MA-SafeDiffuser: Safe Multi-Agent Planning with Diffusion Probabilistic Models

Kiran Ravish
IIIT Hyderabad
Hyderabad, India
kiran.ravish@research.iiit.ac.in

Ankita Kushwaha
IIIT Hyderabad
Hyderabad, India
ankita.kushwaha@research.iiit.ac.in

Preeti
IIIT Hyderabad
Hyderabad, India
preeti.preeti@research.iiit.ac.in

Pawan Kumar
IIIT Hyderabad
Hyderabad, India
pawan.kumar@iiit.ac.in

ABSTRACT

We propose *MA-SafeDiffuser*, a multi-agent extension of SafeDiffuser that equips diffusion-based trajectory planners with *finite-time diffusion invariance* guarantees for joint-agent safety specifications. Building on the single-agent construction that embeds control barrier function (CBF) constraints into reverse-diffusion updates, we: (i) formalize joint safe sets as intersections of per-agent and pairwise barrier sets; (ii) derive centralized and decentralized (communication-aware) constrained denoising procedures with provable invariance under mild assumptions; (iii) address local-trap and deadlock phenomena via time-varying specifications and liveness CBFs; and (iv) develop a lightweight benchmarking suite including a multi-agent Maze2D domain. Empirically, MA-SafeDiffuser reduces violation counts relative to unconstrained diffusion baselines, while retaining planning quality. We provide algorithms, proofs, and reference implementations.

KEYWORDS

Multiagent; Planning; Diffusion Model

ACM Reference Format:
Kiran Ravish, Ankita Kushwaha, Preeti, and Pawan Kumar. 2026. MA-SafeDiffuser: Safe Multi-Agent Planning with Diffusion Probabilistic Models. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/PAPW1165

1 INTRODUCTION

Diffusion probabilistic models (DDPMs) [10] were recently proposed as a powerful class of generative models whose iterative denoising dynamics can be repurposed for *trajectory synthesis* and *planning*. Recent exciting work such as Diffuser [11] and subsequent follow-ups on classifier/energy guidance [5] demonstrate that diffusion models can indeed produce long-horizon behaviors that emulates task-level structure and can also leverage heterogeneous offline experience. When applied casually, however, unconstrained denoising provides no formal safety guarantees: for instance, an optimized sample may graze or penetrate obstacles, violate actuation limits, or collide with other agents.

Extension from diffusion to safe diffusion. The SafeDiffuser framework [20] addresses this issue in the *single-agent* setting by embedding *Control Barrier Function* (CBF) inequalities [1] into each reverse step of the diffusion process. Conceptually, the unconstrained denoiser proposes an update and a safety projector modifies it to satisfy linearized CBF rows; an execution-time filter then mirrors the same CBF constraints in receding horizon. This yields *finite-time diffusion invariance* for the safe set while retaining the modeling flexibility of diffusion planning. Extensions of CBFs, e.g., exponential CBFs for higher relative degree [16] and nonsmooth barrier sets [9] further enlarge the class of safety specifications that can be handled.

Motivation for Multi-agent. Many real systems such as robot swarms, UAV traffic management, multi-arm manipulation, automated warehouses are inherently *multi-agent*. Safety must hold (i) for each agent with respect to walls, bounds, and actuation limits; (ii) *pairwise* to prevent inter-agent collisions; and sometimes (iii) *globally* for task-level constraints. Classical reactive methods such as artificial potential fields [12], velocity obstacles and RVO/ORCA [8, 19], and model-predictive control with hand-crafted penalties provide strong baselines in practice, but do not yield end-to-end invariance guarantees when coupled with learned generative planners and can deadlock under symmetric interactions. On the learning side, multi-agent RL (e.g., MADDPG [14], MAPPO [21]) excels at flexible coordination but typically lacks hard safety certificates and may require extensive online exploration; recent surveys on SafeRL and SafeMARL further highlight these limitations and open challenges in constrained multi-agent settings [13].

Our Contributions. We develop **MA–SafeDiffuser**, a *multi-agent* extension of SafeDiffuser that equips diffusion planning with centralized and decentralized safety enforcement and theoretical guarantees. The key idea is to formalize the *joint safe set* at each waypoint as the intersection of per-agent, pairwise, and (optional) task barriers, and to enforce the corresponding linearized CBF rows during denoising. A two-time formulation (reverse step j over planning waypoints k) enables *receding-horizon* enforcement at $k = 0$ together with temporal regularization and short multi-k seeding to distribute margin across all k.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/PAPW1165

Key Contributions.

- **Joint safety with diffusion.** A centralized projector that enforces the *conjunction* of per-agent and pairwise CBF rows throughout reverse denoising, with optional task constraints. A receding–horizon execution-time filter mirrors these rows to certify the applied control.
- **Finite-time diffusion invariance.** Under standard $C^{1,1}$ regularity, step-size bounds, and feasibility assumptions, we prove that the denoising process is *finite-time diffusion invariant* for the joint safe set, and we provide a discrete-time CBF comparison lemma for reverse-Euler updates.
- **Decentralized enforcement with neighbor messaging.** A communication–aware variant in which each agent enforces local self/task rows and *robust* pairwise rows using predicted neighbor states, responsibility shares, and a computable robustness margin that compensates for bounded prediction errors; invariance carries over under clear Lipschitz and step-bound conditions.
- **Practical deadlock mitigations.** Time-varying specifications (TVS) that relax early reverse steps, small symmetry-breaking biases in pairwise shares, and progress/liveness barriers toward corridor waypoints to avoid local traps; these mechanisms are orthogonal to the safety proof and improve success rates in practice.
- **Evaluation and differentiable enforcement.** A light Maze2D benchmark with door-aware geometry and per-step action clamps for fair Safe/Unsafe comparisons; enforcement can be implemented via differentiable QP layers (OptNet, CVXPY-Layers) [2, 6] or via parameter–free Dykstra projections onto half-spaces [3, 7], and solves in milliseconds in our experiments.

Our approach complements the growing literature on diffusion for decision making and control [5, 10, 11], including goal/skill-conditioned and score-guided variants, by *hardening* the generative sampler with CBF certificates. Relative to multi-agent navigation baselines (potential fields, VO/ORCA) [8, 12, 19] and MARL [14, 21], MA SafeDiffuser provides a *model-agnostic* safety layer that can be used with hand-crafted (score) denoisers or learned score networks and yields invariance guarantees at both denoising and execution time.

Organization of this paper. Section 2 formalizes the two-time indexing, barrier classes, and joint safe sets. Section 3 presents centralized and decentralized projectors and TVS/liveness devices. Section 5 establishes diffusion invariance and decentralized robustness. Section 6 reports Maze2D results and ablative analyses. Section 8 concludes the paper by summarizing limitations and future directions (larger agent counts, richer tasks, tighter learned scores).

