MULTI-AGENT FLOW MATCHING WITH DECOUPLED GENERATIVE GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative modeling is a widely used framework for producing diverse objects from complex, multimodal distributions. However, its expressivity does not, in general, come with formal guarantees that the generated objects satisfy hard constraints or requirements. In multi-agent generation, this problem becomes more challenging because a hard requirement can depend on multiple agents, while each agent may need to determine its own guidance input without relying on the simultaneously computed guidance inputs of other agents. To this end, we introduce DeGG-Flow, a general framework for multi-agent flow matching with decoupled generative guidance. By representing the generative process as a control-affine dynamical system, we develop guidance conditions for two classes of coupled requirements: shared requirements whose satisfaction depends on multiple agents together, and private requirements associated with each individual agent dependent on its neighbors. For both classes, we establish feasibility conditions and finite-horizon convergence guarantees. We further derive a Wasserstein bound that characterizes the distributional deviation induced by the guidance. We demonstrate DeGG-Flow on multi-robot collaboration for crossing a spatial gap by reconfiguring the environment, and on multi-object scene generation with affordance requirements. Across both applications, DeGG-Flow directly generates objects that satisfy all corresponding hard requirements, including at team sizes unseen during training.

1 INTRODUCTION

Generative models have achieved remarkable success across a wide range of domains, from language and vision to robotics and scientific discovery (Brown et al., 2020; Rombach et al., 2022; Chi et al., 2025; Lipman et al., 2022; Albergo et al., 2025; Janner et al., 2022; Tang et al., 2024). A core strength of these models is their ability to characterize complex, multimodal distributions and generate diverse objects from them. However, such expressivity, in general, does not imply that every generated object satisfies required hard constraints or requirements. For example, a generated motion may violate safety requirements, a generated molecular structure may violate physical or chemical constraints, and new requirements introduced after training may not be represented in the training data. This gap is especially important when a learned generative model is already useful but new hard requirements arise after training. In such cases, it is desirable to impose constraints at generation time without collecting new data or retraining the model.

Guidance in the generative process provides a natural mechanism for enforcing constraints (Bansal et al., 2024; Feng et al., 2025). Recent work has made this idea increasingly principled by incorporating control barrier functions (CBF) (Ames et al., 2016) into generative process. SafeDiffuser (Xiao et al., 2025) introduces CBF into the denoising process of diffusion probabilistic model (Ho et al., 2020); SafeFlow (Dai et al., 2025) targets robot motion planning by imposing CBF constraints at the waypoints of a generated trajectory; Gadginmath et al. (2026) formulate constrained sampling through a stochastic differential equation and implement discrete-time CBF through a first-order approximation; and SafeFlowMatcher (Yang et al., 2026) addresses robotic path planning by first generating a candidate path and then applying a separate CBF-based correction.

Multi-agent generative modeling changes the structure of such guidance, as hard constraints or requirements may depend jointly on multiple agents. A direct approach is to treat all agents as a single-agent system and compute the guidance together. However, a key advantage of multi-agent

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

systems is that the team need not depend on a single central decision maker, whose failure could affect all agents at once. This motivates decentralized or agent-wise guidance, where each agent determines its own guidance without relying on any other agent's guidance. Existing multi-agent generative models address different aspects of this problem. MADiff (Zhu et al., 2024) models multimodal multi-agent behavior and supports decentralized execution; Shaoul et al. (2025) combine diffusion models with search-based planning to generate collision-free multi-robot trajectories; Parimi & Williams (2025) decompose multi-arm planning into single-arm trajectory generation and pairwise collision resolution; Liang et al. (2025) integrate constrained projection into the denoising process (Ho et al., 2020) to satisfy collision avoidance and kinematic constraints; and MAC-Flow (Lee et al., 2026) learns joint behaviors with flow matching and distills them into decentralized one-step policies. These works provide different mechanisms for generating, coordinating, and adapting multi-agent behavior. What remains unresolved is how to guarantee a hard requirement that depends on multiple agents when each agent determines only its own guidance.

The constrained generative methods discussed earlier do not directly resolve this problem either. Applied to the full multi-agent state, they naturally lead to a joint guidance problem. The challenge is thus to formulate a guidance rule for each agent that it can satisfy, while ensuring that these agent-level rules together provide a provable guarantee on the team-level coupled requirement.

To this end, we introduce **DeGG-Flow**, a general framework for multi-agent flow matching with decoupled generative guidance. Its central idea is to preserve agent-wise guidance while providing formal guarantees for requirements that can couple multiple agents. DeGG-Flow handles two classes of requirements: shared requirements for which multiple agents collaboratively contribute to their satisfaction, and private requirements associated with each individual agent whose satisfaction depends on its neighbors. The guidance is applied during the generative process, without collecting new training data, retraining the learned generative model, or repairing the generated objects afterward. The distributional deviation induced by the guidance is further shown to be theoretically bounded rather than uncharacterized. Our main contributions are summarized as follows.

- **Agent-wise guidance with team-level requirement.** We formulate the generative process of multi-agent flow matching as a control-affine dynamical system and adapt disentangled control (Lin et al., 2026) to decoupled guidance, so that each agent determines only its own guidance input without relying on the simultaneously computed guidance input of any other agent, while retaining formal guarantees for requirements that couple multiple agents.
- **Feasibility and finite-horizon convergence guarantees.** For shared requirements, we introduce an adaptive constraint allocation approach to ensure feasibility of the corresponding decoupled guidance, and propose a time-varying bound that guarantees the generated object reaches a desired set in the generative state space within the finite horizon. For private requirements, we also establish feasibility conditions for the corresponding decoupled guidance and propose an approach to guarantee finite-horizon convergence.
- **Bounded distributional deviation under guidance.** We derive a Wasserstein bound that characterizes how far the guided joint generative distribution can deviate from the unguided one, and empirically evaluate the resulting distributional deviation in the experiments.
- **Applications with shared and private requirements.** We demonstrate DeGG-Flow on multi-robot collaboration with shared coupled requirements for crossing a spatial gap by reconfiguring the environment, and on multi-object scene generation with private coupled requirements, where each object's affordance requirements, i.e., its access and use requirements, depend on neighboring objects. In both tasks, DeGG-Flow directly generates objects that satisfy all hard requirements without post-generation correction, including for numbers of agents beyond those used during training.

