COMPOSITIONAL DIFFUSION WITH MAXIMAL-NOISE TRANSITIONS FOR LONG-HORIZON PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Compositional diffusion planners generate long-horizon trajectories by composing short local plans that are denoised jointly and kept consistent on their overlaps. However, when local plan distributions are multimodal, averaging incompatible local modes can produce infeasible composed plans. We attribute the persistence of mode-averaging to the predicted noise direction retained in the sampling transition. To address this challenge, we propose *Maximal-Noise Compositional Diffusion* (MaxCD), a *training-free* method that modifies sampling transitions on the overlaps. MaxCD constructs the next noisy state from the denoised estimate and fresh Gaussian noise, removing the predicted noise direction without additional denoiser evaluations. We additionally introduce a spectral split that applies maximal-noise transitions to low-frequency components and retains the original transition on high-frequency components to preserve fine details. Experiments demonstrate that MaxCD substantially improves long-horizon planning performance on OGBench and the visual consistency of compositional panorama generation. Project website at https://max-comp-diffusion.github.io/.

1 INTRODUCTION

Diffusion planners model distributions over entire trajectories and have shown strong performance on offline reinforcement learning benchmarks (Janner et al., 2022; Ajay et al., 2023; Liang et al., 2023). However, directly training these models for long-horizon tasks is challenging, as collecting demonstrations covering diverse start and goal configurations is expensive (Lee & Choi, 2025b; Chen et al., 2026; Luo et al., 2025). Building on compositional generative modeling (Du et al., 2020; 2023), recent approaches compose overlapping local plans from pretrained short-horizon diffusion models at inference time (Mishra et al., 2023; Luo et al., 2025; Mishra et al., 2026; Sun et al., 2026). The local plans are denoised jointly, with information propagated through their overlaps to encourage consistency. This approach also applies to panoramic image generation, where overlapping image patches are jointly denoised using a pretrained image diffusion model (Bar-Tal et al., 2023; Zhang et al., 2023; Lee et al., 2023b; Kim et al., 2024).

However, when local plan distributions are multimodal, neighboring local plans may be denoised towards incompatible modes. For example, in a maze with multiple corridors connecting the start and the goal, the local plans containing these endpoints may follow different corridors. A local plan conditioned on both through its overlaps may then be denoised towards an average of these incompatible local modes, producing a trajectory in a low-density region between the modes. This failure, known as *mode-averaging* (Mishra et al., 2026), produces infeasible composed plans even when the individual modes correspond to feasible local behavior. Existing methods address these inconsistencies through search, guidance, or changes to the composition rule. CDGS (Mishra et al., 2026) combines population-based search with iterative resampling and likelihood-based pruning, requiring additional denoiser evaluations. ECD (Sun et al., 2026) constructs a global energy from local bridge potentials and uses its negative gradient to correct the composed plan. Its efficient approximation computes a boundary reaction term through a block-tridiagonal solve per chunk. For panoramic image generation, SyncDiffusion (Lee et al., 2023b) encourages consistency across windows using gradients of a perceptual loss computed between their denoised image predictions.

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

In this paper, we investigate how the sampling transition affects the persistence of mode-averaging. The predicted noise direction retained by the transition carries information from the current noisy state into the next. Our analysis shows that retaining this direction can increase the probability of remaining in a mode-averaged region across denoising steps. To address this challenge, we propose *Maximal-Noise Compositional Diffusion* (MaxCD), a *training-free* method that applies maximal-noise transitions to the overlaps of local plans. MaxCD constructs the next noisy state on these overlaps from the denoised estimate and fresh Gaussian noise, followed by denoising conditioned on neighboring local plans. We additionally introduce a spectral split that applies maximal-noise transitions to low-frequency components and retains the original transition on high-frequency components to preserve fine details. MaxCD retains the original composition rule and requires no additional denoiser evaluations.

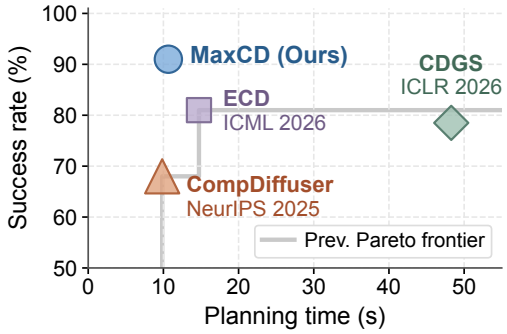


Figure 1: **Planning success and runtime.** Success rates and per-plan runtimes averaged over `PointMaze-Giant-Stitch` and `AntMaze-Giant-Stitch`, using the results in Table 10.

Our main contributions are as follows: **(1)** We analyze how retaining the predicted noise direction contributes to the persistence of mode-averaging. **(2)** We propose *Maximal-Noise Compositional Diffusion* (MaxCD), a training-free method that modifies transitions on the overlaps of local plans and uses a spectral split to preserve fine details, without additional denoiser evaluations. **(3)** We demonstrate that MaxCD improves the success rate of CompDiffuser on long-horizon planning tasks from OGBench (Park et al., 2025) and the consistency of panoramic image generation.

2 Background

2.1 Planning with Diffusion Models

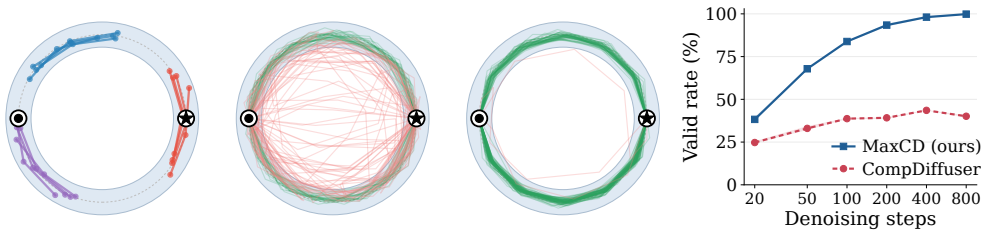
Diffusion probabilistic models (Ho et al., 2020) learn a data distribution by defining a forward process that gradually adds Gaussian noise and a reverse process that recovers the clean data. Given a clean trajectory $\boldsymbol{\tau}^{(0)} \sim p(\boldsymbol{\tau})$, the forward process produces increasingly 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 decreasing noise schedule. A network $\boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^{(t)}, t)$ is trained to predict the noise by minimizing $\mathbb{E}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^{(t)}, t)\|^2]$, and a sample is generated by starting from $\boldsymbol{\tau}^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applying a learned reverse transition T times. Diffuser (Janner et al., 2022) applies this framework to trajectory planning by training the model on trajectories $\boldsymbol{\tau} = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \dots)$ from offline data. Goal-conditioned planning generates trajectories conditioned on specified start and goal states. These constraints can be imposed through inpainting, which fixes the corresponding state coordinates at each denoising step (Janner et al., 2022).

2.2 Compositional Diffusion with Local Plans

Compositional diffusion generates long-horizon plans by composing overlapping local plans from a pretrained short-horizon diffusion model (Mishra et al., 2023; Zhang et al., 2023; Luo et al., 2025). Consider a global plan $\boldsymbol{\tau} = (x_1, \dots, x_N) \in \mathbb{R}^{N \times D}$ composed of M overlapping local plans $y_1, \dots, y_M$, where each $x_i \in \mathbb{R}^D$ is a state or a state-action pair. Each local plan covers a contiguous subsequence of $\boldsymbol{\tau}$, and adjacent local plans share variables over their overlaps. The local plans are denoised jointly using a shared noise-prediction network $\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)$. Some methods couple the local plans by averaging noise predictions in overlapping regions at each denoising step (Mishra et al., 2023; Zhang et al., 2023; Mishra et al., 2026). In CompDiffuser (Luo et al., 2025), each local plan maintains its own noisy state and is conditioned on the noisy overlaps of its neighbors. The first and last local plans are additionally conditioned on the start and goal states, respectively. When



(a) Training Segments (b) CompDiffuser (c) MaxCD **(ours)** (d) Valid Rate vs. Steps

Figure 2: **Toy illustration of persistent mode-averaging. (a)** Training data consists of short trajectory segments, each represented by three positions along a 60° circular arc. At test time, three overlapping local plans are composed to connect the start ◉ and goal ⊛. A trajectory is valid when all its states and segment midpoints lie within the shaded ring. **(b)** CompDiffuser produces many invalid (red) trajectories passing through the interior of the circle, illustrating mode-averaging. **(c)** MaxCD generates mostly valid (green) trajectories along the circle using the same trained model and composition rule. **(d)** As the number of denoising steps increases, the valid rate improves substantially for MaxCD but remains low for CompDiffuser. Panels (b) and (c) use 400 denoising steps.

local plan distributions are multimodal, the denoised estimates on the overlaps can lie between incompatible modes in low-density regions, resulting in infeasible composed plans. This failure is known as *mode-averaging* (Mishra et al., 2026).

2.3 Denoised Estimates and Sampling Transitions

Given a noisy local plan $y_j^{(t)}$, Tweedie's formula (Robbins, 1992; Chung et al., 2023) gives the posterior mean of the clean local plan under the forward process in Equation (1):

$$\begin{aligned}\mathbb{E}[y_j^{(0)} \mid y_j^{(t)}] &= \frac{1}{\sqrt{\alpha_t}}\big(y_j^{(t)} + (1-\alpha_t)\nabla_{y_j^{(t)}} \log p(y_j^{(t)})\big) \\ &\approx \frac{1}{\sqrt{\alpha_t}}\big(y_j^{(t)} - \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)\big) =: \hat{y}_{j,0}^{(t)},\end{aligned} \tag{2}$$

where $\hat{y}_{j,0}^{(t)}$ denotes the denoised estimate of the clean local plan. We consider sampling transitions that combine the denoised estimate, a predicted noise direction, and fresh Gaussian noise (Ho et al., 2020; Song et al., 2021):

$$y_j^{(t-1)} = \underbrace{\sqrt{\alpha_{t-1}}\,\hat{y}_{j,0}^{(t)}}_{\text{denoised estimate}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\,\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)}_{\text{predicted noise direction}} + \underbrace{\sigma_t\,\mathbf{z}}_{\text{fresh noise}}, \tag{3}$$

where $\alpha_0 = 1$, $0 \le \sigma_t \le \sqrt{1-\alpha_{t-1}}$, and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is sampled independently at each transition. The second term retains the predicted noise direction, while the third introduces fresh noise with standard deviation σ_t. Increasing σ_t reduces the coefficient of the retained direction without changing the denoised estimate in the update. This family includes deterministic DDIM sampling at $\sigma_t = 0$ and DDPM sampling with the posterior variance (Song et al., 2021).

3 Maximal-Noise Compositional Diffusion

3.1 Mode-Averaging Problem in Compositional Diffusion

When local plan distributions are multimodal, neighboring local plans may be denoised towards incompatible modes. A local plan conditioned on these neighbors through its overlaps can then have a denoised estimate between the modes, in a low-density region. The composed plan can therefore be infeasible even when each mode represents feasible local behavior. In the circular-arc example in Figure 2a, the local diffusion model is trained on short segments along the circle. At test time, feasible plans connect the start and goal along either the upper or lower half. However, CompDiffuser often generates trajectories through the interior of the circle in Figure 2b, where they

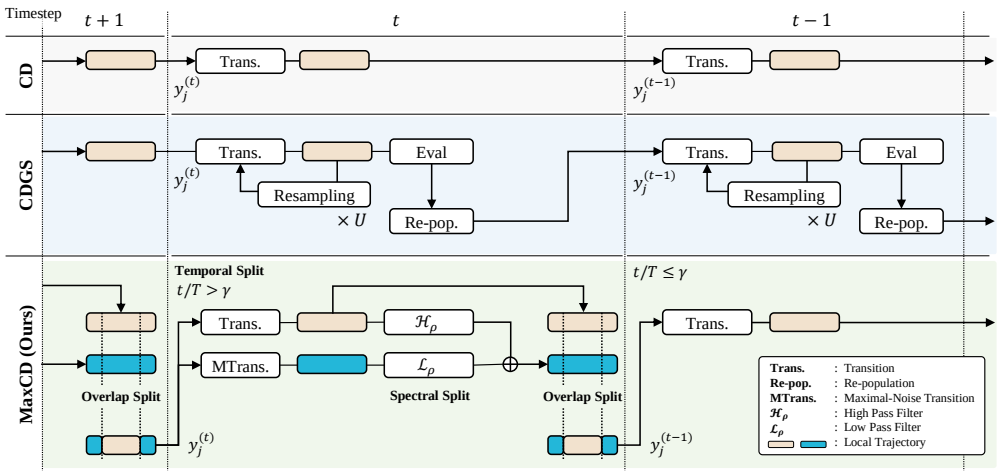


Figure 3: **Overview of MaxCD, compared with CompDiffuser and CDGS. Top:** CompDiffuser applies the original transition at each denoising step. **Middle:** CDGS performs U resampling iterations before evaluating and repopulating candidate plans. **Bottom:** MaxCD replaces the predicted noise direction with fresh Gaussian noise in the low-frequency components, while retaining the original transition for high-frequency components. The update applies to overlap coordinates (cyan) only when $t/T > \gamma$, with the original transition retained elsewhere and during final refinement.

depart from the regions supported by the training data. Increasing the number of denoising steps does not reliably resolve this problem, as shown in Figure 2d. We next examine how the sampling transition contributes to the persistence of these mode-averaged states.

