SafeFlowMatcher

Safe and Fast Planning using Flow Matching with Control Barrier Functions

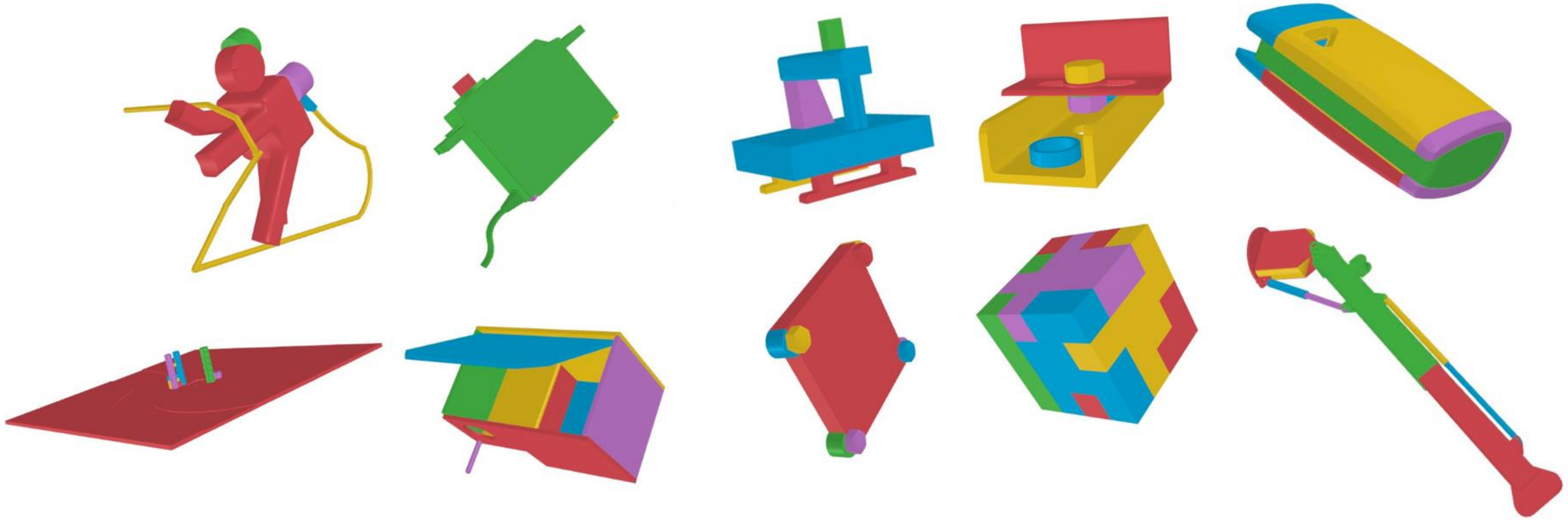
2025.06.04

Team 1
Jiwon Park & Jeongyong Yang

KAIST

Review: Towards Efficient Assembly Part Motion Planning for Robotic Assembly

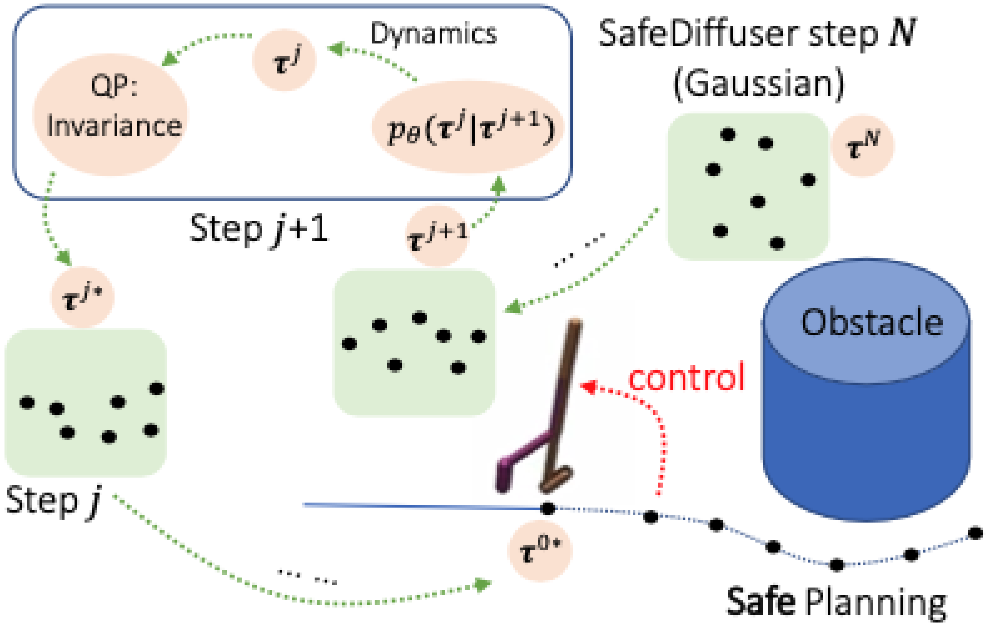
- **Problem**: Random action sampling lacks efficiency.
- **Key Idea:**
  Use VLMs (e.g., GPT-4o) to predict promising disassembly directions from 3 orthographic images and a text prompt
- **Experiments:**
  - Fewer actions sampling → Faster plan
  - Validate in narrow-passage scenarios