2 Multi-Agent Flow Matching Guided by Disentangled Control

2.1 Multi-agent flow matching

Consider a team of $N \in \mathbb{Z}^+$ agents, indexed by $i \in \mathcal{N} \coloneqq \{1, \dots, N\}$, whose joint generative process produces an output $y \in \mathcal{Y}_N$ for a given task, where $\mathcal{Y}_N$ denotes the corresponding output space. Let $\chi \in \mathcal{C}_N$ denote the generation condition, where $\mathcal{C}_N$ is the corresponding condition space.

The condition contains information such as initial states, agent or object properties, environment information, or task specifications. The output y depends on specific applications. For example, in Section 4.1, y represents a motion policy for N robots, and in Section 4.2, y represents an arrangement of N objects. Instead of generating y directly, each agent i has a generative state $z_i(\tau) \in \mathcal{Z} \subseteq \mathbb{R}^{d_z}$ at generative time $\tau \in [0,1]$, which is different from physical time $t \in \mathbb{R}_{\geq 0}$. Denote the joint generative state as $\mathbf{z}(\tau) \coloneqq [z_1(\tau)^\top, \dots, z_N(\tau)^\top]^\top \in \mathcal{Z}^N \subseteq \mathbb{R}^{Nd_z}$. The task-specific decoder and encoder map between the joint generative state space and the output space are denoted by

$$\mathcal{D}_N : \mathcal{Z}^N \times \mathcal{C}_N \to \mathcal{Y}_N, \quad y = \mathcal{D}_N(\mathbf{z} \,|\, \chi), \tag{1}$$

$$\mathcal{E}_N : \mathcal{Y}_N \times \mathcal{C}_N \to \mathcal{Z}^N, \quad \mathbf{z} = \mathcal{E}_N(y \,|\, \chi). \tag{2}$$

For a training data y, $\mathcal{E}_N(y \,|\, \chi)$ gives its target generative state for flow matching, and $\mathcal{D}_N(\mathbf{z}(1) \,|\, \chi)$ represents the final generated object.

Given N and χ, let $p_{\text{data}}(\mathbf{z} \,|\, \chi)$ denote the joint conditional distribution of the encoded training data in $\mathcal{Z}^N$, and let $p_0(\mathbf{x})$ denote a simple initial distribution in $\mathcal{Z}^N$ from which samples can be drawn efficiently. Generation starts from $\mathbf{z}(0) \sim p_0$ and evolves over $\tau \in [0,1]$ governed by the dynamical system $\dot{\mathbf{z}} = f(\tau, \mathbf{z} \,|\, \chi)$. The objective of flow matching (Lipman et al., 2022) is to learn a vector field whose flow transports p_0 to $p_{\text{data}}(\mathbf{z} \,|\, \chi)$. To learn such a vector field, for each target $\mathbf{z} \in \mathcal{Z}^N$, consider a conditional probability path $p(\tau, \mathbf{x} \,|\, \mathbf{z}, \chi)$ satisfying $p(0, \mathbf{x} \,|\, \mathbf{z}, \chi) = p_0(\mathbf{x})$ and $p(1, \mathbf{x} \,|\, \mathbf{z}, \chi) = \delta(\mathbf{x} - \mathbf{z})$, where $\mathbf{x} \in \mathcal{Z}^N$ is a realization of the intermediate random variable $\mathbf{X}(\tau)$ and $\delta(\mathbf{x} - \mathbf{z})$ is the Dirac delta concentrated at $\mathbf{z}$. Let $v(\tau, \mathbf{x} \,|\, \mathbf{z}, \chi) \in \mathbb{R}^{Nd_z}$ denote a conditional time-varying vector field that generates this path, then the marginal probability path is $p(\tau, \mathbf{x} \,|\, \chi) = \int_{\mathcal{Z}^N} p(\tau, \mathbf{x} \,|\, \mathbf{z}, \chi) p_{\text{data}}(\mathbf{z} \,|\, \chi) \, \mathrm{d}\mathbf{z}$, and the corresponding marginal vector field is

$$f(\tau, \mathbf{x} \,|\, \chi) \coloneqq \int_{\mathcal{Z}^N} v(\tau, \mathbf{x} \,|\, \mathbf{z}, \chi) \, \frac{p(\tau, \mathbf{x} \,|\, \mathbf{z}, \chi) p_{\text{data}}(\mathbf{z} \,|\, \chi)}{p(\tau, \mathbf{x} \,|\, \chi)} \, \mathrm{d}\mathbf{z}. \tag{3}$$

Since (3) depends on the unknown data distribution, in this paper, (3) is represented by a message passing graph neural network (GNN) (Gilmer et al., 2017)

$$f^\theta : [0,1] \times \mathcal{Z}^N \times \mathcal{C}_N \to \mathbb{R}^{Nd_z}, \tag{4}$$

which we refer to as the **nominal vector field**. The same neural network parameters θ are shared across all agents. Each agent computes its output from its own state and messages aggregated from neighboring agents so that the GNN (4) is permutation equivariant. Denote $f^\theta = [(f_1^\theta)^\top, \dots, (f_N^\theta)^\top]^\top$, where $f_i^\theta \in \mathbb{R}^{d_z}$ is the nominal vector field of agent i, which is trained using the multi-agent conditional flow matching loss

$$\mathcal{L}_{\text{MAC}}(\theta \,|\, \chi) \coloneqq \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \left\| f_i^\theta(\tau, \mathbf{X}(\tau) \,|\, \chi) - v_i(\tau, \mathbf{X}(\tau) \,|\, \mathbf{Z}, \chi) \right\|^2 \right], \tag{5}$$

in which the expectation is taken over $\tau \sim \text{Unif}[0,1]$, the encoded data random variable $\mathbf{Z} \sim p_{\text{data}}(\mathbf{z} \,|\, \chi)$, and the intermediate state random variable $\mathbf{X}(\tau) \sim p(\tau, \mathbf{x} \,|\, \mathbf{Z}, \chi)$ along the conditional probability path. After training, f^θ is the nominal vector field governing the generative dynamics. We next introduce a guidance mechanism modifying such dynamics.