2 PROBLEM FORMULATION AND NOTATION

States, trajectories, indices. We use the two-time notation of [20]: *diffusion time* $j \in \{0, \dots, N\}$ indexes the reverse steps of the denoiser, the *planning time* $k \in \{0, \dots, H\}$ indexes the $H+1$ waypoints of a trajectory. For M agents, we define agent a has state space $\mathcal{X}_a \subseteq \mathbb{R}^{n_a}$ and we have state $x_{a,k}^j \in \mathcal{X}_a$ at (j,k). We stack joint states as

$$\boldsymbol{x}_k^j \;=\; \big(x_{1,k}^j, \dots, x_{M,k}^j\big) \in \mathcal{X} \;:=\; \mathcal{X}_1 \times \cdots \times \mathcal{X}_M, \quad n \;=\; \sum_{a=1}^{M} n_a.$$

We define a joint trajectory at diffusion step j to be $\tau^j = (\boldsymbol{x}_0^j, \dots, \boldsymbol{x}_H^j) \in \mathcal{X}^{H+1}$. We use the notation $\mathcal{T} := \mathcal{X}^{H+1}$ for the trajectory space and use bold symbols for joint quantities.

Reverse-time update. Let $u^j \in \mathbb{R}^{n(H+1)}$ denote the stacked reverse-time update at step j (arranged by waypoints and agents). The unconstrained reverse dynamics are formalized via the Euler limit as follows

$$\dot{\tau}^j \;=\; \lim_{\Delta\tau \to 0} \frac{\tau^j - \tau^{j+1}}{\Delta\tau} \;=\; u^j, \text{ equivalently, } \tau^j \;=\; \tau^{j+1} + \Delta\tau\, u^j, \tag{1}$$

where u^j is produced by the denoiser's drift (reference interpolation + temporal smoothing) and this is subsequently *projected* to enforce safety constraints [11, 20].

Safety specifications. At each waypoint k, we have three classes of (Clarke-regular) barrier functions as follows:

- *Per-agent barriers* $b_{a,k}^{\text{self}} : \mathcal{X}_a \to \mathbb{R}$ with safe super-level sets $\{x_{a,k} : b_{a,k}^{\text{self}}(x_{a,k}) \geq 0\}$,
- *Pairwise barriers* $b_{ab,k}^{\text{pair}} : \mathcal{X}_a \times \mathcal{X}_b \to \mathbb{R}$ $(a < b)$ with safe sets $\{(x_{a,k}, x_{b,k}) : b_{ab,k}^{\text{pair}} \geq 0\}$,
- (optional) *Task or global barriers* $b_k^{\text{task}} : \mathcal{X} \to \mathbb{R}$ with safe set defined as $\{\boldsymbol{x}_k : b_k^{\text{task}}(\boldsymbol{x}_k) \geq 0\}$.

The *joint waypoint-wise safe set* now can be defined as follows

$$\begin{aligned} C_k \;:=\; & \bigcap_{a=1}^{M} \big\{x_{a,k} : b_{a,k}^{\text{self}}(x_{a,k}) \geq 0\big\} \\ & \cap \bigcap_{a<b} \big\{(x_{a,k}, x_{b,k}) : b_{ab,k}^{\text{pair}}(x_{a,k}, x_{b,k}) \geq 0\big\} \\ & \cap \big\{\boldsymbol{x}_k : b_k^{\text{task}}(\boldsymbol{x}_k) \geq 0\big\}. \end{aligned} \tag{2}$$

We can now extend this safe waypoint notion to safe trajectories: a trajectory $\tau = (\boldsymbol{x}_0, \dots, \boldsymbol{x}_H) \in \mathcal{T}$ is *safe* iff $\boldsymbol{x}_k \in C_k$ for all k. The *trajectory-level safe set* is then written as follows

$$C \;:=\; \big\{\tau \in \mathcal{T} :\; \boldsymbol{x}_k \in C_k \text{ for all } k = 0, \dots, H\big\}.$$

Definition 1 (Multi-agent finite-time diffusion invariance)**.** The reverse-time denoising process is *finite-time diffusion invariant* for C if there exists an index $i \in \{0, \dots, N\}$ such that, for every $j \leq i$, the denoised trajectory τ^j is safe: $\tau^j \in C$, i.e., $\boldsymbol{x}_k^j \in C_k$ for all k.

Remark 2 (Enforcement index and receding horizon)**.** When it comes to implementing CBFs, we usually apply the linearized constraints at the very first point in time, or $k = 0$, which is referred to as the receding horizon, and then rely on temporal regularization of the reverse model and a very short period of multi-k enforcement to get a safety net that covers the rest of the planning horizon. See §3 and §5).

3 METHOD: MA-SAFEDIFFUSER

In our multi-agent extension, we enforce safety by *projecting* the unconstrained reverse-diffusion drift onto linearized CBF half-spaces at a designated enforcement index. By default we choose $k = 0$, as

Algorithm 1 MA-SafeDiffuser: Centralized projector (with one reverse step at j). Here $\gamma : [0, N] \to \mathbb{R}_{\geq 0}$ is the TVS *relaxation schedule*-a nonincreasing function of the reverse index j (with $\gamma(0) = 0$) that temporarily offsets the barrier and whose (discrete) diffusion-time derivative is denoted $\dot{\gamma}(j)$.

Require: Previous trajectory $\tau^{j+1} = (\boldsymbol{x}_0^{j+1}, \ldots, \boldsymbol{x}_H^{j+1})$; unconstrained reverse drift u_0^j (from (1)); barrier rows $\{b_\ell\}$ at enforcement index $k = 0$; variant $\in$ {RoS, ReS, TVS}; step $\Delta\tau$; step bound $u_{\max}$; weights (λ, ω^j)

Ensure: Projected update u^j and next trajectory τ^j

1: **Linearize barriers at $k = 0$:** for each active row b_ℓ (self/pair/task) form gradient $\nabla b_\ell(\boldsymbol{x}_0^j)$ and right-hand side h_ℓ^j: **RoS/ReS:** $h_\ell^j \leftarrow \alpha_\ell(b_\ell(\boldsymbol{x}_0^j))$;

2: **TVS:** $h_\ell^j \leftarrow \alpha_\ell(b_\ell(\boldsymbol{x}_0^j) - \gamma(j)) - \dot{\gamma}(j)$.

3: **Build convex program** in decision variables (u^j, r^j) as follows

$$\min \ \|u^j - u_0^j\|_2^2 + \lambda\|r^j\|_2^2$$
$$\text{s.t.} \quad \nabla b_\ell(\boldsymbol{x}_0^j)^\top u^j \ \geq \ -\omega_\ell^j r_\ell^j + h_\ell^j, \ \|u_a^j\|_2 \leq u_{\max} \ \forall a.$$

4: **Solve** (QP or Dykstra on half–spaces) to obtain (u^j, r^j); set $r^j = 0$ in RoS.

5: **Update reverse step:** $\tau^j \leftarrow \tau^{j+1} + \Delta\tau\, u^j$.

