PRESTO:

Fast Motion Planning Using Diffusion Models Based on Key-Configuration Environment Representation

Mingyo Seo*[1], Yoonyoung Cho*[2], Yoonchang Sung[1],
Peter Stone [1], Yuke Zhu†[1], and Beomjoon Kim†[2]

1 The University of Texas at Austin

2 KAIST

Motion Planning in Practical Environments

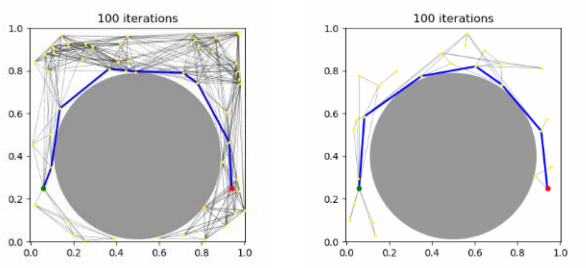
© Dexterity Inc.

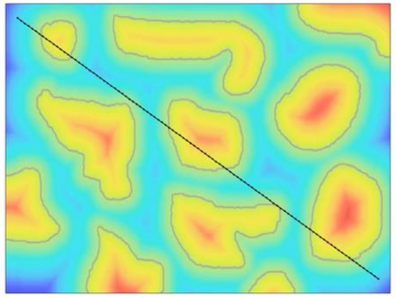
🤔 Complex geometry due to cluttered objects

🤔 Limited computational time

💡 Consistent fixture setup

Motivation: Challenges in Motion Planning



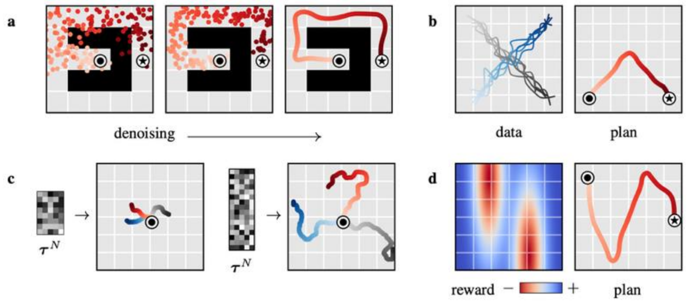


sample-based planners [1,2]

🔍 narrow-passage, high dimensionality

optimization-based planners [3]

🔍 dependence on seed trajectories, local minima



generative models [4]

✓ data-driven trajectory sampling priors

🔍 poor constraint handling, weak generalization

[1] Kavraki *et al.*, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, 1996

[2] LaValle *et al.*, Rapidly-exploring random trees: A new tool for path planning, 1998

[3] Schulman et al., Motion Planning with Sequential Convex Optimization and Convex Collision Checking, 2013

[4] Janner et al., Planning with Diffusion for Flexible Behavior Synthesis, 2022

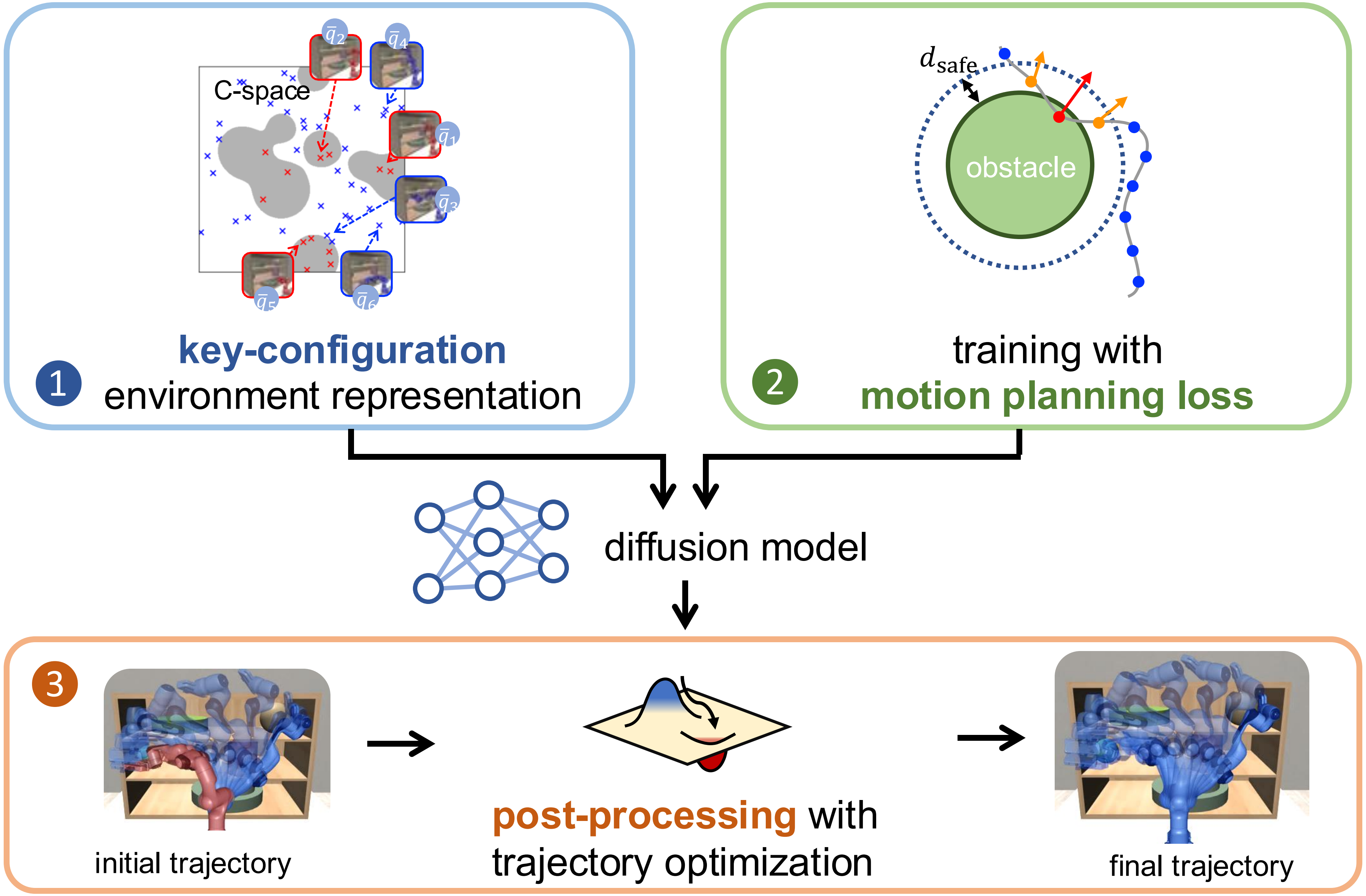
Research Questions

🤔 What are **good representations** for generalizing to unseen environments?