Contents

Introduction: Safe Generative Model Planning

- SafeDiffuser incorporates with control barrier functions to guarantee safety



[The SafeDiffuser Workflow]

Limitation 1: Slow Planning

- Diffusion-based planner like Diffuser needs a lot of denoising steps, leading to high computation load and slow generation (planning).

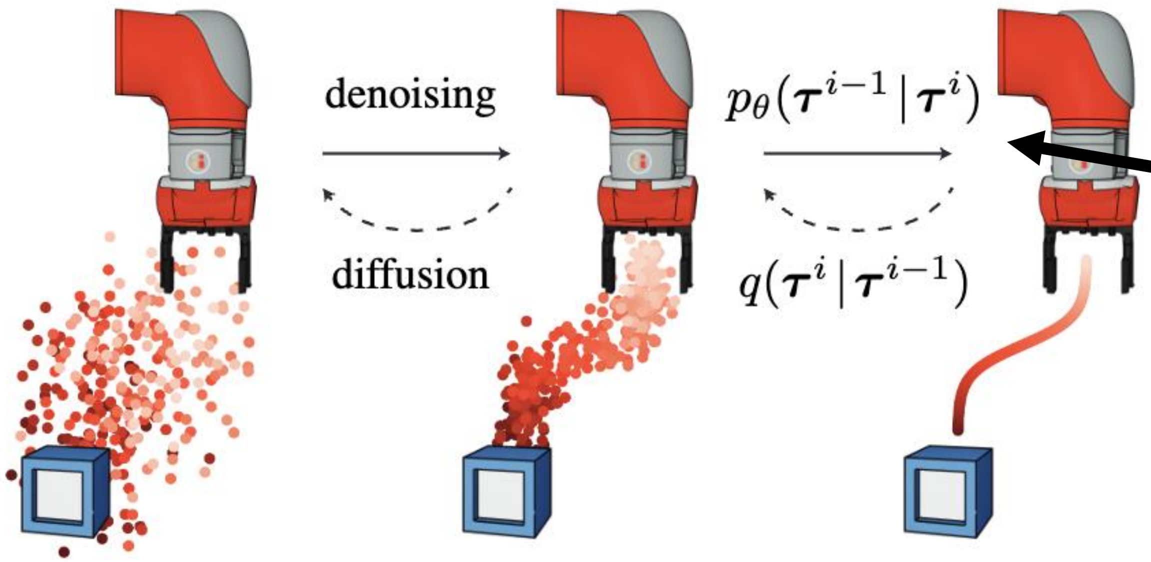


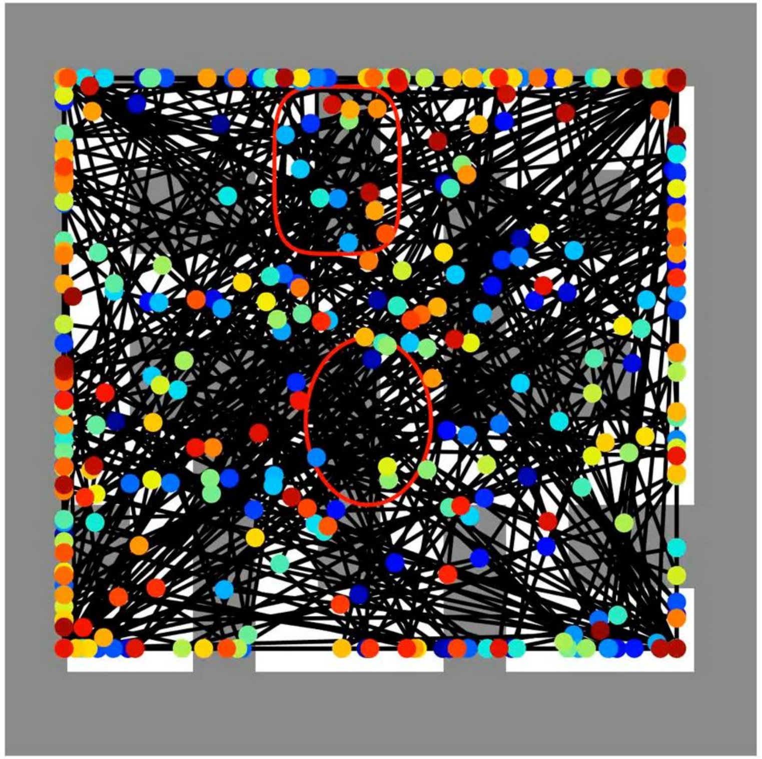
Figure 1. Diffuser is a diffusion probabilistic model that plans by iteratively refining trajectories.

Many denoising steps

Slow

Limitation 2: Local Traps

- Local traps occur when trajectories are safe but unable to reach the goal.



Limitation 2: Local Traps

- They addressed this local problem, but the local rates are still high!

Method	S-spec(↑ & ≥ 0)	C-spec(↑ & ≥ 0)	Score (↑)	Time	NLL	Trap rate 1 (↓)	Trap rate 2 (↓)
Diffuser Janner et al. (2022)	-0.983	-0.894	1.598±0.174	0.006	4.501±0.475		