6: **(Optional all-k enforcement)** If required (cf. §5), repeat 1–5 with rows at all k for a fixed number of reverse steps.

7: **return** τ^j.

receding horizon at *every* reverse step j. The same filter is applied at execution time to the instantaneous control. In our proposal we have the following: (i) a *central* projector that is responsible for enforcing all per-agent, pairwise, and task rows jointly (see §4.1); (ii) a *decentralized* variant with neighbor messaging and robust margins (see §4.2); and (iii) a set of time-varying specifications and liveness mechanisms that avoids deadlock (see §4.4). The regularization of the reverse model and/or a brief period of enforcing the model at different values of k send the margin to all values of k, when the constraints are only applied at $k = 0$ (see cf. §5).

4 ALGORITHMS

4.1 Centralized joint enforcement

For the reverse step j we let $u_0^j \in \mathbb{R}^{n(H+1)}$ denote the stacked unconstrained drift (cf. (1)). We solve a minimum-deviation objective at index $k = 0$, by linearizing the active barriers as follows.

$$\min_{u^j, r^j} \ \|u^j - u_0^j\|_2^2 \ + \ \lambda \|r^j\|_2^2 \qquad \text{s.t.}$$

$$\text{(self)} \ \nabla b_{a,0}^{\text{self}}(x_{a,0}^j)^\top u_a^j \ + \ \alpha\big(b_{a,0}^{\text{self}}(x_{a,0}^j)\big) \ \geq \ -\omega_a^j r_a^j, \qquad \forall a \in \{1, \ldots, M\}, \tag{3}$$

$$\text{(pair)} \ \nabla_{x_a} b_{ab,0}^{\text{pair}}(x_{a,0}^j, x_{b,0}^j)^\top u_a^j \ + \ \nabla_{x_b} b_{ab,0}^{\text{pair}}(x_{a,0}^j, x_{b,0}^j)^\top u_b^j \ + \ \alpha\big(b_{ab,0}^{\text{pair}}(\cdot)\big) \ \geq \ -\omega_{ab}^j r_{ab}^j, \quad \forall a < b, \tag{4}$$

$$\text{(task)} \ \nabla b_0^{\text{task}}(\boldsymbol{x}_0^j)^\top u^j \ + \ \alpha\big(b_0^{\text{task}}(\boldsymbol{x}_0^j)\big) \ \geq \ -\omega_0^j r_0^j \tag{5}$$

here $u^j = (u_1^j, \ldots, u_M^j)$, α is an extended class-$\mathcal{K}$ function, and r^j collects relaxation variables. The variants are:

- **RoS** (robust-safe): here we consider hard enforcement with $r^j \equiv 0$ and all $\omega^j \equiv 0$.
- **ReS** (relaxed-safe): here we allow $r^j \neq 0$ with diffusion–time weights $\omega^j \downarrow 0$ to recover feasibility; feasibility gaps vanish as j decreases.
- **TVS** (time-varying-safe): here we replace each b by $b - \gamma(j)$ and augment the RHS by $-\dot{\gamma}(j)$ (the linearization is applied to $b - \gamma$).

During numerical simulations, imposing step bounds (e.g., $\|u_a^j\|_2 \leq u_{\max}$ for each agent block) is useful for controlling truncation errors and ensuring numerical stability. The resulting QP can be solved using any sparse convex QP solver (e.g., OSQP or CVXPY-Layers [6]). When all constraints are linearized half-spaces, one may alternatively employ Dykstra-type projection methods [3, 7] (see §4.3).

All-k enforcement (optional). In case, the denoiser lacks a temporal contraction, we may enforce (3)–(5) for all $k \in \{0, \ldots, H\}$ during the first J reverse steps; this will ensure a safety margin at every waypoint that persists under the subsequent $k = 0$-only enforcement (cf. 7), however, at the expense of more possibly more compute.

4.2 Decentralized enforcement with neighbor messaging

In an environment with multiple agents and limited long range communication, we may consider a decentralized variant of our method. To this end, each agent a solves a *local* projector using *predicted* neighbor states denoted by $\widehat{x}_{b,0}^j$. These predicted neighbor states are obtained from the previous communication round, for example. In the decentralized setting, we need to localize the barriers. For pairwise barriers, we linearize locally at $(x_{a,0}^j, \widehat{x}_{b,0}^j)$ and we enforce a *robust* inequality with *responsibility sharing* $\rho_{ab}, \rho_{ba} \in (0, 1)$, $\rho_{ab} + \rho_{ba} = 1$:

$$\nabla_{x_a} b_{ab,0}^{\text{pair}}(x_{a,0}^j, \widehat{x}_{b,0}^j)^\top u_a^j \ + \ \rho_{ab}\, \alpha\big(b_{ab,0}^{\text{pair}}(x_{a,0}^j, \widehat{x}_{b,0}^j)\big) \ \geq \ m_{\text{rob}},$$

(and similarly for agent b). With one to three Jacobi and ADMM-style rounds [4] per diffusion step, messages are updated and the local projections are recomputed. Under prediction error $\|\widehat{x}_{b,0}^j - x_{b,0}^j\| \leq \varepsilon$, Lipschitz constants L_b (barrier value), L_∇ (barrier gradients), L_α (class-$\mathcal{K}$), and per–step bounds $\|u_a^j\| \leq u_{\max}$, it suffices (see Theorem 8) to pick

$$m_{\text{rob}} \ \geq \ \tfrac{1}{2}\Big(2\, L_\nabla\, u_{\max} \ + \ L_\alpha\, L_b\Big)\, \varepsilon, \tag{6}$$

to guarantee that the *true* centralized pairwise inequalities (with $(x_{a,0}^j, x_{b,0}^j)$) are satisfied (see Theorem 8 for a detailed proof). We note here that self and task barriers do not depend on predictions and they can be enforced locally similar to the centralized case.

4.3 Execution-time safety filter

On completion of the reverse steps at index j, we get a first waypoint proposal, namely, $x_0^j \mapsto x_1^j$. Before execution at task time t, we solve the *instantaneous* CBF projection for $u_t = x_{t+1} - x_t$

with the same rows (now evaluated at x_t) and alongwith explicit step bounds, e.g., $\|u_t\|_2 \le u_{\max}$ (or for example, $\|u_t\|_\infty \le u_{\max}$). As mentioned before, this receding-horizon filter prevents rare large steps from creating violations and mirrors SafeDiffuser's deployment [20]. When constraints are linearized half-spaces, Dykstra's algorithm provides a parameter-free projection onto the intersection of half–spaces [3, 7]; otherwise, we may prefer to solve a small convex QP problem. The execution-time filter is essential to preserve invariance for the applied control (cf. 7).