🤔 How can generative models **learn task constraints** like collision avoidance?

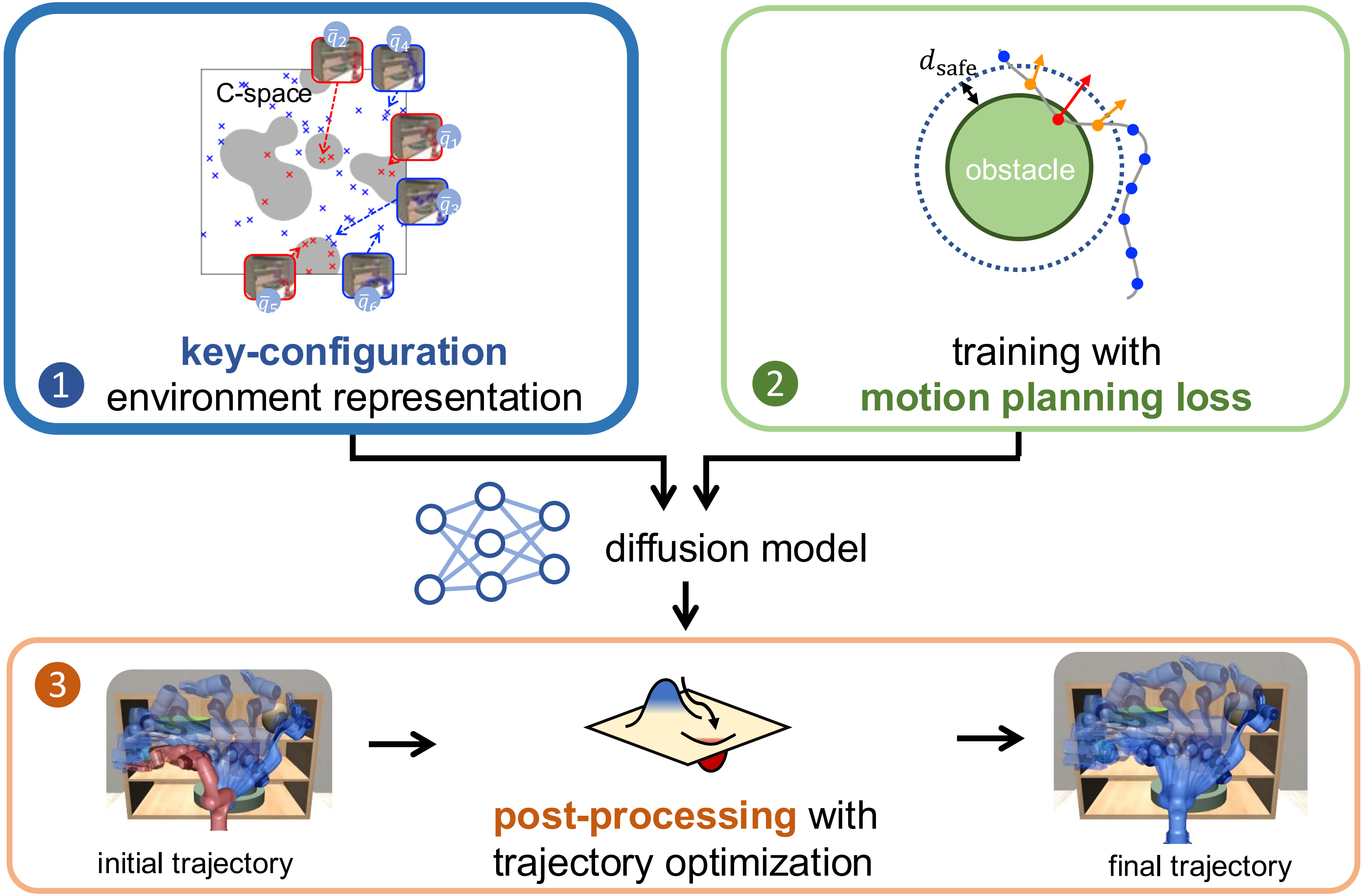
PRESTO:

Planning with Environment Representation, Sampling, and Trajectory Optimization



PRESTO:

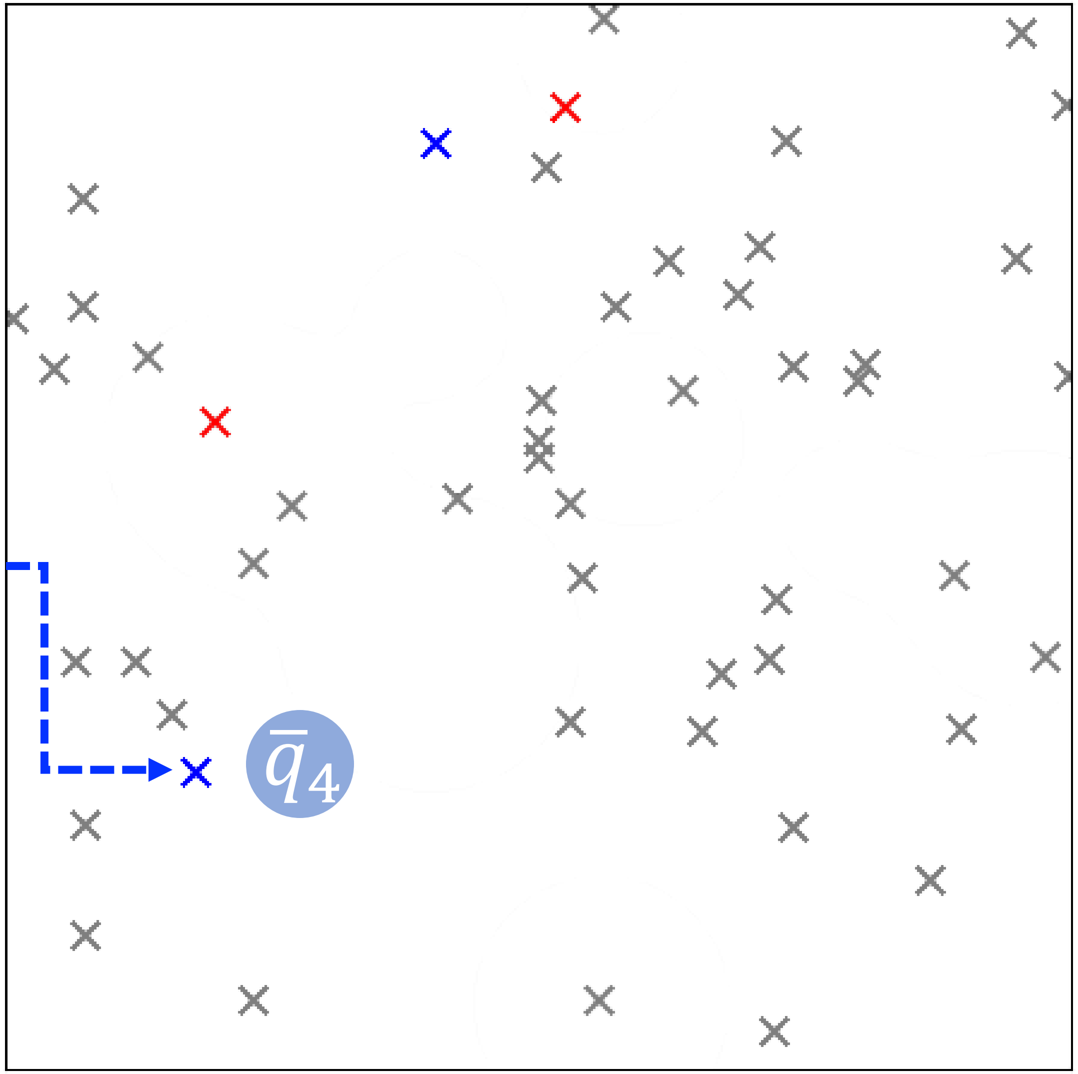
Planning with Environment Representation, Sampling, and Trajectory Optimization