Trunc. Brockman et al. (2016)	$-1.192e^{-7}$	-0.759	1.577±0.242	0.024	4.494±0.465		
CG Dhariwal & Nichol (2021)	-0.789	-0.979	0.384±0.020	0.053	6.962±0.350		
CG−ε Dhariwal & Nichol (2021)	-0.853	-0.995	0.383±0.017	0.061	6.975±0.343		
InvODE Xiao et al. (2023b)	14.000	$1.657e^{-5}$	-0.025±0.000	0.018	–		
RoS-diffuser (Ours)	0.010	0.010	1.519±0.330	0.106	4.584±0.646	100%	100%
RoS-diffuser-CF (Ours)	0.010	0.010	1.536±0.306	0.007	4.481±0.298	100%	100%
ReS-diffuser (Ours)	0.010	0.010	1.557±0.289	0.107	4.434±0.561	46%	17%
ReS-diffuser-CF (Ours)	0.010	0.010	1.544±0.280	0.007	4.619±0.652	36%	16%
TVS-diffuser (Ours)	0.003	0.003	1.543±0.303	0.107	4.533±0.494	47%	21%
TVS-diffuser-CF (Ours)	0.003	0.003	1.588±0.231	0.007	4.462±0.431	48%	18%
ReS-diffuser-L10 (Ours)	0.010	0.010	1.527±0.291	0.011	4.571±0.693	39%	8%

[Results of SafeDiffuser]

Goal: Safe and Fast Planner

Limitations

Diffuser

Slow

CBF from SafeDiffuser

Unsafe
(still frequent local traps)

