SafeFlowMatcher

Safe and Fast Planning using Flow Matching with Control Barrier Functions

2025.05.14

Team 1
Jiwon Park & Jeongyong Yang

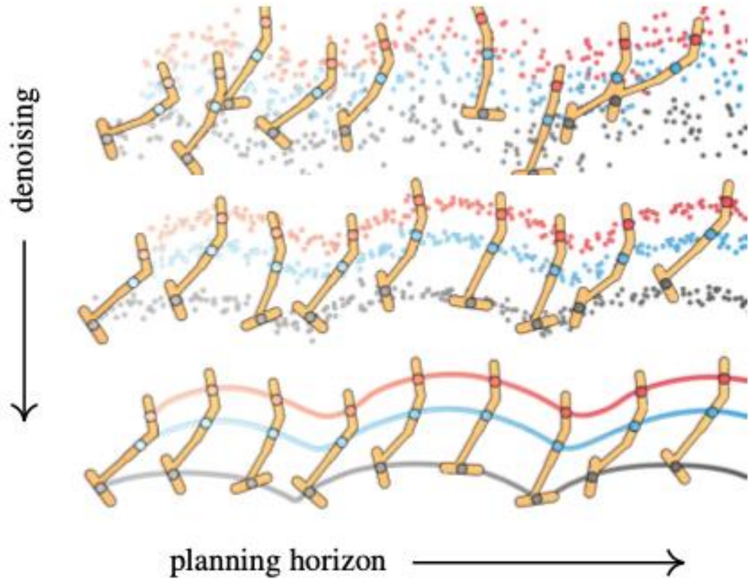
KAIST

Contents

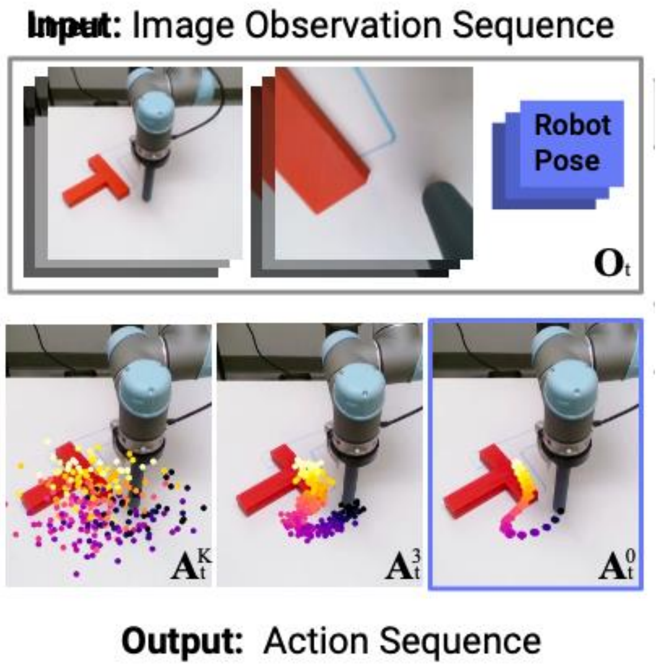
1. Introduction
2. Baseline: Diffuser
3. Baseline : SafeDiffuser
4. Limitations
5. Proposed Methodology
6. Experimental Results

Introduction

- Probabilistic Generative Models + Planning



Planning with Diffusion for Flexible Behavior Synthesis (2022 ICML)



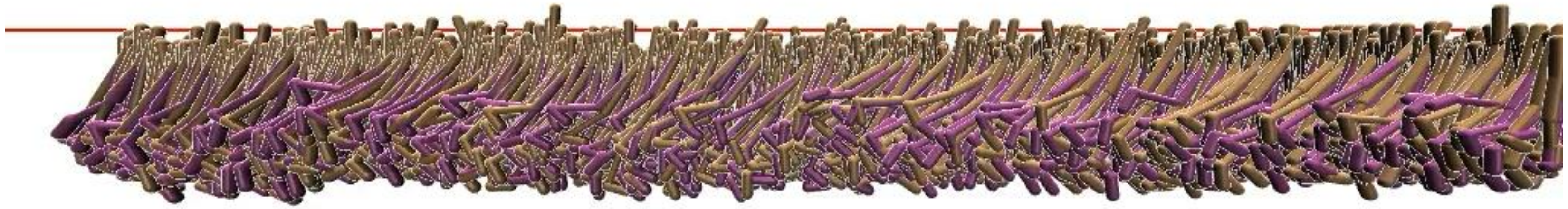
Diffusion Policy (2023 RSS)

NoMaD : Goal Masking Diffusion Policies for Navigation and Exploration (2024 ICRA)

Introduction

- Probabilistic Generative Models cannot guarantee safety.

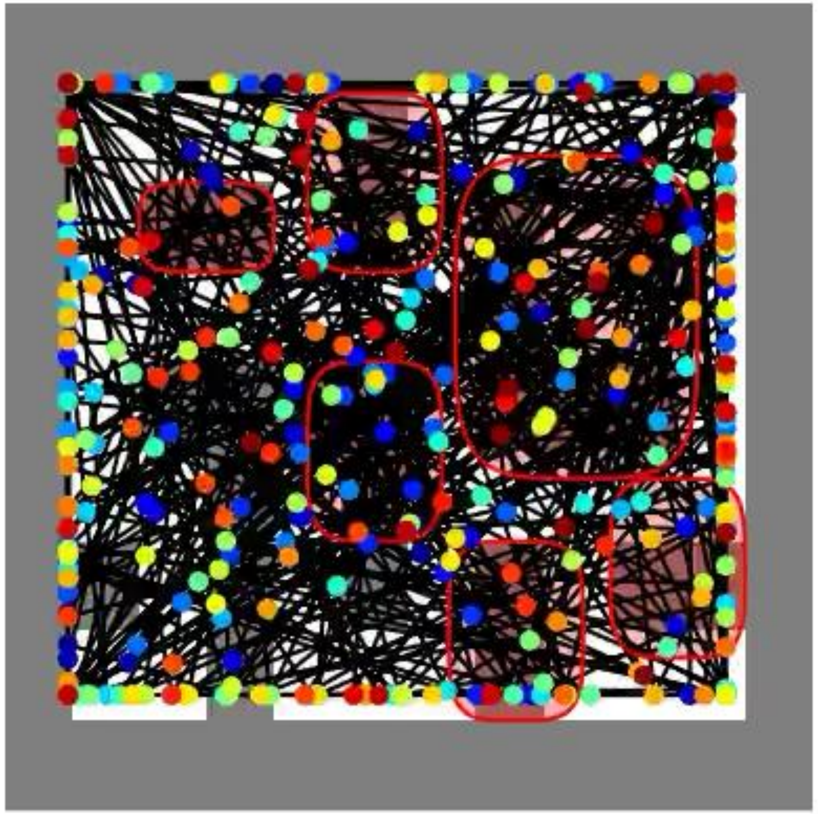
Walker2D



Hopper

Introduction

- Probabilistic Generative Models cannot guarantee safety.