Key-configuration environment representation

$\bar{q}_4$ $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ \\ \\ \end{bmatrix}$

collision-checking
at key configurations

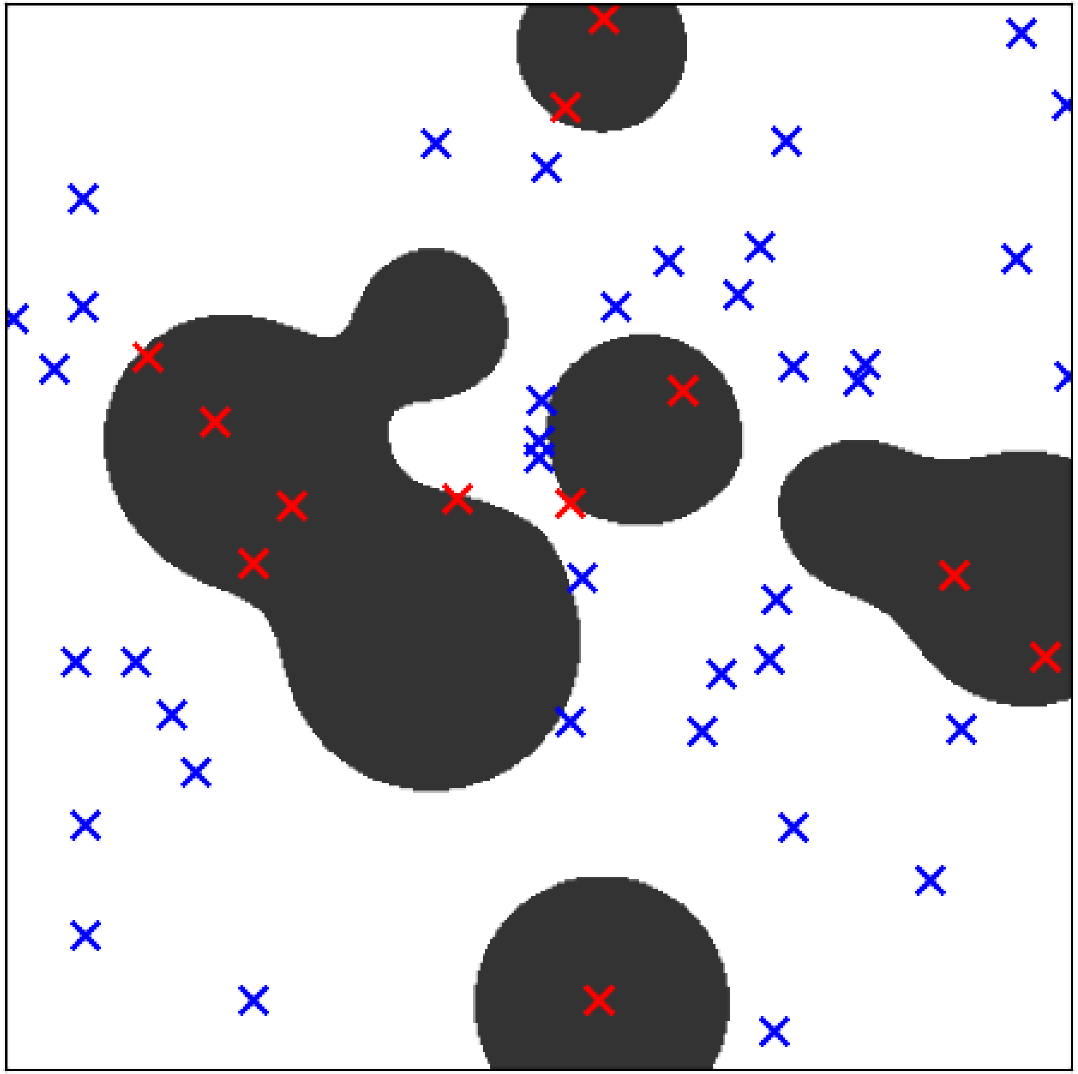


unknown C-space

Key-configuration environment representation

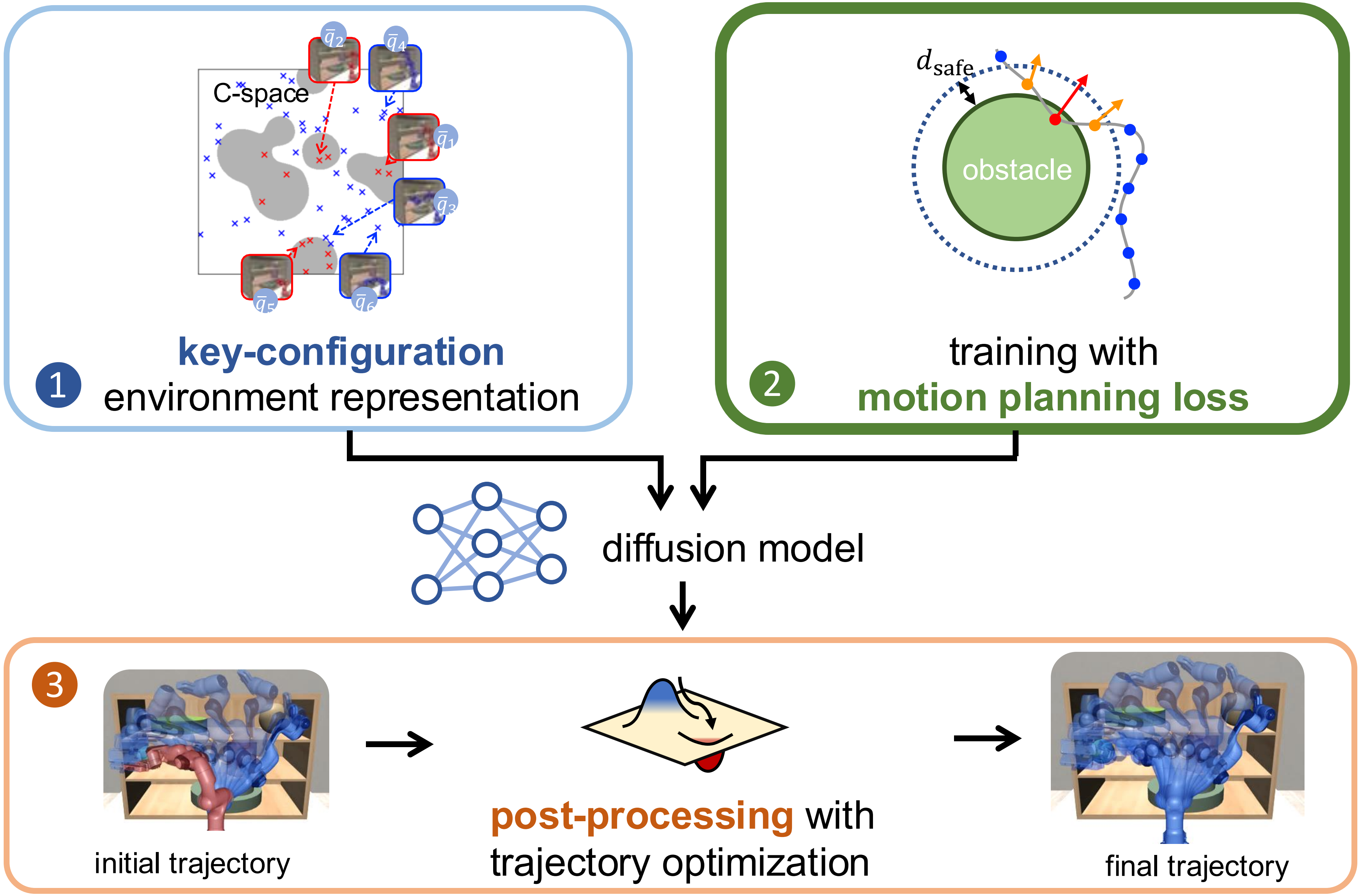
$$\phi = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

environment representation

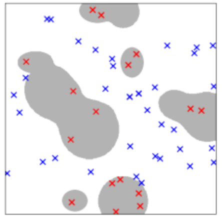


PRESTO:

Planning with Environment Representation, Sampling, and Trajectory Optimization



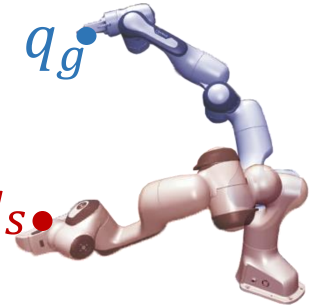
Training with TrajOpt-based **motion planning loss**