Proposed Method: SafeFlowMatcher

FlowMatcher

Fast

Finite-time CBF

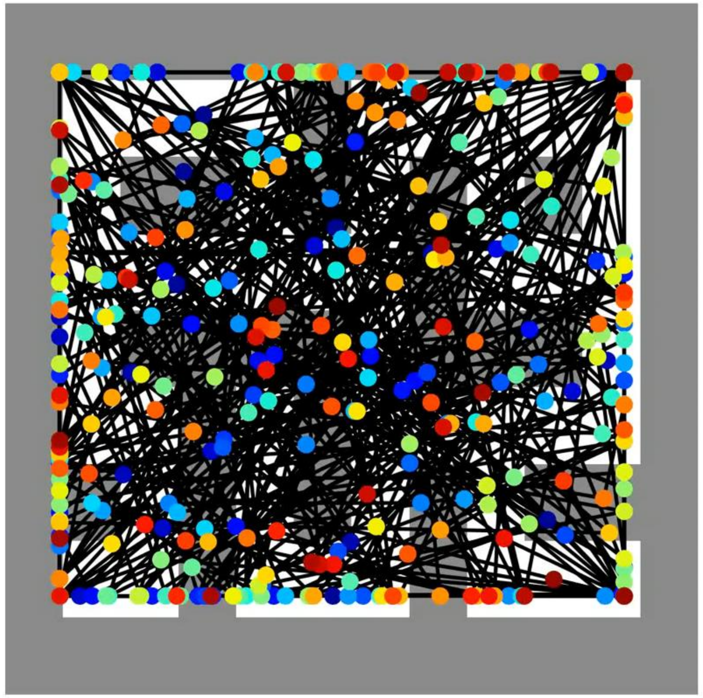
Safe
(guarantee safety within finite-time)

Adaptive Scheduling

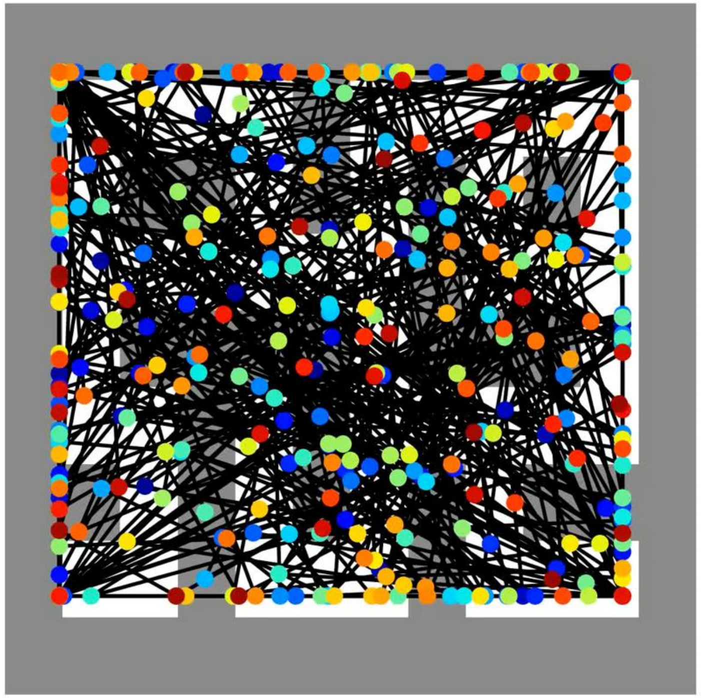
Safe
(escape local traps effectively)

Method 1: FlowMatcher

- We implemented a flow-matching-based planner called **FlowMatcher**, based on conditional flow matching theory and inspired by Diffuser paper.
- FlowMatcher can generate paths **FAST**.



Diffuser



FlowMatcher

Method 2: Finite-time CBF with Relaxation

- **SafeDiffuser** (Relaxation form)

$$\min_{u,r} \|u - u_{des}\|^2 + \|r\|^2$$
$$s.t.\ \nabla_x h(x)u + \alpha h(x) \geq \boxed{-w(t)r}$$

- **SafeFlowMatcher**

Relaxation

$$\min_{u,r} \|u - u_{des}\|^2 + \|r\|^2$$
$$s.t.\ \nabla_x h(x)u + \alpha |h(x)|^{\rho} \geq \boxed{-w(t)r}$$

- $\boldsymbol{w(i)}$ is monotonically decreasing function where i is denoising step.