Baseline: Diffuser

- Diffuser [1] is the first generative

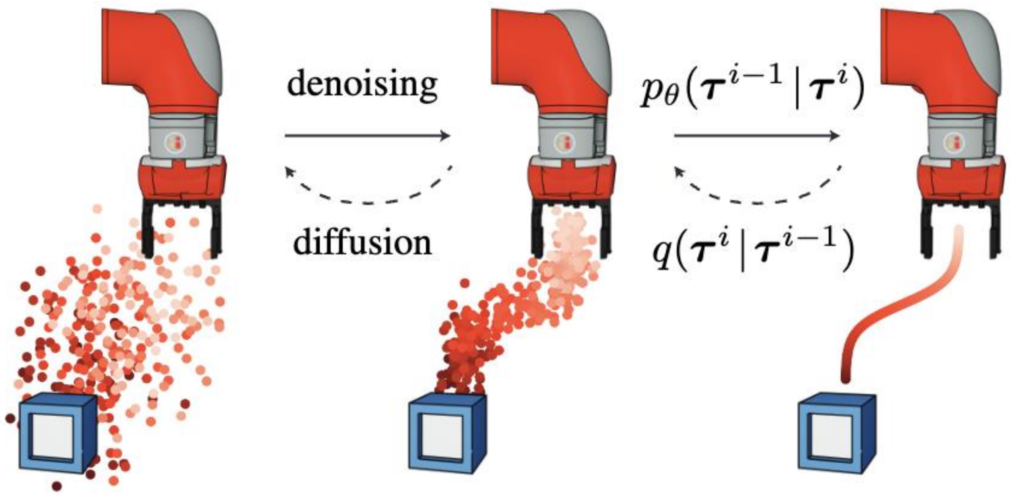
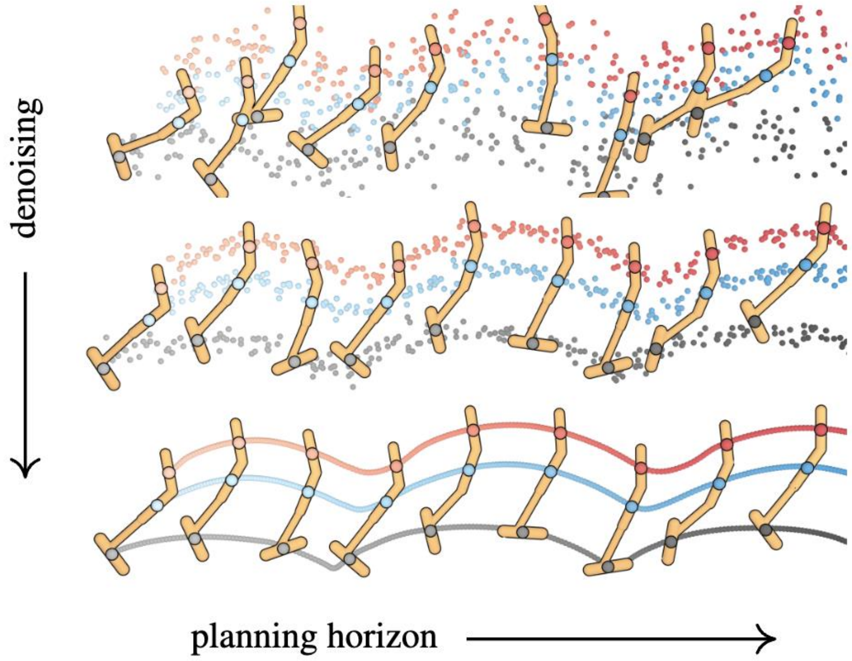
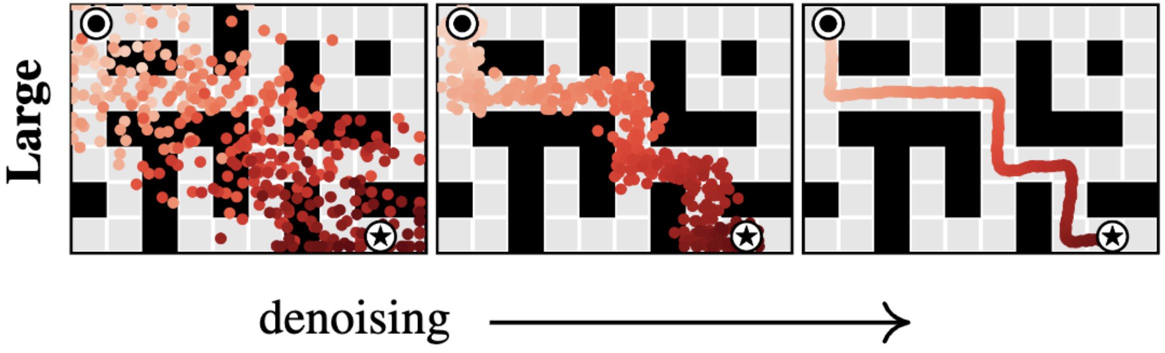


Figure 1. Diffuser is a diffusion probabilistic model that plans by iteratively refining trajectories.





Baseline: SafeDiffuser

- SafeDiffuser [2] incorporate with control barrier functions to guarantee safety.

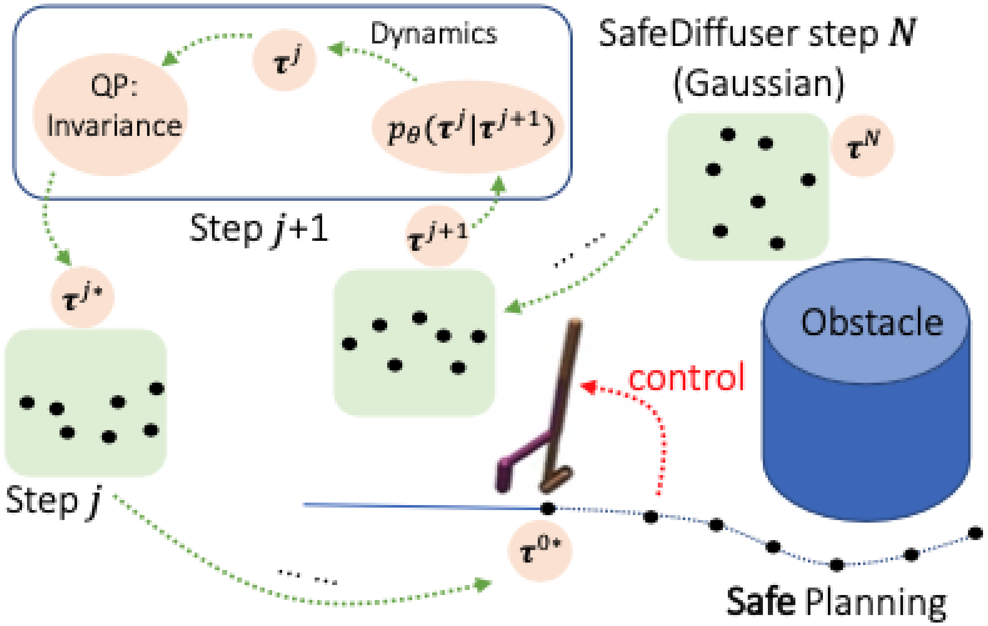
SafeDiffuser: Safe Planning with Diffusion Probabilistic Models

diffusion models, termed diffusers Janner et al. (2022), have shown promise in trajectory planning for a variety of robotic tasks. Diffusers enable flexible behavior synthesis that makes it well generalized in novel environments.

Fig. 1: Our proposed SafeDiffuser (lower) generates safe trajectories with guarantees, while the diffuser (upper) fails (from ⊙ to ⊗).

During inference, the diffuser, conditioned on the current state and objectives, starts from Gaussian noise to generate clean planning trajectories based on which we can get a control policy. After applying this control policy one step forward, we get a new state and run the diffusion procedure again to get a new planning trajectory. This process is repeated until the objective is achieved. However, one big challenge in this method is that there are no safety guarantees. For instance, the planning trajectory could easily violate safety constraints in the maze (as shown in Fig. 1). This shortcoming demands a fundamental fix to diffusion models to ensure the safe generation of planning trajectories in safety-critical applications such as trustworthy policy learning and optimization.