3.2 Persistence of Mode-Averaging Under the Sampling Transition

The predicted noise direction in Equation (3) is computed from the current noisy state and reused to construct the next state. Rearranging the definition of the denoised estimate in Equation (2) makes this dependence explicit:

$$\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t) = \frac{y_j^{(t)} - \sqrt{\alpha_t}\,\hat{y}_{j,0}^{(t)}}{\sqrt{1-\alpha_t}}. \tag{4}$$

Thus, the noise prediction is the difference between the current noisy state and the scaled denoised estimate, divided by the noise standard deviation. Substituting Equation (4) into Equation (3) gives

$$\begin{aligned} y_j^{(t-1)} &= \sqrt{\alpha_{t-1}}\,\hat{y}_{j,0}^{(t)} + c_t\big(y_j^{(t)} - \sqrt{\alpha_t}\,\hat{y}_{j,0}^{(t)}\big) + \sigma_t \mathbf{z} \\ &= \underbrace{c_t y_j^{(t)}}_{\text{retained state component}} + \big(\sqrt{\alpha_{t-1}} - c_t\sqrt{\alpha_t}\big)\hat{y}_{j,0}^{(t)} + \sigma_t \mathbf{z}, \end{aligned} \tag{5}$$

where $c_t = \sqrt{(1-\alpha_{t-1}-\sigma_t^2)/(1-\alpha_t)}$. For a fixed denoised estimate, $c_t y_j^{(t)}$ shows that part of the current noisy state is carried into the next state, even when fresh Gaussian noise is added. Increasing σ_t reduces c_t, and at the maximal variance $\sigma_t^2 = 1-\alpha_{t-1}$ this coefficient is zero. This removes the retained state component, while the denoised estimate still depends on the current plan and its neighbors.

To isolate the effect of the transition on persistence, we consider a local model in which the denoised estimate remains at the midpoint between two incompatible modes. Let $y^{(t)}$ be a scalar coordinate on an overlap of local plan $y_j^{(t)}$, $\hat{y}_0^{(t)}$ its corresponding denoised estimate, and $s_t = \sqrt{1-\alpha_t}$. We place the midpoint at zero for notational simplicity and model the mode-averaged region by

$$\hat{y}_0^{(t)} = 0 \quad \text{whenever} \quad |y^{(t)}| \leq b s_t, \qquad b > 0. \tag{6}$$

The estimate is fixed at the midpoint while the coordinate lies in $[-bs_t, bs_t]$, whose width decreases with the noise level. Its value outside the region is unrestricted. The same analysis applies to any fixed midpoint after shifting the coordinates, as shown in Section C.1.

Let $\epsilon_\theta^{(t)}$ denote the corresponding scalar component of the noise prediction $\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)$. Within the region, Equations (4) and (6) give

$$\epsilon_\theta^{(t)} = \frac{y^{(t)} - \sqrt{\alpha_t}\,\hat{y}_0^{(t)}}{s_t} = \frac{y^{(t)}}{s_t}. \tag{7}$$

Substituting this expression into the scalar coordinate of Equation (3) gives

$$y^{(t-1)} = \sqrt{\alpha_{t-1}}\,\hat{y}_0^{(t)} + \sqrt{s_{t-1}^2 - \sigma_t^2}\,\epsilon_\theta^{(t)} + \sigma_t z_t = \frac{\sqrt{s_{t-1}^2 - \sigma_t^2}}{s_t}\,y^{(t)} + \sigma_t z_t. \tag{8}$$

Here $0 \le \sigma_t \le s_{t-1}$ and $z_t \sim \mathcal{N}(0,1)$ is sampled independently at each transition. The following proposition compares the probability that a coordinate initially in this region remains inside throughout a sequence of denoising steps.

Proposition 1 (Probability of remaining in a mode-averaged region)**.** *Consider Equation* (8) *under the local model in Equation* (6) *over* $K = t_{\mathrm{hi}} - t_{\mathrm{lo}} \ge 1$ *transitions, with* $0 < \alpha_{t_{\mathrm{hi}}} < \cdots < \alpha_{t_{\mathrm{lo}}} < 1$. *Let the standard deviations* σ_t *be deterministic, with* $0 \le \sigma_t \le s_{t-1}$. *Suppose* $y^{(t_{\mathrm{hi}})} \sim \mathcal{N}(0, s_{t_{\mathrm{hi}}}^2)$ *and the variables* $z_t \sim \mathcal{N}(0,1)$ *are mutually independent and independent of* $y^{(t_{\mathrm{hi}})}$. *Let* $q_b = 2\Phi(b) - 1$, *where* Φ *is the standard normal cumulative distribution function. Then*

$$\Pr\left(\begin{array}{c} |y^{(t)}| \le b s_t \text{ for all} \\ t_{\mathrm{lo}} \le t < t_{\mathrm{hi}} \end{array} \,\middle|\, |y^{(t_{\mathrm{hi}})}| \le b s_{t_{\mathrm{hi}}} \right) \ge q_b^K. \tag{9}$$

Equality holds if and only if $\sigma_t = s_{t-1}$ *for every* $t_{\mathrm{lo}} < t \le t_{\mathrm{hi}}$.

The proof can be found in Section C.1. Under this local model, deterministic DDIM ($\sigma_t = 0$) scales both the coordinate and the region width by s_{t-1}/s_t, so a coordinate starting inside remains inside. At maximal variance ($\sigma_t = s_{t-1}$), each transition from inside the region draws $y^{(t-1)} = s_{t-1} z_t$, which stays inside with probability q_b. The probability of remaining inside for all K transitions is therefore q_b^K, the minimum in Proposition 1. Retaining the predicted noise direction at any of these transitions gives a larger probability of remaining inside for the entire sequence.

3.3 Maximal-Noise Transitions on Overlaps

Motivated by the analysis in Section 3.2, we propose *Maximal-Noise Compositional Diffusion* (MaxCD), which applies maximal-noise transitions to the overlaps of local plans. Setting $\sigma_t = \sqrt{1 - \alpha_{t-1}}$ in Equation (3) removes the predicted noise direction and gives

$$\bar{y}_j^{(t-1)} = \sqrt{\alpha_{t-1}}\hat{y}_{j,0}^{(t)} + \sqrt{1 - \alpha_{t-1}}\mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{10}$$

Subsequent denoising conditioned on neighboring local plans can then move the overlap estimate away from a mode-averaged state towards a supported mode.

Preserving fine details. The denoised estimate used in Equation (10) approximates a posterior mean, in which uncertain fine details can be averaged out. Repeatedly applying this transition across all frequency components can therefore suppress fine details, as illustrated in Figure 4. To preserve these details, we apply maximal-noise transitions to low-frequency components, which can revise coarse trajectory geometry, and retain the original transition for high-frequency components. Let $\mathcal{L}_\rho$ be the orthogonal projection onto normalized frequencies $|f| \le \rho$, and $\mathcal{H}_\rho = I - \mathcal{L}_\rho$. We apply these projections along the local-plan horizon for each state dimension in planning, and along the spatial dimensions in panorama generation. The spectral split gives the candidate $\bar{y}_{j,\rho}^{(t-1)}$:

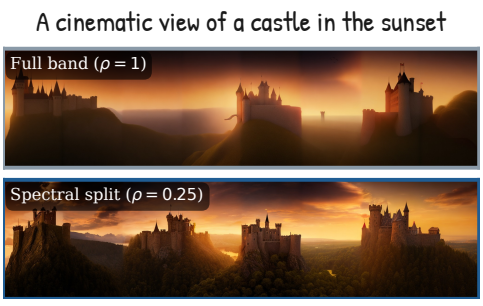


Figure 4: **Effect of the spectral split.** Same prompt and seed. **Top:** Full-band maximal noise over-smooths the image. **Bottom:** The spectral split preserves fine details.

$$\begin{aligned} \bar{y}_{j,\rho}^{(t-1)} = \sqrt{\alpha_{t-1}}\,\hat{y}_{j,0}^{(t)} &+ \sqrt{1 - \alpha_{t-1}}\,\mathcal{L}_\rho \mathbf{z} \\ &+ \mathcal{H}_\rho\Big(\sqrt{1 - \alpha_{t-1} - \sigma_t^2}\,\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t) + \sigma_t \mathbf{z}\Big), \end{aligned} \tag{11}$$

where σ_t is the noise standard deviation of the original transition. This candidate retains the denoised estimate and removes the predicted noise direction only from the low-frequency components.

Compositional update. Neighboring local plans exchange information through their overlaps. We therefore apply the candidate in Equation (11) to the overlap coordinates of interior local plans and retain the original transition elsewhere. For $y_j^{(t)} \in \mathbb{R}^{h \times D}$, let $\mathbf{m}_j \in \{0,1\}^h$ be one on the overlaps of an interior local plan and zero elsewhere, where h is the local-plan horizon. We set $\mathbf{m}_1 = \mathbf{m}_M = \mathbf{0}$ to retain the original transition in the start- and goal-conditioned plans. To reserve the low-noise stages for refinement, we use this update only when $t/T > \gamma$:

$$y_j^{(t-1)} = \mathbf{m}_j \odot \bar{y}_{j,\rho}^{(t-1)} + (1 - \mathbf{m}_j) \odot \tilde{y}_j^{(t-1)}, \tag{12}$$

where $\tilde{y}_j^{(t-1)}$ is the original transition in Equation (3), evaluated with $\mathbf{z}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ independent of $\mathbf{z}$. The mask is broadcast across state dimensions. For $t/T \leq \gamma$, all coordinates use the original transition $\tilde{y}_j^{(t-1)}$. Diffusion editing methods similarly restrict modifications to selected denoising steps to balance structural preservation and adaptation (Hertz et al., 2023; Kwon et al., 2023; Tumanyan et al., 2023). Both candidates reuse the same denoised estimate and noise prediction, so MaxCD requires no retraining or additional denoiser evaluations. An overview of MaxCD is provided in Figure 3, and the full procedure is given in Algorithm 1.

4 Experiments

In this section, we evaluate MaxCD on long-horizon planning tasks from OGBench (Park et al., 2025) and on panoramic image generation. Specifically, we demonstrate **(1)** that MaxCD improves the feasibility of composed plans, **(2)** that it improves planning performance on locomotion and object manipulation tasks, and **(3)** that the maximal-noise transition also improves consistency in panoramic image generation. Additional details on the experimental setup are provided in Section E, and further results in Section F.

Datasets and Environments. We consider two categories of long-horizon tasks from OGBench, a benchmark for offline goal-conditioned reinforcement learning. **(1) Locomotion:** `PointMaze`, `AntMaze`, and `HumanoidMaze` on the `Stitch` datasets across `Medium`, `Large`, and `Giant` maze sizes. **(2) Object Manipulation:** `Cube` manipulation (`Single`, `Double`, `Triple`, `Quadruple`) on the `Play` datasets, and `AntSoccer` (`Arena`, `Medium`) on the `Stitch` datasets. The `Stitch` datasets contain local trajectory segments that cover only parts of the evaluation tasks. Solving these tasks requires composing multiple segments into a long-horizon plan that connects the start and goal. The `Play` datasets contain unstructured object interactions. For panorama generation, we compose 17 overlapping 512×512 windows of Stable Diffusion 2.0 (Rombach et al., 2022) into 512×4608 panoramas with 50% overlap, following the protocol of Mishra et al. (2026).