4.4 Avoiding traps and deadlock

Despite the safety constraints using control barriers, we may encounter deadlocks, and local traps. To this end, we propose to employ three practical devices to reduce deadlock and local traps without sacrificing (safety) invariance:

(1) **Time-varying specifications (TVS).** Use $b - \gamma(j)$ with γ nonincreasing in reverse time and $\gamma(0) = 0$; constraints are relaxed early and tightened near $j = 0$.
(2) **Symmetry breaking.** We add small bias offsets or priorities in pairwise shares (ρ_{ab}, ρ_{ba}) to avoid stalemates in symmetric configurations. For example, $\rho_{ab} > \rho_{ba}$ will avoid the symmetry.
(3) **Liveness/progress CBFs.** We include auxiliary barriers that encode progress toward intermediate subgoals for example, corridor waypoints, ensuring nonzero "driving" terms when pure safety is noninformative.

As we show later, these mechanisms are orthogonal to the core safety projector and are compatible with the invariance proofs we derive in Section 5.

5 THEORETICAL RESULTS

We prove finite-time diffusion invariance under standard smoothness and feasibility assumptions, and we also provide a decentralized robustness guarantee.

Assumption 3 (Regularity and feasibility)**.** All active barriers are assumed to be b_ℓ are $C^{1,1}$ on an open set $\mathcal{U} \supset C$, i.e., b_ℓ are continuously differentiable and we assume that their gradients are locally Lipschitz on $\mathcal{U}$; moreover, for each barrier, there exists an extended class-$\mathcal{K}$ function α_ℓ. At every reverse step j, the projector (3)-(5) (RoS) is feasible or admits relaxations (ReS/TVS) with $\omega^j \downarrow 0$. The unconstrained drift u_0^j is assumed to be locally Lipschitz in τ^j and that it satisfies per-step bounds $\|u_a^j\| \le u_{\max}$.

Lemma 4 (Barrier conjunction). *Let $C = \bigcap_{\ell=1}^{L}\{z : b_\ell(z) \ge 0\}$ and consider $\dot z = u(z)$. If for all ℓ and all z, $\nabla b_\ell(z)^\top u(z) + \alpha_\ell(b_\ell(z)) \ge 0$, then C is forward invariant, that is, we have $z(0) \in C \Rightarrow z(t) \in C$ for all $t \ge 0$.*

Proof Sketch. We know that each single set $S_\ell = \{b_\ell \ge 0\}$ is forward invariant by the CBF condition; as we know that at boundary points $\{b_\ell = 0\}$ the field is tangent–inward [1, 9, 15]. For smooth inequality sets, the tangent cone of the intersection equals the intersection of tangent cones, so $u(z) \in T_{S_\ell}(z)$ for all active ℓ implies $u(z) \in T_C(z)$, and invariance will follow from Nagumo's theorem [15]. The same conclusion will hold for locally Lipschitz b_ℓ when using Clarke generalized gradients. More detail can be found in supplementary. □

Algorithm 2 MA-SafeDiffuser-DEC: Decentralized projector (one reverse step at j)

Require: Neighbor graph $\mathcal{G}$; last–round neighbor predictions $\widehat{x}_{b,0}^j$; robust margin m_{rob}; shares $\rho_{ab}, \rho_{ba} \in (0,1)$, $\rho_{ab}+\rho_{ba} = 1$; per–agent bound $u_{\max}$
Ensure: Local updates $\{u_a^j\}_{a=1}^M$ and aggregated u^j
1: **for** $a \in \{1, \dots, M\}$ **in parallel do**
2: **Self/task rows:** build $\nabla b_{a,0}^{\text{self}}(x_{a,0}^j)$ and (if present) global rows as in Alg. 1.
3: **Pairwise rows (robust):** for each $b \in \mathcal{N}(a)$, linearize at $(x_{a,0}^j, \widehat{x}_{b,0}^j)$ and enforce
$$\nabla_{x_a} b_{ab,0}^{\text{pair}}(x_{a,0}^j, \widehat{x}_{b,0}^j)^\top u_a^j \;+\; \rho_{ab}\,\alpha\big(b_{ab,0}^{\text{pair}}(x_{a,0}^j, \widehat{x}_{b,0}^j)\big) \;\ge\; m_{\text{rob}}.$$
4: **Local solve:** minimize $\|u_a^j - u_{0,a}^j\|_2^2$ subject to the above and $\|u_a^j\|_2 \le u_{\max}$; broadcast updated $x_{a,0}^j$ to $\mathcal{N}(a)$.
5: **end for**
6: **Repeat** the local step for $R \in \{1, 2, 3\}$ message–passing rounds.
7: **Aggregate** $u^j \leftarrow (u_1^j, \dots, u_M^j)$ and **update** $\tau^j \leftarrow \tau^{j+1} + \Delta\tau\, u^j$.
8: **return** τ^j.

A discrete-time CBF lemma (reverse time). We use the following Euler-discretized comparison, which basically follows from $C^{1,1}$ regularity.

Lemma 5 (Discrete-time CBF monotonicity). *Let $b \in C^{1,1}$ in a neighborhood of C with locally Lipschitz gradient constant $L_{\nabla b}$ and let α be extended class-$\mathcal{K}$. Suppose u satisfies $\nabla b(x)^\top u + \alpha(b(x)) \ge 0$ and consider the reverse-time Euler step $x^j = x^{j+1} + \Delta\tau\, u^j$ with $\|u^j\| \le u_{\max}$. Then for $\Delta\tau \le \bar\Delta := \min\{1, \alpha'(0)/(2L_{\nabla b}u_{\max})\}$ there exists $c > 0$ such that*
$$b(x^j) \;\ge\; b(x^{j+1}) \;-\; \Delta\tau\,\alpha\big(b(x^j)\big) \;-\; c\,\Delta\tau^2$$
In particular, we have that if $b(x^{j+1}) \ge 0$ then $b(x^j) \ge 0$ for all $\Delta\tau \le \bar\Delta$.

Proof Sketch. By $C^{1,1}$,
$$b(x^j) = b(x^{j+1}) + \nabla b(x^j)^\top (x^j - x^{j+1}) + O(\|x^j - x^{j+1}\|^2)$$
with the remainder bounded by $L_{\nabla b}\Delta\tau^2\|u^j\|^2$. We substitute $x^j - x^{j+1} = \Delta\tau\, u^j$ and then use $\nabla b(x^j)^\top u^j \ge -\alpha(b(x^j))$. The step-size condition ensures the implicit inequality preserves nonnegativity when $b(x^{j+1}) \ge 0$ (discrete comparison). □

Remark 6 (Finite-time reach under RoS/TVS)**.** If $b(x^N) < 0$ (RoS), one can choose α with finite-time property $\int_{-a}^{0} \frac{ds}{\alpha(s)} < \infty$ on $[-a, 0]$ and $\Delta\tau$ small so that the $O(\Delta\tau^2)$ truncation is dominated by $\Delta\tau\,\alpha(\cdot)$; then the discrete recursion crosses into $b \ge 0$ in finitely many steps. Under TVS with $\tilde b := b - \gamma(j)$, $\gamma(0) = 0$, the same argument would apply to $\tilde b$, giving $\tilde b(x^0) \ge 0$.