ϕ

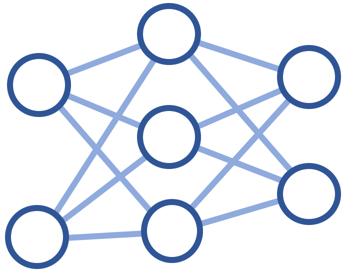
environment representation

goal q_g

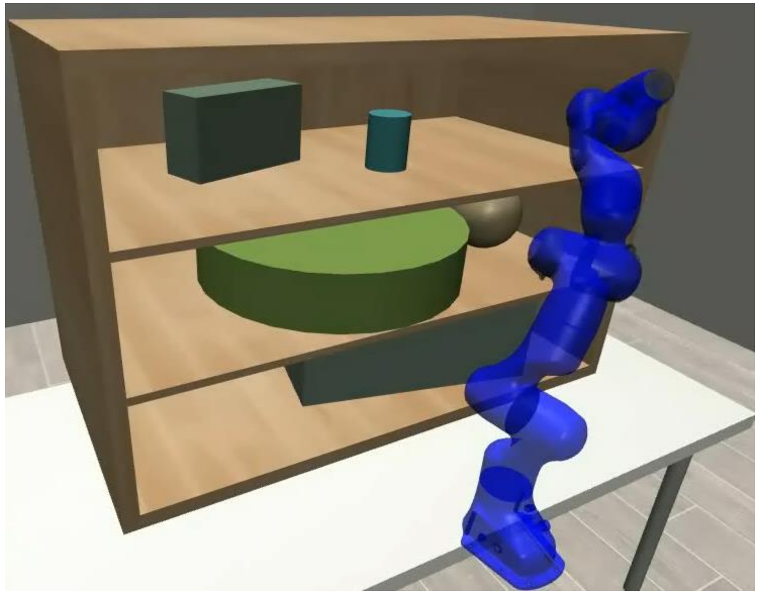


start q_s

endpoint constraints



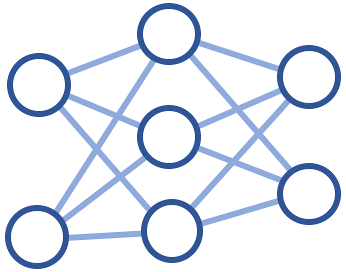
conditional diffusion model



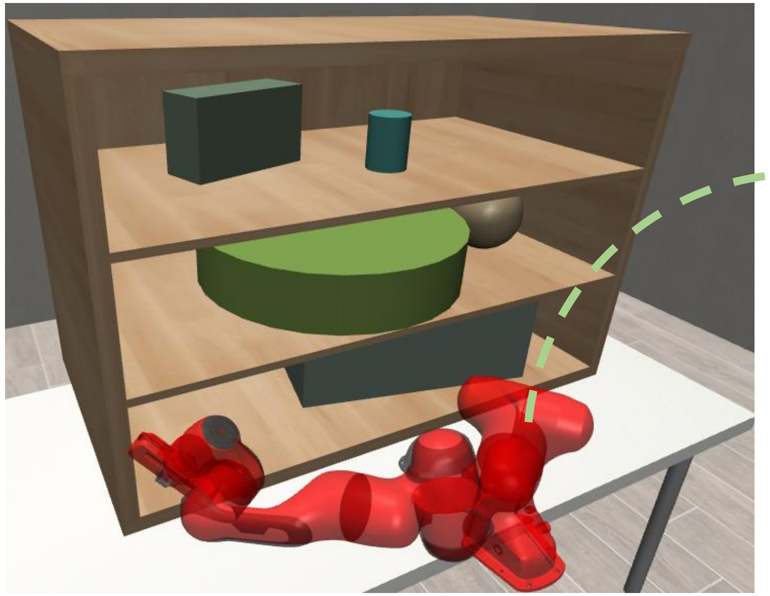
generated trajectory

Training with TrajOpt-based **motion planning loss**

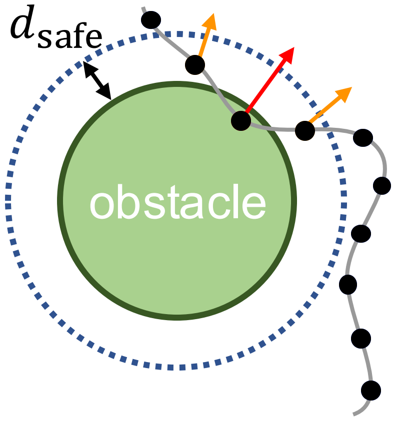
training
(i-th denoising step)