2.2 Disentangled control guidance

The nominal vector field f^θ transports samples from a simple initial distribution toward a complex, multimodal distribution that approximates the encoded data distribution. However, the resulting generated objects are not, in general, guaranteed to satisfy hard constraints or requirements. We therefore add a guidance input to the dynamics of each agent in the generative process. Let $u_i(\tau) \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ be the guidance input of agent i, and $g_i : [0,1] \times \mathcal{Z}^N \times \mathcal{C}_N \to \mathbb{R}^{d_z \times m_i}$ be the map determining how the guidance input affects the generative dynamics. Then, the guided generative dynamics are in the control-affine form

$$\dot{z}_i = f_i^\theta(\tau, \mathbf{z} \,|\, \chi) + g_i(\tau, \mathbf{z} \,|\, \chi) u_i, \quad \forall i \in \mathcal{N}. \tag{6}$$

We encode the hard requirement by a C^1 (i.e., continuously differentiable) function, and consider two classes of decoupled guidance by adapting the **disentangled control** (Lin et al., 2026) from multi-agent control in physical space into the generative process as a form of decoupled guidance, i.e., guidance for shared-entangled requirements (SE guidance) and guidance for private-entangled requirements (PE guidance), to enable agent i to compute u_i without access to the simultaneously computed guidance input $u_j, \forall j \in \mathcal{N} \setminus \{i\}$.

SE guidance. Let $\mathcal{G}_{\mathrm{F}}^{\mathrm{SE}} = (\mathcal{N}, \mathcal{A}^{\mathrm{SE}}, \mathcal{I}^{\mathrm{SE}})$ be a factor graph (Kschischang et al., 2001), where $\mathcal{A}^{\mathrm{SE}}$ is a finite set of factors and $\mathcal{I}^{\mathrm{SE}} \subseteq \mathcal{N} \times \mathcal{A}^{\mathrm{SE}}$ is the incidence relation. For each factor $a \in \mathcal{A}^{\mathrm{SE}}$, define the incident agent set as $\mathcal{S}_a^{\mathrm{SE}} \coloneqq \{i \in \mathcal{N} : (i,a) \in \mathcal{I}^{\mathrm{SE}}\} \neq \varnothing$, and let $\mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}$ denote the corresponding subvector of the joint generative state. Let $V_a^{\mathrm{SE}} : [0,1] \times \prod_{i \in \mathcal{S}_a^{\mathrm{SE}}} \mathcal{Z} \times \mathcal{C}_N \to \mathbb{R}$ be C^1. The guidance input u_i of agent $i \in \mathcal{S}_a^{\mathrm{SE}}$ is such that

$$\nabla_{z_i} V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)^\top \left(f_i^\theta(\tau, \mathbf{z} \,|\, \chi) + g_i(\tau, \mathbf{z} \,|\, \chi) u_i \right) + \zeta_{a,i}^{\mathrm{SE}}(\tau, \mathbf{z} \,|\, \chi) \leq 0, \tag{7}$$

in which

$$\sum_{i \in \mathcal{S}_a^{\mathrm{SE}}} \zeta_{a,i}^{\mathrm{SE}}(\tau, \mathbf{z} \,|\, \chi) = \partial_\tau V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi) + \alpha_a^{\mathrm{SE}}\big(V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)\big), \tag{8}$$

where $\alpha_a^{\mathrm{SE}} : \mathbb{R} \to \mathbb{R}$ is an extended class $\mathcal{K}_\infty$ function.

PE guidance. For each agent $i \in \mathcal{N}$, let $V_i^{\mathrm{PE}} : [0,1] \times \mathcal{Z}^N \times \mathcal{C}_N \to \mathbb{R}_{\geq 0}$ be C^1. The guidance input u_i of agent i is such that

$$\nabla_{z_i} V^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi)^\top \left(f_i^\theta(\tau, \mathbf{z} \,|\, \chi) + g_i(\tau, \mathbf{z} \,|\, \chi) u_i \right) + \zeta_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi) \leq 0, \quad \forall i \in \mathcal{N}, \tag{9}$$

in which $V^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi) \coloneqq \sum_{i \in \mathcal{N}} V_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi)$, and

$$\zeta_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi) \coloneqq \partial_\tau V_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi) + \omega^{\mathrm{PE}}(\tau)\, \alpha^{\mathrm{PE}}\big(V_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi)\big), \tag{10}$$

where $\omega^{\mathrm{PE}} : [0,1] \to \mathbb{R}_{\geq 0}$, and $\alpha^{\mathrm{PE}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a subadditive class $\mathcal{K}_\infty$ function.

Intuitively, SE guidance is suitable for scenarios where multiple agents have certain shared goals or requirements, such as the task of multi-robot collaborative policy generation in Section 4.1, while PE guidance is suitable for scenarios where each agent has its own requirement, which can be affected by its neighbors, such as the task of multi-object scene generation in Section 4.2.

3 Adaptive Guidance and Formal Guarantees

Section 2 introduces two classes of decoupled guidance for multi-agent generation, under which each agent computes its own guidance input without relying on the simultaneously computed guidance inputs of other agents. Since the generative process evolves over the finite horizon $\tau \in [0,1]$, it is necessary to ensure that the final generated object directly satisfies the desired hard requirements. In this section, we propose separate finite-horizon convergence methods for SE and PE guidance. We further characterize the distributional deviation induced by guidance through a Wasserstein bound on the guided and nominal joint generative distributions.

3.1 SE guidance with finite-horizon convergence

Let the C^1 function $q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)$ represent a shared requirement, where $q_a^{\mathrm{SE}} \leq 0$ means that the requirement is satisfied. An initial sample $\mathbf{z}(0) \sim p_0$ need not satisfy this requirement. Thus, we introduce a time-varying upper bound $\beta_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0))$ that is initialized from $\mathbf{z}(0)$. The bound is chosen to satisfy

$$\beta_a^{\mathrm{SE}}(0 \,|\, \chi, \mathbf{z}(0)) \geq q_a^{\mathrm{SE}}(0, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(0) \,|\, \chi), \tag{11}$$

and is tightened toward a prescribed terminal value. In particular, choosing $\beta_a^{\mathrm{SE}}(1 \,|\, \chi, \mathbf{z}(0)) = 0$ ensures that the desired hard requirement is satisfied at the end of the generative process.