Baselines. We compare with offline goal-conditioned RL methods: goal-conditioned behavioral cloning (GCBC) (Lynch et al., 2020), goal-conditioned implicit V-learning (GCIVL) and Q-learning (GCIQL) (Kostrikov et al., 2022), Quasimetric RL (QRL) (Wang et al., 2023), Contrastive RL (CRL) (Eysenbach et al., 2022), and Hierarchical implicit Q-learning (HIQL) (Park et al., 2023). We also include compositional diffusion planners: Generative Skill Chaining (GSC) (Mishra et al., 2023), CompDiffuser (CD) (Luo et al., 2025), CDGS (Mishra et al., 2026), and ECD (Sun et al., 2026). For each environment, CD, CDGS, ECD, and MaxCD use the same pretrained local planners and low-level controllers. We use a single default configuration for MaxCD across OGBench and panorama generation, with noise-level threshold $\gamma = 0.2$ and spectral bandwidth $\rho = 0.25$. Sampling and replanning settings are provided in Section E.5. For panorama generation, we compare with GSC, SyncDiffusion (Lee et al., 2023b), CDGS, and ECD. The image-domain analogue of GSC is MultiDiffusion (Bar-Tal et al., 2023), which averages predictions in overlapping regions. SyncDiffusion uses LPIPS-based perceptual guidance to encourage consistency between windows.

4.1 Plan Feasibility Analysis

Table 1: **Quantitative results on OGBench locomotion tasks.** We report success rates as mean ± standard deviation. Our evaluations use 5 seeds, with 20 episodes per task across 5 tasks. Sources of baseline results are provided in Appendix G.

Env	Type	Size	GCBC	GCIVL	GCIQL	QRL	CRL	HIQL	GSC	CD	CDGS	ECD	MaxCD
pointmaze	stitch	Medium	23±18	70±14	21±9	80±12	0±1	74±6	**100**±0	**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	**100**±0
		Giant	0±0	0±0	0±0	50±8	0±0	0±0	28±3	69±3	74±3	81±2	**94**±2
antmaze	stitch	Medium	45±11	44±6	29±6	59±7	53±6	94±1	96±2	91±2	93±3	**97**±2	**97**±2
		Large	3±3	18±2	7±2	18±2	11±2	67±5	70±2	**89**±2	63±4	85±2	87±2
		Giant	0±0	0±0	0±0	0±0	0±0	2±2	25±2	67±3	83±3	81±2	**88**±2
humanoid maze	stitch	Medium	29±5	12±2	12±3	18±2	36±2	88±2	92±1	92±2	90±2	89±2	**94**±2
		Large	6±3	1±1	0±0	3±1	4±1	28±3	68±3	74±3	63±4	76±2	**80**±3
		Giant	0±0	0±0	0±0	0±0	0±0	3±2	8±1	44±4	35±4	50±3	**71**±2

Table 2: **Quantitative results on OGBench object manipulation tasks.** We report success rates as mean ± standard deviation. Our evaluations use 5 seeds, with 20 episodes per task across 5 tasks.

Env	Type	Size	GCBC	GCIVL	GCIQL	QRL	CRL	HIQL	GSC	CD	CDGS	ECD	MaxCD
cube	play	Single	6±2	53±4	68±6	5±1	19±2	15±3	**100**±0	**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	82±3	**91**±4
		Triple	1±1	1±0	3±1	0±0	4±1	3±1	45±4	51±4	60±4	58±4	**64**±3
		Quadruple	0±0	0±0	0±0	0±0	0±0	0±0	30±4	48±4	51±4	52±3	**57**±3
antsoccer	stitch	Arena	24±8	21±3	2±0	1±1	1±0	15±1	65±2	69±3	61±3	66±2	**75**±2
		Medium	2±1	1±0	0±0	0±0	0±0	4±1	12±2	17±3	11±2	18±2	**20**±2

We first assess whether MaxCD improves the feasibility of plans generated by CompDiffuser. We sample 20 plans for each of the 5 test-time tasks in `AntMaze-Giant-Stitch` and measure the *valid plan rate*, the fraction of plans that do not pass through maze walls. In Figure 5, we compare CD, ECD, CDGS, and MaxCD on Task 4 using the same pretrained local model and start and goal conditions. CompDiffuser produces many infeasible plans that pass through maze walls, while MaxCD generates more valid plans that follow the maze corridors. ECD and CDGS improve validity on this task, although some plans still cross walls. Across the 5 tasks, MaxCD increases the mean valid plan rate from 36% to 86%. Additional AntMaze and PointMaze comparisons are provided in Section F.2.

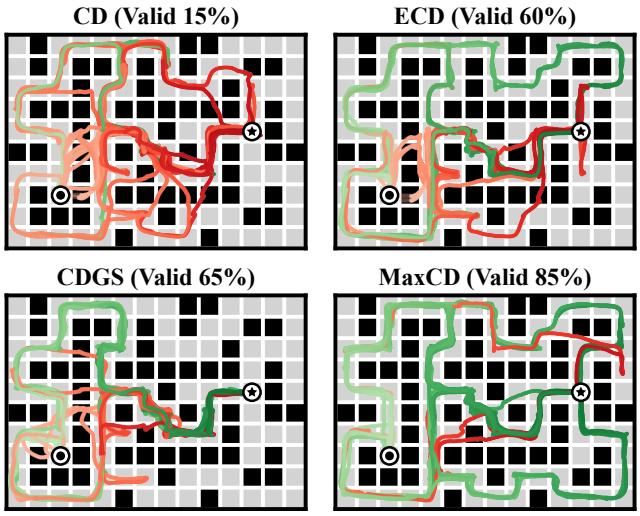


Figure 5: **Plan quality on `AntMaze-Giant-Stitch`.** Each panel shows 20 plans for Task 4. **Top:** CD and ECD. **Bottom:** CDGS and MaxCD. Red plans pass through walls, while green plans are valid. Percentages give valid rates. All five tasks are shown in Section F.2.

4.2 Enhancing Long-Horizon Planning Performance

Locomotion. Table 1 reports success rates on `PointMaze-Stitch`, `AntMaze-Stitch`, and `HumanoidMaze-Stitch`, where the compositional planners generate trajectories in the 2D x-y position space. Following Luo et al. (2025), `PointMaze` plans are executed with a PD controller, while `AntMaze` and `HumanoidMaze` plans use learned inverse dynamics models. MaxCD achieves the highest success rates on the most challenging `Giant` tasks across all three environments, where successful planning requires composing many short local plans. Together with the feasibility results in Section 4.1, these results show that maximal-noise transitions improve long-horizon planning with the same pretrained local models.

Table 3: **Quantitative results on panoramic image generation.** Results are reported as mean ± standard deviation. Evaluation details are given in Section E.6.

Metric	GSC	SyncDiffusion	CDGS	ECD	MaxCD
Intra-LPIPS ↓	0.70±0.06	0.61±0.06	0.65±0.05	0.69±0.08	**0.57**±0.06
Intra-Style-L ↓	3.98±3.44	2.46±2.49	**1.72**±1.32	3.06±2.71	2.56±2.35
CLIP-S ↑	32.47±1.85	32.36±2.02	32.88±2.06	32.22±1.80	**32.92**±2.24
Time (s) ↓	**33.30**±0.30	84.30±0.20	3014.90±0.90	33.80±0.30	33.40±0.20



Figure 6: **Panoramic image generation.** Qualitative examples from GSC and MaxCD on landscape and corgi prompts. Additional comparisons are provided in Figure 13.

Object Manipulation. Table 2 reports success rates on `Cube` and `AntSoccer`. `Cube` tasks require arranging one to four cubes through pick-and-place sequences, with longer planning horizons as the number of cubes increases. The planner operates on the full state, including robot arm joint positions and cube poses, and the plans are executed with goal-conditioned Diffusion Q-Learning (DQL) policies (Wang et al., 2022). In `AntSoccer`, the ant must reach the ball and then dribble it to the goal. Following Luo et al. (2025), we use a 17D planner that includes the x-y positions of the ant and ball and the joint positions of the ant, with a learned inverse dynamics model for execution. Offline goal-conditioned RL methods struggle on `Triple` and `Quadruple`, while compositional planners achieve higher success rates by combining short local plans. MaxCD achieves the highest success rates across both environments, improving on CompDiffuser on all multi-cube tasks and both `AntSoccer` layouts. These results show that maximal-noise transitions also improve planning for tasks that require coordinated robot and object motion.

4.3 Panoramic Image Generation

We further evaluate MaxCD on panoramic image generation by composing overlapping windows of a pretrained image diffusion model (Bar-Tal et al., 2023; Zhang et al., 2023). We apply MaxCD to GSC with the spatial-frequency spectral split described in Section 3.3. Table 3 compares GSC, SyncDiffusion, CDGS, ECD, and MaxCD on visual consistency, prompt alignment, and generation time. Following Lee et al. (2023b) and Mishra et al. (2026), Intra-LPIPS and Intra-Style-L measure perceptual and style consistency between non-overlapping crops, while CLIP-S measures alignment with the text prompt. MaxCD improves all three quality metrics over GSC with nearly the same generation time, without additional denoiser evaluations. Compared with CDGS, it achieves lower Intra-LPIPS and higher CLIP-S. CDGS achieves the lowest Intra-Style-L but requires approximately $90\times$ the generation time of MaxCD. In the landscape example in Figure 6, applying maximal-noise

transitions produces a more coherent visual style across distant windows. For the corgi prompt, GSC generates groups of people in several regions, while MaxCD depicts corgis more consistently across the panorama. Additional visualizations are provided in Figure 13.

4.4 Effect of the Spectral Split

Table 4 isolates the effect of the spectral split using the 2D, 15D, and 29D planners of Luo et al. (2025), with the pretrained planner and all other sampling settings fixed within each comparison. Without the split, the 15D and 29D planners achieve success rates close to CD. Adding the split raises success rates on the `Giant` maze from 38% to 61% for the 15D planner and from 27% to 45% for the 29D planner, compared with a smaller change from 83% to 88% for the 2D planner. These results support preserving high-frequency information when revising the composed plan through maximal-noise transitions. The same motivation is illustrated in Figure 4, where the split preserves fine details in panoramas that are over-smoothed by full-band transitions. Additional ablations and a planning-time comparison are provided in Sections F.5 and F.6.

Table 4: **Ablation on the spectral split.** Success rates on `AntMaze-Stitch`.

Size	Dim.	CD	MaxCD w/o split	MaxCD
`Large`	2D	**89**± 2	86± 2	87± 2
	15D	63± 4	65± 5	**77**± 3
	29D	65± 5	66± 5	**71**± 4
`Giant`	2D	67± 3	83± 2	**88**± 2
	15D	40± 3	38± 4	**61**± 4
	29D	25± 3	27± 4	**45**± 3

5 Related Work

Planning with Diffusion Models. Diffusion planners model distributions over trajectories from offline data. Diffuser (Janner et al., 2022) guides trajectory sampling with a learned value function, while Decision Diffuser (Ajay et al., 2023) conditions on returns and constraints through classifier-free guidance. AdaptDiffuser (Liang et al., 2023) improves the planner by fine-tuning on generated trajectories. LoMAP (Lee & Choi, 2025a) improves plan feasibility by projecting intermediate samples onto local approximations of the data manifold. Monte Carlo Tree Diffusion (MCTD) (Yoon et al., 2025) combines diffusion planning with Monte Carlo tree search to evaluate and refine partially denoised plans. Hierarchical Diffuser (Chen et al., 2024) uses a high-level diffusion planner to propose subgoals and a low-level diffusion planner to generate short-horizon trajectories between them. To extend the training horizon, SCoTS (Lee & Choi, 2025b) constructs longer trajectories by stitching short segments and trains a planner on the resulting dataset.

Compositional Planning. GSC (Mishra et al., 2023) extends the planning horizon at inference time through skill score composition, while CompDiffuser (Luo et al., 2025) conditions local plans on the noisy overlaps of their neighbors. To address mode-averaging, CDGS (Mishra et al., 2026) introduces population-based search with resampling and pruning. ECD (Sun et al., 2026) derives a correction from a global energy built from local bridge potentials. MaxCD investigates how the retained noise direction affects the persistence of mode-averaging and changes the transition on overlaps without retraining or additional denoiser evaluations. An extended discussion, including compositional image generation and stochastic sampling, is provided in Section D.