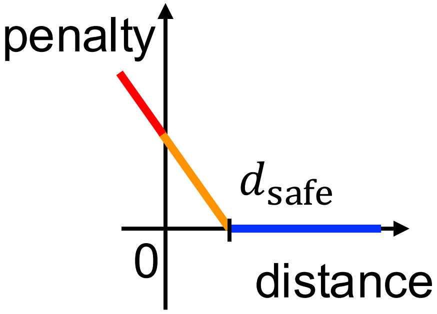


conditional
diffusion model



generated trajectory



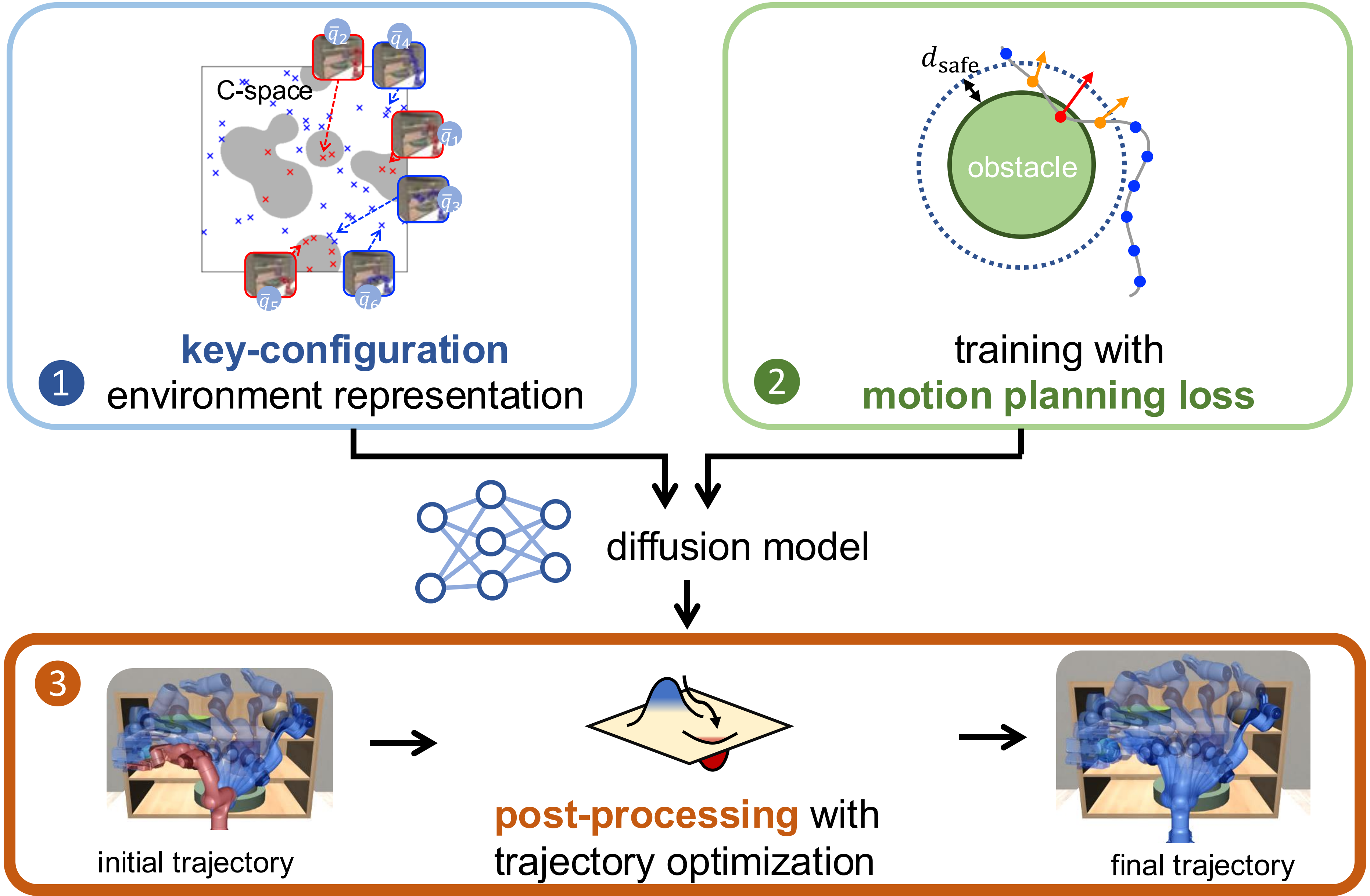


motion planning
loss

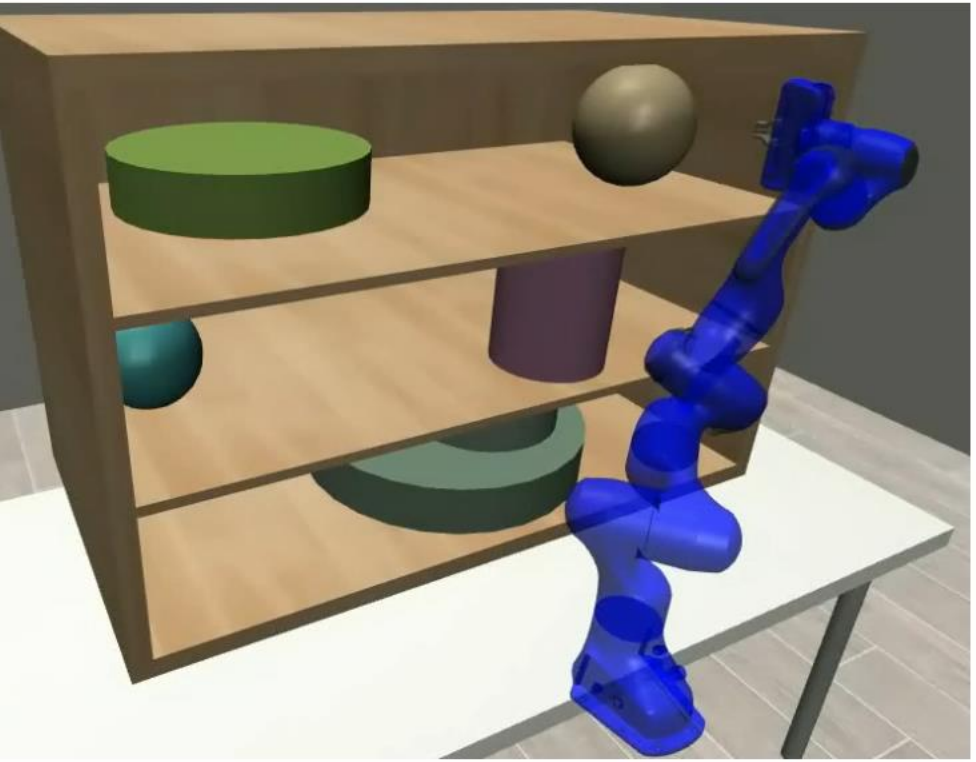