Accordingly, define

$$V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi) \coloneqq q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi) - \beta_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0)), \tag{12}$$

so that $V_a^{\mathrm{SE}} \leq 0$ is equivalent to $q_a^{\mathrm{SE}} \leq \beta_a^{\mathrm{SE}}$. Since factor a involves multiple agents, $\partial_\tau V_a^{\mathrm{SE}} + \alpha_a^{\mathrm{SE}}(V_a^{\mathrm{SE}})$ can be allocated among the involved agents using $w_{a,i}^{\mathrm{SE}} \geq 0$ with $\sum_{i \in \mathcal{S}_a^{\mathrm{SE}}} w_{a,i}^{\mathrm{SE}} = 1$, i.e.,

$$\zeta_{a,i}^{\mathrm{SE}}(\tau, \mathbf{z} \,|\, \chi) \coloneqq w_{a,i}^{\mathrm{SE}} \Big(\partial_\tau q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi) - \dot{\beta}_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0)) + \alpha_a^{\mathrm{SE}}\big(V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)\big) \Big). \tag{13}$$

Since $\partial_\tau V_a^{\mathrm{SE}} = \partial_\tau q_a^{\mathrm{SE}} - \dot{\beta}_a^{\mathrm{SE}}$, this construction satisfies (8). The following theorem shows that satisfying the agent-wise constraint of SE guidance keeps the shared requirement below its prescribed bound throughout the generative process.

Theorem 3.1 (Finite-horizon convergence for SE guidance)**.** *Let q_a^{SE} and β_a^{SE} be C^1. Suppose* (11) *holds, and let V_a^{SE} and $\zeta_{a,i}^{\mathrm{SE}}$ be defined by* (12) *and* (13)*, with $w_{a,i}^{\mathrm{SE}} \geq 0$ and $\sum_{i\in\mathcal{S}_a^{\mathrm{SE}}} w_{a,i}^{\mathrm{SE}} = 1$. If every agent $i \in \mathcal{S}_a^{\mathrm{SE}}$, along a trajectory of* (6)*, satisfies* (7)*, $\forall \tau \in [0,1]$, then*

$$q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) \leq \beta_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0)), \quad \forall \tau \in [0,1].$$

In particular, if $\beta_a^{\mathrm{SE}}(1 \,|\, \chi, \mathbf{z}(0)) = 0$, then the final generated state satisfies the shared requirement, i.e., $q_a^{\mathrm{SE}}(1, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(1) \,|\, \chi) \leq 0$.

Proof. See Appendix A.1. □

To ensure that the constraint (7) is feasible for each agent, we adapt the coefficients $w_{a,i}^{\mathrm{SE}}$ according to how each agent can affect the shared requirements. Define $\ell_{a,i}^{\mathrm{SE}} \coloneqq g_i(\tau, \mathbf{z} \,|\, \chi)^\top \nabla_{z_i} q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)$, which determines how the guidance input u_i affects the shared requirement q_a^{SE}. Based on (13), the constraint (7) can be written as $(\ell_{a,i}^{\mathrm{SE}})^\top u_i + b_{a,i}^{\mathrm{SE}} \leq 0$ with $b_{a,i}^{\mathrm{SE}} \coloneqq \nabla_{z_i} q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)^\top f_i^\theta(\tau, \mathbf{z} \,|\, \chi) + \zeta_{a,i}^{\mathrm{SE}}(\tau, \mathbf{z} \,|\, \chi)$. When $b_{a,i}^{\mathrm{SE}} \leq 0$, $u_i = 0$ already satisfies this inequality. When $b_{a,i}^{\mathrm{SE}} > 0$, agent i must apply a guidance input that reduces the corresponding requirement. When two SE factors are simultaneously active at agent i, their local guidance conditions may compete for the same input u_i. To address this issue, let $a = \{1,2\}$ and $\hat{\ell}_{a,i}^{\mathrm{SE}} \coloneqq \ell_{a,i}^{\mathrm{SE}} / \|\ell_{a,i}^{\mathrm{SE}}\|$ if $\ell_{a,i}^{\mathrm{SE}} \neq 0$ and $\hat{\ell}_{a,i}^{\mathrm{SE}} \coloneqq 0$ otherwise. Then, $\eta_i^{\mathrm{SE}} \coloneqq (\hat{\ell}_{1,i}^{\mathrm{SE}})^\top \hat{\ell}_{2,i}^{\mathrm{SE}}$ characterizes how aligned the effects of u_i on the two active requirements is. For $\delta \in (0,1)$, let $\phi_\delta : [-1,1] \to [-1,0]$ be a C^2 function satisfying $\phi_\delta(\eta) = \eta$ for $\eta \leq -\delta$, $\phi_\delta(\eta) = 0$ for $\eta \geq \delta$, and $\phi_\delta(\eta) \leq \min\{\eta, 0\}$ for $\eta \in [-1,1]$. Then, we define the effective directions $d_{1,i}^{\mathrm{SE}} \coloneqq \ell_{1,i}^{\mathrm{SE}} - \|\ell_{1,i}^{\mathrm{SE}}\| \phi_\delta(\eta_i^{\mathrm{SE}}) \hat{\ell}_{2,i}^{\mathrm{SE}}$ and $d_{2,i}^{\mathrm{SE}} \coloneqq \ell_{2,i}^{\mathrm{SE}} - \|\ell_{2,i}^{\mathrm{SE}}\| \phi_\delta(\eta_i^{\mathrm{SE}}) \hat{\ell}_{1,i}^{\mathrm{SE}}$. When only one SE factor is active at agent i, we set $d_{a,i}^{\mathrm{SE}} \coloneqq \ell_{a,i}^{\mathrm{SE}}$. The modification is inactive when the two effective directions are sufficiently aligned and increases as they become opposed. As such, for each active SE factor, we choose the adaptive constraint allocation coefficient in (13) as

$$w_{a,i}^{\mathrm{SE}} \coloneqq \frac{\|d_{a,i}^{\mathrm{SE}}\|^2}{\sum_{j\in\mathcal{S}_a^{\mathrm{SE}}} \|d_{a,j}^{\mathrm{SE}}\|^2}. \tag{14}$$

Then, the following proposition establishes the existence of guidance inputs such that (7) is satisfied.

Proposition 3.2 (Adaptive SE guidance)**.** *If each agent is incident to at most two simultaneously active SE factors, and for every active SE factor the denominator of* (14) *is nonzero and $d_{a,i}^{\mathrm{SE}} \neq 0$ when $b_{a,i}^{\mathrm{SE}} > 0$, then, there exists a guidance input $u_i(\tau)$ satisfying* (7)*, $\forall \tau \in [0,1]$. Such a guidance input can be obtained from*

$$\begin{aligned} \min_{u_i \in \mathbb{R}^{m_i}} \quad & u_i^\top H_i u_i \\ \text{s.t.} \quad & (\ell_{a,i}^{\mathrm{SE}})^\top u_i + b_{a,i}^{\mathrm{SE}} \leq 0, \quad \forall a \in \mathcal{A}_i^{\mathrm{SE}}, \end{aligned} \tag{15}$$

which admits a unique minimizer when $H_i \succ 0$, where $\mathcal{A}_i^{\mathrm{SE}} \coloneqq \{a \in \mathcal{A}^{\mathrm{SE}} : i \in \mathcal{S}_a^{\mathrm{SE}}\}$.