6 Conclusion

We presented Maximal-Noise Compositional Diffusion (MaxCD), a training-free method that addresses persistent mode-averaging in compositional diffusion planning. Our analysis shows that retaining the predicted noise direction can increase the probability of remaining in a mode-averaged region across denoising steps. MaxCD applies maximal-noise transitions to low-frequency components on overlaps and retains the original transition on high-frequency components to preserve fine details, without additional denoiser evaluations. Experiments on OGBench demonstrate improved plan feasibility in maze navigation and higher success rates on challenging locomotion and object manipulation tasks, with substantial gains on the `Giant` mazes. The same transition improves visual consistency and prompt alignment in panorama generation with nearly the same generation time as the base sampler. We discuss limitations and future directions in Appendix B.

AI USE STATEMENT

We used generative AI tools to assist with the initial setup of experimental environments, implementing and debugging code, and reviewing and editing proofs. We also used these tools for manuscript drafting and editing and schematic figure design. The authors take responsibility for the final text, mathematical claims, code, figures, and reported results.

REPRODUCIBILITY STATEMENT

The full sampling procedure is given in Algorithm 1, and the proof of Proposition 1 is provided in Section C.1. Details of the datasets, pretrained models, low-level controllers, hyperparameters, and evaluation protocols for planning and panorama generation are provided in Section E. Sources of baseline results are listed in Section G. The supplementary material includes the sampler implementation, code for the planning and panorama experiments, evaluation scripts, and reproduction instructions.

ETHICS STATEMENT

This work aims to advance machine learning by improving the reliability of compositional diffusion planning. We do not identify direct negative societal impacts that require specific discussion. We encourage practitioners to assess safety and reliability before deploying these planners in real-world applications.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *International Conference on Learning Representations (ICLR)*, 2023.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *International Conference on Machine Learning (ICML)*, 2023.

Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion. *arXiv preprint arXiv:2401.02644*, 2024.

Chang Chen, Hany Hamed, Doojin Baek, Taegu Kang, Samyeul Noh, Yoshua Bengio, and Sungjin Ahn. Extendable planning via multiscale diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning (ICML)*, 2023.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *International Conference on Learning Representations (ICLR)*, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. Synctweedies: A general generative framework based on synchronized diffusions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *International Conference on Learning Representations (ICLR)*, 2023.

Kyowoon Lee and Jaesik Choi. Local manifold approximation and projection for manifold-aware diffusion planning. In *International Conference on Machine Learning (ICML)*, 2025a.

Kyowoon Lee and Jaesik Choi. State-covering trajectory stitching for diffusion planners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Kyowoon Lee, Seongun Kim, and Jaesik Choi. Refining diffusion planner for reliable behavior synthesis by automatic detection of infeasible plans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. In *International Conference on Machine Learning (ICML)*, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yunhao Luo, Utkarsh A Mishra, Yilun Du, and Danfei Xu. Generative trajectory stitching through diffusion composition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, 2020.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022.

Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pp. 2905–2925. PMLR, 2023.

Utkarsh Aashu Mishra, David He, Yongxin Chen, and Danfei Xu. Compositional diffusion with guided search for long-horizon planning. In *International Conference on Learning Representations (ICLR)*, 2026.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American statistical association*, 62(318):626–633, 1967.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Tao Sun, Utkarsh Aashu Mishra, Jiaxin Lu, Danfei Xu, and Iro Armeni. Energy-based compositional diffusion planning. In *International Conference on Machine Learning (ICML)*, 2026.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning (ICML)*, 2023.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Jaesik Yoon, Hyeonseo Cho, Doojin Baek, Yoshua Bengio, and Sungjin Ahn. Monte carlo tree diffusion for system 2 planning. In *International Conference on Machine Learning (ICML)*, 2025.

Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. 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.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Appendix:

Compositional Diffusion with Maximal-Noise Transitions for Long-Horizon Planning

CONTENTS

A Algorithm 15

B Limitations and Future work 15

C Proofs 16

C.1 Proof of Proposition 1 16

C.2 Effect of the Denoising Schedule 18

D Extended Related Work 19

E Implementation Details 20

E.1 Environments and Datasets 20

E.2 Pretrained Planners and Low-Level Controllers 20

E.3 Evaluation Protocol 21

E.4 Hyperparameters of MaxCD 21

E.5 Baseline Implementations 22

E.6 Panorama Protocol and Metrics 22

E.7 Compute Resources 22

F Additional Results 23

F.1 Visual Comparison of Denoising Processes 23

F.2 Additional Plan Quality Visualizations 24

F.3 Full-State Plan Visualizations 26

F.4 Cube Manipulation Rollouts 27

F.5 Additional Ablation Studies 28

F.6 Planning-Time Comparison 29

F.7 Additional Qualitative Results 29

G Baseline Result Sources 32

A ALGORITHM

Algorithm 1 summarizes the MaxCD transition. At each step, the noise prediction is computed once for each local plan and reused in both transition candidates, together with the denoised estimate. The candidates use independent Gaussian draws. Orthogonality allows the spectral candidate to use one Gaussian draw for both bands. Setting $\rho = 1$ recovers the full-band maximal-noise transition in Equation (10) on the selected overlaps.

Algorithm 1 Maximal-Noise Compositional Diffusion

Input: denoiser $\boldsymbol{\epsilon}_\theta$, number of local plans M, noise-level threshold γ, spectral bandwidth ρ
Output: global plan $\boldsymbol{\tau}$
`// overlap masks of the interior local plans`
$\mathbf{m}_j \leftarrow$ one on the overlaps and zero elsewhere for $1 < j < M$, with $\mathbf{m}_1 = \mathbf{m}_M = \mathbf{0}$
`// initialize the local plans`
$y_j^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $j = 1, \dots, M$
for $t = T, \dots, 1$ **do**
  for $j = 1, \dots, M$ **in parallel do**
    `// independent Gaussian noise for the two candidates`
    $\mathbf{z}, \mathbf{z}' \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$
    `// denoised estimate with neighbor conditioning (Eq. (2))`
    $\hat{y}_{j,0}^{(t)} \leftarrow \frac{1}{\sqrt{\alpha_t}}\big(y_j^{(t)} - \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t)\big)$
    `// original transition of the sampler (Eq. (3))`
    $\tilde{y}_j^{(t-1)} \leftarrow \sqrt{\alpha_{t-1}}\,\hat{y}_{j,0}^{(t)} + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\,\boldsymbol{\epsilon}_\theta(y_j^{(t)}, t) + \sigma_t\,\mathbf{z}'$
    if $t/T > \gamma$ **then**
      `// spectral split: original high-frequency update (Eq. (11))`
      $\bar{y}_{j,\rho}^{(t-1)} \leftarrow \sqrt{\alpha_{t-1}}\,\hat{y}_{j,0}^{(t)} + \sqrt{1-\alpha_{t-1}}\,\mathcal{L}_\rho \mathbf{z} + \mathcal{H}_\rho\Big(\sqrt{1-\alpha_{t-1}-\sigma_t^2}\,\epsilon_\theta(y_j^{(t)}, t) + \sigma_t \mathbf{z}\Big)$
      `// apply the transition to overlaps (Eq. (12))`
      $y_j^{(t-1)} \leftarrow \mathbf{m}_j \odot \bar{y}_{j,\rho}^{(t-1)} + (1 - \mathbf{m}_j) \odot \tilde{y}_j^{(t-1)}$
    else
      $y_j^{(t-1)} \leftarrow \tilde{y}_j^{(t-1)}$
    end if
  end for
end for
$\boldsymbol{\tau} \leftarrow$ blend the final local plans $y_1^{(0)}, \dots, y_M^{(0)}$ over their overlaps
return $\boldsymbol{\tau}$

B LIMITATIONS AND FUTURE WORK

MaxCD operates on pretrained local diffusion models, so the quality of the composed plans depends on the local behaviors learned from the training data. As in existing compositional planners (Luo et al., 2025; Mishra et al., 2026), we specify the number of local plans before sampling, which determines the total planning horizon. Future work could estimate the temporal distance between the start and goal using representations learned by methods such as METRA (Park et al., 2024), and use this estimate to adaptively choose the number of local plans. Extending our sampling strategy to compositional planning with flow-matching models (Lipman et al., 2022) is another interesting direction for future work.

C PROOFS

C.1 PROOF OF PROPOSITION 1

Proposition 1 (Probability of remaining in a mode-averaged region)**.** *Consider Equation* (8) *under the local model in Equation* (6) *over* $K = t_{\mathrm{hi}} - t_{\mathrm{lo}} \geq 1$ *transitions, with* $0 < \alpha_{t_{\mathrm{hi}}} < \cdots < \alpha_{t_{\mathrm{lo}}} < 1$*. Let the standard deviations* σ_t *be deterministic, with* $0 \leq \sigma_t \leq s_{t-1}$*. Suppose* $y^{(t_{\mathrm{hi}})} \sim \mathcal{N}(0, s_{t_{\mathrm{hi}}}^2)$ *and the variables* $z_t \sim \mathcal{N}(0,1)$ *are mutually independent and independent of* $y^{(t_{\mathrm{hi}})}$*. Let* $q_b = 2\Phi(b) - 1$*, where* Φ *is the standard normal cumulative distribution function. Then*

$$\Pr\left(\begin{array}{c} |y^{(t)}| \leq b s_t \text{ for all} \\ t_{\mathrm{lo}} \leq t < t_{\mathrm{hi}} \end{array} \;\middle|\; |y^{(t_{\mathrm{hi}})}| \leq b s_{t_{\mathrm{hi}}} \right) \geq q_b^K. \tag{9}$$

Equality holds if and only if $\sigma_t = s_{t-1}$ *for every* $t_{\mathrm{lo}} < t \leq t_{\mathrm{hi}}$*.*

Proof. Index the states in denoising order by $t_k = t_{\mathrm{hi}} - k$ for $k = 0, \ldots, K$, so that $t_0 = t_{\mathrm{hi}}$ and $t_K = t_{\mathrm{lo}}$. Within this proof, write $s_k = \sqrt{1 - \alpha_{t_k}}$ for the noise standard deviation at state k, and use σ_k and Z_k for the variables σ_{t_k} and z_{t_k} of transition $k = 0, \ldots, K-1$.

A Gaussian reference process. By Equations (2) and (6), the corresponding scalar noise prediction inside the region is $y^{(t_k)}/s_k$. The denoised-estimate term is zero there, so Equation (3) reduces to the linear update in Equation (8). To study the first exit, define a reference process Y_k by extending this update to all coordinate values:

$$Y_0 = y^{(t_0)}, \qquad Y_{k+1} = \frac{\sqrt{s_{k+1}^2 - \sigma_k^2}}{s_k} Y_k + \sigma_k Z_k. \tag{13}$$

This process uses the original coordinates and the same Gaussian noise samples as $y^{(t_k)}$. If the initial state is inside the region, the two processes follow the same update until and including the first exit. They therefore leave the region at the same step, although their later states can differ. Consequently,

$$\left\{ |y^{(t_k)}| \leq b s_k \text{ for all } 0 \leq k \leq K \right\} = \left\{ |Y_k| \leq b s_k \text{ for all } 0 \leq k \leq K \right\}. \tag{14}$$

It suffices to compute the probability for Y_k. Write $A_k = \{|Y_k| \leq b s_k\}$ and $q_b = 2\Phi(b) - 1$.

