent methods avoid the need for a time-dependent classifier by operating directly on the Tweedie estimate. Given a differentiable loss function $\mathcal{L}(\hat{\boldsymbol{\tau}}_0^{(t)})$ defined on the clean-data support, the likelihood term is approximated as [11, 70]:

$$\nabla_{\boldsymbol{\tau}^{(t)}} \log p(y|\boldsymbol{\tau}^{(t)}) \approx -\gamma \, \nabla_{\boldsymbol{\tau}^{(t)}} \mathcal{L}\big(\hat{\boldsymbol{\tau}}_0^{(t)}\big), \tag{13}$$

where $\gamma > 0$ is a guidance scale and the gradient is taken with respect to $\boldsymbol{\tau}^{(t)}$ through the Tweedie estimate. This approach requires only a pretrained diffusion model and a differentiable objective, making it applicable to a wide range of tasks without additional training [77, 27].

D Proofs

D.1 Proof of Proposition 1 (Reconstruction Error as Density Proxy)

We provide the full derivation of the ELBO connection.

Per-segment ELBO. For a single segment y_j with clean sample $y_j^{(0)}$, the per-segment reconstruction error at probe timestep s is:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\|y_j^{(0)} - \hat{y}_{j,0}^{(s)}\|^2\Big] &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\left\| y_j^{(0)} - \frac{\sqrt{\alpha_s}\, y_j^{(0)} + \sqrt{1-\alpha_s}\, \boldsymbol{\epsilon} - \sqrt{1-\alpha_s}\, \boldsymbol{\epsilon}_\theta(y_j^{(s)}, s)}{\sqrt{\alpha_s}} \right\|^2\right] \\ &= \frac{1-\alpha_s}{\alpha_s}\, \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(y_j^{(s)}, s)\|^2\Big]. \end{aligned} \tag{14}$$

Multiplying both sides by $\alpha_s/(1-\alpha_s)$ and summing over $s = 1, \ldots, T$:

$$\sum_{s=1}^{T} \frac{\alpha_s}{1-\alpha_s}\, \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\|y_j^{(0)} - \hat{y}_{j,0}^{(s)}\|^2\Big] = \sum_{s=1}^{T} \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(y_j^{(s)}, s)\|^2\Big]. \tag{15}$$

The right-hand side is the negative ELBO for the local diffusion model [29], providing an upper bound on $-\log p_\theta(y_j^{(0)})$.

Composed trajectory. Let $\bar{\boldsymbol{\epsilon}}_\theta(\hat{\boldsymbol{\tau}}_s, s)$ denote the *composed noise prediction*: equal to $\boldsymbol{\epsilon}_\theta^{(j)}$ on non-overlap variables $x_i \in y_j$ and to $\frac{1}{2}\big(\boldsymbol{\epsilon}_\theta^{(k)} + \boldsymbol{\epsilon}_\theta^{(k+1)}\big)$ on overlap variables $x_i \in y_k \cap y_{k+1}$. By the definition of the composed Tweedie reconstruction $\hat{\boldsymbol{\tau}}_0^{\mathrm{rec}}$ in Definition 1, a direct algebraic manipulation with $\hat{\boldsymbol{\tau}}_s = \sqrt{\alpha_s}\, \hat{\boldsymbol{\tau}}_0 + \sqrt{1-\alpha_s}\, \boldsymbol{\epsilon}$ yields

$$\hat{\boldsymbol{\tau}}_0 - \hat{\boldsymbol{\tau}}_0^{\mathrm{rec}} = -\sqrt{\tfrac{1-\alpha_s}{\alpha_s}}\big(\boldsymbol{\epsilon} - \bar{\boldsymbol{\epsilon}}_\theta(\hat{\boldsymbol{\tau}}_s, s)\big). \tag{16}$$

Squaring and taking the expectation over $\boldsymbol{\epsilon}$ gives

$$\frac{\alpha_s}{1-\alpha_s}\, \mathcal{E}_{\mathrm{recon}}(\hat{\boldsymbol{\tau}}_0;\, s) = \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon} - \bar{\boldsymbol{\epsilon}}_\theta\big(\sqrt{\alpha_s}\, \hat{\boldsymbol{\tau}}_0 + \sqrt{1-\alpha_s}\, \boldsymbol{\epsilon},\, s\big)\big\|^2\Big]. \tag{17}$$

Summing over $s = 1, \ldots, T$, the right-hand side is the DDPM simple loss L_{simple} for the composed denoiser $\bar{\boldsymbol{\epsilon}}_\theta$. By the standard variational argument [29], L_{simple} upper-bounds $-\log p_\theta(\hat{\boldsymbol{\tau}}_0)$ up to a timestep re-weighting, where p_θ is the distribution implicitly defined by the composed denoiser $\bar{\boldsymbol{\epsilon}}_\theta$. This establishes the inequality in Equation (7). □

D.2 Proof of Proposition 2 (Overlap Consistency Measures Score Disagreement)

Both segments k and $k{+}1$ receive the same noisy trajectory $\hat{\boldsymbol{\tau}}_s$ in the overlap region $y_k \cap y_{k+1}$. Let $z := \hat{\boldsymbol{\tau}}_s\big|_{y_k \cap y_{k+1}}$ be the common noisy overlap. The Tweedie estimates from each segment in the

overlap are:

$$\hat{y}_{k,0}\big|_{y_k \cap y_{k+1}} = \frac{z - \sqrt{1-\alpha_s}\,\boldsymbol{\epsilon}_\theta^{(k)}(y_k^{(s)}, s)}{\sqrt{\alpha_s}}, \tag{18}$$

$$\hat{y}_{k+1,0}\big|_{y_k \cap y_{k+1}} = \frac{z - \sqrt{1-\alpha_s}\,\boldsymbol{\epsilon}_\theta^{(k+1)}(y_{k+1}^{(s)}, s)}{\sqrt{\alpha_s}}, \tag{19}$$

where $\boldsymbol{\epsilon}_\theta^{(j)}(y_j^{(s)}, s)$ is the noise prediction from segment j evaluated on its full noisy input $y_j^{(s)}$. Taking the difference:

$$\hat{y}_{k,0}\big|_{y_k \cap y_{k+1}} - \hat{y}_{k+1,0}\big|_{y_k \cap y_{k+1}} = \frac{\sqrt{1-\alpha_s}}{\sqrt{\alpha_s}} \left(\boldsymbol{\epsilon}_\theta^{(k+1)} - \boldsymbol{\epsilon}_\theta^{(k)}\right)\big|_{y_k \cap y_{k+1}}. \tag{20}$$