Proof. See Appendix A.2. □

3.2 PE GUIDANCE WITH FINITE-HORIZON CONVERGENCE

Next, we propose a method to ensure finite-horizon convergence for PE guidance. Recall that, unlike the real-valued $V_a^{\mathrm{SE}} \in \mathbb{R}$ in the SE guidance scenario, here each agent has a nonnegative $V_i^{\mathrm{PE}} \in \mathbb{R}_{\geq 0}$ representing the violation of its private requirement that can be dependent on its neighbors. Let $\partial_\tau V_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi) = 0$ and choose the total violation to be $V^{\mathrm{PE}}(\mathbf{z} \,|\, \chi) = \sum_{i\in\mathcal{N}} V_i^{\mathrm{PE}}(\mathbf{z} \,|\, \chi)$, so $V^{\mathrm{PE}} = 0$ means that all private requirements are satisfied. We then choose

$$\alpha^{\mathrm{PE}}(s) \coloneqq c^{\mathrm{PE}} s^{\rho^{\mathrm{PE}}}, \tag{16}$$

where $c^{\mathrm{PE}} \in \mathbb{R}_{>0}$ and $\rho^{\mathrm{PE}} \in (0,1)$. The choice of $\rho^{\mathrm{PE}} \in (0,1)$ enables finite-horizon convergence, and $\omega^{\mathrm{PE}}(\tau)$ in (10) modulates the required decrease of V^{PE} over τ, as shown in the following theorem.

Theorem 3.3 (Finite-horizon convergence for PE guidance)**.** *Let α^{PE} be given by* (16)*. If every agent, along a trajectory of* (6)*, satisfies* (9)*, $\forall \tau \in [0,1]$, then*

$$V^{\mathrm{PE}}(\mathbf{z}(\tau)\,|\,\chi) \le \max\left\{0,\ \left(V^{\mathrm{PE}}(\mathbf{z}(0)\,|\,\chi)\right)^{1-\rho^{\mathrm{PE}}} - c^{\mathrm{PE}}(1-\rho^{\mathrm{PE}})\int_0^\tau \omega^{\mathrm{PE}}(s)\,\mathrm{d}s\right\}^{\frac{1}{1-\rho^{\mathrm{PE}}}}.$$

In particular, if $c^{\mathrm{PE}}(1-\rho^{\mathrm{PE}})\int_0^1 \omega^{\mathrm{PE}}(s)\,\mathrm{d}s \ge \left(V^{\mathrm{PE}}(\mathbf{z}(0)\,|\,\chi)\right)^{1-\rho^{\mathrm{PE}}}$, then the final generated state satisfies all private requirements, i.e., $V^{\mathrm{PE}}(\mathbf{z}(1)\,|\,\chi) = 0$.

Proof. See Appendix A.3. □

Unlike SE guidance, PE guidance assigns each agent a single inequality constraint. Define

$$\ell_i^{\mathrm{PE}} \coloneqq g_i(\tau, \mathbf{z}\,|\,\chi)^\top \nabla_{z_i} V^{\mathrm{PE}}(\mathbf{z}\,|\,\chi),$$
$$b_i^{\mathrm{PE}} \coloneqq \nabla_{z_i} V^{\mathrm{PE}}(\mathbf{z}\,|\,\chi)^\top f_i^\theta(\tau, \mathbf{z}\,|\,\chi) + c^{\mathrm{PE}}\omega^{\mathrm{PE}}(\tau)\left(V_i^{\mathrm{PE}}(\mathbf{z}\,|\,\chi)\right)^{\rho^{\mathrm{PE}}}.$$

Then the constraint (9) becomes $(\ell_i^{\mathrm{PE}})^\top u_i + b_i^{\mathrm{PE}} \le 0$. If $b_i^{\mathrm{PE}} \le 0$, the nominal input $u_i = 0$ already satisfies the constraint. If $b_i^{\mathrm{PE}} > 0$, a feasible guidance input exists when $\ell_i^{\mathrm{PE}} \ne 0$. This gives the following feasibility result.

Corollary 3.4 (Feasibility of PE guidance)**.** *Per Theorem 3.3, if $b_i^{\mathrm{PE}} \le 0$, or if $b_i^{\mathrm{PE}} > 0$ and $\ell_i^{\mathrm{PE}} \ne 0$, then there exists a guidance input $u_i(\tau)$ satisfying* (9)*, $\forall \tau \in [0,1]$. Such a guidance input can be obtained from*

$$\begin{aligned} \min_{u_i \in \mathbb{R}^{m_i}} \quad & u_i^\top H_i u_i \\ \text{s.t.} \quad & (\ell_i^{\mathrm{PE}})^\top u_i + b_i^{\mathrm{PE}} \le 0, \end{aligned} \tag{17}$$

which admits a unique minimizer when $H_i \succ 0$.

Proof. See Appendix A.4. □

3.3 Distributional deviation

For generated objects to satisfy certain hard constraints or reach a desired set in the generative space, the nominal vector field is modified by SE or PE guidance. We then establish a Wasserstein bound to characterize the resulting deviation from the nominal joint generative distribution. The bound relates this deviation to the magnitude of the guidance correction, providing a theoretical characterization of such intervention. Specifically, consider nominal and guided joint generative state trajectories initialized from the same random sample, i.e., $\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{X}^{\mathrm{nom}}(\tau) = f^\theta(\tau, \mathbf{X}^{\mathrm{nom}}(\tau)\,|\,\chi)$ and $\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{X}^{\mathrm{gui}}(\tau) = f^\theta(\tau, \mathbf{X}^{\mathrm{gui}}(\tau)\,|\,\chi) + \Gamma(\tau\,|\,\xi)$, with $\mathbf{X}^{\mathrm{nom}}(0) = \mathbf{X}^{\mathrm{gui}}(0) = \xi \sim p_0$, where the joint guidance correction is denoted by $\Gamma(\tau\,|\,\xi) \coloneqq \left[(g_1(\tau, \mathbf{X}^{\mathrm{gui}}(\tau)\,|\,\chi)u_1(\tau))^\top, \ldots, (g_N(\tau, \mathbf{X}^{\mathrm{gui}}(\tau)\,|\,\chi)u_N(\tau))^\top\right]^\top$. Let μ_1 and ν_1 denote the probability distributions of $\mathbf{X}^{\mathrm{nom}}(1)$ and $\mathbf{X}^{\mathrm{gui}}(1)$, respectively.