Theorem 7 (Centralized multi–agent finite–time diffusion invariance). *Under Assumption 3, if at every reverse step j the centralized projector enforces* (3)-(5) *(RoS or TVS) for all per-agent, pairwise, and task barriers at the enforcement index, which by default*

is $k = 0$, then the denoising process is finite-time diffusion invariant for C with probability 1 (w.r.t. model noise). Moreover, the execution-time safety filter preserves invariance for the applied control u_t.

Sketch of Proof. We fix any barrier b_ℓ at $k = 0$. By Lemma 5, the enforced inequality implies no boundary crossing in reverse time (considering small $\Delta\tau$), so $b_\ell(x_0^j) \geq 0$ for all $j \leq i$ for some $i \leq N$ (note that finite-time reach holds by Remark 6). By Lemma 4, the conjunction over all active rows is forward invariant at $k = 0$. If all k are enforced, we are basically done. Otherwise, we either (i) enforce all k for finitely many reverse steps to seed a margin everywhere, or (ii) we rely on temporal regularization of the reverse model to propagate a margin from $k = 0$ to all k (both are standard in the baseline SafeDiffuser [20] paper). The "probability 1" qualifier here is conditional on the event that each QP (or projection) is feasible; by Assumption 3 this event has probability 1 since the projection is a deterministic measurable map of the model noise. The execution–time CBF filter then basically satisfies $\nabla b_\ell(x_t)^\top u_t + \alpha_\ell(b_\ell(x_t)) \geq 0$, hence preserves invariance per a discrete-time CBF argument identical to Lemma 5. More detail can be found in appendix. □

Theorem 8 (Decentralized invariance under bounded error). *Assume neighbor messages yield predictions $\widehat{x}_{b,0}^j$ with bounded error $\|\widehat{x}_{b,0}^j - x_{b,0}^j\| \leq \varepsilon$. Suppose each agent enforces its local pairwise CBF with a robust margin $m_{\text{rob}} > 0$ and responsibility shares $\rho_{ab}, \rho_{ba} \in (0,1)$, $\rho_{ab}+\rho_{ba} = 1$. Let L_b and L_∇ be local Lipschitz constants for b_{ab} and its gradients, L_α for α, and assume per-step bounds $\|u_a^j\| \leq u_{\max}$. If we have*

$$m_{\text{rob}} \;\geq\; \tfrac{1}{2}\left(2\,L_\nabla\,u_{\max} \;+\; L_\alpha\,L_b\right)\varepsilon \tag{7}$$

then the decentralized projected updates satisfy the true *centralized pairwise inequalities. Consequently, the conclusions of 7 hold.*

Sketch of Proof. By Summing the two local inequalities (one at agent a, one at agent b) built with predicted neighbor states, and compare them to the centralized inequality evaluated at the *true* joint state. Use Lipschitz bounds on barrier values and gradients and the per-step bound to show the centralized left-hand side exceeds the predicted one minus the error term $(2L_\nabla u_{\max}+L_\alpha L_b)\varepsilon$. Choosing m_{rob} as in (7) makes the sum nonnegative, hence the true centralized pairwise inequality holds. Taking the finite conjunction over all pairs and combining with self/task rows (which do not require predictions) gives the desired centralized feasibility, so 7 applies. □

Remark 9 (Complexity and solvers)**.** The centralized QP has dimension $(H+1)n$ and $O(M^2)$ sparse rows from pairwise constraints. First-order operator-splitting solvers (e.g., OSQP [17]) and differentiable layers (OptNet [2], CVXPY-Layers [6]) are suitable; when constraints are linearized half-spaces, Dykstra-type projections offer a parameter-free alternative [3, 7].

6 NUMERICAL EXPERIMENTS

We evaluate **MA-SafeDiffuser** in **Maze2D** domain to validate multi-agent safety and study the effect of reward shaping and fairness clamps. All experiments were done on RTX 4090 GPUs using python and standard pytorch libraries like pytorch and numpy in double precision arithmetic.

Algorithm 3 Execution-time safety filter (receding horizon at task time t)

Require: Current state x_t; proposal x_{t+1}^{prop}; bound $u_{\max}$
Ensure: Safe control u_t and applied next state x_{t+1}
1: $u_t^0 \leftarrow x_{t+1}^{\text{prop}} - x_t$
2: Project u_t^0 onto the intersection of instantaneous rows $\nabla b_\ell(x_t)^\top u_t \geq \alpha_\ell(b_\ell(x_t))$ and $\|u_t\|_2 \leq u_{\max}$ (QP or Dykstra)
3: $x_{t+1} \leftarrow x_t + u_t$; **return** (u_t, x_{t+1})

6.1 Maze2D (Multi-Agent) setup

Environment. The Maze2D domain is a continuous planar environment bounded within the axis–aligned box $\mathcal{B} = \{x \in \mathbb{R}^2 : (-1.5,-1.5) \leq x \leq (1.5,1.5)\}$. The interior of $\mathcal{B}$ is partitioned into a 6×6 grid of square rooms, each separated by solid wall segments except for small open passages (doors) centered on the walls. Each wall is represented as a line segment $w_\ell = [a_\ell, b_\ell]$, while each door is modeled as a short open gap (p_0, p_1) on the same line. Walls are impassable obstacles, whereas doors define narrow traversable corridors through which agents can pass. In the Figures 1, 2, walls are shown in **black**, door openings in **green**, and goal regions as **orange boxes**. Agents are depicted as colored points that begin near random valid start positions (blue markers) and must navigate to their respective goal boxes while avoiding wall and inter-agent collisions. We evaluate configurations with $M \in \{2, 4, 8\}$ agents. The kinematic update rule follows $x_{i,t+1} = x_{i,t} + u_{i,t}$ with bounded control $\|u_{i,t}\|_2 \leq u_{\max}$, and all motion takes place continuously in $\mathbb{R}^2$.

Safety constraints. (i) **Box invariance**: $x_{i,t} \in \mathcal{B}$; (ii) **Wall clearance**: $d(x_{i,t}, w_\ell) \geq \rho_{\text{agent}}+\rho_{\text{wall}}$ with $(\rho_{\text{agent}}, \rho_{\text{wall}}) = (0.05, 0.06)$; (iii) **No wall crossing**: $[x_{i,t}, x_{i,t+1}] \cap w_\ell = \varnothing$; (iv) **Pairwise separation**: $\|x_{i,t} - x_{j,t}\|_2 \geq \rho_{\text{pair}} = 0.25$ for $i \neq j$. The joint waypoint–wise safe set C_k is given by (2).

Planner. At each reverse step, MA-SafeDiffuser performs a score-guided denoising update followed by a projection onto the linearized safety set via Dykstra iterations. Both centralized and decentralized projection modes are supported: the centralized variant enforces all inter-agent constraints jointly, while the decentralized variant performs per-agent local projections using neighbor information within a communication radius of 0.9. The reference trajectory is generated via door-aware interpolation between start and goal waypoints.