Squaring yields:

$$\|\hat{y}_{k,0} - \hat{y}_{k+1,0}\|^2\big|_{y_k \cap y_{k+1}} = \frac{1-\alpha_s}{\alpha_s} \|\boldsymbol{\epsilon}_\theta^{(k+1)} - \boldsymbol{\epsilon}_\theta^{(k)}\|^2\big|_{y_k \cap y_{k+1}}. \tag{21}$$

Substituting the score identity $\boldsymbol{\epsilon}_\theta^{(j)} = -\sqrt{1-\alpha_s}\,\mathbf{s}_\theta^{(j)}$ from Section 2.3 yields Equation (9). □

Connection to marginal consistency. In the Bethe approximation, marginal consistency requires $p_k(x_i) = p_{k+1}(x_i)$ for all $x_i \in y_k \cap y_{k+1}$. Taking gradients: $\nabla_{x_i} \log p_k(x_i) = \nabla_{x_i} \log p_{k+1}(x_i)$, i.e., scores from both segments must agree at overlap variables. $\mathcal{E}_{\text{ov}}$ is a relaxation of this hard constraint, penalizing violations via the squared score difference.

D.3 Proof of Proposition 3 (Guided Distribution)

Proposition 3 (Guided Distribution)**.** *Let $p_\theta(\boldsymbol{\tau})$ denote the composed distribution obtained via score averaging. The RCD-guided reverse process targets the modified distribution*

$$\tilde{p}(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) \cdot \exp\big(-w \cdot \mathcal{E}_{\text{RCD}}(\boldsymbol{\tau})\big), \tag{22}$$

which up-weights high-density, globally coherent plans and down-weights mode-averaged plans.

Proof. Consider the reverse-time SDE for the composed distribution $p_\theta(\boldsymbol{\tau})$ [71]:

$$\mathrm{d}\boldsymbol{\tau} = \left[f(\boldsymbol{\tau}, t) - g(t)^2 \nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau})\right] \mathrm{d}t + g(t)\,\mathrm{d}\bar{\boldsymbol{w}}, \tag{23}$$

where f and g are the drift and diffusion coefficients and $\bar{\boldsymbol{w}}$ is a reverse-time Wiener process.

The RCD guidance modifies the score by adding $-w\nabla_{\boldsymbol{\tau}}\mathcal{E}_{\text{RCD}}(\boldsymbol{\tau})$:

$$\mathrm{d}\boldsymbol{\tau} = \left[f(\boldsymbol{\tau}, t) - g(t)^2 \big(\nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}) - w\nabla_{\boldsymbol{\tau}}\mathcal{E}_{\text{RCD}}(\boldsymbol{\tau})\big)\right] \mathrm{d}t + g(t)\,\mathrm{d}\bar{\boldsymbol{w}}. \tag{24}$$

This is the reverse SDE for the modified distribution:

$$\tilde{p}_t(\boldsymbol{\tau}) \propto p_t(\boldsymbol{\tau}) \cdot \exp\big(-w \cdot \mathcal{E}_{\text{RCD}}(\boldsymbol{\tau})\big), \tag{25}$$

since $\nabla_{\boldsymbol{\tau}} \log \tilde{p}_t(\boldsymbol{\tau}) = \nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}) - w\nabla_{\boldsymbol{\tau}}\mathcal{E}_{\text{RCD}}(\boldsymbol{\tau})$.

At $t = 0$, the marginal is $\tilde{p}(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) \cdot \exp(-w \cdot \mathcal{E}_{\text{RCD}}(\boldsymbol{\tau}))$. Since $\mathcal{E}_{\text{RCD}} \geq 0$ with equality for perfectly reconstructed, globally coherent plans, the exponential factor $\exp(-w\,\mathcal{E}_{\text{RCD}})$ up-weights plans with low reconstruction error and high overlap consistency, while down-weighting mode-averaged plans. □

D.4 Decomposition into Local Reconstruction Errors

The reconstruction and overlap terms are complementary because averaging can hide opposing reconstruction errors at segment boundaries. We make this relation explicit for the chain structure in Section 2.2, where each variable belongs to at most two segments. Fix a candidate trajectory $\hat{\boldsymbol{\tau}}_0$ and a probe timestep s. For a noise sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, let $r_{\text{recon}}(\boldsymbol{\epsilon}) = \|\hat{\boldsymbol{\tau}}_0 - \hat{\boldsymbol{\tau}}_0^{\text{rec}}\|^2$ and let $r_{\text{ov}}(\boldsymbol{\epsilon})$ be the overlap term in Equation (8) for the same reconstruction. For a segment j, define

$$r_j(\boldsymbol{\epsilon}) = \sum_{i : x_i \in y_j} \frac{1}{d_i} \big\|\hat{\boldsymbol{\tau}}_0\big|_{x_i} - \hat{y}_{j,0}\big|_{x_i}\big\|^2, \tag{26}$$

where $d_i \in \{1, 2\}$ is the degree of variable x_i and $\cdot\big|_{x_i}$ denotes restriction to the coordinates of the global variable x_i. Thus overlap coordinates receive weight $1/2$ in each segment. For a shared variable with target x and two local reconstructions a and b,

$$\frac{1}{2}\|x-a\|^2 + \frac{1}{2}\|x-b\|^2 = \left\|x - \frac{a+b}{2}\right\|^2 + \frac{1}{4}\|a-b\|^2. \tag{27}$$

Summing over shared and non-shared coordinates gives the samplewise identity

$$r_{\text{recon}}(\boldsymbol{\epsilon}) + \frac{M-1}{4}\, r_{\text{ov}}(\boldsymbol{\epsilon}) = \sum_{j=1}^{M} r_j(\boldsymbol{\epsilon}). \tag{28}$$

Taking the expectation over $\boldsymbol{\epsilon}$ and using Equation (6) gives

$$\mathcal{E}_{\text{recon}}(\hat{\boldsymbol{\tau}}_0;\, s) + \frac{M-1}{4}\, \mathbb{E}_{\boldsymbol{\epsilon}}[r_{\text{ov}}(\boldsymbol{\epsilon})] = \sum_{j=1}^{M} \mathbb{E}_{\boldsymbol{\epsilon}}[r_j(\boldsymbol{\epsilon})]. \tag{29}$$

The factor $M-1$ follows from the normalization of the overlap term in Equation (8). This decomposition shows how overlap consistency accounts for local reconstruction errors that cancel in the merged estimate, but does not establish calibrated density under the global trajectory distribution.