Theorem 3.5 (Wasserstein bound on guidance-induced distributional deviation)**.** *Assume μ_1 and ν_1 have finite second moments and $\left\|f^\theta(\tau, \mathbf{y}\,|\,\chi) - f^\theta(\tau, \mathbf{z}\,|\,\chi)\right\| \le L(\tau)\,\|\mathbf{y} - \mathbf{z}\|$, where $L : [0,1] \to \mathbb{R}_{\ge 0}$ is integrable. If $\int_0^1 \sqrt{\int_{\mathcal{Z}^N} \|\Gamma(s\,|\,\xi)\|^2 p_0(\xi)\,\mathrm{d}\xi}\,\mathrm{d}s < \infty$, where $\Gamma(s\,|\,\xi)$ denotes the guidance correction corresponding to the initial sample $\xi \sim p_0$, then*

$$W_2(\mu_1, \nu_1) \le \int_0^1 \exp\left(\int_s^1 L(r)\,\mathrm{d}r\right)\sqrt{\int_{\mathcal{Z}^N} \|\Gamma(s\,|\,\xi)\|^2 p_0(\xi)\,\mathrm{d}\xi}\,\mathrm{d}s. \tag{18}$$

In addition, given χ, assume the decoder satisfies $\|\mathcal{D}_N(\mathbf{y}\,|\,\chi) - \mathcal{D}_N(\mathbf{z}\,|\,\chi)\| \le L_D\,\|\mathbf{y} - \mathbf{z}\|$ for some $L_D \in \mathbb{R}_{\ge 0}$. Let $\bar{\mu}_1$ and $\bar{\nu}_1$ denote the probability distributions of $\mathcal{D}_N(\mathbf{X}^{\mathrm{nom}}(1)\,|\,\chi)$ and $\mathcal{D}_N(\mathbf{X}^{\mathrm{gui}}(1)\,|\,\chi)$, respectively. Then

$$W_2(\bar{\mu}_1, \bar{\nu}_1) \le L_D W_2(\mu_1, \nu_1). \tag{19}$$

Proof. See Appendix A.5. □

Empirically evaluation of the distributional deviation between the nominal vector field and the guided one is in Appendix C.2.

3.4 ALGORITHM

After establishing the finite-horizon convergence guarantees for SE and PE guidance, Algorithm 1 summarizes the decoupled guidance procedure. The nominal vector field f_i^θ may be provided by a pretrained model, or learned using (5) that can be augmented with task-specific training losses. Such augmentation affects only the learned nominal vector field and is independent of the SE or PE guidance applied during the generative process. Note that the proposed decoupled guidance requires neither retraining nor a particular neural network architecture.

Algorithm 1 Multi-agent flow matching with SE/PE guidance

Require: Condition χ, decoder $\mathcal{D}_N$, and either a pretrained f_i^θ or training data with encoder $\mathcal{E}_N$.
if f_i^θ is unavailable **then**
  $\theta \leftarrow \arg\min_\theta \mathcal{L}_{\mathrm{MAC}}(\theta \mid \chi)$ using (5)
end if
$\mathbf{z}(0) \sim p_0$.
if SE guidance **then**
  Specify q_a^{SE}, β_a^{SE}, α_a^{SE}, and $w_{a,i}^{\mathrm{SE}}$, and construct V_a^{SE} and $\zeta_{a,i}^{\mathrm{SE}}$ using (12)–(13).
  for all $i \in \mathcal{N}$ **do**
    $u_i^*(\tau) \leftarrow$ solution of (15).
  end for
else if PE guidance **then**
  Choose α^{PE} and ω^{PE} according to Theorem 3.3.
  for all $i \in \mathcal{N}$ **do**
    $u_i^*(\tau) \leftarrow$ solution of (17).
  end for
end if
Evolve $\mathbf{z}(\tau)$ over $\tau \in [0, 1]$ according to (6) with $u_i = u_i^*$.
return $\mathcal{D}_N(\mathbf{z}(1) \mid \chi)$.

4 APPLICATIONS

In this section, we apply DeGG-Flow to two tasks with shared and private coupled requirements, respectively. Both tasks are visualized in MuJoCo (Todorov et al., 2012).

4.1 TASK 1: MULTI-ROBOT COLLABORATION USING SE GUIDANCE

Motivated by disaster response and search-and-rescue tasks, we study how multiple robots can collaborate and exploit tools in the environment to accomplish tasks beyond the capability of an individual robot. In Task 1, a team of robots faces a spatial gap that cannot be crossed directly and collaboratively uses a plank to construct a traversable bridge. Since all robots act on the same rigid body, their generated motions must remain mutually compatible while satisfying shared task and safety requirements. Therefore, SE guidance is suitable for Task 1.

The generated object is a motion policy specifying the robot contact point velocities on the plank. The condition χ contains the current robot and plank states and environment information. Only the first few actions of each generated policy are executed before a new policy is generated. Task 1 contains two SE factors: $q_{\mathrm{task}}(\mathbf{z} \mid \chi)$, which encodes task progress at the plank pose reached after the executed actions, and $q_{\mathrm{safe}}(\mathbf{z} \mid \chi)$, which evaluates safety along these actions. The rigid-body representation, decoder, conditioning, and detailed factor construction are provided in Appendix B.