KAIST

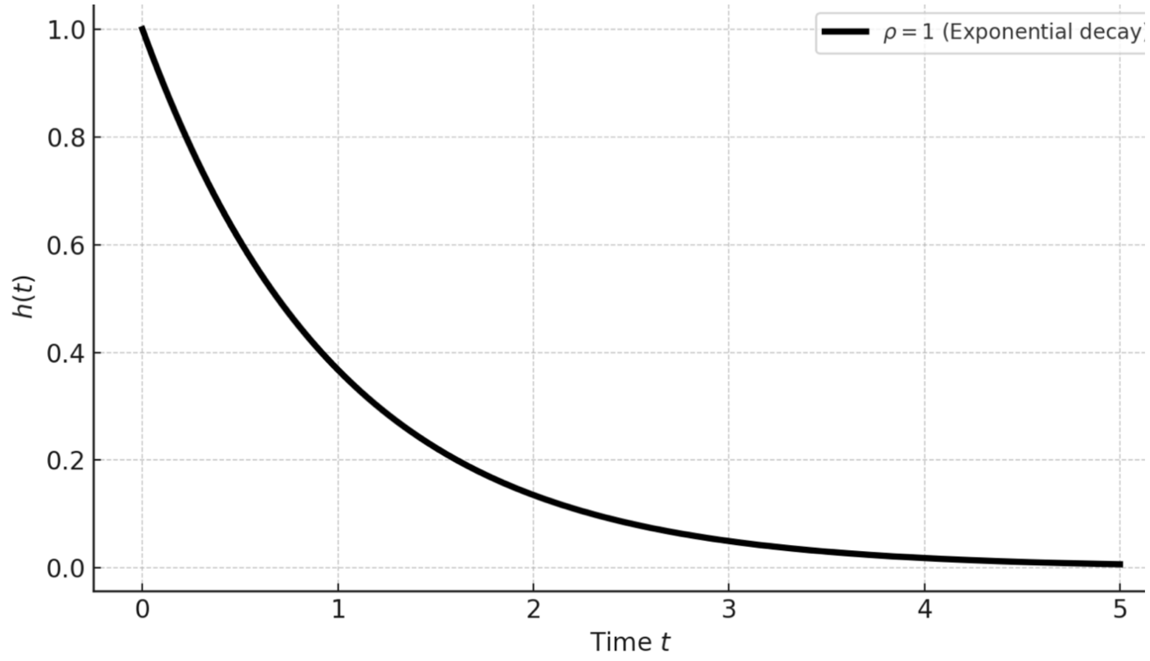
*Here, we wrote the equation as simple as possible to clearly deliver the concept. It's not mathematical rigor.

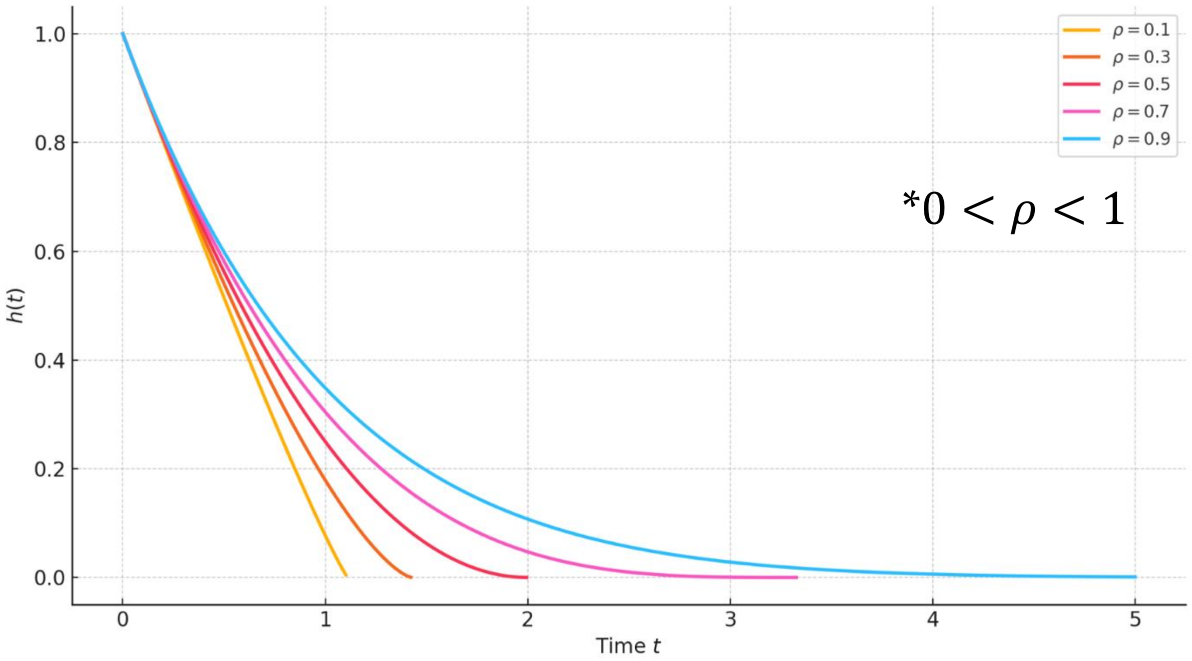
Method 2: Finite-time CBF with Relaxation