E Algorithm

Each RCD guidance step requires one additional forward pass through the score network (to compute the reconstruction at probe level s) and one backward pass for the gradient. The expectation in Equation (6) is approximated with a single Monte Carlo noise sample. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Refining Compositional Diffusion

Require: Shared diffusion model $\boldsymbol{\epsilon}_\theta$, guidance weight w, overlap weight λ_{ov}, probe level s
Observe current state $\boldsymbol{s}$; initialize $\boldsymbol{\tau}^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  `// compositional denoising via score averaging over segments` $\{y_j\}$
  $\bar{\boldsymbol{\epsilon}}_\theta(\boldsymbol{\tau}^{(t)}, t) \leftarrow$ average per-segment predictions $\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)$, sharing in overlaps
  $\hat{\boldsymbol{\tau}}_0^{(t)} \leftarrow \big(\boldsymbol{\tau}^{(t)} - \sqrt{1-\alpha_t}\, \bar{\boldsymbol{\epsilon}}_\theta(\boldsymbol{\tau}^{(t)}, t)\big)/\sqrt{\alpha_t}$ `// Tweedie estimate`
  `// self-reconstruction: perturb and reconstruct`
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\hat{\boldsymbol{\tau}}_s \leftarrow \sqrt{\alpha_s}\, \hat{\boldsymbol{\tau}}_0^{(t)} + \sqrt{1-\alpha_s}\, \boldsymbol{\epsilon}$
  for each segment $j = 1, \ldots, M$ **do**
    $\hat{y}_{j,0} \leftarrow \big(\hat{y}_{j,s} - \sqrt{1-\alpha_s}\, \boldsymbol{\epsilon}_\theta(\hat{y}_{j,s}, s)\big)/\sqrt{\alpha_s}$ `// per-segment Tweedie`
  end for
  `// compose per-segment estimates via overlap averaging (Eq. 3)`
  $\hat{\boldsymbol{\tau}}_0^{\text{rec}} \leftarrow$ merge $\{\hat{y}_{j,0}\}_{j=1}^{M}$, averaging in overlap regions
  `// RCD guidance objective (Eq. 10)`
  $\mathcal{E}_{\text{recon}} \leftarrow \|\hat{\boldsymbol{\tau}}_0^{(t)} - \hat{\boldsymbol{\tau}}_0^{\text{rec}}\|^2$ `// self-reconstruction error`
  $\mathcal{E}_{\text{ov}} \leftarrow \frac{1}{M-1}\sum_{k=1}^{M-1} \|\hat{y}_{k,0} - \hat{y}_{k+1,0}\|^2\big|_{y_k \cap y_{k+1}}$ `// overlap consistency`
  $\mathbf{g}^{(t)} \leftarrow \nabla_{\boldsymbol{\tau}^{(t)}}\big(\mathcal{E}_{\text{recon}} + \lambda_{\text{ov}}\, \mathcal{E}_{\text{ov}}\big)$
  $\tilde{\mathbf{g}}^{(t)} \leftarrow \mathbf{g}^{(t)}/\|\mathbf{g}^{(t)}\|_\infty$ `// normalized guidance`
  `// reverse step with guidance (Eq. 12)`
  $\boldsymbol{\tau}^{(t-1)} \leftarrow \boldsymbol{\mu}_\theta(\boldsymbol{\tau}^{(t)}, t) + \sigma_t\, \mathbf{z} - w\, \sigma_t^2\, \tilde{\mathbf{g}}^{(t)}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\boldsymbol{\tau}_{\boldsymbol{s}_0}^{(t-1)} \leftarrow \boldsymbol{s}$ `// constrain start state`
end for
return $\boldsymbol{\tau}^{(0)}$

F Implementation Details

F.1 Environments and Datasets

We evaluate on OGBench [58], which provides diverse long-horizon goal-conditioned tasks with `stitch` and `play` datasets. **Locomotion.** `pointmaze`, `antmaze`, and `humanoid maze` require an agent to navigate from a start position to a goal in mazes of increasing size (`Medium`, `Large`, `Giant`). The `stitch` datasets consist of short, disconnected trajectory segments that do not individually span start-to-goal pairs, requiring methods to compose multiple segments at inference time. **Object Manipulation.** `cube` tasks require a 6-DoF UR5e robot arm to pick and place one to four cubes into target configurations. The `play` datasets contain diverse, unstructured manipulation trajectories. `antsoccer` requires an ant agent to dribble a ball to a goal location, combining locomotion and object interaction. **Visual.** `visual antmaze` provides 64×64 RGB image observations. We pretrain a VAE to encode observations into a 16-dimensional latent space, and all planning and inverse dynamics operate in this latent space.

Table 8 lists all evaluation environments with their corresponding OGBench dataset names and the maximum environment steps per episode.

Table 8: **Evaluation environments and datasets in OGBench.** We follow the evaluation setup of CompDiffuser [49].

Environment	Type	Size	Dataset Name	Env Steps
`pointmaze`	`stitch`	`Medium`	pointmaze-medium-stitch-v0	1000
		`Large`	pointmaze-large-stitch-v0	1000
		`Giant`	pointmaze-giant-stitch-v0	1000
`antmaze`	`stitch`	`Medium`	antmaze-medium-stitch-v0	1000
		`Large`	antmaze-large-stitch-v0	2000
		`Giant`	antmaze-giant-stitch-v0	2000
`humanoid maze`	`stitch`	`Medium`	humanoidmaze-medium-stitch-v0	5000
		`Large`	humanoidmaze-large-stitch-v0	5000
		`Giant`	humanoidmaze-giant-stitch-v0	8000
`antsoccer`	`stitch`	`Arena`	antsoccer-arena-stitch-v0	5000
		`Medium`	antsoccer-medium-stitch-v0	5000
`cube`	`play`	`Single`	cube-single-play-v0	2000
		`Double`	cube-double-play-v0	2000
		`Triple`	cube-triple-play-v0	2000
		`Quadruple`	cube-quadruple-play-v0	2000
`visual antmaze`	`stitch`	`Medium`	antmaze-medium-stitch-v0	2000
		`Large`	antmaze-large-stitch-v0	2000

F.2 Diffusion Model Architecture and Training

Our implementation is built on top of the public CompDiffuser codebase[1] [49], and we use the same pretrained local diffusion model as CompDiffuser for all experiments. For `pointmaze`, `antmaze`, and `humanoid maze`, whose planners operate in the 2D x-y space, the model architecture follows a 1D temporal U-Net with residual blocks, group normalization, and sinusoidal timestep embeddings. For `antsoccer` and `cube`, we use a DiT-based architecture [49] with environment-specific hidden dimensions and patch sizes. Training uses the standard DDPM [29] objective with a linear noise schedule. All models are trained until convergence using the Adam optimizer with learning rate 2×10^{-4}.

