Appendix

1 Incorporating Guidance During Sampling

In an unconditional diffusion model, test-time guidance [4, 9] constrains trajectories to specific environments and start/goal configurations. While prior works [2, 3] rely on test-time guidance for collision avoidance and endpoint constraints, we use only conditional diffusion models and trajectory optimization for strict constraint satisfaction. Here, we also include an ablation study on the complementary use of guidance steps during sampling to enhance motion planning performance.

Guidance Function Implementation As in MPD [2], the guidance function includes collision and smoothness costs (same as Section IV-B). Collision costs are computed with cuRobo [7], and smoothness costs use a Gaussian Process prior [1, 2]. To ensure stability during guidance, we first smooth the sampled trajectory with a Gaussian kernel ($\sigma = 4.0$) before computing costs, allowing collision-cost gradients to affect neighboring points. We then compute the clamped collision cost ($d_{\texttt{max}} = 0.1$) and the smoothness cost, summing them as $k_{\text{smooth}} c_{\text{smooth}} + k_{\text{coll}} c_{\text{coll}}$, with $k_{\text{smooth}} = 1e-9$ and $k_{\text{coll}} = 1e-2$. Gradients are computed with PyTorch [6], clamped ($g_{\texttt{max}} = 1.0$) to prevent erratic updates, zeroed at endpoints, and added directly to the trajectory.

Results Figure 6 compares our model with a variant that includes guidance steps during denoising iterations. Overall, guided-sampling enhances the quality of initial trajectories (black dots represent PRESTO without guidance, and gray dots represent PRESTO with guidance). For example, the success rate of the denoised trajectory before optimization reaches 92.8% in Level 4, compared to 51.1% for PRESTO without guidance. However, incorporating guidance requires gradient evaluations for the costs at each diffusion iteration, resulting in computational overhead. In Level 3, this overhead accumulates to an average of 0.38 seconds, indicating that the added cost of guidance steps may occasionally degrade performance within a given time frame. Despite this, guidance generally improves performance across Levels 1-4 in terms of all three metrics: success rate, collision rate, and penetration depth, given the same number of trajectory optimization iterations.

We also report the effects of guidance on variants of our model in Figure 5. Performance across all baselines improves when guidance steps are applied compared to the original results in Figure 4. Notably, the success rate gap between PRESTO and its variants widens with the application of guidance. For instance, across Level 1-4, the gap

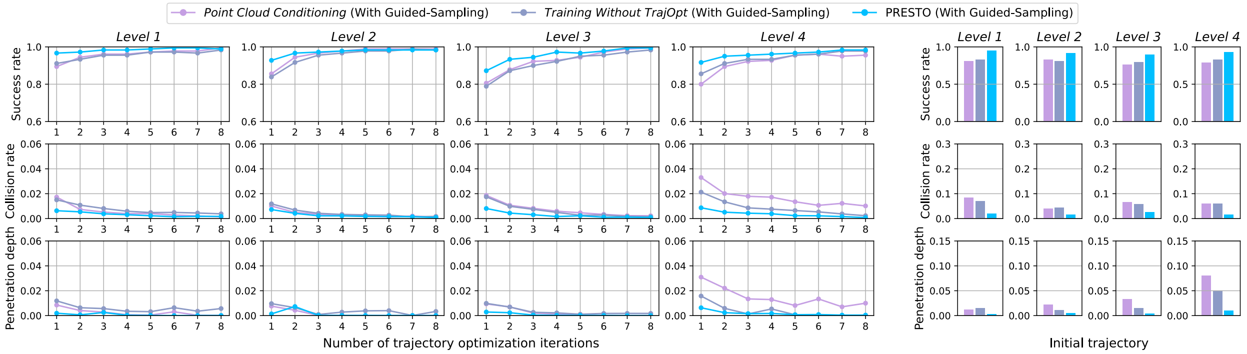

Figure 5: Ablation studies with guided-sampling. We report the success rate (%), collision rate (%), and penetration depth (m) averaged across 180 problems for PRESTO and the self-variant baselines with guided-sampling. (**Left**) We show performance changes with varying post-processing iterations. (**Right**) We present performance of trajectories directly generated by the diffusion models, without post-processing.

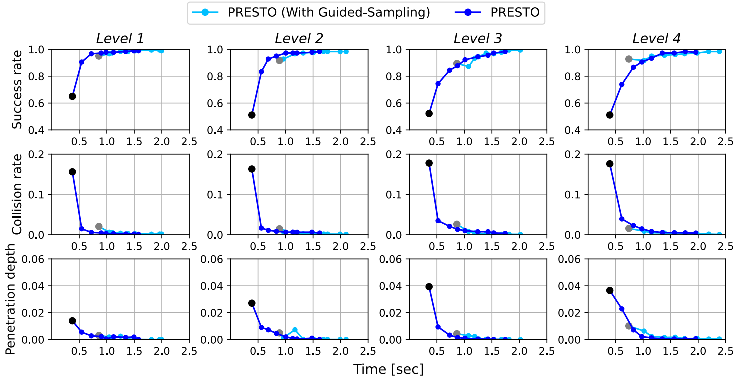

Figure 6: Results with guided-sampling. We report the success rate (%), collision rate (%), and penetration depth (m) averaged across 180 problems for PRESTO and the self-variant baselines. Black dots represent PRESTO without post-processing, while gray dots represent PRESTO with Guided Sampling, also without post-processing.

between PRESTO and the closest baseline (*Point-Cloud Conditioning*) increases from 2.4% to 4.0%. As PRESTO generates higher-quality initial trajectories with smaller penetration depths, spurious collisions are resolved with just a few guidance steps, leading to greater success rate gains compared to the ablations of PRESTO.

2 Point Cloud Encoder Architecture

For our ablation with point-cloud inputs (Section V-C), we design the point-cloud encoder based on recent patch-based transformers [5, 8]. We divide the $\mathbb{R}^{1024\times 3}$ point cloud into 8 patches using farthest-point sampling and k-nearest neighbors ($k = 128$). Each patch is normalized, flattened, and projected into shape embeddings via a 3-layer MLP with GeLU and Layer Normalization. Positional embeddings, computed from patch centers using a 2-layer MLP, are added before processing with a 4-layer transformer to extract geometric features, which serve as additional input tokens for the DiT in the diffusion model.

References

[1] T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression,” in *Robotics: Science and Systems*, 2014.

[2] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, “Motion planning diffusion: Learning and planning of robot motions with diffusion models,” *arXiv preprint arXiv:2308.01557*, 2023.

[3] S. Huang *et al.*, “Diffusion-based generation, optimization, and planning in 3d scenes,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[4] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” in *International Conference on Machine Learning*, 2022.

[5] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, “Masked autoencoders for point cloud self-supervised learning,” in *European Conference on Computer Vision*, 2022.

[6] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, 2019.

[7] B. Sundaralingam *et al.*, “cuRobo: Parallelized collision-free minimum-jerk robot motion generation,” *arXiv preprint arXiv:2310.17274*, 2023.

[8] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, “Point-bert: Pre-training 3d point cloud transformers with masked point modeling,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[9] M. Zhang *et al.*, “Motiondiffuse: Text-driven human motion generation with diffusion model,” *arXiv preprint arXiv:2208.15001*, 2022.