For evaluation, every robot uses the same trained GNN representing the nominal vector field f_i^θ in (6). We refer to the generative process with $u_i = 0$ as *Nominal*, and to the same nominal vector field with SE guidance $g_i u_i^*$, where u_i^* is obtained from (15), as *Guided*. Nominal and Guided use the same initial physical condition, initial sample $\mathbf{z}(0)$, conditioning information χ, and decoder $\mathcal{D}_N$. The nominal vector field is trained with $N \in \{4, 5, 6\}$, and we evaluate 50 trials for each $N \in \{4, 5, 6, 7, 8\}$, with $N = 7, 8$ evaluating generalization to team sizes unseen in the training data. As shown in Table 1, Nominal fails in 32 trials, whereas Guided successfully constructs a

Table 1: Success rates for Task 1

N	Nominal	Guided
4	46/50 (92%)	50/50 (100%)
5	45/50 (90%)	50/50 (100%)
6	43/50 (86%)	50/50 (100%)
7 (unseen)	42/50 (84%)	50/50 (100%)
8 (unseen)	42/50 (84%)	50/50 (100%)
Overall	218/250 (87.2%)	250/250 (100%)

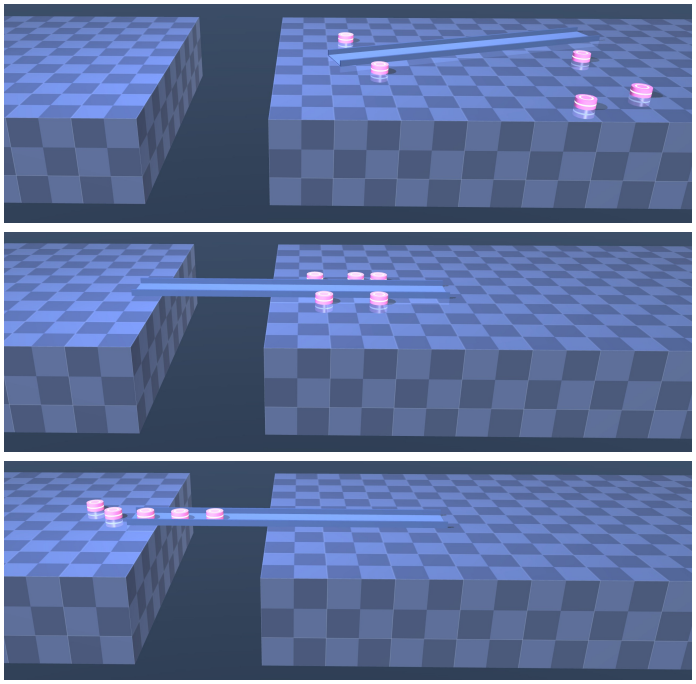

Figure 1: Snapshots of an $N = 5$ trial of Task 1 with SE guidance, showing the initial configuration, successful bridge construction, and spatial gap crossing.

traversable bridge in all 250 trials, including the 100 trials with unseen team sizes of $N = 7, 8$. Figure 1 visualizes an $N = 5$ Guided trial. These results show that SE guidance can recover failures of the nominal generative policy while preserving successful collaboration at team sizes beyond those used during training. Additional results, including the q_{task}–β_{task} and q_{safe}–β_{safe} trajectories, an $N = 7$ rescue example, and multi-seed training and validation curves, are provided in Appendix B.3.

4.2 Task 2: Multi-Object Scene Generation Using PE Guidance

Section 4.1 demonstrates SE guidance in multi-robot collaboration. We next consider PE guidance in multi-object scene generation. A useful scene must satisfy not only geometric validity, such as collision avoidance and workspace containment, but also how individual objects are intended to be accessed or used. Each object therefore has its own affordance requirement, while whether that requirement is satisfied can depend on neighboring objects. Thus, PE guidance is suitable for Task 2.

Task 2 generates a tabletop arrangement of N objects. For object $i \in \mathcal{N}$, denote its pose by $g_{O_i} = (R_i, p_i) \in \mathbb{SE}(2)$, and let $\mathcal{D}_N(\mathbf{z} \mid \chi) = \{g_{O_i}\}_{i \in \mathcal{N}}$. The condition χ specifies the selected objects, their geometry, the tabletop, and their affordance requirements. For object i, let $h_{ik}(\mathbf{z} \mid \chi)$ denote its kth geometric requirement, where $h_{ik} \geq 0$ means that the requirement is satisfied. These requirements include object separation, unobstructed affordance regions, and containment of both objects and affordance regions within the tabletop. Let $r_{ik} = -h_{ik}$ and choose a buffer $\delta^{\text{PE}} > 0$.

Table 2: Success rates for Task 2

N	Natural Nominal	Private Nominal	Guided
4	44/50 (88%)	36/50 (72%)	50/50 (100%)
5	35/50 (70%)	31/50 (62%)	50/50 (100%)
6	32/50 (64%)	26/50 (52%)	50/50 (100%)
7 (unseen)	16/50 (32%)	13/50 (26%)	50/50 (100%)
8 (unseen)	8/50 (16%)	6/50 (12%)	50/50 (100%)
Overall	135/250 (54.0%)	112/250 (44.8%)	250/250 (100%)

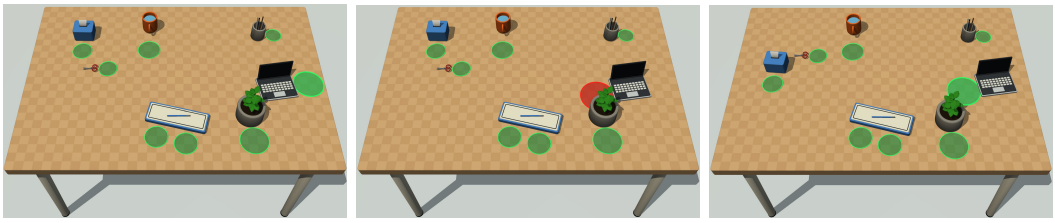

Figure 2: An $N = 7$ example for Task 2. Green circles represent the objects' affordance regions that are unobstructed, while the red circle represents one that is obstructed. Left: the scene generated using the nominal vector field satisfies the default right-handed laptop use requirement. Middle: the same scene evaluated for a left-handed user, where the required mouse operating region on the left side of the laptop is occupied by a potted plant. Right: PE guidance generates a scene satisfying the left-handed use requirement.

We define $V^{\mathrm{PE}} = \sum_{i=1}^{N} V_i^{\mathrm{PE}}$ with $V_i^{\mathrm{PE}}(\mathbf{z} \,|\, \chi) = \sum_k \frac{1}{2} \max\{r_{ik}(\mathbf{z} \,|\, \chi) + \delta^{\mathrm{PE}}, 0\}^2$. The object geometries, affordance regions, requirement parameterization, and other implementation details are provided in Appendix C.