Reward shaping and metrics. We measure three metrics per episode: (i) total violation count $V = \sum_t N_t^{\text{viol}}$ (primary safety metric); (ii) total return $R = \sum_t r_t$ based in goal progress, success bonus (+15) and step penalty (-0.002); (iii) success rate S= fraction of agents that reach goals before horizon T=700. To equalize control effort across planners, both Safe and Unsafe variants use identical action clamp $||u_t||_2 \leq u_{max} = 0.085$.

We use two shaping variants:

- *Progress-dominant shaping* rewards forward motion toward the goal and completion bonuses with mild step costs.

- *Safety shaping* augments with small penalties for wall proximity and intersection events, enabling finer differentiation between near-miss and hard violation.

Baselines and fairness. Unsafe = unconstrained denoising (no safety filter). Safe = MA–SafeDiffuser diffusion planner with CBF-style projection. Centralized vs. Decentralized = comparison between full-state projection and limited-neighbor projection under identical seeds and random noise. All runs use shared random seeds to ensure fair comparison under equivalent stochasticity.

Protocol and hyperparameters. Each configuration is run for three random seeds, across diffusion horizon $H \in \{20, 28, 36\}$, $N \in \{10, 16, 24\}$. Planner hyperparameters are fixed as $\beta = 0.70$, smoothing $\lambda_{smooth} = 0.35$, $\alpha_k = 3.0$, projection iterations = 80, $\Delta\tau = 0.06$. We evaluate both Safe and Unsafe planners over a 700-step horizon, resulting in 54 configurations across 3 seeds in total.

Violation counting. We follow a strict safety accounting scheme in all Maze2D experiments. At each timestep, the environment monitors three types of safety infractions: (i) *soft proximity violations* when an agent enters the near-wall buffer region $d(x_i, w_\ell) < \rho_{\text{agent}} + \rho_{\text{wall}} - 0.02$, (ii) *hard wall-cross violations* when the straight-line segment $[x_{i,t}, x_{i,t+1}]$ intersects any wall segment w_ℓ, and (iii) *boundary violations* when an agent exits the box domain $B = [-1.5, 1.5]^2$. Even visually collision-free trajectories (e.g., 2-agent Safe case in Figure. 2) may register a small number of soft violations when agents pass within a few centimeters of a wall or corner. This conservative counting ensures that the safety metric V penalizes both hard collisions and near-wall grazing events, encouraging overcautious rather than risky planning behavior.

6.2 Results and discussion

Quantitative Results. Across all settings, total returns scale approximately linearly with the number of agents M, as each agent receives an independent goal-progress reward and success bonus. The Safe variant consistently achieves comparable total reward to the Unsafe baseline while reducing violation counts by 40-60%. This demonstrates that enforcing control-barrier projections in the reverse denoising steps preserves planning performance without sacrificing efficiency.

Remark 10 (Ablations and fairness)**.** Without a uniform step bound, Unsafe can "jump" aggressively and *avoid* repeated proximity ticks, while Safe moves conservatively and may accumulate soft proximity penalties when sliding along walls. Enforcing the same instantaneous clamp for both methods, counting proximity only when *approaching* the wall, and using a short refractory period for proximity events restores a fair comparison and highlights the safety advantage of MA–SafeDiffuser. Increasing H or N yields smoother trajectories and fewer boundary contacts, confirming that longer reverse horizons enhance constraint satisfaction. However, very high values of H, N incur diminishing returns and longer runtime. Table 1 illustrates mean violations for Safe vs. Unsafe across all M, revealing clear safety gains that scale with the number of agents. Across all settings, total returns scale approximately linearly with the number of agents M, as each agent receives an independent goal-progress reward and success bonus.

Qualitative Results: Trajectory rollouts in Figure 2 show that MA-Safe Diffuser agents maintain safe clearances when crossing doorways and avoid wall clipping even under dense interactions. Unsafe denoisers exhibit oscillations and frequent minor breaches near corners.

6.2.1 Learned Score Denoiser (MLP MA-SafeDiffuser). To test generalization beyond handcrafted priors, we train a goal-augmented MLP score network that approximates the scaled denoising residual ϵ/σ from synthetic diffusion pairs.

Dataset. We generate 1500 training samples by adding Gaussian noise at random diffusion times to reference door-guided trajectories and record the normalized denoising targets. The input vector includes noisy trajectory, reference, goal offset, and sinusoidal time embedding.

Network: The MLP has two hidden layers of size $(256, 256)$ with ReLU activation and is trained using Adam for 1000 epochs with learning rate $1e-3$, weight decay $1e-6$, and per-feature normalization. The final training loss stabilized around 0.40 (normalized MSE), confirming moderate convergence.

Evaluation: The trained MLP is used within the MA-Safe Diffuser planner ($H = 28$, $N = 40$) for both Safe and Unsafe variants. Representative results are in Table 2.

The safe variant reduces violation count by $\sim 50\%$ while maintaining identical return and success rate, confirming that the learned denoiser preserves CBF-based safety enforcement. Trajectory visualizations as shown in Figure 1 show smoother door transitions and fewer wall contacts compared to the unconstrained baseline. Our results on MLP based denoiser is preliminary, and very likely needs finetuning. For the 2DMaze experiment, our method is able to learn the trajectories well without using MLP.

Comparison with SafeRL Methods. Table 3 compares MA-Safe Diffuser with MAPPO-Lagrangian under identical door-based waypoint guidance. While MAPPO-Lagrangian enforces safety via a learned soft-penalty mechanism, this approach leads to stalling behavior in the narrow 6×6 maze, resulting in a 0% success rate even after 600 training episodes within the 700-step horizon. In contrast, MA-SafeDiffuser applies zero-shot geometric projections with corridor relaxation, achieving 100% success. These results demonstrate that explicit geometric feasibility provides greater robustness and sample efficiency than learned soft-constraint policies in constrained multi-agent environments.

7 RELATED WORK

Diffusion for planning and control. The paper Diffuser [11] adapts denoising diffusion probabilistic models (DDPMs) [1] for goal-conditioned trajectory synthesis via iterative denoising. Extensions explore task-conditioned guidance using classifier or energy-based methods [5].

Safe diffusion. SafeDiffuser [20] integrates control barrier functions (CBFs) into diffusion steps for *single-agent* safety under robust or time-varying conditions. We extend this to *multi-agent* systems, supporting self, pairwise, and task-level safety with decentralized enforcement and execution-time filtering.