$$\mathcal{L} = w_1 \mathcal{L}_{\text{diffusion}} + w_2 \mathcal{L}_{\text{coll}} + w_3 \mathcal{L}_{\text{smooth}}$$

PRESTO:

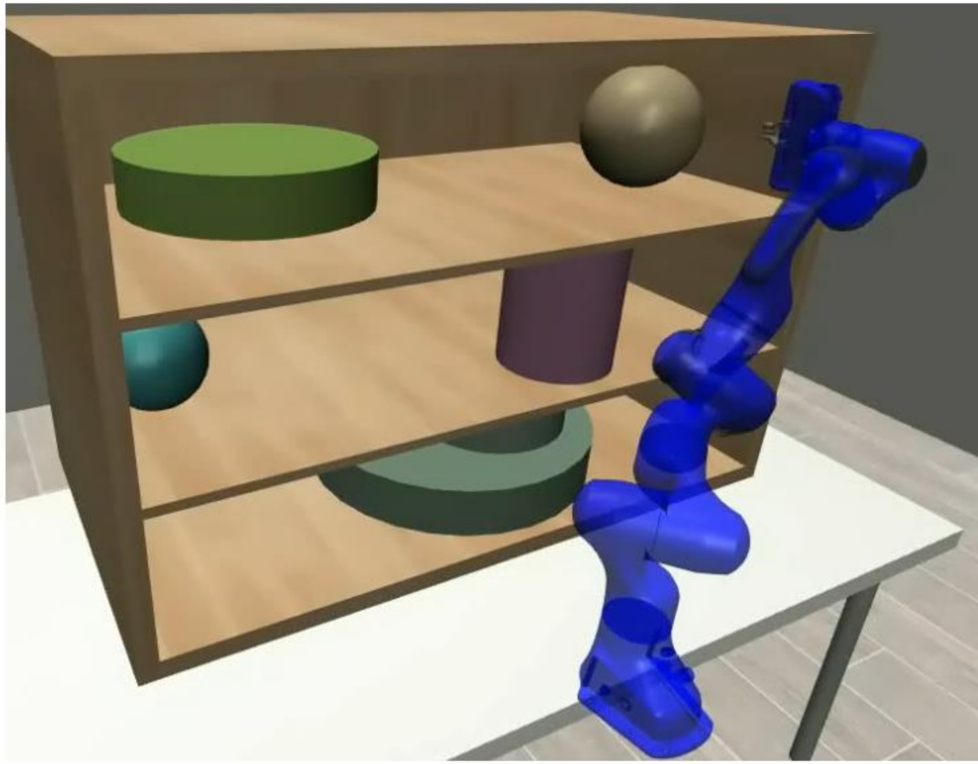
Planning with Environment Representation, Sampling, and Trajectory Optimization