Gaussian marginals and the probability bound. We first use the unconditioned initial distribution $Y_0 \sim \mathcal{N}(0, s_0^2)$ and condition on A_0 after obtaining the joint probability bound. The recursion in Equation (13) is linear with deterministic coefficients, so $(Y_0, \ldots, Y_K)$ is a centered Gaussian vector. Since Y_k depends only on $Y_0, Z_0, \ldots, Z_{k-1}$, it is independent of Z_k. Starting from $\mathrm{Var}(Y_0) = s_0^2$, induction gives

$$\begin{aligned} \mathrm{Var}(Y_{k+1}) &= \frac{s_{k+1}^2 - \sigma_k^2}{s_k^2} \mathrm{Var}(Y_k) + \sigma_k^2 \\ &= s_{k+1}^2 - \sigma_k^2 + \sigma_k^2 = s_{k+1}^2. \end{aligned}$$

Thus $Y_k \sim \mathcal{N}(0, s_k^2)$ and

$$\Pr(A_k) = \Pr(-b s_k \leq Y_k \leq b s_k) = \Phi(b) - \Phi(-b) = q_b$$

for every k. Applying the rectangle inequality of Šidák (1967, Corollary 1) gives

$$\Pr\left(\bigcap_{k=0}^{K} A_k \right) \geq \prod_{k=0}^{K} \Pr(A_k) = q_b^{K+1}.$$

The inequality also applies to singular Gaussian vectors. Since $q_b > 0$, conditioning on A_0 yields

$$\Pr\left(\bigcap_{k=1}^{K} A_k \;\middle|\; A_0 \right) = \frac{\Pr(\bigcap_{k=0}^{K} A_k)}{q_b} \geq q_b^K.$$

Together with Equation (14), this proves Equation (9) for the original coordinate.

Equality condition. If $\sigma_k = s_{k+1}$ for every k, then $Y_{k+1} = s_{k+1}Z_k$ and all reference states are independent. The conditional probability is therefore q_b^K. For the converse, the covariance of two successive states is

$$\mathrm{Cov}(Y_k, Y_{k+1}) = \frac{\sqrt{s_{k+1}^2 - \sigma_k^2}}{s_k}\,\mathrm{Var}(Y_k) = s_k\sqrt{s_{k+1}^2 - \sigma_k^2}.$$

Their correlation is therefore

$$\rho_k = \frac{\mathrm{Cov}(Y_k, Y_{k+1})}{s_k s_{k+1}} = \sqrt{1 - \frac{\sigma_k^2}{s_{k+1}^2}}. \tag{15}$$

It is positive whenever $\sigma_k < s_{k+1}$. Let $F(\rho)$ be the probability that a standard bivariate normal pair with correlation ρ lies in $[-b, b]^2$. For $0 \le \rho < 1$, denote its density by $\phi_2(x, y; \rho)$. Direct differentiation gives $\partial_\rho \phi_2 = \partial_x \partial_y \phi_2$. Integrating over $[-b, b]^2$ yields

$$\begin{aligned} F'(\rho) &= \phi_2(b, b; \rho) - \phi_2(b, -b; \rho) - \phi_2(-b, b; \rho) + \phi_2(-b, -b; \rho) \\ &= 2\phi_2(b, b; \rho)\left(1 - \exp\left[-\frac{2\rho b^2}{1 - \rho^2}\right]\right). \end{aligned}$$

Here we used $\phi_2(b, -b; \rho) = \phi_2(b, b; \rho)\exp[-2\rho b^2/(1 - \rho^2)]$. Thus $F(0) = q_b^2$ and $F'(\rho) > 0$ for $0 < \rho < 1$. At $\rho = 1$, the two coordinates are equal almost surely, so $F(1) = q_b > q_b^2$. It follows that $F(\rho) > q_b^2$ whenever $\rho > 0$.

Now suppose $\sigma_k < s_{k+1}$ for some k, so that $\rho_k > 0$. The pair $(Y_k/s_k, Y_{k+1}/s_{k+1})$ has standard normal marginals and correlation ρ_k, so $\Pr(A_k \cap A_{k+1}) = F(\rho_k)$. Applying Šidák (1967, Theorem 1) repeatedly to the coordinates outside this pair gives

$$\Pr\left(\bigcap_{i=0}^{K} A_i\right) \ge q_b^{K-1} F(\rho_k) > q_b^{K+1}.$$

Dividing by q_b shows that Equation (9) is strict. Therefore equality holds if and only if $\sigma_k = s_{k+1}$ for every k.

Finally, if $\sigma_k = 0$ for every k, then Equation (13) gives

$$Y_{k+1} = \frac{s_{k+1}}{s_k} Y_k, \qquad Y_k = \frac{s_k}{s_0} Y_0.$$

Hence $|Y_k| \le b s_k$ is equivalent to $|Y_0| \le b s_0$ for every k. Conditioned on A_0, the process remains inside with probability one. □

Equivalent normalized formulation. Define $R_k = Y_k/s_k$, which measures the same coordinate in units of the current noise standard deviation. Dividing Equation (13) by the *next* noise standard deviation s_{k+1} gives

$$\begin{aligned} R_{k+1} = \frac{Y_{k+1}}{s_{k+1}} &= \frac{\sqrt{s_{k+1}^2 - \sigma_k^2}}{s_k s_{k+1}} Y_k + \frac{\sigma_k}{s_{k+1}} Z_k \\ &= \sqrt{1 - \frac{\sigma_k^2}{s_{k+1}^2}}\, R_k + \frac{\sigma_k}{s_{k+1}} Z_k. \end{aligned}$$

With ρ_k defined in Equation (15), we have $\sigma_k/s_{k+1} = \sqrt{1 - \rho_k^2}$. The update can therefore be written as

$$R_{k+1} = \rho_k R_k + \sqrt{1 - \rho_k^2}\, Z_k, \qquad \rho_k = \sqrt{1 - \frac{\sigma_k^2}{1 - \alpha_{t_{k+1}}}}. \tag{16}$$

Before the first exit, $R_k = y^{(t_k)}/s_k$. The condition $|Y_k| \le b s_k$ is equivalent to $|R_k| \le b$, so normalization leaves the first-exit event unchanged. It makes the region $[-b, b]$ and the marginal distribution $\mathcal{N}(0, 1)$ the same at every step. These marginals refer to the unconditioned reference process, whereas Proposition 1 conditions on starting inside the region.

Nonzero averaged estimates. The choice of zero for the averaged estimate can be replaced by any fixed value m. Suppose $\hat{y}_0^{(t_k)} = m$ whenever $|y^{(t_k)} - \sqrt{\alpha_{t_k}}\, m| \le b s_k$. Inside this shifted region, the same substitution gives

$$y^{(t_{k+1})} - \sqrt{\alpha_{t_{k+1}}}\, m = \frac{\sqrt{s_{k+1}^2 - \sigma_k^2}}{s_k}\big(y^{(t_k)} - \sqrt{\alpha_{t_k}}\, m\big) + \sigma_k Z_k.$$

The proof then applies to regions $|y^{(t_k)} - \sqrt{\alpha_{t_k}}\, m| \le b s_k$, with initial law $y^{(t_0)} \sim \mathcal{N}(\sqrt{\alpha_{t_0}}\, m, s_0^2)$. Thus setting the midpoint to zero is a choice of coordinates. The assumption in Equation (6) is that the estimate remains at this fixed midpoint throughout the region.

C.2 Effect of the Denoising Schedule

We examine the dependence between the initial and final states of the reference process when the number of denoising steps increases. Consider DDPM sampling with the posterior variance, corresponding to $\eta = 1$ in the parameterization of Song et al. (2021). This is the stochastic transition used for CompDiffuser in Figure 2. Here, let $t_0 > \cdots > t_K$ be any denoising grid with $0 < \alpha_{t_0} < \cdots < \alpha_{t_K} < 1$. The grid contains K transitions, and adjacent selected times need not differ by one. Write σ_k for the standard deviation of the transition from t_k to t_{k+1}, with $R_0 \sim \mathcal{N}(0,1)$ independent of the mutually independent $Z_k \sim \mathcal{N}(0,1)$. Let $\ell_k = \log\big(\alpha_{t_k}/(1-\alpha_{t_k})\big)$ denote the log signal-to-noise ratio. For the reference process in Equation (16),

$$\mathrm{Corr}(R_0, R_K) = \prod_{k=0}^{K-1} \rho_k = e^{-(\ell_K - \ell_0)/2}. \tag{17}$$

The variance for a transition from t_k to t_{k+1} is

$$\sigma_k^2 = \frac{1-\alpha_{t_{k+1}}}{1-\alpha_{t_k}}\left(1 - \frac{\alpha_{t_k}}{\alpha_{t_{k+1}}}\right).$$

Substituting into Equation (16) gives

$$\begin{aligned}\rho_k^2 &= 1 - \frac{\sigma_k^2}{1-\alpha_{t_{k+1}}} \\ &= \frac{\alpha_{t_k}(1-\alpha_{t_{k+1}})}{\alpha_{t_{k+1}}(1-\alpha_{t_k})} \\ &= e^{-(\ell_{k+1}-\ell_k)}.\end{aligned}$$

Since $\rho_k \ge 0$, we have $\rho_k = e^{-(\ell_{k+1}-\ell_k)/2}$.

For $i \le j$, the independence of Z_j from the previous states gives

$$\mathrm{Cov}(R_i, R_{j+1}) = \rho_j\, \mathrm{Cov}(R_i, R_j).$$

Iterating this identity and using the unit marginal variances yields, for $i < j$,

$$\mathrm{Corr}(R_i, R_j) = \prod_{m=i}^{j-1} \rho_m = e^{-(\ell_j - \ell_i)/2}.$$

Taking $i = 0$ and $j = K$ proves Equation (17). The endpoint correlation depends only on the initial and final noise levels, so refining the grid between them leaves it unchanged. This provides a possible explanation for the limited improvement in Figure 2d, although the probability of escape can still change with the intermediate noise levels.

D Extended Related Work

Trajectory Stitching and Horizon Extension. Trajectory stitching can extend the training data as well as the plans generated at inference time. SCoTS (Lee & Choi, 2025b) constructs a dataset of longer trajectories from short segments for training a long-horizon planner. Extendable Planning via Multiscale Diffusion (Chen et al., 2026) combines progressive trajectory extension with hierarchical planning across temporal scales. These approaches address the limited horizon of the training trajectories. MaxCD operates with pretrained local planners at inference time and requires no additional training data or retraining.

Plan Refinement and Composition. Plan-refinement methods use different signals to identify and correct infeasible trajectories. RGG (Lee et al., 2023a) learns to predict the restoration gap and uses its guidance to refine sampled plans, with an attribution-map regularizer to stabilize the guidance. LoMAP (Lee & Choi, 2025a) uses local approximations of the data manifold to project intermediate samples. ECD (Sun et al., 2026) derives a global energy from local bridge potentials and uses its negative gradient, including the dependence of the interior of each plan on its boundaries. MaxCD examines how the transition between noise levels affects mode-averaging while retaining the denoised estimates and composition rule of the base sampler.

Compositional Image Generation. Pretrained image models can generate outputs beyond their training resolution by composing overlapping windows. MultiDiffusion (Bar-Tal et al., 2023) fuses local diffusion updates through an optimization over the shared image. DiffCollage (Zhang et al., 2023) represents large outputs with a factor graph and composes local scores. SyncDiffusion (Lee et al., 2023b) uses gradients of a perceptual loss between denoised windows to improve global consistency. SyncTweedies (Kim et al., 2024) synchronizes the outputs of Tweedie's formula in a canonical space shared by multiple diffusion processes. MaxCD retains the denoised estimates and composition rule of the base sampler, and applies the spectral split to spatial frequencies when generating panoramas.

Stochastic Sampling and Re-noising. Karras et al. (2022) explain how the Langevin component of stochastic sampling can correct errors accumulated in earlier steps. SDEdit (Meng et al., 2022) adds noise to an input image and denoises it with a pretrained model to balance fidelity to the input and realism. RePaint (Lugmayr et al., 2022) repeatedly re-noises and denoises samples while conditioning on the known image regions. Restart sampling (Xu et al., 2023) alternates additional forward noising steps with backward ODE sampling to reduce accumulated errors. Our analysis focuses on the persistence of mode-averaging in compositional planning.

E Implementation Details

E.1 Environments and Datasets

We use the `stitch` and `play` datasets from OGBench (Park et al., 2025), a benchmark for offline goal-conditioned reinforcement learning. The environments and evaluation settings are summarized in Table 5.

Locomotion. We evaluate maze navigation in `pointmaze`, `antmaze`, and `humanoidmaze` across `Medium`, `Large`, and `Giant` layouts. The agent must reach a specified goal position, while the `stitch` datasets provide short navigation segments that must be combined to cover longer distances.

Object Manipulation. In `cube`, a robot arm arranges one to four cubes into target configurations through pick-and-place actions. The `play` datasets contain unstructured object interactions. We also evaluate `antsoccer`, where an ant must approach a ball and dribble it to a goal. Its `stitch` datasets contain separate navigation and dribbling trajectories, requiring these behaviors to be combined during evaluation.