[1] `https://github.com/devinluo27/comp_diffuser_release`

F.3 Hyperparameters

Table 9 lists the diffusion model and RCD hyperparameters used across all environments. We use a *single fixed set of RCD hyperparameters* (w=0.25, λ_{ov}=0.5, s/T=0.4) across every benchmark in this paper, including locomotion mazes of three different scales, high-dimensional object manipulation, and visual AntMaze. The hyperparameter ablations in Appendix G.2 confirm that RCD is robust to the choice of these hyperparameters.

Table 9: **Hyperparameters across environments.**

Parameter	**PointMaze**	**AntMaze**	**Humanoid Maze**	**Cube**	**AntSoccer**	**Visual AntMaze**
Architecture	U-Net	U-Net	U-Net	DiT	DiT	U-Net
Segment length H	160	160	336	160	160	160
Overlap size	64	64	128	56	56	64
Diffusion steps T	512	512	1000	512	512	512
Probe ratio s/T	0.40	0.40	0.40	0.40	0.40	0.40
Guidance weight w	0.25	0.25	0.25	0.25	0.25	0.25
Overlap weight λ_{ov}	0.5	0.5	0.5	0.5	0.5	0.5

Table 10: **Number of composed segments M per environment.** For locomotion mazes and `antsoccer`, values follow the same setting as CompDiffuser [49].

Environment	**Type**	**Size**	**Dataset Name**	**# Segments M**
`pointmaze`	`stitch`	`Medium`	pointmaze-medium-stitch-v0	3
		`Large`	pointmaze-large-stitch-v0	6
		`Giant`	pointmaze-giant-stitch-v0	8
`antmaze`	`stitch`	`Medium`	antmaze-medium-stitch-v0	3
		`Large`	antmaze-large-stitch-v0	6
		`Giant`	antmaze-giant-stitch-v0	9
`humanoid maze`	`stitch`	`Medium`	humanoidmaze-medium-stitch-v0	4
		`Large`	humanoidmaze-large-stitch-v0	6
		`Giant`	humanoidmaze-giant-stitch-v0	11
`antsoccer`	`stitch`	`Arena`	antsoccer-arena-stitch-v0	5
		`Medium`	antsoccer-medium-stitch-v0	6
`cube`	`play`	`Single`	cube-single-play-v0	2
		`Double`	cube-double-play-v0	5
		`Triple`	cube-triple-play-v0	10
		`Quadruple`	cube-quadruple-play-v0	10
`visual antmaze`	`stitch`	`Medium`	antmaze-medium-stitch-v0	2
		`Large`	antmaze-large-stitch-v0	5

F.4 Evaluation Protocol

We follow the OGBench evaluation protocol [58]. Each environment specifies 5 test-time start-goal pairs. For each pair, we run 20 episodes and report the binary success rate (reaching within a threshold distance of the goal). Results are averaged over 5 random seeds, and we report mean $\pm$ standard deviation. For replanning experiments, we adopt the adaptive replanning strategy proposed by CompDiffuser [49]: replanning is triggered when the agent deviates from its current subgoal beyond a threshold distance, and a receding scheme progressively reduces the number of composed segments based on how much of the current plan has been executed to encourage faster convergence toward the goal.

F.5 Compute Resources

All experiments, including training of the local diffusion models and all evaluations are conducted on a single NVIDIA H100 GPU.

F.6 Low-Level Controllers

The diffusion planner generates state-space trajectories, and a separate low-level controller converts these into executable actions. **Locomotion and AntSoccer.** Following CompDiffuser [49], we train an MLP-based inverse dynamics model that takes consecutive states (s_t, s_{t+1}) as input and predicts the action a_t. The same model architecture and training procedure from CompDiffuser is used. **Cube Manipulation.** We use a DQL-based value-learning policy [75] that takes the current state and a planned subgoal as input and outputs actions. The DQL policy is trained on the same offline dataset used for the diffusion planner. **Visual AntMaze.** We train an inverse dynamics model in the VAE latent space that maps consecutive latent states (z_t, z_{t+1}) to actions, following a similar architecture to the state-based inverse dynamics but operating on 16-dimensional latent representations.

G Additional Results

G.1 Additional Plan Quality Visualizations

Figure 5 extends the plan quality analysis of Section 4.1 to `PointMaze-Giant-Stitch`, confirming that the improvement in plan feasibility observed on `AntMaze` generalizes across environments. CompDiffuser produces valid rates as low as 10% on the hardest task (Task 3), while CDGS shows improvement on some tasks but remains inconsistent (15–100%). RCD achieves 85–100% valid rates across all 5 tasks, consistent with the `AntMaze-Giant-Stitch` case presented in Section 4.1.

G.2 Additional Ablation Studies

We vary the three RCD-specific hyperparameters one at a time on `PointMaze-Giant-Stitch` and `AntMaze-Giant-Stitch`, fixing the remaining ones at the defaults listed in Table 9. Reported numbers follow the same evaluation protocol as the main results (Appendix F.4). Success rates are averaged over 5 random seeds, each evaluated on 5 start-goal pairs with 20 episodes per pair. The defaults (w=0.25, λ_{ov}=0.5, s/T=0.4) are highlighted with **bold** values.

Ablation on guidance weight w. Table 11 varies the RCD guidance weight $w \in \{0, 0.25, 0.5\}$. Setting w=0 disables RCD guidance entirely and recovers a score close to plain compositional denoising, while both non-zero weights lead to large improvements on both environments. The default w=0.25 attains the best average success rate.

Table 11: **Ablation on guidance weight w.**

Env	Type	Size	Baselines		RCD		
			CD	CDGS	w=0	w=0.25 (default)	w=0.5
`pointmaze`	`stitch`	`Giant`	$69_{\pm 3}$	$74_{\pm 3}$	$59_{\pm 5}$	$\mathbf{100}_{\pm 0}$	$100_{\pm 0}$
`antmaze`	`stitch`	`Giant`	$67_{\pm 3}$	$83_{\pm 3}$	$56_{\pm 4}$	$\mathbf{89}_{\pm 2}$	$87_{\pm 3}$
Average			68.0	78.5	57.5	**94.5**	93.5

Ablation on overlap consistency weight λ_{ov}. Table 12 varies the overlap consistency weight $\lambda_{\text{ov}} \in \{0, 0.25, 0.5\}$. Even with λ_{ov}=0 (reconstruction error only) the guided sampler already outperforms the compositional baseline, but adding the overlap consistency term consistently improves `AntMaze-Giant-Stitch` and yields the best overall success rate at the default λ_{ov}=0.5.