Post-processing with trajectory optimization



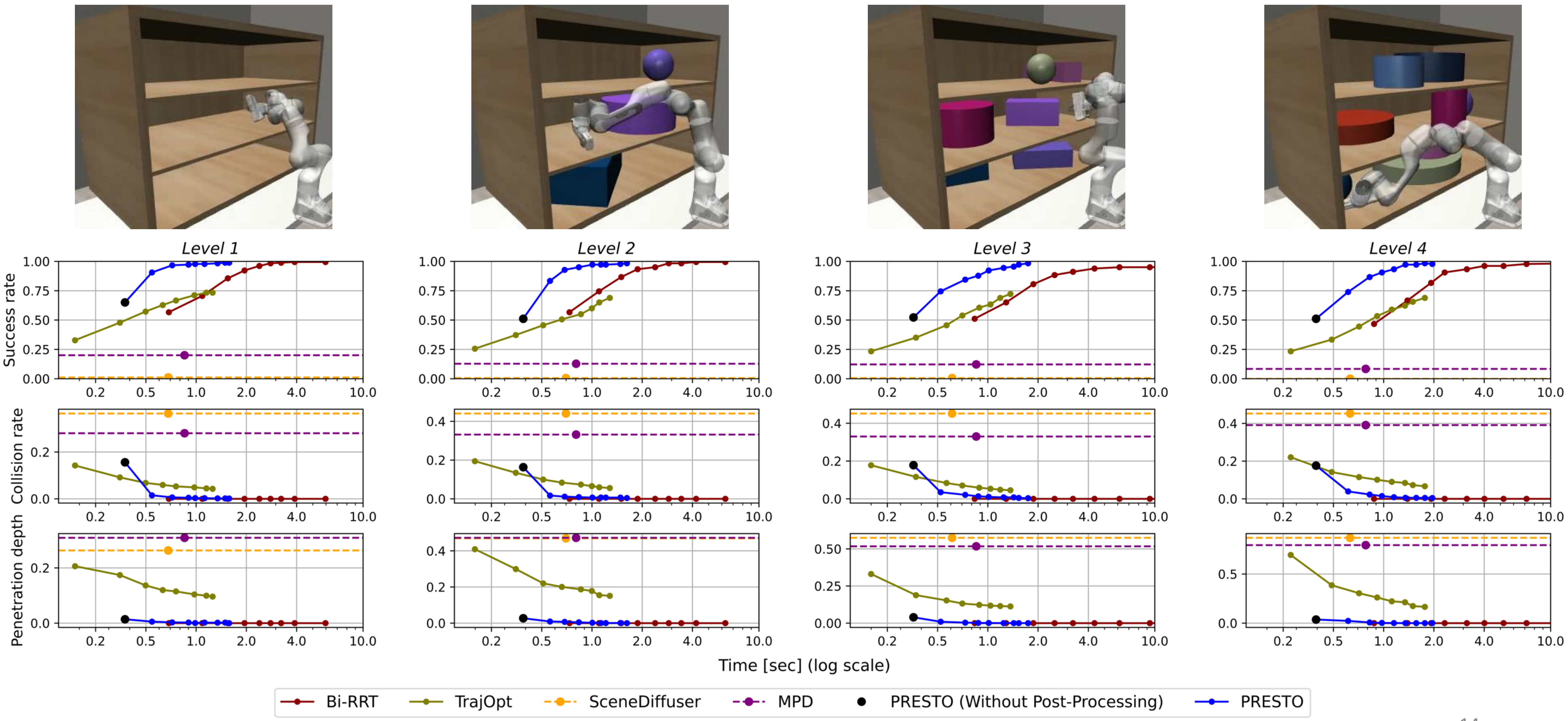
trajectory
optimization



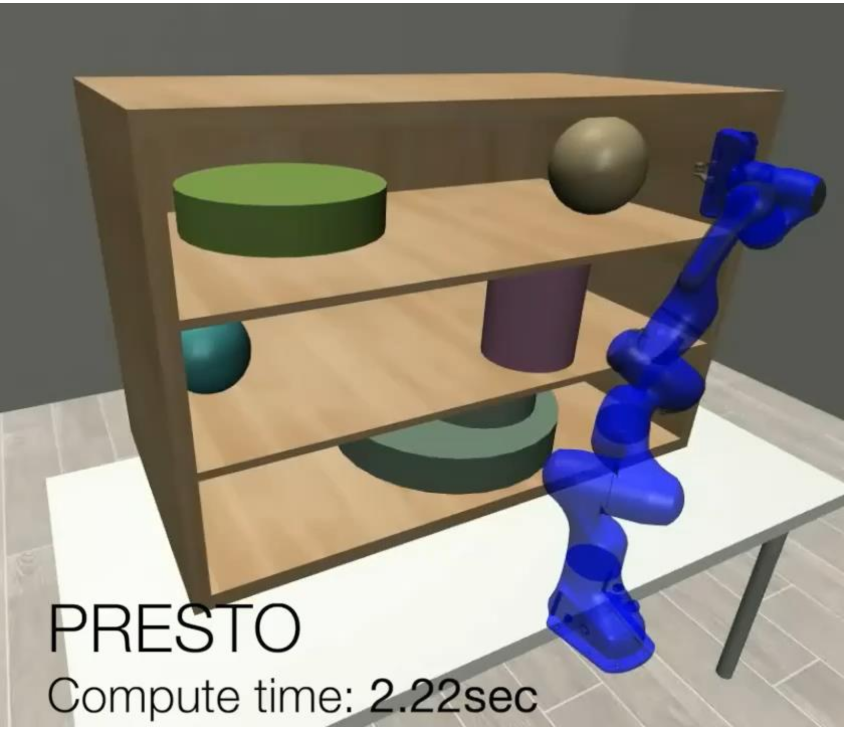
initial trajectory

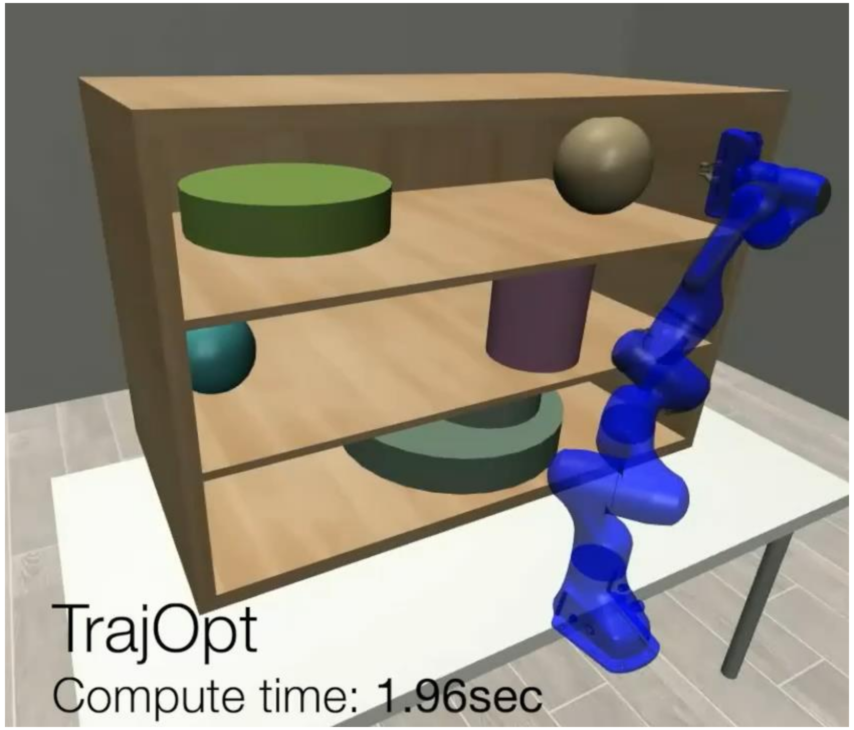
final trajectory

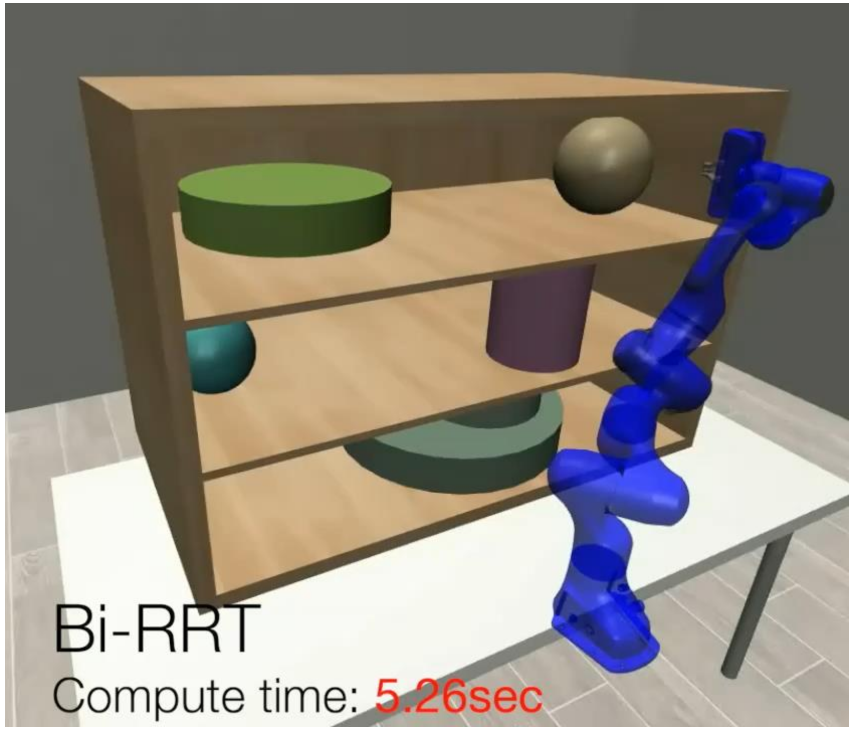
Quantitative Evaluation



Qualitative Evaluation





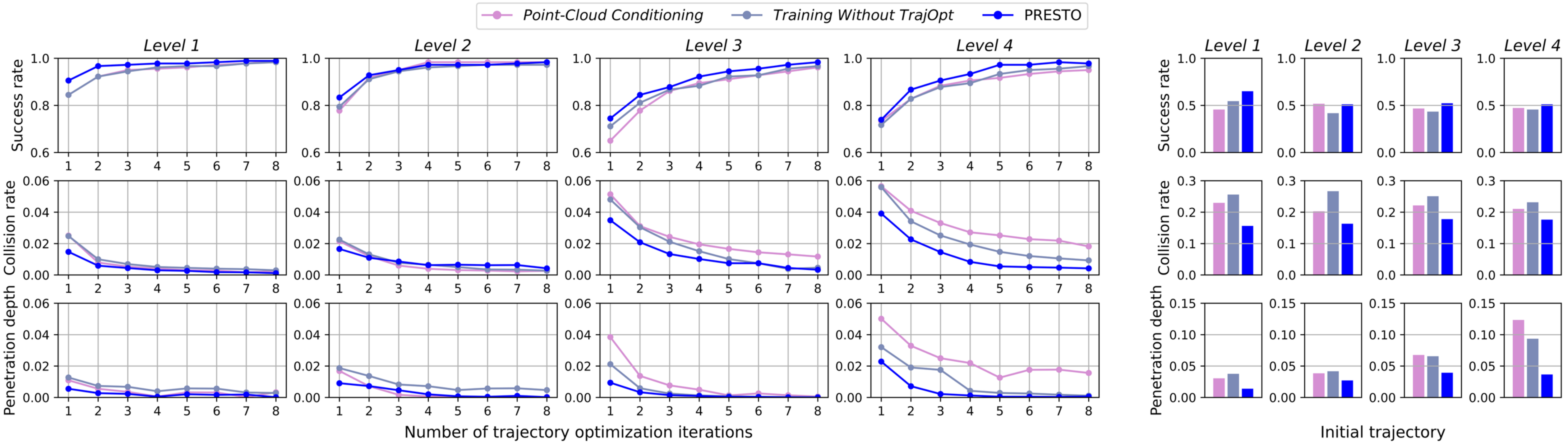


colliding/collision-free

Ablation Studies

🤔 **Key-configuration representations** vs **Point-cloud representations**?

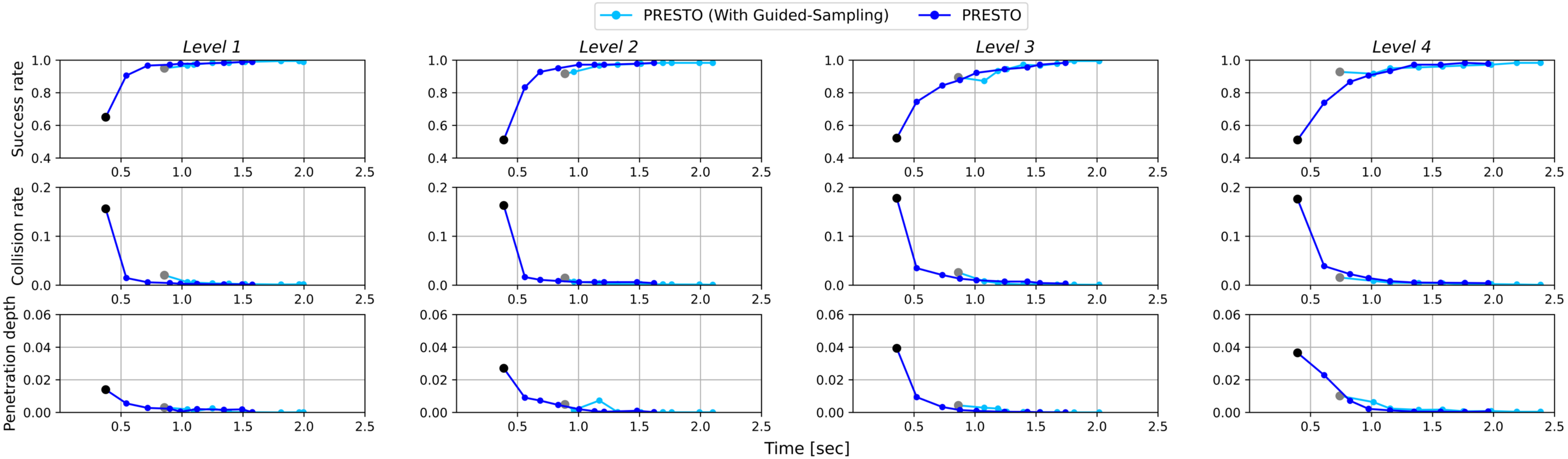
🤔 Is **training with motion-planning costs** helpful?



Test-Time Guidance

💡 Improvement on trajectory generation through test-time guidance

🤔 Additional computational overhead for guidance



Conclusion

Key ideas

- 💡 **task-related** environment representation in **C-space**
- 💡 training with **motion-planning loss**
- 💡 **post-processing** with trajectory optimization

Main results

- ✓ faster computation compared with pure planners
- ✓ outperforming generative-model-based motion planners

* Project page: https://kiwi-sherget.github.io/PRESTO