Videos can be found in the anonymous website: https://safediffuser.github.io/safediffuser/



[The SafeDiffuser Workflow]

Limitations: Diffuser

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

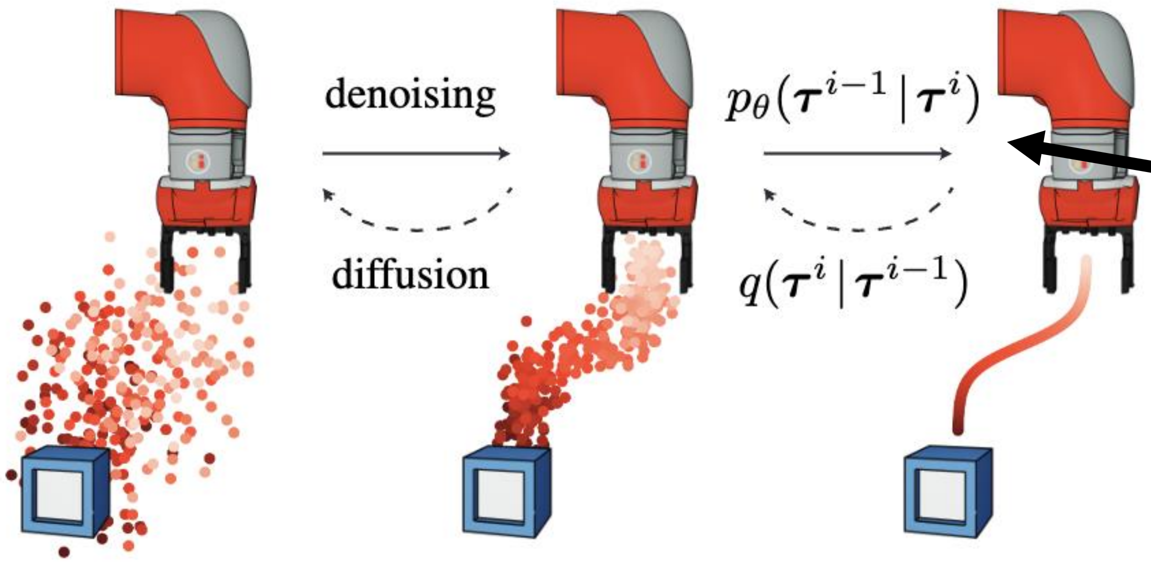


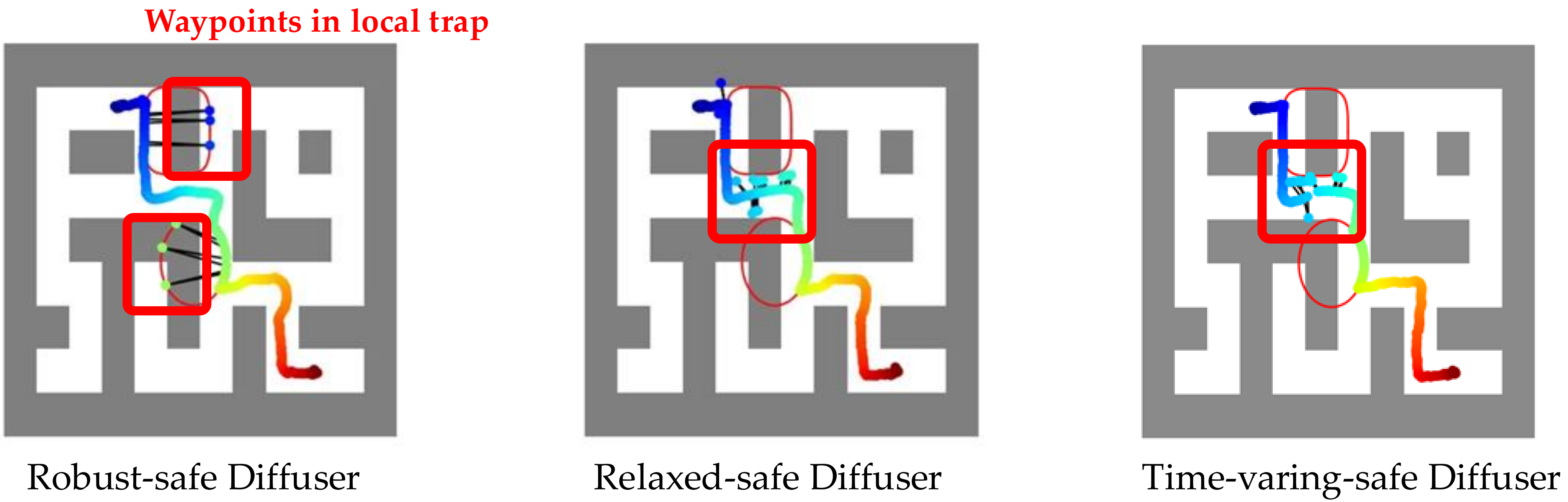
Figure 1. Diffuser is a diffusion probabilistic model that plans by iteratively refining trajectories.

Many denoising steps

Slow

Limitations: SafeDiffuser

- SafeDiffuser proposed three different architecture to enforce safety constraints using CBF.
- However, it needs to be improved to avoid local traps and ensure safety.



[Results of SafeDiffuser]

Limitations: SafeDiffuser

- They enforce safety constraints, but still some states are stuck in local trap.

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]

KAIST

Goal: Safe and Fast Planner

Limitations

Diffuser [1]

Slow

CBF from SafeDiffuser [2]

Unsafe
(still frequent local traps)

Proposed Method

Flow Matching [3] in Generative Model Field

FlowMatcher

Fast

Finite-time CBF

Safe

Finite-time Stability Theory [4] in Control Field

Fast and Safe Planner
SafeFlowMatcher

Flow Matching Recap from Student Lecture

- Flow Matching requires fewer sampling steps to generate an image or trajectory.
- Since neural network inference is required at each step, fewer steps can help reduce the total generation (or planning) time.

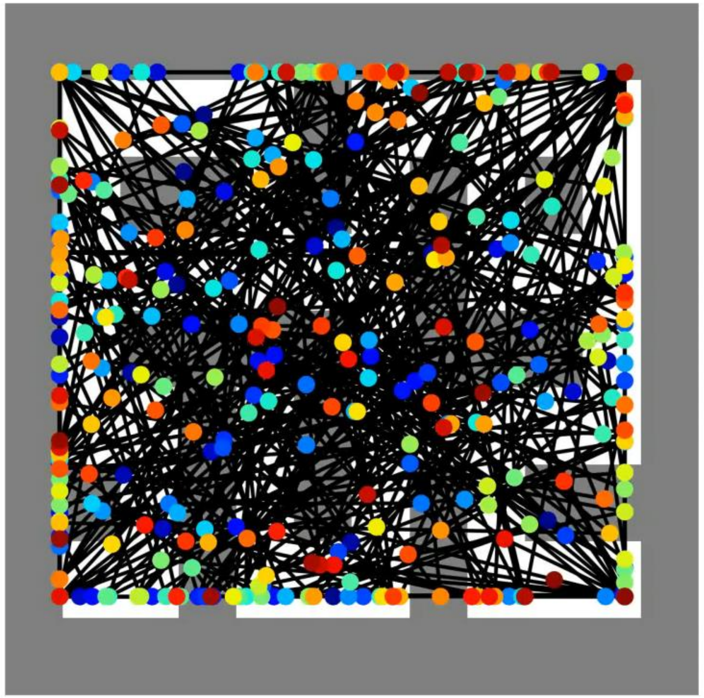
	Diffusion	**Flow Matching**
Process	Step-by-step noise addition and denoising	Continuous transformation via **Velocity fields**
Mathematical Base	Stochastic process	Deterministic ODE
Sampling	Many steps	**Few steps**
Best for	High-fidelity, complex generation	Fast, controllable planning

[Comparison between Diffusion and Flow Matching]