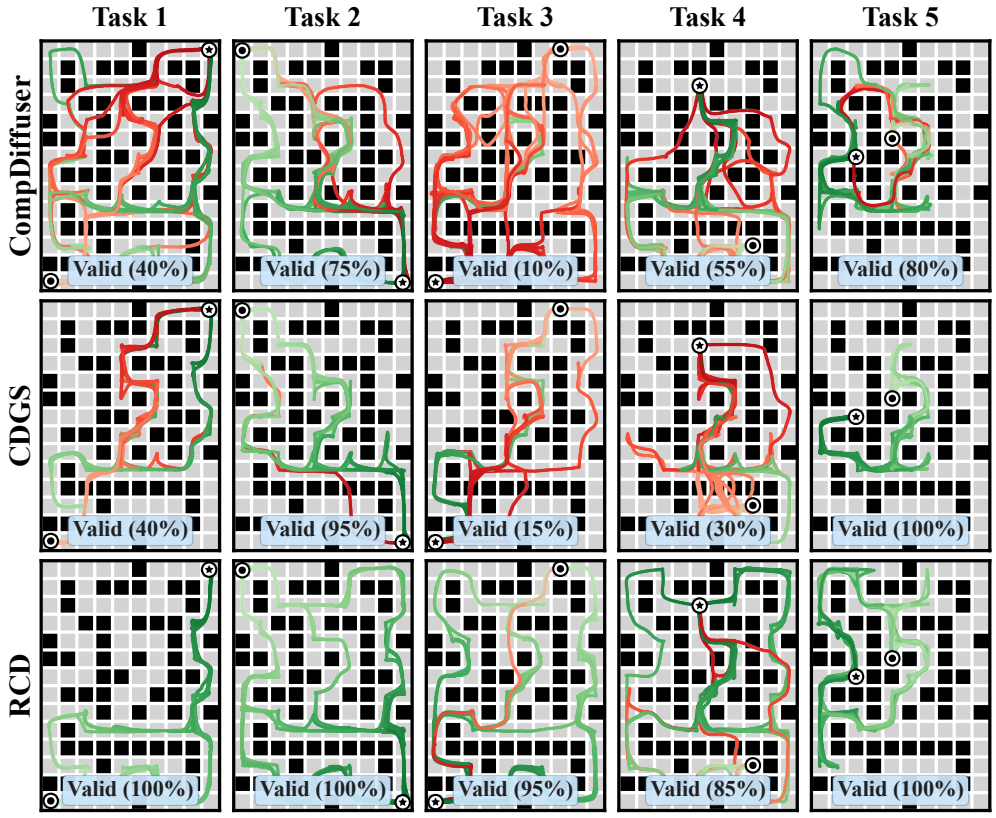


Figure 5: **Plan quality comparisons on** `PointMaze-Giant-Stitch`**.** Each column shows 20 sampled plans from CompDiffuser, CDGS, and RCD for 5 test-time tasks defined in OGBench. Plans that violate environment constraints (wall penetration) are shown in red; feasible plans in green.

Table 12: **Ablation on overlap consistency weight λ_{ov}.**

Env	Type	Size	Baselines		RCD		
			CD	CDGS	λ_{ov}=0	λ_{ov}=0.25	λ_{ov}=0.5 (default)
pointmaze	stitch	Giant	$69_{\pm 3}$	$74_{\pm 3}$	$95_{\pm 2}$	$92_{\pm 3}$	$\mathbf{100}_{\pm 0}$
antmaze	stitch	Giant	$67_{\pm 3}$	$83_{\pm 3}$	$79_{\pm 3}$	$85_{\pm 3}$	$\mathbf{89}_{\pm 2}$
Average			68.0	78.5	87.0	88.5	**94.5**

Ablation on probe level s/T. Table 13 varies the probe level $s/T \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. Performance is stable across a wide range of probe levels, and we adopt s/T=0.4 as the default across all environments.

Table 13: **Ablation on probe level s/T.**

Env	Type	Size	Baselines		RCD					
			CD	CDGS	s/T=0.1	0.2	0.3	0.4 (default)	0.5	0.6
pointmaze	stitch	Giant	$69_{\pm 3}$	$74_{\pm 3}$	$98_{\pm 2}$	$98_{\pm 2}$	$98_{\pm 2}$	$\mathbf{100}_{\pm 0}$	$98_{\pm 2}$	$98_{\pm 2}$
antmaze	stitch	Giant	$67_{\pm 3}$	$83_{\pm 3}$	$81_{\pm 4}$	$86_{\pm 3}$	$84_{\pm 3}$	$\mathbf{89}_{\pm 2}$	$\mathbf{89}_{\pm 2}$	$83_{\pm 3}$
Average			68.0	78.5	89.5	92.0	91.0	**94.5**	93.5	90.5

G.3 Reconstruction Error and Plan Feasibility

We further evaluate whether the RCD guidance objective reflects the physical feasibility of composed plans. We sample 100 plans from unguided CD in each of `AntMaze-Giant-Stitch` and `PointMaze-Giant-Stitch`, and compute the self-reconstruction error $\mathcal{E}_{\text{recon}}$, overlap consistency $\mathcal{E}_{\text{ov}}$, and combined objective $\mathcal{E}_{\text{RCD}}$ for each plan. Following Section 4.1, a plan is valid if it does not pass through maze walls.

Table 14: **RCD guidance components and plan feasibility.** AUROC for detecting wall collisions on 100 unguided CD plans per environment. Colliding plans form the positive class.

Environment	$\mathcal{E}_{\text{recon}}$	$\mathcal{E}_{\text{ov}}$	$\mathcal{E}_{\text{RCD}}$
`antmaze-Giant`	0.83	0.92	0.92
`pointmaze-Giant`	0.76	0.88	0.85

Table 15: **Valid plan rates by RCD objective value.** Plans are sorted by increasing $\mathcal{E}_{\text{RCD}}$ and divided into five groups of 20 per environment. Rates are reported as percentages.