Table 5: **OGBench environments and evaluation settings.** Episode limits are measured in environment steps, and M is the initial number of composed local plans.

Environment	Type	Size	Dataset Name	Max. Steps	M
pointmaze	stitch	Medium	pointmaze-medium-stitch-v0	1000	3
		Large	pointmaze-large-stitch-v0	1000	6
		Giant	pointmaze-giant-stitch-v0	1000	8
antmaze	stitch	Medium	antmaze-medium-stitch-v0	1000	3
		Large	antmaze-large-stitch-v0	1000	6
		Giant	antmaze-giant-stitch-v0	2000	9
humanoid maze	stitch	Medium	humanoidmaze-medium-stitch-v0	5000	4
		Large	humanoidmaze-large-stitch-v0	5000	6
		Giant	humanoidmaze-giant-stitch-v0	8000	11
antsoccer	stitch	Arena	antsoccer-arena-stitch-v0	5000	5
		Medium	antsoccer-medium-stitch-v0	5000	6
cube	play	Single	cube-single-play-v0	2000	2
		Double	cube-double-play-v0	2000	5
		Triple	cube-triple-play-v0	2000	10
		Quadruple	cube-quadruple-play-v0	2000	10

E.2 Pretrained Planners and Low-Level Controllers

We build on the public CompDiffuser implementation[1] (Luo et al., 2025). For each environment, the compositional planners use the same pretrained local model, whose parameters remain fixed during evaluation. The 2D `pointmaze`, `antmaze`, and `humanoidmaze` planners use conditional temporal U-Nets. The 15D and 29D `antmaze` planners and the 17D `antsoccer` planner use DiT-based denoisers. The models encode the noisy overlaps of neighboring local plans as conditioning inputs and directly predict the clean local plan. For the sampling transitions, we recover the corresponding noise prediction from the clean estimate using Equation (2).

`PointMaze`, `AntMaze`, and `HumanoidMaze` planning in Table 1 uses the 2D x-y position. The spectral split ablation in Table 4 additionally uses the 15D `antmaze` state, which includes joint positions, and the full 29D state, which also includes velocities. The 17D `antsoccer` state contains the x-y positions of the ant and ball and the joint positions of the ant. Local plan lengths and overlap lengths are listed in Table 6. For `cube`, we use a DiT-based planner on the full state, including robot arm joint positions and cube poses.

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

The generated state trajectories are executed by low-level controllers. Following Luo et al. (2025), we use a proportional-derivative (PD) controller for `pointmaze` and learned inverse dynamics models for `antmaze`, `humanoidmaze`, and `antsoccer`. These inverse dynamics models map the current observation and a planned subgoal to an action. For `cube`, we use Diffusion Q-Learning (DQL) policies (Wang et al., 2022) conditioned on the current state and a planned subgoal.

E.3 Evaluation Protocol

We evaluate on the 5 tasks provided for each OGBench environment (Park et al., 2025). For each random seed, we run 20 episodes per task and average the success rate over all 100 episodes. Success is determined by the goal-reaching criterion of the environment. We report the mean and standard deviation of these success rates across 5 random seeds. The base compositional sampler uses 50 DDIM steps with classifier-free guidance weight 2. Initial candidate budgets are listed in Table 6. We select the candidate with the smallest average discrepancy between overlapping local plans and merge its overlaps by exponential blending, following Luo et al. (2025). Method-specific sampling settings are described in Section E.5. During execution, we use the adaptive replanning procedure of CompDiffuser. A new plan is generated from the current state when the distance to the current subgoal exceeds a prescribed threshold or the current plan is exhausted. The number of local plans is reduced according to the executed portion of the trajectory as replanning proceeds. The episode ends when the goal is reached or the environment-step limit in Table 5 is reached.

E.4 Hyperparameters of MaxCD

The two hyperparameters of MaxCD are the noise-level threshold γ and the spectral bandwidth ρ. We use a single default configuration, $\gamma = 0.2$ and $\rho = 0.25$, for OGBench and panorama generation. Table 6 summarizes these values together with the sampler settings for each environment. In the ablations of Section F.5, we vary one parameter at a time while keeping the other at its default.

For planning, we apply the spectral transition on the overlaps of local plans with two neighbors when $t/T > \gamma$, and retain the original transition on the remaining coordinates and at lower noise levels. The projection $\mathcal{L}_\rho$ is implemented by a discrete Fourier transform along the temporal axis, independently for each state coordinate, with frequencies normalized by the Nyquist frequency. For panoramas, we average overlapping window noise predictions to obtain a noise prediction for the shared panorama latent. We apply the spectral transition to this shared latent using a two-dimensional Fourier transform and a radial low-frequency mask, with frequencies normalized by the Nyquist frequency.

Table 6: **Default sampler and MaxCD settings by environment.** Local lengths and overlaps count states for planning. Panorama window sizes are in pixels, and overlap is horizontal. AntMaze settings apply to the 2D, 15D, and 29D planners. Cube uses 16 candidates for Single and Double, and 8 for Triple and Quadruple.

Parameter	**PointMaze**	**AntMaze**	**Humanoid Maze**	**Cube**	**AntSoccer**	**Panorama**
Local length / window	160	160	336	160	160	512×512
Overlap	56	56	128	56	56	50%
DDIM steps	50	50	50	50	50	50
Guidance weight	2.0	2.0	2.0	2.0	2.0	7.5
Base DDIM η	1.0	1.0	1.0	1.0	1.0	0.0
Initial candidates	40	40	40	16 / 8	40	1
Noise-level threshold γ	**0.2**	**0.2**	**0.2**	**0.2**	**0.2**	**0.2**
Spectral bandwidth ρ	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**	**0.25**

E.5 Baseline Implementations

GSC and CompDiffuser. We use the CompDiffuser implementation and pretrained local models described in Section E.2. GSC composes the same local planners by averaging their noise predictions on the overlaps.

CDGS. We use the official CDGS implementation[2] with the same pretrained local planners. CDGS searches over candidate plans through iterative resampling and likelihood-based pruning, which require additional denoiser evaluations.

ECD. We apply the bridge and boundary-reaction corrections from the official ECD implementation[3] after each local sampling update. For the 2D maze planners, the boundary reaction uses a Laplacian approximation computed by a tridiagonal solve along the planning horizon. For panoramas, the correction operates along the horizontal axis of each window.

SyncDiffusion. For panorama generation, SyncDiffusion (Lee et al., 2023b) applies gradients of an LPIPS loss between decoded denoised window predictions to encourage perceptual consistency.

E.6 Panorama Protocol and Metrics

Following Mishra et al. (2026), we generate 512×4608 panoramas with Stable Diffusion 2.0 base (Rombach et al., 2022) by composing 17 windows of 512×512 with 50% horizontal overlap. The quantitative evaluation uses 14 prompts and 10 random seeds per prompt, yielding 140 panoramas per method. All methods use the same prompts and seeds, a 50-step diffusion schedule, and classifier-free guidance weight 7.5.

Following SyncDiffusion (Lee et al., 2023b), we divide each panorama into 9 non-overlapping 512×512 crops for evaluation. Intra-LPIPS (Zhang et al., 2018) measures perceptual consistency as the mean VGG-LPIPS distance over all 36 crop pairs. Intra-Style-L (Gatys et al., 2016) measures style consistency as the mean VGG-19 Gram-matrix loss over these pairs. We report the latter in units of 10^{-3}. CLIP-S (Radford et al., 2021; Hessel et al., 2021) measures alignment with the text prompt using ViT-B/16 CLIP scores averaged over the crops. For each quality metric, Table 3 reports the mean and standard deviation across the 140 panoramas.

Generation time is reported as the mean and standard deviation over 10 runs for each method.

E.7 Compute Resources

All planning evaluations and panorama generation experiments use a single NVIDIA RTX A6000 GPU.

[2] https://github.com/UtkarshMishra04/CDGS_ogbench

[3] https://github.com/GradientSpaces/ECD

F Additional Results

F.1 Visual Comparison of Denoising Processes

Figure 7 compares the denoising processes of CompDiffuser and MaxCD on Task 4 of `AntMaze-Giant-Stitch`, using the same pretrained local model and start and goal conditions. As the noise level decreases, CompDiffuser retains an infeasible trajectory through maze walls, illustrating persistent mode-averaging. In MaxCD, maximal-noise transitions allow the overlap states to be revised through denoising conditioned on neighboring local plans, as described in Section 3.3. At low noise levels, the original transition refines the trajectory into a coherent plan along the maze corridors.

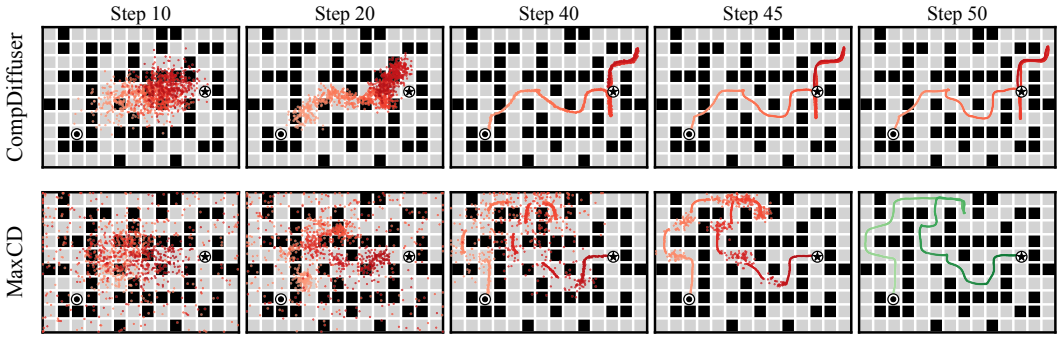


Figure 7: **Resolving mode-averaging with maximal-noise transitions.** CompDiffuser (top) and MaxCD (bottom) on `AntMaze-Giant-Stitch` Task 4 with the same start and goal. Columns show noisy trajectory states after the indicated denoising steps and the final plans at step 50. CompDiffuser retains an infeasible trajectory through walls, while MaxCD produces a coherent plan along the corridors. Invalid plans are red and valid plans are green.

F.2 Additional Plan Quality Visualizations

Figure 8 extends the Task 4 comparison in Figure 5 to all 5 tasks of `AntMaze-Giant-Stitch`, using the same MaxCD setting across tasks. In these visualizations, MaxCD produces a higher valid plan rate than CompDiffuser on every task, including Task 3, where the rate increases from 5% to 60%. Figure 9 provides the corresponding four-method comparisons on all 5 tasks of `PointMaze-Giant-Stitch`.