- What does ρ stand for?
- ρ enforces the system converge faster, even in finite time.

$$\min_{u,r} \|u - u_{des}\|^2$$
$$s.t.\ \dot{h}(x) + \alpha h(x) \geq 0$$

$$\min_{u,r} \|u - u_{des}\|^2$$
$$s.t.\ \dot{h}(x) + \alpha |h(x)|^{\rho} \geq 0$$





*$0 < \rho < 1$

Method 2: Finite-time CBF with Relaxation

- We analytically derived the finite-time bound for convergence.
- Here, t_0 is the time when $w(t)$=0.

$$\boldsymbol{T \leq t_0 + \frac{|h(x)|^{1-\rho}}{\alpha(1-\rho)}}$$

- The Key point is that **if we select proper hyperparameters, we can guarantee the finite-time convergence to the safe set**.

KAIST

Method 3: Adaptive Time Scheduling

- Diffusion-based methods denoise over fixed steps.
- However, flow matching-based method can adopt adaptive step sizes thanks to their continuous-time formulation.

Feature	**Diffusion Models**	**Flow Matching Methods**
Time Domain	Discrete time steps (e.g., 256 steps)	Continuous time, modeled via ODEs
Step Size	Fixed by noise schedule	Arbitrary

- We select Δt decreasingly to address the local trap problems.

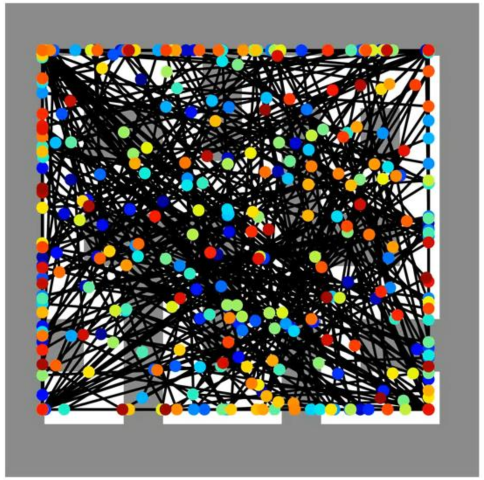
KAIST

Method 3: Adaptive Time Scheduling

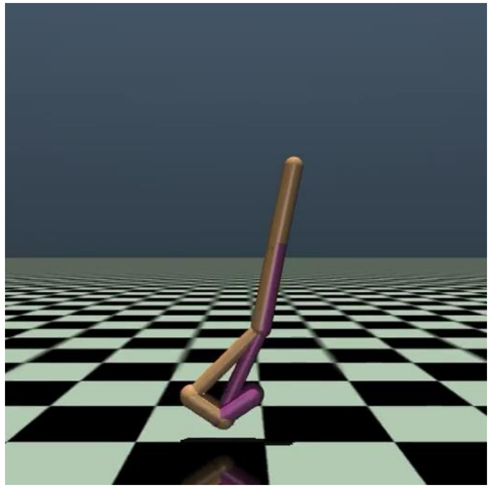
- Local traps occur when CBF constraints dominate early.
- We first go over obstacles with a **few large steps** under weak CBF.
- Then, we safely refine trajectory using **many small steps** with strong CBF.