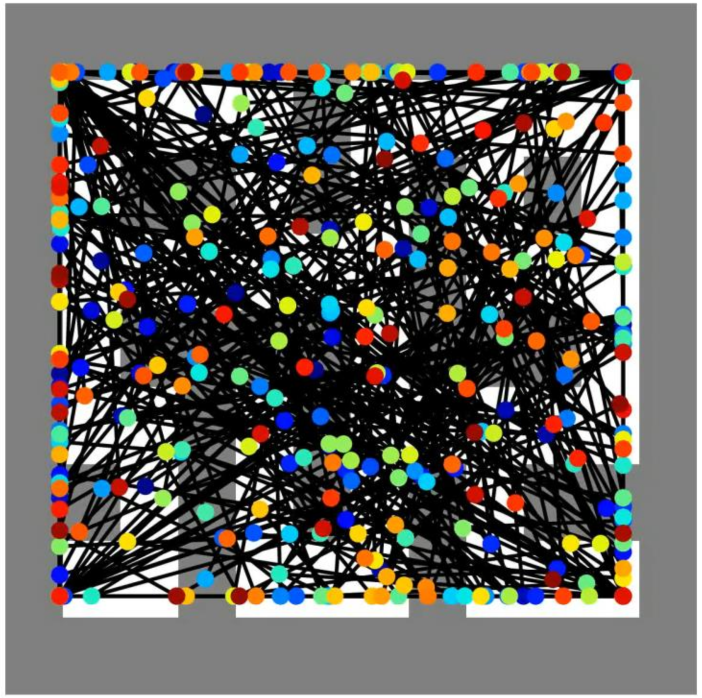
KAIST

FlowMatcher

- We implemented a flow-matching-based planner called **FlowMatcher**, built on conditional flow matching theory and inspired by Diffuser [1].
- FlowMatcher can generate paths **FAST**.



Diffuser



FlowMatcher

Brief Introduction to Finite-time CBF

- From finite-time stability theory, we can derive Finite-time CBF (FT-CBF).

Definition 1 (Finite-Time CBF) *Given the affine system* $\dot{\mathbf{x}}_t = f(\mathbf{x}_t) + g(\mathbf{x}_t)\mathbf{u}_t$ *and the safe set* $\mathcal{C} \triangleq \{\mathbf{x}_t \in \mathbb{R}^n \mid b(\mathbf{x}_t) \geq 0\}$, C^1 *function* b *is called a finite-time convergence CBF if there exist parameters* $\rho \in [0,1)$ *and* $\epsilon > 0$ *such that for all* $\mathbf{x}_t \in \mathcal{D}$,

$$\sup_{\mathbf{u} \in \mathcal{U}} [L_f b(\mathbf{x}_t) + L_g b(\mathbf{x}_t)\mathbf{u}_t + \epsilon \cdot \mathrm{sign}(b(\mathbf{x}_t))|b(\mathbf{x}_t)|^\rho] \geq 0, \tag{5}$$

where $L_f b(\mathbf{x}_t) \triangleq \nabla b(\mathbf{x}_t)^\top f(\mathbf{x}_t)$ *and* $L_g b(\mathbf{x}_t) \triangleq \nabla b(\mathbf{x}_t)^\top g(\mathbf{x}_t)$ *denote the Lie derivatives of* b *along* f *and* g*, respectively.*

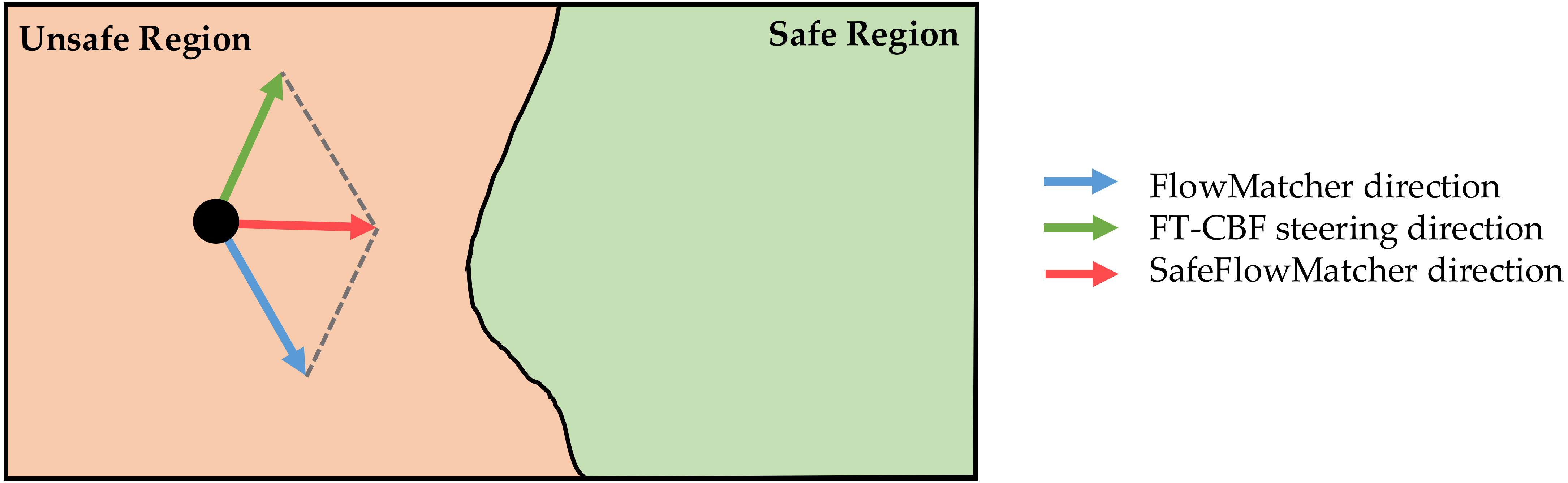
Lemma 1 (Forward Invariance of the Safe Set) *Define CBF* b *as in Definition 1, such that the initial state satisfies* $b(\mathbf{x}_0) \geq 0$. *Any Lipschitz continuous controller* $\mathbf{u}_t$ *that satisfies condition (5) ensures forward invariance of the safe set* $\mathcal{C}$, *i.e.,* $b(\mathbf{x}_t) \geq 0$ *for all* $t \geq 0$.

- We will explore CBF and finite-time CBF in more detail later in the paper presentation. Focus on the **key concecpt of CBF** here.

KAIST

Brief Introduction to Finite-time CBF

- Unlike nominal CBF, finite-time convergence CBF guarantees that the states converge to a safe set within a finite time.
- Since FlowMatcher generates trajectories over a time horizon $t \in [0,1]$, it is important that the states converge to the safe set by $t = 1$.



SafeFlowMatcher

- SafeFlowMatcher Dynamics:

$$\overbrace{\frac{d\boldsymbol{\tau}_t}{dt}}^{\textbf{Trajectory State}} = \overbrace{v_t(\boldsymbol{\tau}_t;\theta)}^{\textbf{Velocity Field from FlowMatcher}} + \underbrace{\Delta\mathbf{u}_t}_{\textbf{Perturbation from FT-CBF}} \triangleq \mathbf{u}_t,$$

- Thus, u_t is new control input to generate safe trajectories.

SafeFlowMatcher

- We can derive the optimal control input u_t^* using convex optimization (QP).

$$\mathbf{u}_t^{k*}, r_t^{k*} = \underset{\mathbf{u}_t^k, r_t^k}{\operatorname{argmin}} \|\mathbf{u}_t^k - v_t(\boldsymbol{\tau}_t^k;\theta)\| + \|r_t^k\| \quad \text{subject to} \quad (9).$$

$$\frac{db(\boldsymbol{\tau}_t^k)}{d\boldsymbol{\tau}_t^k}\mathbf{u}_t^k + \epsilon \cdot \operatorname{sign}(b(\boldsymbol{\tau}_t^k) - \delta)|b(\boldsymbol{\tau}_t^k) - \delta|^\rho + w_t^k r_t^k \geq 0, \forall k \in \mathcal{H}, \forall t \in [0,1]. \tag{9}$$

- The optimal control input is minimally modified input from FlowMatcher.
- The slack variable relax the constraint in the initial phase.