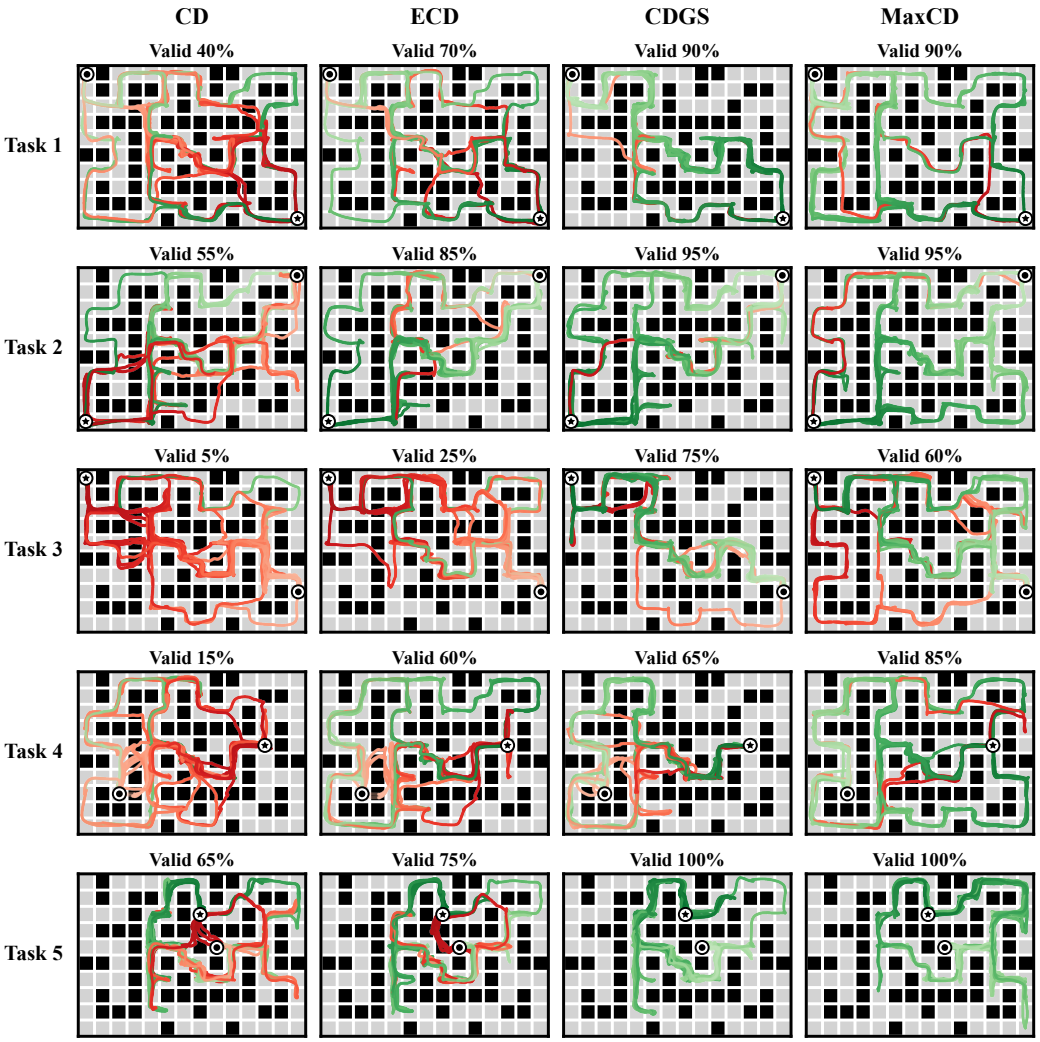


Figure 8: **Additional plan quality comparisons on `AntMaze-Giant-Stitch`.** Rows show Tasks 1–5, and columns show CD, ECD, CDGS, and MaxCD from left to right. Each panel shows 20 plans. Plans crossing walls are red, and valid plans are green. Percentages indicate the valid plan rate.

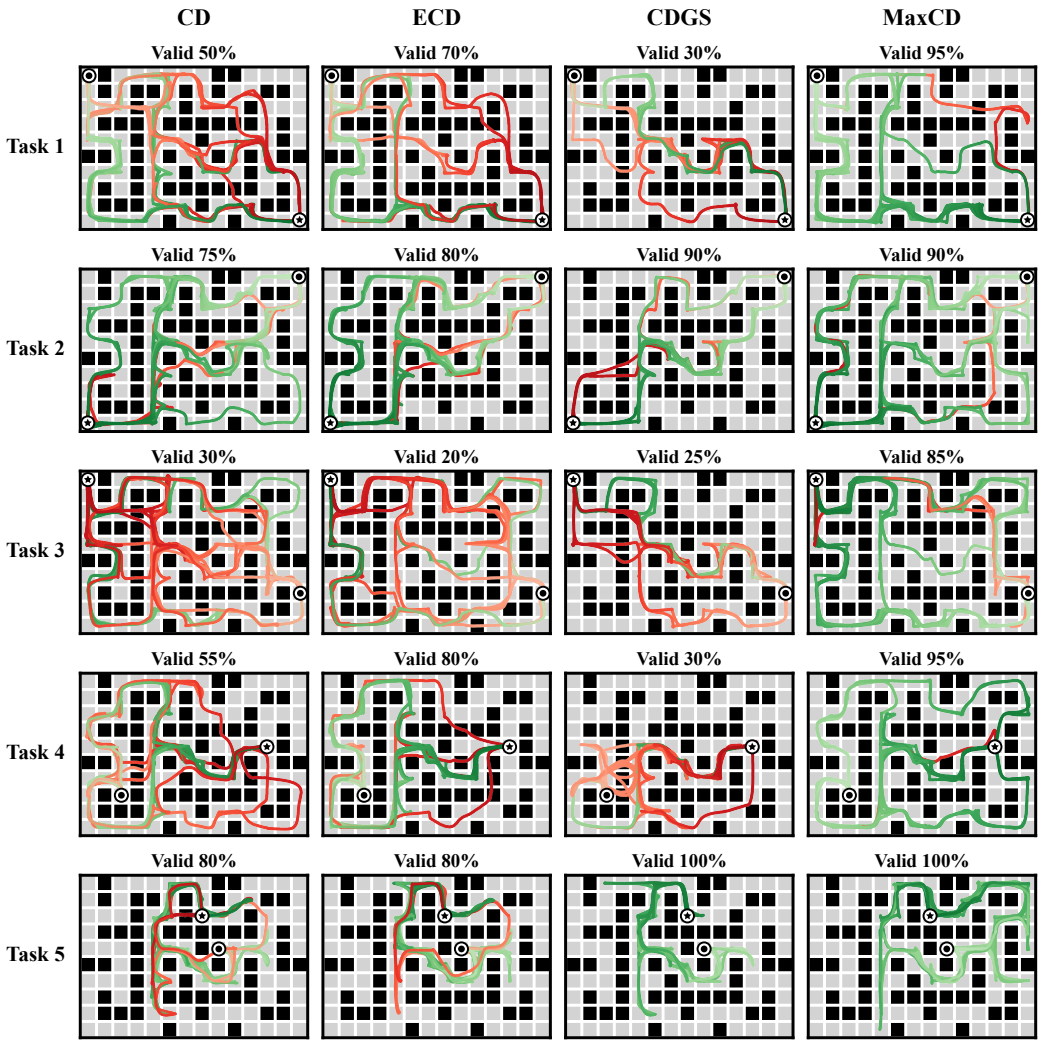


Figure 9: **Additional plan quality comparisons on `PointMaze-Giant-Stitch`.** Rows show Tasks 1–5, and columns show CD, ECD, CDGS, and MaxCD from left to right. Each panel shows 20 plans. Plans crossing walls are red, and valid plans are green. Percentages indicate the valid plan rate.

F.3 Full-State Plan Visualizations

Figure 10 compares full-state plans generated by CompDiffuser and MaxCD on `AntMaze-Large-Stitch`, using the 29D pretrained local model evaluated in Table 4.

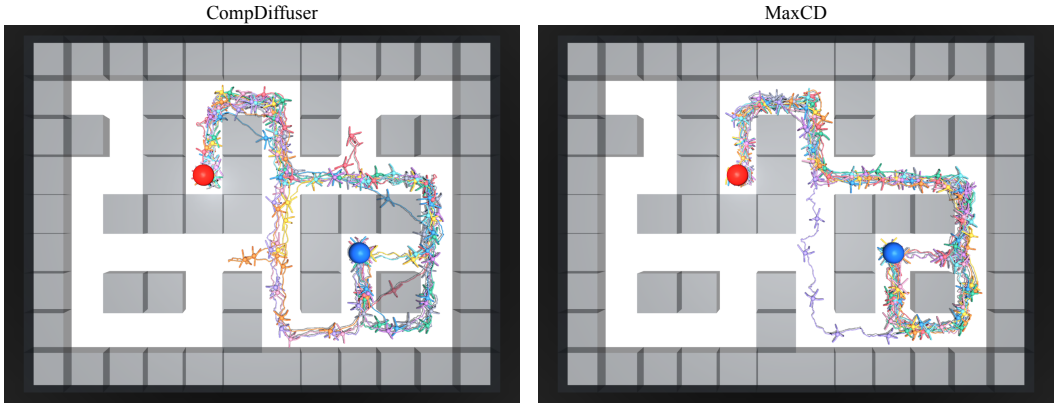


Figure 10: **Full-state plans on `AntMaze-Large-Stitch`.** Ten plans per method on Task 4, with a different color for each plan. Applying maximal-noise transitions helps generate coherent long-horizon plans that follow the maze corridors. Ant configurations are reconstructed from the predicted positions and joint angles at regularly spaced locations along each path. Blue and red spheres mark the start and goal cells.

F.4 Cube Manipulation Rollouts

Figures 11 and 12 show `Cube-Triple` rollouts executed with MaxCD, complementing the success rates in Table 2. A goal-conditioned DQL controller executes the composed plans, as described in Section E.2. The arm moves the cubes to their target positions in Figure 11 and stacks them in Figure 12.

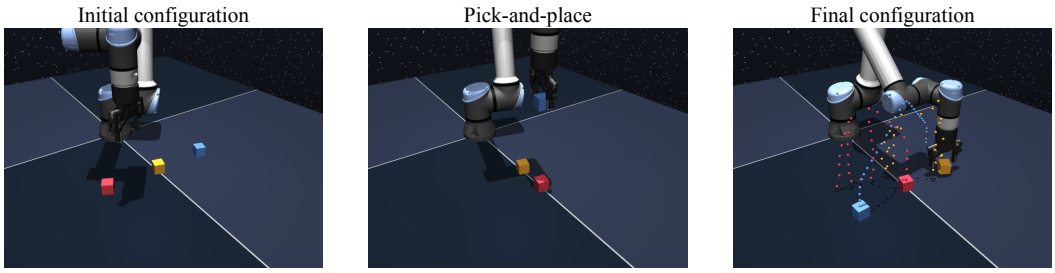


Figure 11: **Cube manipulation with MaxCD.** Initial, intermediate, and final configurations on `Cube-Triple` Task 2. Colored points in the final panel trace the executed cube trajectories.

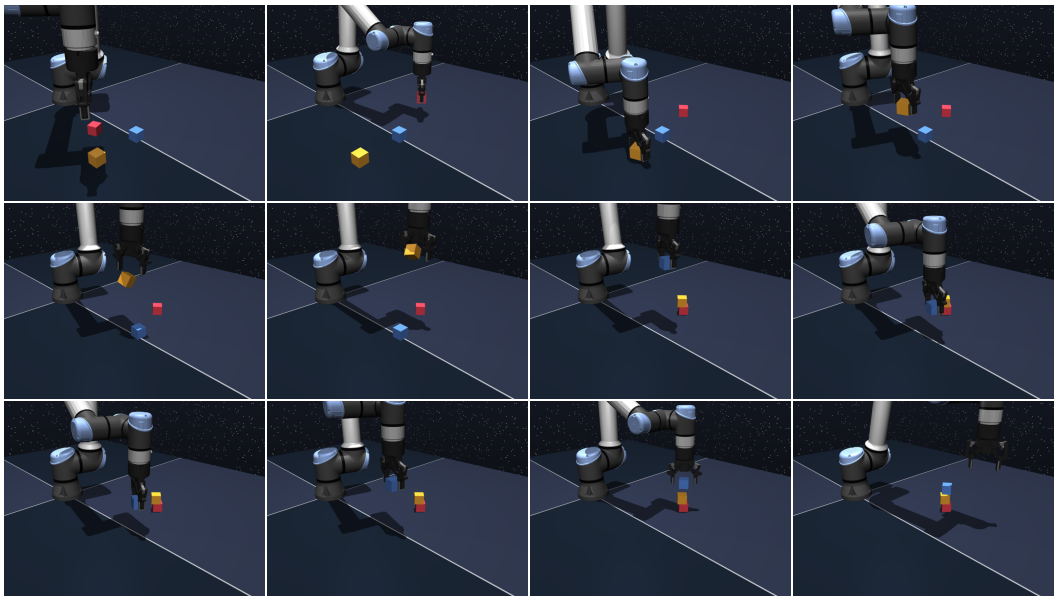


Figure 12: **Stacking three cubes with MaxCD.** Frames from a `Cube-Triple` Task 5 rollout, ordered from left to right and top to bottom. The UR5e arm stacks three cubes through pick-and-place actions.

F.5 Additional Ablation Studies

We vary γ and ρ one at a time, keeping the other at its default in Table 6. Both ablations use the `Stitch` datasets, with 2D position planners on `PointMaze-Giant` and `AntMaze-Giant`, and 15D and 29D state planners on `AntMaze-Large`. The default values, $\gamma = 0.2$ and $\rho = 0.25$, are shared across OGBench and panorama generation. The highest mean success rates in each row are shown in **bold**, including ties.

Noise-Level Threshold. Table 7 compares $\gamma \in \{0, 0.1, 0.2, 0.3\}$ with $\rho = 0.25$ fixed. Larger values of γ leave more final denoising steps for the original transition, while $\gamma = 0$ applies maximal-noise transitions throughout. This comparison examines how retaining the original transition at low noise levels affects planning performance.