Environment	**Lowest 20%**	**20–40%**	**40–60%**	**60–80%**	**Highest 20%**
`antmaze-Giant`	95	70	25	10	5
`pointmaze-Giant`	95	75	50	25	15

Table 14 reports AUROC for detecting wall collisions. The combined objective achieves 0.92 on `AntMaze-Giant-Stitch` and 0.85 on `PointMaze-Giant-Stitch`, indicating that infeasible plans tend to receive larger objective values. Table 15 further shows that the valid plan rate decreases monotonically across groups of increasing $\mathcal{E}_{\text{RCD}}$. These results support the use of the RCD objective as a signal of plan feasibility. The overlap consistency term achieves higher collision AUROC than the self-reconstruction error in both environments. When used for guidance, however, the self-reconstruction error alone gives higher success rates. In Table 4, using only $\mathcal{E}_{\text{recon}}$ achieves 79% on `AntMaze-Giant-Stitch` and 95% on `PointMaze-Giant-Stitch`, compared with 56% and 59% using only $\mathcal{E}_{\text{ov}}$. Overlap consistency targets agreement at segment boundaries, which can hold even for an infeasible plan, while the self-reconstruction error guides the entire trajectory toward high-density regions. These results are consistent with the complementary roles of the two guidance components.

G.4 RCD with Unconditional Score Composition

We further evaluate RCD with unconditional score composition in the style of GSC [51] and DiffCollage [81]. We compose local denoisers without conditioning on noisy neighboring segments and use the same RCD hyperparameters as in the OGBench experiments (Appendix F.3): $w = 0.25$, $\lambda_{\text{ov}} = 0.5$, and $s/T = 0.4$. Success rates are averaged over 5 random seeds, with 100 evaluation episodes per seed.

Table 16: **RCD with unconditional score composition.** Success rates (mean $\pm$ std) with and without RCD guidance.

Environment	**Size**	**Without RCD**	**With RCD**
`pointmaze`	`Medium`	$100_{\pm 0}$	$100_{\pm 0}$
	`Large`	$100_{\pm 0}$	$100_{\pm 0}$
	`Giant`	$28_{\pm 3}$	$\mathbf{51}_{\pm 3}$
`antmaze`	`Medium`	$95_{\pm 2}$	$94_{\pm 2}$
	`Large`	$70_{\pm 2}$	$\mathbf{90}_{\pm 2}$
	`Giant`	$22_{\pm 1}$	$\mathbf{59}_{\pm 2}$

Table 16 reports success rates with and without RCD guidance. RCD improves success rates by 23 percentage points on `PointMaze-Giant-Stitch`, 20 on `AntMaze-Large-Stitch`, and 37 on

AntMaze-Giant-Stitch. The remaining tasks show little change, with a decrease of one percentage point on AntMaze-Medium-Stitch. These results demonstrate that RCD can improve compositional planning without bidirectional neighbor conditioning.

G.5 Composition with Out-of-Distribution Constraints

The OGBench stitch tasks require composing familiar local segments into globally novel trajectories. We further evaluate RCD under a local distribution shift using the bimodal toy domain in Figure 1, where the training modes are centered at -1 and $+1$.

We clamp one interior variable to 0.0, 1.5, or 2.0 throughout sampling and generate 200 trajectories at horizon 12 for each condition. A sample is valid if all free interior variables lie within three training standard deviations of the same mode. We exclude the clamped variable from this criterion, so the valid rate measures the consistency of the remaining variables. We use 100 denoising steps, guidance weight $w = 5$, overlap consistency weight $\lambda_{\text{ov}} = 1$, probe level $s/T = 0.35$, and four Monte Carlo noise samples per guidance step. These settings are fixed across constraints.

Table 17: **Toy composition with a clamped interior variable.** Valid rates are evaluated on the free interior variables. Reconstruction error is measured on CD samples, averaged over all coordinates and eight probe noise draws.

Clamped value	CD valid rate	RCD valid rate	Reconstruction error (CD)
None	0.27	0.92	0.32
0.0 (between modes)	0.27	0.92	0.34
1.5	0.25	0.84	0.36
2.0	0.23	0.78	0.42

Table 17 shows that RCD achieves higher valid rates than CD under every constraint. At a clamped value of 2.0, RCD retains a valid rate of 0.78, compared with 0.23 for CD. The mean reconstruction error on CD samples increases from 0.32 without an interior constraint to 0.42 at a clamped value of 2.0. These results show that RCD can improve the consistency of the free variables even when one interior variable is constrained away from the training modes.

G.6 Failure Analysis

We analyze failure cases in Visual-AntMaze, Cube, and AntSoccer, focusing on low-level execution and candidate selection in the visual setting.

Low-level execution. To evaluate execution independently of generated plan quality, we provide the low-level controllers (Appendix F.6) with ground-truth trajectory segments. We measure the fraction of segments for which the agent reaches within one spatial unit of the demonstrated endpoint. For state-based and visual AntMaze, the reference paths are approximately 15 units long. For AntSoccer, we evaluate the ball position, using dribbling segments with demonstrated ball displacements of 2–5 units.

Table 18: **Low-level execution of ground-truth trajectories.** Endpoint tracking rates with dataset trajectories as references.

Setting	Endpoint criterion	Rate (%)
State-based AntMaze	Agent within 1 unit	93
Visual AntMaze (16D latent)	Agent within 1 unit	30
AntSoccer	Ball within 1 unit	33

Table 18 reports the endpoint tracking rates. The state-based AntMaze controller achieves 93%, while the visual controller achieves 30% on paths of comparable length. The visual setting uses inverse dynamics in a 16-dimensional VAE latent space, where execution can fail even when the reference trajectory is feasible. On AntSoccer, the controller reaches the demonstrated ball endpoint in only 33% of cases, despite the short reference displacements. Qualitative inspection of Cube

rollouts similarly shows that goal-reaching plans can fail during execution when the controller cannot reach, grasp, or place the target cube. These results indicate that low-level execution limits how improvements in plan quality translate into task success.

Stability of latent-space rankings. We evaluate the stability of candidate rankings under the random noise used to compute the density proxy. For each of 5 batches of 20 candidates, we compute the scores twice with independent noise and measure the fraction of top-2 candidates retained. The retention rate is 40% in the 16-dimensional VAE latent space, compared with 70% in the state space. The lower retention rate indicates that candidate selection is more sensitive to evaluation noise in the visual setting. Since CDGS uses these scores to prune candidates, this sensitivity may contribute to its low success rates in Table 2. RCD uses gradient guidance to refine trajectories throughout denoising. Its gains over CD in Table 2 suggest that the signal remains useful in the latent space, although its ability to reflect physical feasibility depends on the information preserved by the VAE.

H Baseline Result Sources

H.1 Locomotion

We obtain scores for GCBC, GCIVL, GCIQL, QRL, CRL, and HIQL from Table 2 in [58]. Scores for GSC is taken from Tables 1 and 2 in [49]. CDGS and RCD are from our experiments using the same pretrained models as CompDiffuser (CD).

H.2 Object Manipulation

For `antsoccer`, scores for GCBC through HIQL are from Table 2 in [58], and GSC from Table 3 in [49]. For `cube`, GCBC through HIQL are from Table 2 in [58].