SafeFlowMatcher

- We built some Theorem and Proposition for SafeFlowMatcher.

$$\frac{d\boldsymbol{\tau}_t}{dt} = v_t(\boldsymbol{\tau}_t;\theta) + \Delta\mathbf{u}_t \triangleq \mathbf{u}_t, \qquad (8)$$

Definition 2 (Finite-Time Flow Invariance) *Let C^1 CBF b be such that $b(\boldsymbol{\tau}_t^k) \geq 0$. The system (8) is finite-time flow invariant if there exists $t_f \in [0,1]$ such that $b(\boldsymbol{\tau}_t^k) \geq 0$ for all $k \in \mathcal{H}$, $\forall t \geq t_f$.*

Theorem 1 (Forward Invariance for SafeFlowMatcher) *Let $b : \mathbb{R}^{d\times H} \to \mathbb{R}$ be a C^1 function, and define the robust safety set $\mathcal{C}_\delta \triangleq \{\boldsymbol{\tau}_t^k | b(\boldsymbol{\tau}_t^k) \geq \delta\}$ for some $\delta > 0$. Suppose the system (8) is controlled by $\mathbf{u}(t)$ satisfying the following barrier certificate for $0 < \rho < 1$, $\epsilon > 0$:*

$$\frac{db(\boldsymbol{\tau}_t^k)}{d\boldsymbol{\tau}_t^k}\mathbf{u}_t^k + \epsilon \cdot \mathrm{sign}(b(\boldsymbol{\tau}_t^k) - \delta)|b(\boldsymbol{\tau}_t^k) - \delta|^\rho + w_t^k r_t^k \geq 0, \forall k \in \mathcal{H}, \forall t \in [0,1]. \qquad (9)$$

Here, $w^k : [0,1] \to \mathbb{R}^{\geq 0}$ is a monotonically decreasing function with $w_s = 0$ for $s \in [t_w, 1]$, and r_t^k is a slack variable. The function w^k and parameters $t_w \in (0,1]$, r_t^k are user-defined. Then SafeFlowMatcher achieves finite-time flow invariance on $\mathcal{C}_\delta$.

KAIST

SafeFlowMatcher

- We built some Theorem and Proposition for SafeFlowMatcher.

Proposition 1 (Finite Convergence Time for SafeFlowMatcher) *Suppose Theorem 1 holds. Then for any initial trajectory* $\boldsymbol{\tau}_{t_0}^k \in \mathcal{D} \setminus \mathcal{C}_\delta$, *the state trajectory* $\boldsymbol{\tau}_t^k$ *converges to the safe set* $\mathcal{C}_\delta$ *within finite time*

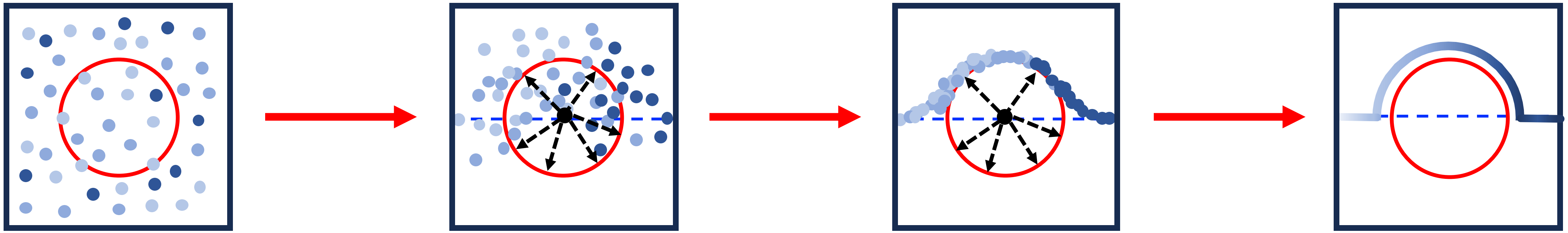
$$T \leq t_0 + \frac{(\delta - b(\boldsymbol{\tau}_{t_0}^k))^{1-\rho}}{\epsilon(1-\rho)}, \tag{10}$$

and remains in the set thereafter.

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

SafeFlowMatcher

- **SafeFlowMatcher**



1. SafeFlowMatcher generates a trajectory from Gaussian noise distribution.
2. In the initial phase, the effect of the CBF is soft.
3. In the final phase, the CBF enforces constraints more strongly.
4. As a result, a safe trajectory is generated.

Additional Technique: Adaptive Time Scheduling

- We proved that the following adaptive time scheduling can reduce global integration error when generating trajectories in flow matching using the Euler integrator.

Theorem 2 (Adaptive Scheduling for Forward Integration) *Suppose $v_t(\cdot;\theta)$ is Lipschitz continuous. If target distribution $q(\boldsymbol{\tau}_1)$ is on a compact set $\mathcal{K} \subset \mathbb{R}^d$ with diameter $R < \infty$, and a trajectory state $\boldsymbol{\tau}_t^k \in \mathcal{D}$ and a set of trajectory states $\mathcal{D}$ with diameter $M < \infty$, then the minimum error of integration for (8) is achieved by selecting timesteps Δt_i according to*

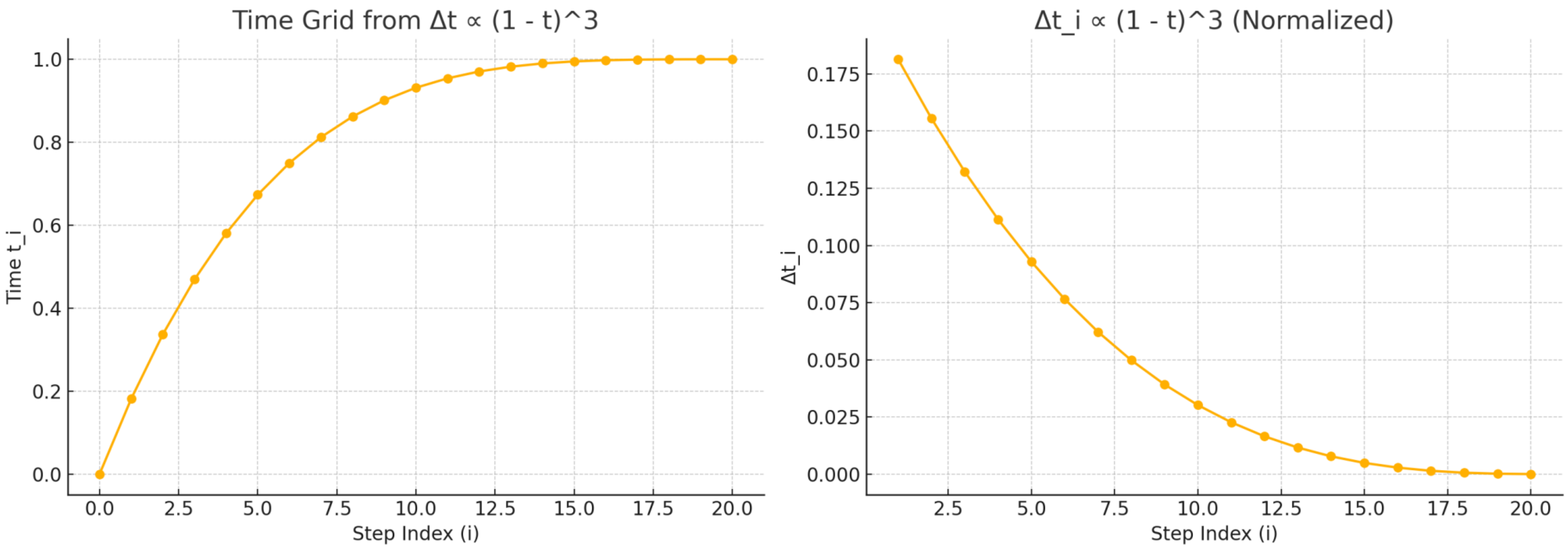
$$\Delta t_i \propto \frac{(1-t_i)^3}{2R(M+t_iR)+(1-t_i)^2}. \tag{14}$$

Since we defined $\Delta t_i \triangleq t_{i+1} - t_i$, Theorem 2 can be used to recursively generate t_{i+1} from t_i over each i. Theorem 2 tells us the step size Δt_i during forward integration can be chosen to be $O((1-t_i)^3)$ to reduce the global error of integration, thereby improving the accuracy of the generated trajectory. The proof of Theorem 2 is provided in Appendix B.