Experimental Results

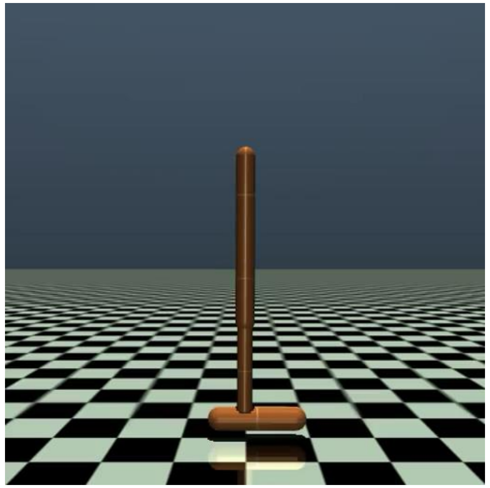
- Experiments
  - Maze2D
    - Qualitative evaluation on Safety, Trap Rate, Computation Time compared to SafeDiffuser
  - Locomotion
    - Qualitative evaluation on Score, Safety, Computation Time compared to SafeDiffuser



Maze2D



Walker2D



Hopper

Experimental Results

- Maze2D
  - SafeFlowMatcher is the only method with **near-zero trap rates**
  - SafeFlowMatcher acheives **the shortest planning time**

Method	S-Spec(↑)	C-Spec(↑)	Score(↑)	Time per Step(↓)	Total Time*(↓)	Trap Rate 1(↓)	Trap Rate 2(↓)
RoS-Diffuser	0.010	0.010	**1.632±0.203**	0.006	1.536	100%	100%
ReS-Diffuser	0.010	0.018	1.504±0.282	0.006	1.536	83%	79%
TVS-Diffuser	-0.018	-0.018	1.569±0.203	0.006	1.536	67%	67%
SafeFlowMatcher (Ours)	0.010	0.010	1.458±0.432	0.006	**0.384**	**1%**	**0%**

[Maze Planning Comparison]

* Number of Steps : 256 (SafeDiffuser), 64 (SafeFlowMatcher)

Experimental Results

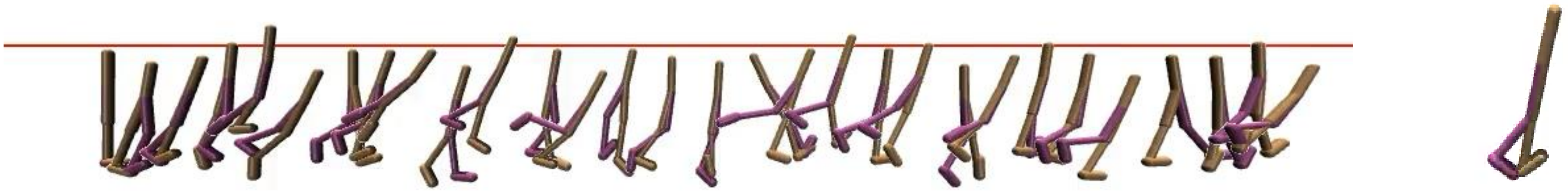
- Locomotion : Experimental Environment

Simulation Platform	MuJoCo Physics Engine
Benchmark Dataset	D4RL
Test Task	Walker2D, Hopper
Model Architecture	TemporalUnet
Horizon	600
Training Epoch	100
Steps per Epoch	5,000
Batch Size	32 (training), 512 (planning)
Learning Rate	2e-4
Max Episode Length	1000
Hardware	H100 (training), RTX3060 (planning)

[Details of Experiments]

Experimental Results

- Locomotion : Walker2D



[SafeDiffuser]

[SafeFlowMatcher]

Experimental Results

- Locomotion : Walker2D
  - SafeFlowMatcher achieves **25.6% improvement** in performance metrics
  - The improvement in performance comes 1.6% less computation time
  - SafeFlowMatcher reduces unsafe risk by 60.75%

[SafeDiffuser*]

[SafeFlowMatcher]

	SafeDiffuser	SafeFlowMatcher
Score	0.39	**0.49**
Computation Time(s)	1.837	**1.807**
Safety	-4.468	-1.754