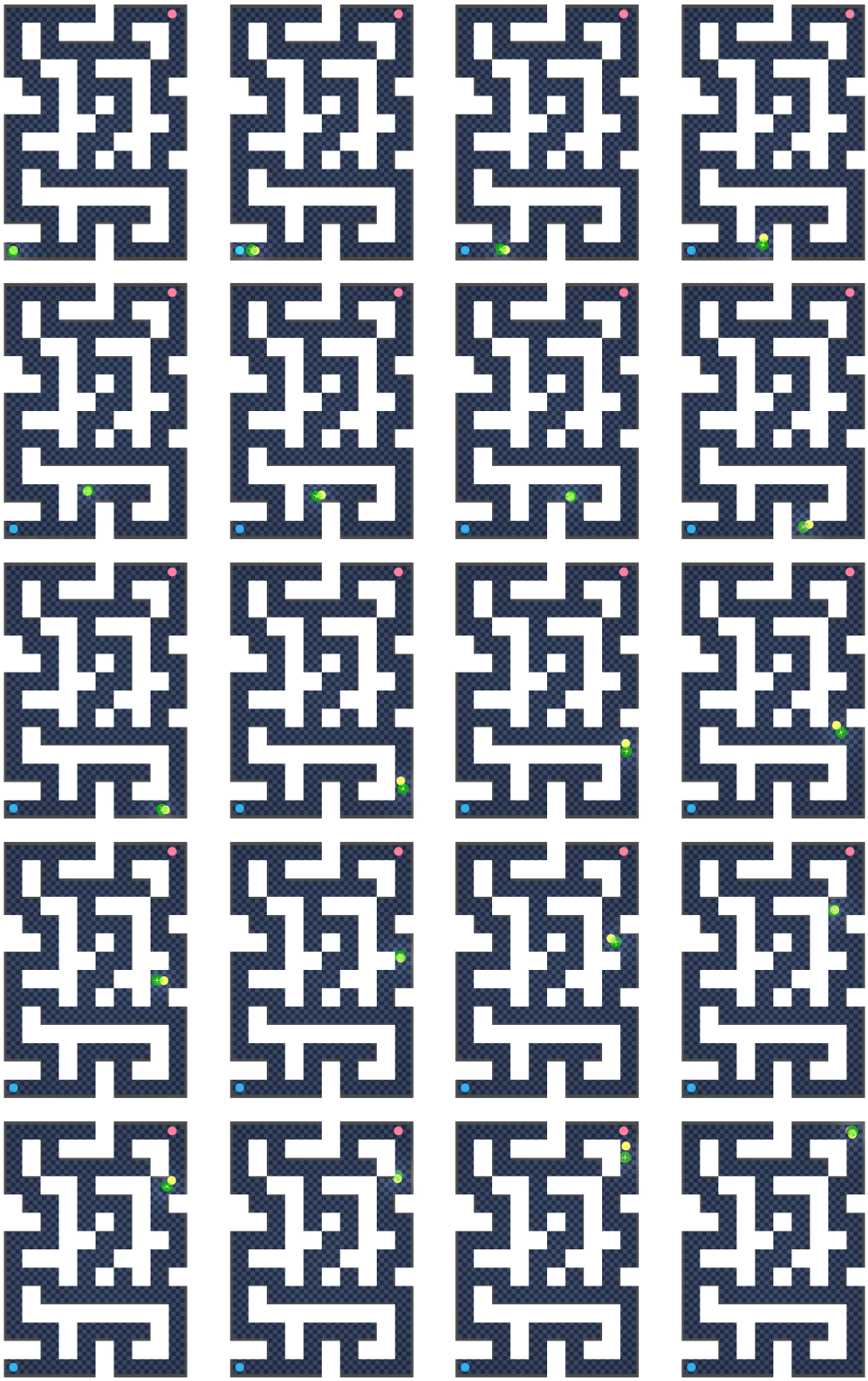


Figure 6: **Visualization of RCD rollout execution on** `AntMaze-Giant-Stitch`**.** The ant agent navigates from the starting region to the pink goal.