- Please refer to Appendix B for the proof of Theorem 2.

Additional Technique: Adaptive Time Scheduling

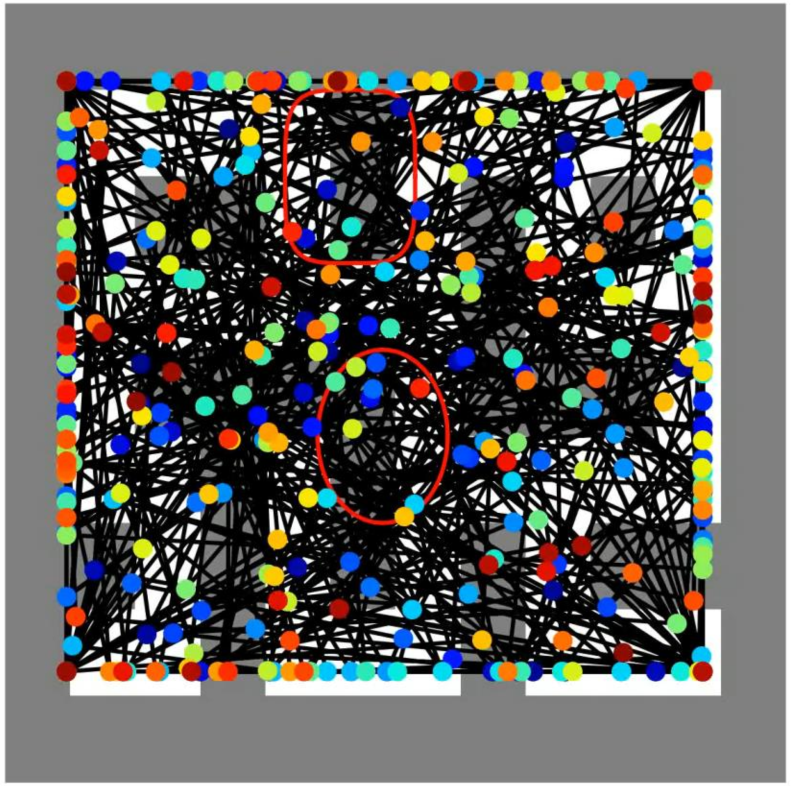
- Instead of using uniform timestep, using adaptive timestep $\boldsymbol{O}((\mathbf{1}-\boldsymbol{t})^{\mathbf{3}})$ in the maze environment leads to **accurate** generation.



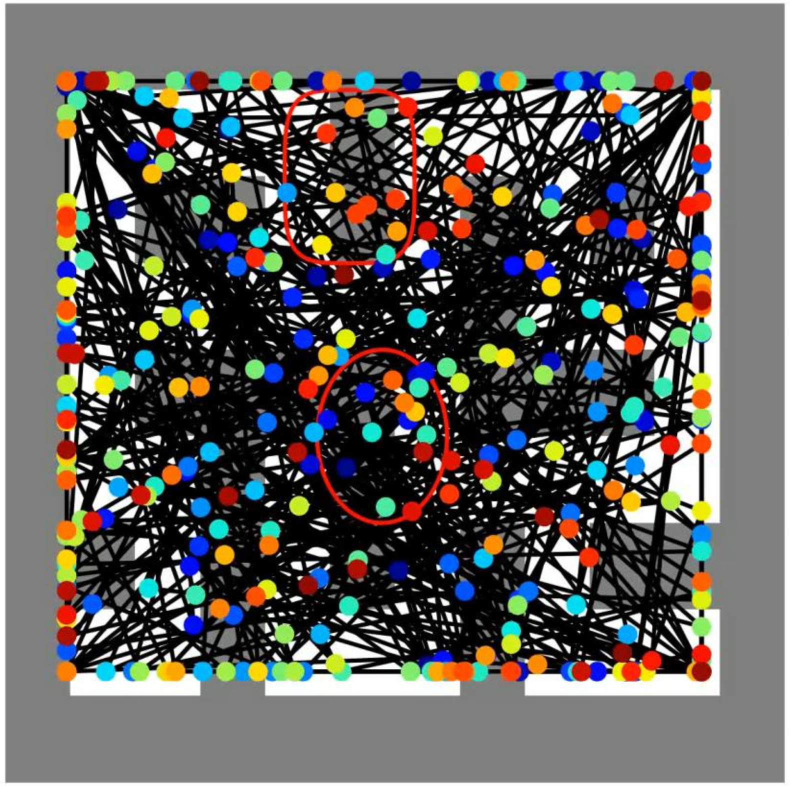
- We are planning to show experimental results in the final presentation.

Experiments: SafeDiffuser vs. SafeFlowMatcher

- We presented an initial prototype of SafeFlowMatcher and provided a preliminary comparison with SafeDiffuser [2].



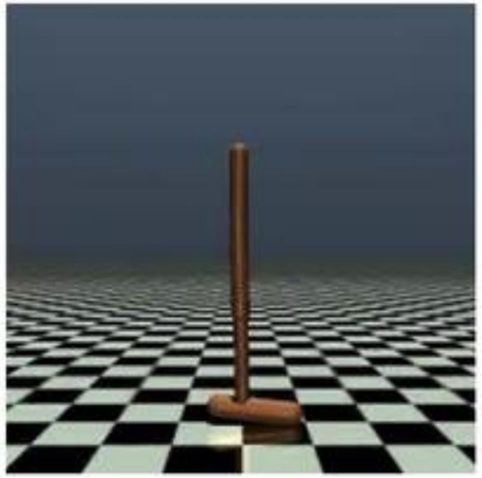
SafeDiffuser



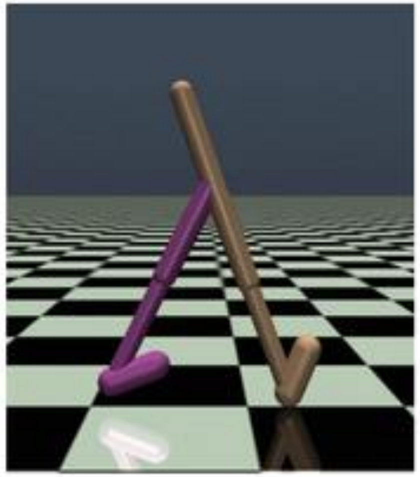
SafeFlowMatcher

Future Plan

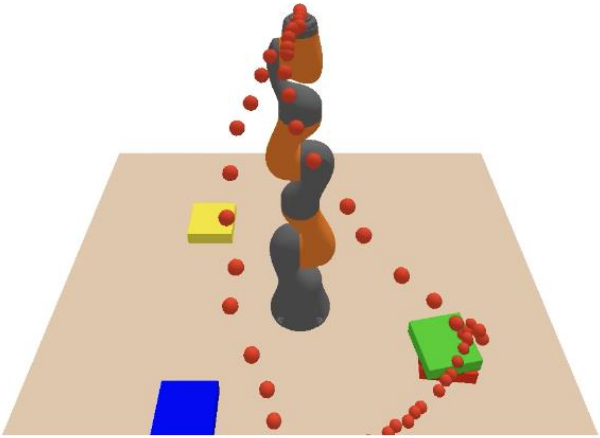
- We plan to extend our experiments to legged locomotion and manipulation tasks, comparing various metrics such as efficiency, safety, and other relevant factors.