[Lomotion Planning Comparison]

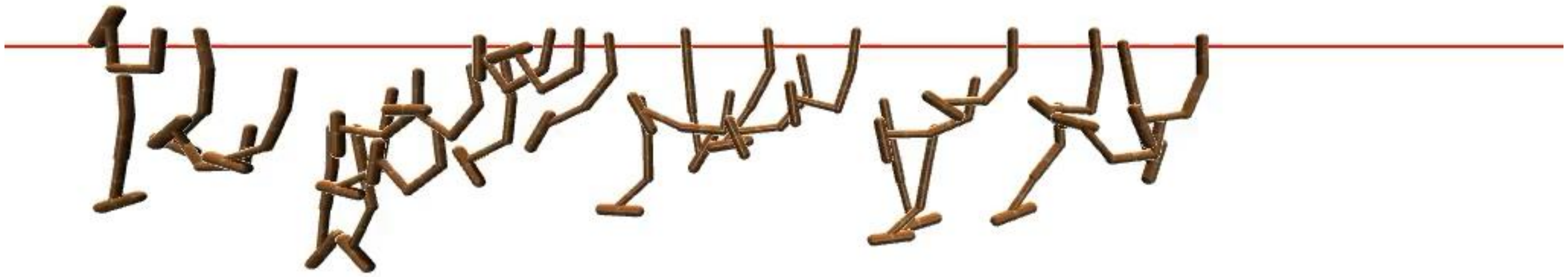
KAIST

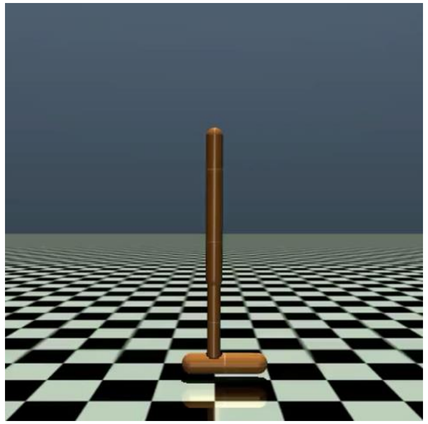
* RoS Diffuser

Experimental Results

- Locomotion : Hopper

[SafeDiffuser]





[SafeFlowMatcher]

Experimental Results

- Locomotion : Hopper
  - SafeFlowMatcher achieves a 59.7% higher score than SafeDiffuser
  - SafeFlowMatcher reduces unsafe risk by 94.6%

[SafeDiffuser]

[SafeFlowMatcher]

	SafeDiffuser	SafeFlowMatcher
Score	0.524	**0.837**
Computation Time(s)	0.434	**0.694**
Safety	-2.918	-0.159

[Lomotion Planning Comparison]

KAIST

Experimental Results

- Summary
  - SafeFlowMatcher outperformes baselines in both Maze2D and Locomotion tasks
    - Acheives **lower trap rate** and **safety improvement**
  - Delivers **faster planning** and supports **scalable step sizes**
  - Demonstrates effectiveness across both **high-dimensional motion** and safety-aware planning

Contributions

	Jiwon Park	Jeongyong Yang
Paper review	O	O
Theory		
SafeFlowMatcher theory	V	O
Adaptive time scheduling	V	O
Existing methods validations		
SafeDiffuser	O	V
Implementation		
FlowMatcher	O	O
Finite-time CBF	V	O
Experiments		
Maze2D	O	O
Legged Locomotion (Warker2D, Hopper)	O	V

O: contribute, V: support

Thank you

Appendix

- Evaluation Metric
  - Specification
    - Calculated by the minimum values of the functions among all runs that define the safety constraints
      - S-Spec : Simple (Quadratic)
      - C-Spec : Complex (Quartic)
  - Local Trap
    - The safety value satisfies $b(x) = 0$ or $b(x) < 0.01$
    - The distance between consecutive points exceeds a predefined threshold δ
  - Trap Rate
    - Trap Rate 1 : at least one trap is encountered
    - Trap Rate 2 : two or more traps are encountered
  - Safety
    - Minimum normalized distance between the agent and the ceiling