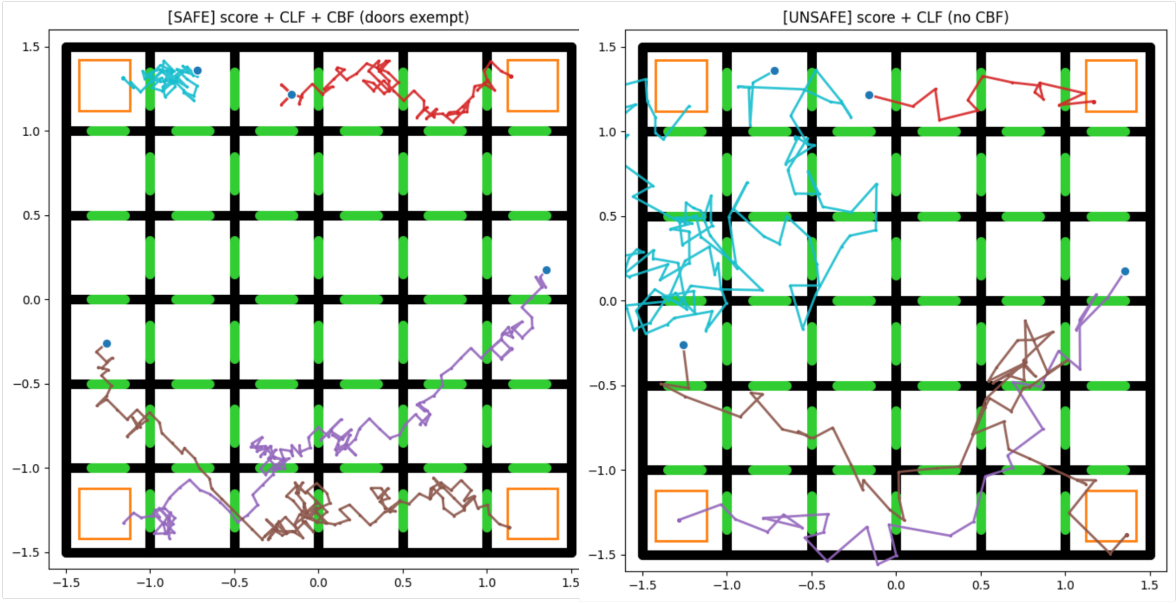

Figure 1: Qualitative rollouts of the learned-score variant of MA-SafeDiffuser with an MLP denoiser (4 agents, planning horizon H=28, diffusion steps N=40, and T=2000 sampling steps; trained on 1500 trajectories for 1000 epochs). (Left) Safe planner with score-based generation combined with CLF (Control Lyapunov Function) and CBF (Control Barrier Function) projection, yielding V_{safe}=62. (Right) Unsafe planner without CBF enforcement, yielding V_{unsafe}=125. The safe variant produces smooth, wall-aware trajectories and avoids collisions, whereas the unsafe variant exhibits frequent constraint violations and unstable motion near obstacles.

Table 1: Representative three-seed outcomes (Maze2D). Reported rewards are total episode returns over $T = 700$ steps.

Setting	Agents (M)	R_{Safe}	R_{Unsafe}	V_{Safe}	V_{Unsafe}	Success
Centralized	2	33.76	33.98	9	23.66	True
	4	66.62	66.73	11.33	42.66	True
	8	133.04	133.16	40	87.33	True
Decentralized	2	33.74	33.80	16	22.66	True
	4	66.70	66.71	45	73.66	True
	8	135.29	135.35	70.66	88.66	True

Table 2: Learned MLP denoiser performance on Maze2D (4 agents, $H = 28$, $N = 40$, $T = 2000$).

Mode	Reward R	Violations V	Success S
Safe	**66.85**	**62**	True
Unsafe (no CBF)	67.00	125	True

Table 3: Comparative performance with MAPPO-Lag on the Maze2D environment (M=4 agents). Results are averaged over 3 random seeds with a T=700 step horizon.

Method	Success Rate	Reward	Violations
MAPPO-Lag	0%	-0.82	532.3
Centralized	**100%**	**66.62**	**11.33**
Decentralized	**100%**	**66.70**	45.00

Control barrier functions and invariance. CBFs offer an careful way to certify forward invariance by setting $\nabla b(x)^\top u + alpha(b(x)) \geq 0$ [1], with extensions for higher relative degree through exponential CBF [16] and nonsmooth sets [9]. We leverage these tools for waypoint-aware safety, establishing discrete-time invariance under diffusion updates.

Multi–agent safety and MARL. Traditional reactive methods (potential fields, velocity obstacles, MPC+CBF) and MARL baselines (MADDPG [14], MAPPO [21]) lack strict safety guarantees. Our