Walker2D



Hopper



Manipulation

KAIST

Contributions

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

References

1. Janner, Y. Du, J. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. In International Conference on Machine Learning, pages 9902–9915. PMLR, 2022.
2. Wei, W. Tsun-Hsuan, G. Chuang, H. Ramin, L. Mathias, and R. Daniela. Safediffuser: Safe planning with diffusion probabilistic models. IEEE, 2025.
3. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2022.
4. P. Bhat and D. S. Bernstein. Finite-time stability of continuous autonomous systems. SIAM Journal on Control and optimization, 38(3):751–766, 2000.

Thank you

Appendix A: Proof of Theorem 1 and Proposition 1

Suppose the Lyapunov candidate function $V(\mathbf{x}_t) \triangleq \max(\delta - b(\mathbf{x}_t), 0)$.

Case 1: If $\mathbf{x}_{t_0} \in \mathcal{C}_\delta$ (i.e., $b(\mathbf{x}_{t_0}) \geq \delta$), then $V(\mathbf{x}_t) = 0$, and from the CBF inequality (9):

$$\dot{V}(\mathbf{x}_t) = -\dot{b}(\mathbf{x}_t) \leq \epsilon(b(\mathbf{x}_t) - \delta)^\rho = 0.$$

So $V(\mathbf{x}_t(t)) = 0$ for all t, which implies $b(\mathbf{x}_t(t)) \geq \delta$; the system stays in $\mathcal{C}_\delta$.

Case 2: If $\mathbf{x}_{t_0} \notin \mathcal{C}_\delta$ (i.e., $b(\mathbf{x}_{t_0}) < \delta$), then $V(\mathbf{x}_t) = \delta - b(\mathbf{x}_t) > 0$. The following finite-stability condition holds

$$\dot{V}(\mathbf{x}_t) = -\dot{b}(\mathbf{x}_t) \leq -\epsilon(\delta - b(\mathbf{x}_t))^\rho = -\epsilon V(\mathbf{x}_t)^\rho.$$

Define the comparison system

$$\dot{\phi}(t) = -\epsilon\phi(t)^\rho, \ \phi(t_0) = V(\mathbf{x}_{t_0}).$$

By the Comparison Lemma [26] (See Lemma 3.4), we have:

$$V(\mathbf{x}_t) \leq \phi(t), \ \forall t \geq t_0.$$

The solution $\phi(t)$ is

$$\phi(t) = \left(V(\mathbf{x}_{t_0})^{1-\rho} - (1-\rho)\epsilon(t - t_0)\right)^{\frac{1}{1-\rho}}, \ \text{for } t \geq t_0.$$

Appendix A: Proof of Theorem 1 and Proposition 1

Thus,

$$V(\mathbf{x}_t) \leq \left(V(\mathbf{x}_{t_0})^{1-\rho} - (1-\rho)\epsilon(t-t_0)\right)^{\frac{1}{1-\rho}}.$$

Hence, the state reaches the robust safe set $\mathcal{C}_\delta$ in finite time T that satisfy $V(\mathbf{x}_T) \leq \phi(T) = 0$. And we get the finite convergence time,

$$T = t_0 + \frac{V(\mathbf{x}_{t_0})^{1-\rho}}{\epsilon(1-\rho)} = t_0 + \frac{(\delta - b(\mathbf{x}_{t_0})^{1-\rho}}{\epsilon(1-\rho)}.$$

Therefore, for all $t \geq T$, we have $V(\mathbf{x}_t) \leq 0$, implying $\mathbf{x} \in \mathcal{C}_\delta$. This completes the proof of both Theorem 1 and Proposition 1.

KAIST

Appendix B: Proof of Theorem 2

Global Error Bound for Adaptive Euler Integration: We consider the initial value problem

$$\frac{d\mathbf{x}}{dt} = v(\mathbf{x}, t), \quad \mathbf{x}(0) = \mathbf{x}_0,$$

and discretize time as $0 = t_0 < t_1 < \cdots < t_T = 1$, with step sizes $\Delta t_k = t_{k+1} - t_k$. The Euler integration scheme is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_k v(\mathbf{x}_k, t_k).$$

A Taylor expansion of the exact solution $\mathbf{x}(t)$ about t_k yields

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \Delta t_k v(\mathbf{x}(t_k), t_k) + \frac{1}{2}\Delta t_k^2 \frac{d^2\mathbf{x}}{dt^2}(\xi_k),$$

for some $\xi_k \in [t_k, t_{k+1}]$.

Define the local error e_k as

$$e_k = \left\| \mathbf{x}(t_{k+1}) - (\mathbf{x}(t_k) + \Delta t_k v(\mathbf{x}(t_k), t_k)) \right\|.$$

From the Taylor expansion, it follows that

$$e_k \leq \left\| \frac{1}{2}\Delta t_k^2 \frac{d^2\mathbf{x}}{dt^2}(\xi_k) \right\| \leq M \Delta t_k^2,$$

where M is a constant.

Now, define the global error E_k as

$$E_k = \|\mathbf{x}(t_k) - \mathbf{x}_k\|.$$

KAIST

Appendix B: Proof of Theorem 2

We compute E_{k+1} as follows:

$$\begin{aligned}
E_{k+1} &= \|\mathbf{x}(t_{k+1}) - \mathbf{x}_{k+1}\| \\
&= \|\mathbf{x}(t_{k+1}) - (\mathbf{x}_k + \Delta t_k v(\mathbf{x}_k, t_k))\| \\
&= \|(\mathbf{x}(t_{k+1}) - \mathbf{x}(t_k) - \Delta t_k v(\mathbf{x}(t_k), t_k)) + \Delta t_k (v(\mathbf{x}(t_k), t_k) - v(\mathbf{x}_k, t_k)) + (\mathbf{x}(t_k) - \mathbf{x}_k)\| \\
&\leq e_k + \Delta t_k \|v(\mathbf{x}(t_k), t_k) - v(\mathbf{x}_k, t_k)\| + E_k.
\end{aligned}$$

Assuming that $v(\mathbf{x}, t)$ is Lipschitz continuous in $\mathbf{x}$ with Lipschitz constant $L(t_k)$, i.e.,

$$\|v(\mathbf{x}(t_k), t_k) - v(\mathbf{x}_k, t_k)\| \leq L(t_k)\|\mathbf{x}(t_k) - \mathbf{x}_k\| = L(t_k)E_k,$$

we obtain

$$E_{k+1} \leq e_k + (1 + \Delta t_k L(t_k))E_k.$$

Substituting the bound for the local error e_k,

$$E_{k+1} \leq M\Delta t_k^2 + (1 + \Delta t_k L(t_k))E_k.$$

Appendix B: Proof of Theorem 2

Applying the discrete Grönwall inequality [27] yields the following bound for the global error at step T:

$$E_T \leq \sum_{j=0}^{T-1} M \Delta t_j^2 \prod_{k=j+1}^{T-1} (1 + L(t_k)\Delta t_k).$$

Thus, the global error accumulates according to both the local errors and the amplification factors induced by the Lipschitz constants of v.