Table 7: **Ablation on the noise-level threshold γ.** Success rates on `Stitch` datasets, with $\rho = 0.25$. State dimensions are shown beside each environment. The last row averages the four settings. Best mean success rates are bold.

Environment	**CD**	$\gamma = 0$	0.1	0.2 (default)	0.3
`PointMaze-Giant` (2D)	$69_{\pm 3}$	$92_{\pm 2}$	$90_{\pm 2}$	$94_{\pm 2}$	$\mathbf{95}_{\pm 2}$
`AntMaze-Giant` (2D)	$67_{\pm 3}$	$80_{\pm 2}$	$79_{\pm 2}$	$\mathbf{88}_{\pm 2}$	$86_{\pm 2}$
`AntMaze-Large` (15D)	$63_{\pm 4}$	$69_{\pm 3}$	$\mathbf{77}_{\pm 3}$	$\mathbf{77}_{\pm 3}$	$67_{\pm 3}$
`AntMaze-Large` (29D)	$65_{\pm 5}$	$69_{\pm 4}$	$62_{\pm 4}$	$\mathbf{71}_{\pm 4}$	$\mathbf{71}_{\pm 4}$
Mean	66.0	77.5	77.0	**82.5**	79.8

Spectral Bandwidth. Table 8 evaluates $\rho \in \{0.1, 0.25, 0.5, 1\}$ with $\gamma = 0.2$ fixed. At $\rho = 1$, the transition is applied to all frequency components, recovering MaxCD without the spectral split in Table 4.

Table 8: **Ablation on the spectral bandwidth ρ.** Success rates on `Stitch` datasets, with $\gamma = 0.2$. State dimensions are shown beside each environment. The last row averages the four settings. Best mean success rates are bold.

Environment	**CD**	$\rho = 0.1$	0.25 (default)	0.5	1
`PointMaze-Giant` (2D)	$69_{\pm 3}$	$\mathbf{94}_{\pm 2}$	$\mathbf{94}_{\pm 2}$	$92_{\pm 2}$	$93_{\pm 2}$
`AntMaze-Giant` (2D)	$67_{\pm 3}$	$87_{\pm 2}$	$\mathbf{88}_{\pm 2}$	$84_{\pm 2}$	$83_{\pm 2}$
`AntMaze-Large` (15D)	$63_{\pm 4}$	$70_{\pm 3}$	$\mathbf{77}_{\pm 3}$	$76_{\pm 3}$	$65_{\pm 5}$
`AntMaze-Large` (29D)	$65_{\pm 5}$	$68_{\pm 4}$	$\mathbf{71}_{\pm 4}$	$66_{\pm 4}$	$66_{\pm 5}$
Mean	66.0	79.8	**82.5**	79.5	76.8

With the fixed configuration in Table 6, $\gamma = 0.2$ and $\rho = 0.25$, MaxCD improves planning performance over CD and panorama quality over GSC across diverse tasks without tuning these hyperparameters separately for each task. The improvements over CD are also maintained across most tested values of γ and ρ in Tables 7 and 8.

Overlap Restriction. We compare applying the spectral transition in Equation (11) to full local plans with restricting it to the overlaps of interior local plans as in Equation (12). Both variants use the same noise-level threshold and spectral bandwidth. In Table 9, overlap restriction increases the average success rate across the six environments from 89.7% to 94.3%. This restriction helps overlap states escape mode-averaging while preserving local plan quality through the original transition elsewhere.

Table 9: **Ablation on overlap restriction.** Success rates as mean $\pm$ standard deviation on the `Stitch` datasets. Both variants use $\gamma = 0.2$ and $\rho = 0.25$. The last row averages the six environment means.

Environment	Full local plans	Overlaps (MaxCD)
`PointMaze-Medium`	$\mathbf{100}_{\pm 0}$	$\mathbf{100}_{\pm 0}$
`PointMaze-Large`	$94_{\pm 4}$	$\mathbf{100}_{\pm 0}$
`PointMaze-Giant`	$\mathbf{95}_{\pm 2}$	$94_{\pm 2}$
`AntMaze-Medium`	$92_{\pm 2}$	$\mathbf{97}_{\pm 2}$
`AntMaze-Large`	$71_{\pm 4}$	$\mathbf{87}_{\pm 2}$
`AntMaze-Giant`	$86_{\pm 2}$	$\mathbf{88}_{\pm 2}$
Mean	89.7	**94.3**

F.6 Planning-Time Comparison

Table 10 compares CD, CDGS, ECD, and MaxCD on the `Stitch` datasets of `PointMaze-Giant` and `AntMaze-Giant`, with planning time measured on the same GPU. Each timing includes sampling, candidate selection, and trajectory blending for the same budget of 40 candidates. CDGS requires additional denoiser evaluations for iterative resampling and DDIM inversion to rank candidate plans (Mishra et al., 2026). ECD adds bridge and boundary-reaction corrections, with the latter computed through a tridiagonal solve in the 2D maze experiments (Sun et al., 2026). MaxCD reuses the denoised estimates from CD and changes only the transition, using noise sampling and Fourier transforms for the spectral split without additional denoiser evaluations. On these tasks, MaxCD achieves higher success rates with only 7–10% more planning time than CD, while reducing planning time by approximately 78% relative to CDGS and 26–29% relative to ECD.

Table 10: **Planning-time comparison.** Success rates from Table 1 and wall-clock time per plan on `PointMaze-Giant-Stitch` and `AntMaze-Giant-Stitch`. Times are mean $\pm$ standard deviation over 10 planning calls on the same GPU.

Method	`PointMaze-Giant`		`AntMaze-Giant`	
	Success (%)	**Time (s)**	**Success (%)**	**Time (s)**
CD	$69_{\pm 3}$	$\mathbf{9.13}_{\pm 0.19}$	$67_{\pm 3}$	$\mathbf{10.55}_{\pm 0.20}$
CDGS	$74_{\pm 3}$	$46.21_{\pm 0.95}$	$83_{\pm 3}$	$50.36_{\pm 0.77}$
ECD	$81_{\pm 2}$	$13.59_{\pm 0.25}$	$81_{\pm 2}$	$15.87_{\pm 0.26}$
MaxCD	$\mathbf{94}_{\pm 2}$	$10.00_{\pm 0.25}$	$\mathbf{88}_{\pm 2}$	$11.32_{\pm 0.28}$

F.7 Additional Qualitative Results

Figure 13 compares the baselines on the landscape and corgi prompts of Figure 6, using the same random seed for all methods. Figure 14 shows additional panoramas generated by MaxCD.

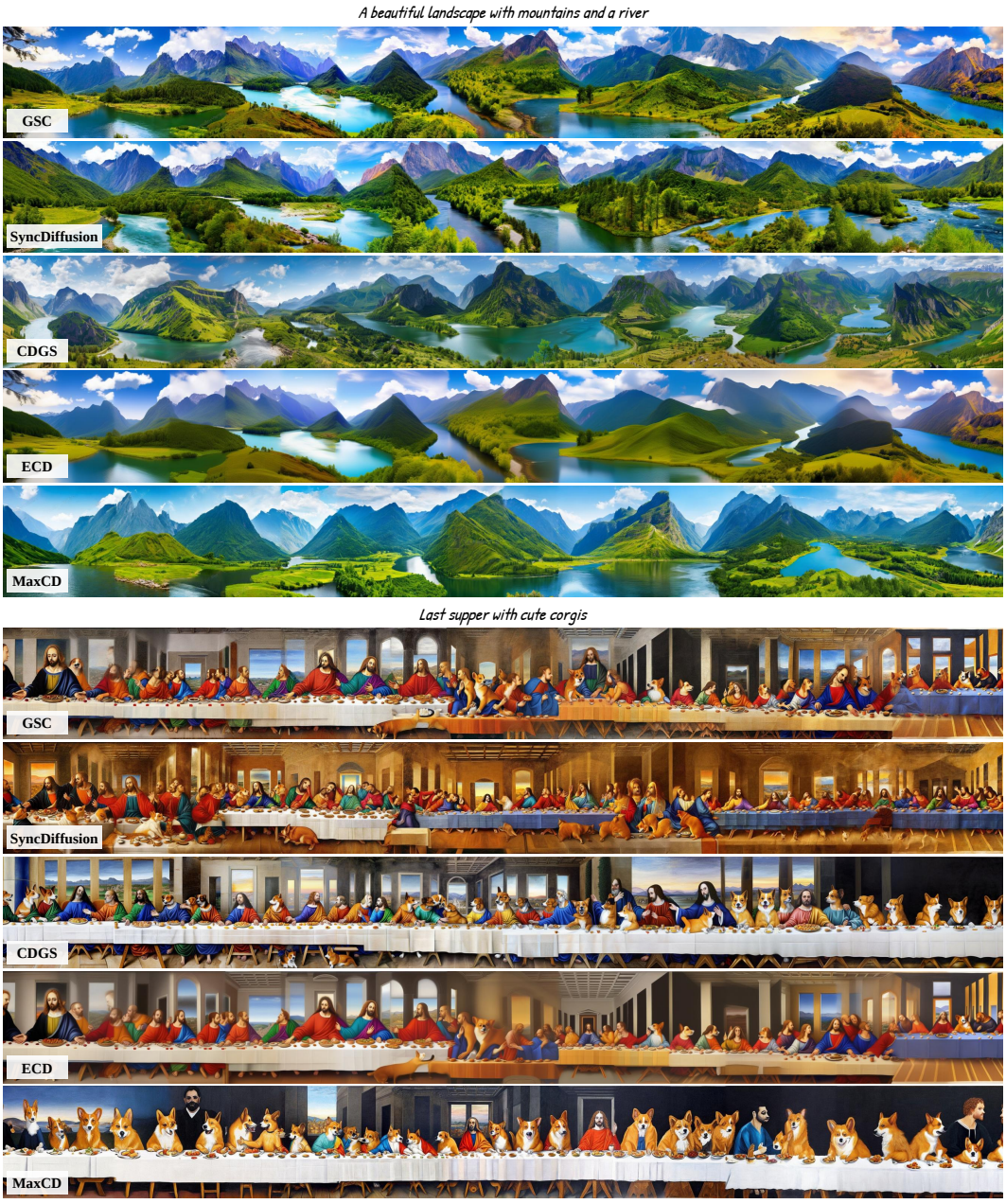


Figure 13: **Additional panorama comparisons.** Landscape and corgi examples for GSC, SyncDiffusion, CDGS, ECD, and MaxCD. All methods use the same prompt and random seed within each example.

A beautiful landscape with mountains and a river



A photo of the dolomites with red lava flowing through the valley



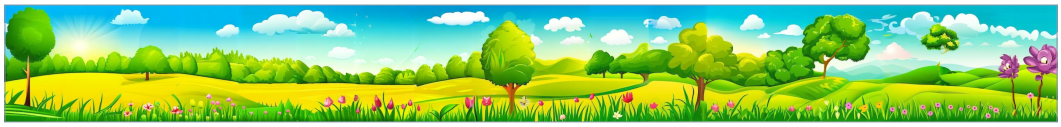
A photo of lush forest with a babbling brook



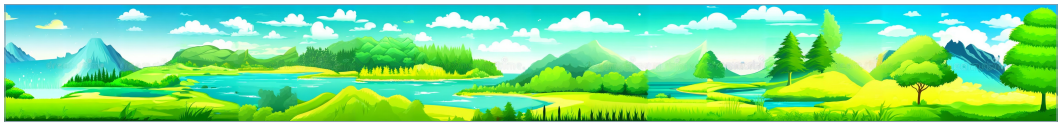
A photo of a forest with a misty fog



Cartoon panorama of spring summer beautiful nature



Natural landscape in anime style illustration



Skyline of a futuristic city with flying cars



A cinematic view of a castle in the sunset



Figure 14: **Additional panoramas generated by MaxCD.** Each panorama is shown with its text prompt.

G BASELINE RESULT SOURCES

Scores for GCBC, GCIVL, GCIQL, QRL, CRL, and HIQL in Tables 1 and 2 are taken from Table 2 of Park et al. (2025). Results for GSC, CD, CDGS, ECD, and MaxCD are obtained from our own evaluations following the protocol in Section E.3.