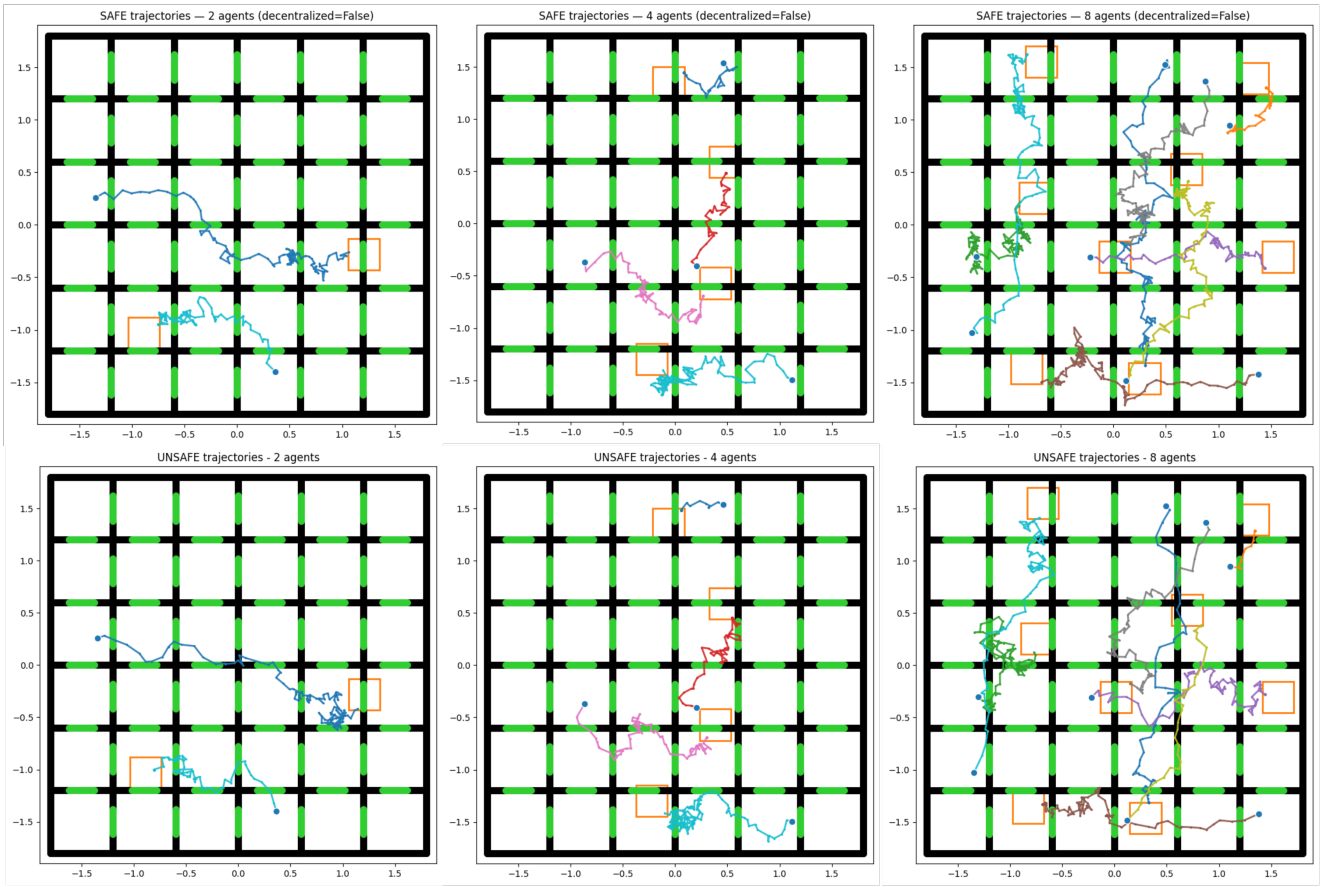

Figure 2: Qualitative results for centralized (decentralized=False) MA-SafeDiffuser. Each panel compares safe and unsafe trajectories across randomized initializations for M=2, 4, 8 agents. For M=2, V_{safe}=8 and V_{unsafe}=25; for M=4, V_{safe}=10 and V_{unsafe}=21; and for M=8, V_{safe}=29 and V_{unsafe}=96. SafeDiffuser agents maintain clearance through doors and corners, whereas unsafe diffusion plans exhibit oscillatory behavior and occasional wall penetration. Start positions (blue dots) and goals (orange boxes) are placed asymmetrically and randomized across runs to emulate a warehouse-like navigation scenario. While safe agents may incur occasional violations due to proximity constraints, their violation rates remain substantially lower than those of unsafe agents.

CBF-with-diffusion projector enforces hard constraints during sampling and execution.

Optimization layers and projections. Differentiable QP layers (OptNet [2], CVXPY-Layers [6]) and operator-splitting solvers (OSQP [17]) enable scalable end-to-end safety. Projection methods (Dykstra [7]; see [3]) provide a parameter-free alternative for linearized constraints.

Benchmarking frameworks. PettingZoo [18] standardizes multi-agent benchmarks (e.g., MPE). We complement this with a light-weight Maze2D testbed for geometry-constrained, safety-aware evaluation.

Our approach MA-SafeDiffuser combines the generative expressiveness of diffusion planning [10, 11] with the formal guaranteed invariance of CBFs [1, 9, 16], SafeDiffuser's single-agent theory [20] is extended to centralized and decentralized multi-agent joint safety and allows employing the latest projection/QP layers [2, 3, 7, 17] for realistic enforcement.

8 CONCLUSION AND LIMITATIONS

We introduced MA-SafeDiffuser, a multi-agent safe diffusion planner with centralized and decentralized enforcement and finite-time diffusion invariance guarantees. Empirically, on Maze2D we see staggering violations reduction over unconstrained denoising. Sub-limitations are: (i) QP feasibility with dense crowds (relaxed by ReS/TVS and introduced reverse steps), (ii) decentralized margins with the need for bounded comms errors, and (iii) shaping sensitivity of absolute returns on very long horizons. We believe that future work may involve scaling our method to larger number of agents and more complex planning tasks.

ACKNOWLEDGEMENTS

This research was financially supported by the University Grants Commission (UGC), Government of India, through the award of the Junior/Senior Research Fellowship (JRF/SRF) under the National Eligibility Test (NET) (Ref. No. 221610037786 & Ref. no. 221610070470). The work was also supported by the Council of Scientific and Industrial Research (CSIR), Government of India, through a CSIR Research Fellowship (09/0917(17235)/2023-EMR-I).

REFERENCES

[1] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. 2017. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Automat. Control* 62, 8 (2017), 3861–3876.

[2] Brandon Amos and J. Zico Kolter. 2017. OptNet: differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, Sydney, Australia, 136–145.

[3] Heinz H. Bauschke and Patrick L. Combettes. 2017. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (2nd ed.). Springer, Cham.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

[5] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat GANs on image synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21)*. Curran Associates Inc., Red Hook, NY, USA, Article 672, 15 pages.

[6] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.

[7] Richard L Dykstra. 1983. An Algorithm for Restricted Least Squares Regression. *J. Amer. Statist. Assoc.* 78, 384 (1983), 837–842.

[8] Paolo Fiorini and Zvi Shiller. 1998. Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research* 17, 7 (1998), 760–772.

[9] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. 2017. Nonsmooth Barrier Functions With Applications to Multi-Robot Systems. *IEEE Control Systems Letters* 1, 2 (2017), 310–315.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada). Curran Associates Inc., Red Hook, NY, USA, 6840–6851.

[11] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. 2022. Planning with Diffusion for Flexible Behavior Synthesis. arXiv:2205.09991 [cs.LG] https://arxiv.org/abs/2205.09991

[12] Oussama Khatib. 1986. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research* 5, 1 (1986), 90–98.

[13] Ankita Kushwaha, Kiran Ravish, Preeti Lamba, and Pawan Kumar. 2025. A Survey of Safe Reinforcement Learning and Constrained MDPs: A Technical Survey on Single-Agent and Multi-Agent Safety. arXiv:2505.17342 [cs.LG] https://arxiv.org/abs/2505.17342

[14] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.

[15] Mitio Nagumo. 1942. Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan, Third Series* 24 (1942), 551–559.

[16] Quan Nguyen and Koushil Sreenath. 2016. Exponential Control Barrier Functions for Enforcing High Relative-Degree Safety-Critical Constraints. In *Proceedings of the 2016 American Control Conference (ACC)*. IEEE, Boston, MA, USA, 322–328.

[17] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. 2020. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation* 12, 4 (2020), 637–672.

[18] Justin K Terry, Benjamin Black, Mario Jayakumar, Ananth Hari, Luis Santos, Clemens Dieffendahl, Niall L Williams, Yashas Lokesh, Ryan Sullivan, Caroline Horsch, and Praveen Ravi. 2021. PettingZoo: Gym for Multi-Agent Reinforcement Learning. https://openreview.net/forum?id=WoLQsYU8aZ

[19] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. 2011. Reciprocal n-Body Collision Avoidance. In *Robotics Research*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3–19.

[20] Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, and Daniela Rus. 2023. SafeDiffuser: Safe Planning with Diffusion Probabilistic Models. arXiv:2306.00148 [cs.LG] https://arxiv.org/abs/2306.00148

[21] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. arXiv:2103.01955 [cs.LG] https://arxiv.org/abs/2103.01955