To control the global error, it is therefore beneficial to adapt the time step Δt_k based on the local Lipschitz constant. Specifically, choosing

$$\Delta t_k \propto \frac{1}{L(t_k)}$$

balances the error contribution at each step. In general, the Lipschitz constant is bounded by the operator (induced) norm of the Jacobian of v with respect to $\mathbf{x}$:

$$L(t_k) = \sup_{\mathbf{x}} \|\nabla_{\mathbf{x}} v(\mathbf{x}(t_k), t_k)\|.$$

Appendix B: Proof of Theorem 2

Lipschitz Constant of Jacobian of Velocity Field: Recall that under the flow matching formulation [5], the velocity field $v_t(\mathbf{x}) \triangleq v(\mathbf{x}(t), t))$ is defined as follows:

$$v_t(\mathbf{x}) = \int v_t(\mathbf{x}|\tilde{\mathbf{x}}_1) p_{1|t}(\tilde{\mathbf{x}}_1|\mathbf{x})\, d\tilde{\mathbf{x}}_1,$$

where the conditional vector field $v_t(\mathbf{x}|\tilde{\mathbf{x}}_1)$ takes the form

$$v_t(\mathbf{x}|\tilde{\mathbf{x}}_1) = \frac{\tilde{\mathbf{x}}_1 - \mathbf{x}}{1-t}.$$

Thus, the velocity field simplifies to

$$v_t(\mathbf{x}) = \frac{\mathbb{E}[\mathbf{x}_1|\mathbf{x}] - \mathbf{x}}{1-t}.$$

The Lipschitz constant of the velocity field v_t is defined as

$$L(t) = \sup_{\mathbf{x}} \|\nabla_{\mathbf{x}} v_t(\mathbf{x})\|.$$

Differentiating $v_t(\mathbf{x})$ with respect to $\mathbf{x}$, we obtain

$$\nabla_{\mathbf{x}} v_t(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} \mathbb{E}[\mathbf{x}_1|\mathbf{x}] - I}{1-t},$$

where I is the identity matrix. Applying the triangle inequality, we get

$$\|\nabla_{\mathbf{x}} v_t(\mathbf{x})\| \leq \frac{\|\nabla_{\mathbf{x}} \mathbb{E}[\mathbf{x}_1|\mathbf{x}]\| + 1}{1-t}.$$

Appendix B: Proof of Theorem 2

Suppose the data distribution $q(\mathbf{x}_1)$ is supported on a compact set $\mathcal{K} \subset \mathbb{R}^d$, and let the diameter of $\mathcal{K}$ be $R < \infty$. Then, by Bayes' rule, we have

$$q(\mathbf{x}_1|\mathbf{x}) = \frac{p_t(\mathbf{x}|\mathbf{x}_1)q(\mathbf{x}_1)}{\int_{\mathcal{K}} p_t(\mathbf{x}|\tilde{\mathbf{x}}_1)q(\tilde{\mathbf{x}}_1)d\tilde{\mathbf{x}}_1},$$

where

$$p_t(\mathbf{x}|\mathbf{x}_1) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{|\mathbf{x} - t\mathbf{x}_1|^2}{2\sigma^2}\right) \quad \text{and} \quad \sigma = 1 - t.$$

The posterior expectation is

$$\mu(\mathbf{x}) \triangleq \mathbb{E}[\mathbf{x}_1|\mathbf{x}] = \frac{\int_{\mathcal{K}} \tilde{\mathbf{x}}_1 p_t(\mathbf{x}|\tilde{\mathbf{x}}_1)q(\tilde{\mathbf{x}}_1)d\tilde{\mathbf{x}}_1}{\int_{\mathcal{K}} p_t(\mathbf{x}|\tilde{\mathbf{x}}_1)q(\tilde{\mathbf{x}}_1)d\tilde{\mathbf{x}}_1} \triangleq \frac{N(\mathbf{x})}{D(\mathbf{x})}.$$

Appendix B: Proof of Theorem 2

Differentiating $\mu(\mathbf{x})$ with respect to $\mathbf{x}$ yields under compact assumption, we can interchange the integral and the gradient

$$\nabla_{\mathbf{x}}\mu(\mathbf{x}) = \frac{\nabla_{\mathbf{x}}N(\mathbf{x})D(\mathbf{x}) - N(\mathbf{x})\nabla_{\mathbf{x}}D(\mathbf{x})}{D(\mathbf{x})^2} = \frac{\nabla_{\mathbf{x}}N(\mathbf{x}) - \mu(\mathbf{x})\nabla_{\mathbf{x}}D(\mathbf{x})}{D(\mathbf{x})}.$$

Due to $\nabla_{\mathbf{x}}p_t(\mathbf{x}|\mathbf{x}_1) = -\frac{\mathbf{x}-t\mathbf{x}_1}{\sigma^2}p_t(\mathbf{x}|\mathbf{x}_1)$, we have

$$\nabla_{\mathbf{x}}N(\mathbf{x}) = -\frac{1}{\sigma^2}\int_{\mathcal{K}} \tilde{\mathbf{x}}_1(\mathbf{x} - t\tilde{\mathbf{x}}_1)^\top p_t(\mathbf{x}|\tilde{\mathbf{x}}_1)q(\tilde{\mathbf{x}}_1)d\tilde{\mathbf{x}}_1,$$
$$\nabla_{\mathbf{x}}D(\mathbf{x}) = -\frac{1}{\sigma^2}\int_{\mathcal{K}} (\mathbf{x} - t\tilde{\mathbf{x}}_1)^\top p_t(\mathbf{x}|\tilde{\mathbf{x}}_1)q(\tilde{\mathbf{x}}_1)d\tilde{\mathbf{x}}_1.$$

Thus,

$$\nabla_{\mathbf{x}}\mathbb{E}[\mathbf{x}_1|\mathbf{x}] = \frac{1}{\sigma^2 D(\mathbf{x})}\int_{\mathcal{K}} (\mathbb{E}[\mathbf{x}_1|\mathbf{x}] - \mathbf{x}_1)(\mathbf{x} - t\mathbf{x}_1)^\top q(\mathbf{x}_1|\mathbf{x})d\mathbf{x}_1.$$

Taking the norm, by Jensen's Inequality and the property of 1-rank matrix, we have

$$\|\nabla_{\mathbf{x}}\mathbb{E}[\mathbf{x}_1|\mathbf{x}]\| \leq \frac{1}{\sigma^2}\mathbb{E}_{q(\cdot|\mathbf{x})}\left[\|\mathbb{E}[\mathbf{x}_1|\mathbf{x}] - \mathbf{x}_1\| \cdot \|\mathbf{x} - t\mathbf{x}_1\|\right] \leq \frac{1}{(1-t)^2}\cdot(2R)\cdot(\|\mathbf{x}\| + tR).$$

KAIST

Appendix B: Proof of Theorem 2

Since we consider a bounded domain $\mathcal{D}$ (e.g., a maze environment) with diameter $M < \infty$, we have $\|\mathbf{x}\| \leq M$ for all $\mathbf{x} \in \mathcal{D}$. Therefore, the above norm is uniformly bounded:

$$\|\nabla_{\mathbf{x}}\mathbb{E}[\mathbf{x}_1|\mathbf{x}]\| \leq \frac{2R(M+tR)}{(1-t)^2}.$$

Since σ is chosen to be $1-(1-\sigma_{\min})t$ in practical optimal transport, instead $1-t$ [5], we have

$$L(t) \leq \frac{2R(M+tR)+(1-(1-\sigma_{\min})t)^2}{(1-(1-\sigma_{\min})t)^3}, \tag{16}$$

where $t \in [0,1]$.

Conclusion: To control the global error, it is desirable to adapt the time step Δt_k based on the local Lipschitz constant. Due to Equation (16), it follows that

$$\Delta t_k \propto \frac{1}{L(t_k)} \propto \frac{(1-t)^3}{2R(M+tR)+(1-t)^2}.$$

Complete the proof.