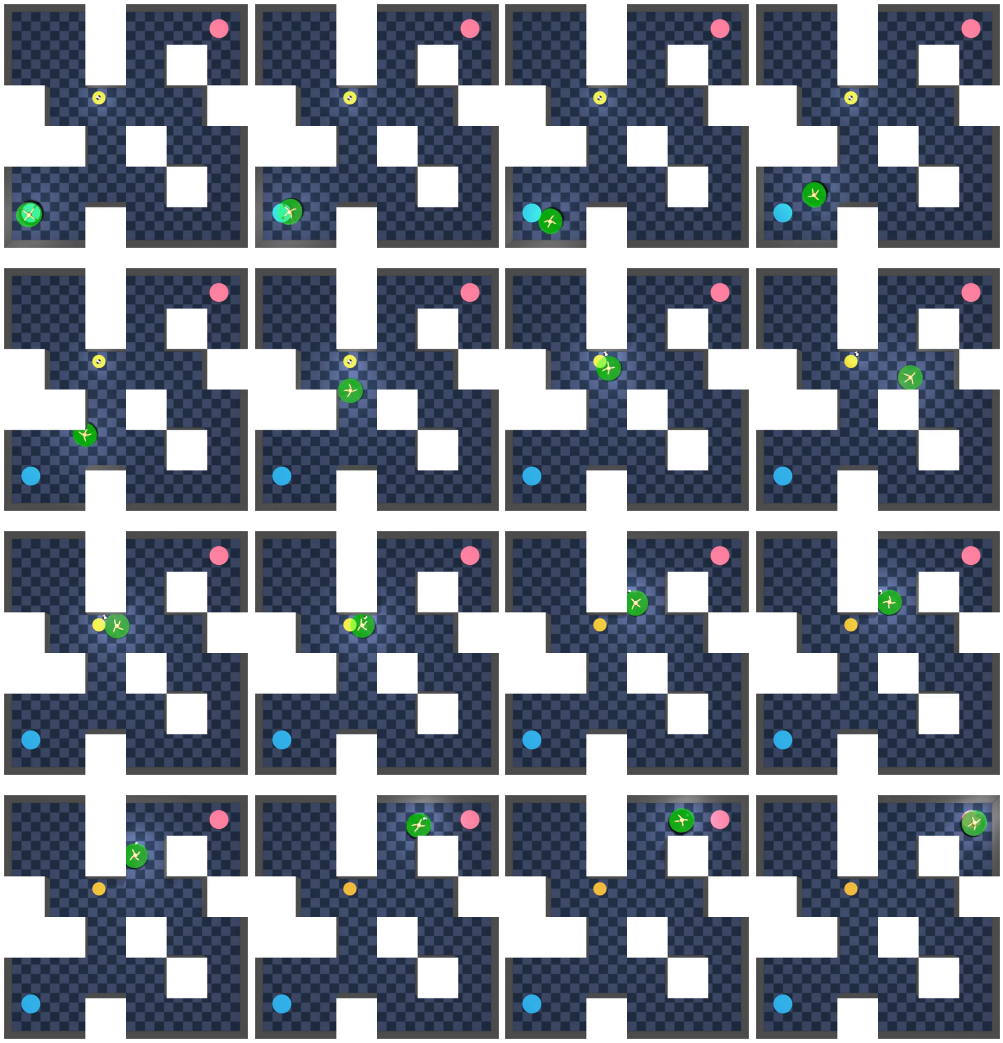


Figure 7: **Visualization of RCD rollout execution on** `AntSoccer-Medium-Stitch`**.** The ant agent dribbles the soccer ball toward the goal location.

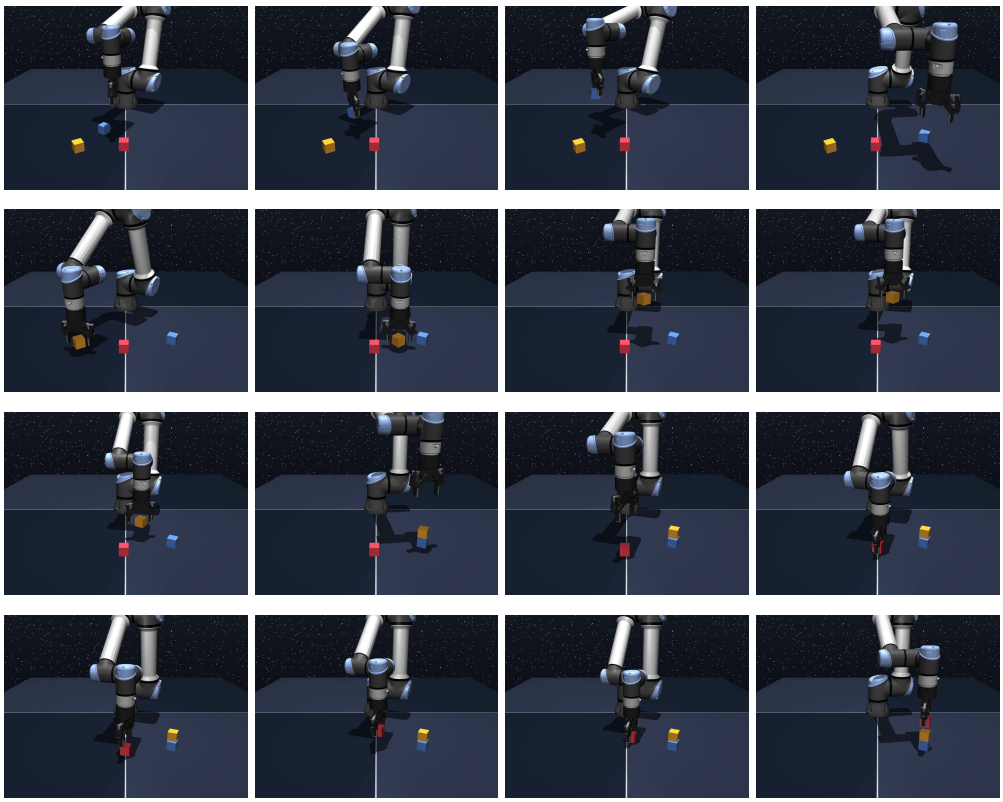


Figure 8: **Visualization of RCD rollout execution on `Cube-Triple`.** The 6-DoF UR5e robot arm arranges three cubes into their target configuration via pick-and-place.

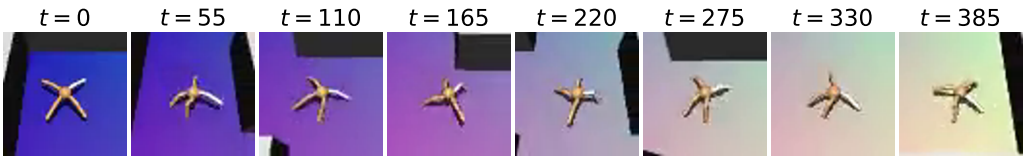


Figure 9: **Visualization of RCD rollout execution on `Visual-AntMaze-Medium-Stitch`.** The ant agent navigates from the starting region to the goal in a long-horizon task where start and goal are at opposite ends of the maze. Frames are the env-rendered 64×64 pixel observations. Planning is performed in a learned 16-dimensional VAE latent space derived from these pixel renders.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions of RCD (training-free compositional diffusion guidance via self-reconstruction error and overlap consistency) and its scope (long-horizon compositional planning), supported by Sections 3 and 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations regarding the Bethe factor graph framework assumption and the pre-specified number of local segments at inference time are discussed explicitly in the *Conclusion* section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper states two propositions (Propositions 1 and 2) with their assumptions in Section 3, and provides full proofs in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental details, including environments, baselines, network architectures, hyperparameters, and the evaluation protocol, are described in Section 4 and Appendix F. Our source code to reproduce the results is also included in the supplemental materials.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include an anonymized source code release in the supplemental materials together with instructions for reproducing the main experimental results. The OGBench datasets [58] used throughout our experiments are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details, including datasets, network architectures, optimizer, RCD guidance hyperparameters, evaluation protocol, and low-level controllers, are described in Section 4 and Appendix F. Hyperparameter choices are further ablated in Section 4.3 and Appendix G.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All success rates in the main and appendix tables are reported as mean $\pm$ standard deviation over 5 random seeds, with 5 start-goal pairs $\times$ 20 episodes per pair on each OGBench task, as described in Section 4 and Appendix F.4.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments, including local diffusion model training and all rollouts for RCD and the baselines, are run on a single NVIDIA H100 GPU, as described in Appendix F.5. Per-plan wall-clock times are reported in Table 6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies fully with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We include an *Impact Statement* paragraph following Section 6, which discusses potential societal impacts and considerations for real-world deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper proposes a training-free inference-time guidance method for offline goal-conditioned planning, and does not release any new pretrained generative models, datasets, or other assets with a high risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite all external assets used, including the OGBench benchmark [58] and the CompDiffuser codebase [49], which serves as our base implementation, and follow their respective terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets or other assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This research does not use large language models as part of its methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.