For evaluation, the nominal vector field is trained on scenes with $N \in \{4, 5, 6\}$, and we evaluate 50 cases for each $N \in \{4, 5, 6, 7, 8\}$. *Natural Nominal* evaluates scenes generated by the nominal vector field under the default affordance requirements. *Private Nominal* evaluates the same nominal scenes after the affordance requirements are changed, while *Guided* starts from the same initial samples and incorporates PE guidance based on the changed requirements. As shown in Table 2, the nominal success rate decreases as the number of objects increases, and Private Nominal further shows that a scene valid under the default requirements can become invalid when the desired access or use of an object changes. In contrast, Guided satisfies all requirements in all 250 cases, including the 100 cases with team sizes $N = 7, 8$ unseen in the training data. Figure 2 shows an $N = 7$ example in which PE guidance adapts a nominal scene from a right-handed to a left-handed laptop use requirement without retraining the generative model. These results demonstrate that PE guidance can result in generated objects satisfying private requirements, including at team sizes beyond those used during training. Additional results, including the evolution of V^{PE}, an $N = 8$ example with multiple simultaneous violations, results across different affordance requirements, and multi-seed training and validation curves are provided in Appendix C.2.

5 Conclusion

This paper proposes DeGG-Flow, a general framework for addressing coupled multi-agent requirements through decoupled generative guidance, without applying a separate correction stage after generation. Each agent determines its own guidance without relying on the simultaneously computed guidance of any other agents, so the team need not rely on a central decision maker. We establish feasibility and finite-horizon guarantees for both SE and PE guidance, and demonstrate the effectiveness of DeGG-Flow in multi-robot collaboration and multi-object scene generation, including numbers of agents beyond those used in training. More broadly, DeGG-Flow allows a pretrained generative model to accommodate new hard requirements or constraints that are unseen in its training data, without the need of collecting new data or retraining the model. Future work will investigate discrete-time theoretical guarantees and large-scale multi-agent systems in real world.

AI USE STATEMENT

Generative AI tools were used to polish the writing and to assist in identifying relevant literature. All AI-assisted work was reviewed by the authors, and any references identified with AI assistance were verified against the original sources. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Complete proofs of all theoretical results are provided in Appendix A. Detailed settings for both applications are provided in Appendices B and C. Code will be released upon acceptance.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE transactions on automatic control*, 62(8): 3861–3876, 2016.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024.

Sanjay P Bhat and Dennis S Bernstein. Finite-time stability of continuous autonomous systems. *SIAM Journal on Control and optimization*, 38(3):751–766, 2000.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Xiaobing Dai, Zewen Yang, Dian Yu, Fangzhou Liu, Hamid Sadeghian, Sami Haddadin, and Sandra Hirche. Safeflow: Safe robot motion planning with flow matching via control barrier functions. *arXiv preprint arXiv:2504.08661*, 2025.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. *arXiv preprint arXiv:2502.02150*, 2025.

Darshan Gadginmath, Ahmed Allibhoy, and Fabio Pasqualetti. Provably safe generative sampling with constricting barrier functions. *arXiv preprint arXiv:2602.21429*, 2026.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

Dongsu Lee, Daehee Lee, and Amy Zhang. Multi-agent coordination via flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 99912–99943, 2026.

Jinhao Liang, Jacob K Christopher, Sven Koenig, and Ferdinando Fioretto. Simultaneous multi-robot motion planning with projected diffusion models. *arXiv preprint arXiv:2502.03607*, 2025.

Ruoyu Lin, Gennaro Notomista, and Magnus Egerstedt. Disentangled control of multi-agent systems. *IEEE Transactions on Control of Network Systems*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Viraj Parimi and Brian C Williams. Diffusion-guided multi-arm motion planning. *arXiv preprint arXiv:2509.08160*, 2025.

Frank C Park. Distance metrics on the rigid-body motions with applications to mechanism design. *Journal of Mechanical Design*, 1995.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Yorai Shaoul, Itamar Mishani, Shivam Vats, Jiaoyang Li, and Maxim Likhachev. Multi-robot motion planning with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 95791–95811, 2025.

Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20507–20518. IEEE, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Wei Xiao, Johnson Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations*, volume 2025, pp. 100391–100412, 2025.

Jeongyong Yang, Seunghwan Jang, and SooJean Han. Safeflowmatcher: Safe and fast planning using flow matching with control barrier functions. In *International Conference on Learning Representations*, volume 2026, pp. 54058–54087, 2026.

Zhengbang Zhu, Minghuan Liu, Liyuan Mao, Bingyi Kang, Minkai Xu, Yong Yu, Stefano Ermon, and Weinan Zhang. Madiff: Offline multi-agent learning with diffusion models. *Advances in Neural Information Processing Systems*, 37:4177–4206, 2024.

A Proofs

A.1 Proof of Theorem 3.1

For any $i \in \mathcal{S}_a^{\mathrm{SE}}$, (12) gives $\nabla_{z_i} V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi) = \nabla_{z_i} q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}} \,|\, \chi)$. Along a trajectory of (6),

$$\frac{\mathrm{d}}{\mathrm{d}\tau} V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) = \partial_\tau q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) - \dot{\beta}_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0)) + \sum_{i \in \mathcal{S}_a^{\mathrm{SE}}} \nabla_{z_i} q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi)^\top \left(f_i^\theta(\tau, \mathbf{z}(\tau) \,|\, \chi) + g_i(\tau, \mathbf{z}(\tau) \,|\, \chi) u_i(\tau) \right). \quad (20)$$

According to (13), (7) becomes

$$\nabla_{z_i} q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi)^\top \left(f_i^\theta(\tau, \mathbf{z}(\tau) \,|\, \chi) + g_i(\tau, \mathbf{z}(\tau) \,|\, \chi) u_i(\tau) \right) + w_{a,i}^{\mathrm{SE}} \left(\partial_\tau q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) - \dot{\beta}_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0)) + \alpha_a^{\mathrm{SE}}\big(V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) \big) \right) \leq 0. \quad (21)$$

Summing (21) over $i \in \mathcal{S}_a^{\mathrm{SE}}$ and using $w_{a,i}^{\mathrm{SE}} \geq 0$ such that $\sum_{i \in \mathcal{S}_a^{\mathrm{SE}}} w_{a,i}^{\mathrm{SE}} = 1$ yields

$$\sum_{i \in \mathcal{S}_a^{\mathrm{SE}}} \nabla_{z_i} q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi)^\top \left(f_i^\theta(\tau, \mathbf{z}(\tau) \,|\, \chi) + g_i(\tau, \mathbf{z}(\tau) \,|\, \chi) u_i(\tau) \right) + \partial_\tau q_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) - \dot{\beta}_a^{\mathrm{SE}}(\tau \,|\, \chi, \mathbf{z}(0)) + \alpha_a^{\mathrm{SE}}\big(V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau) \,|\, \chi) \big) \leq 0. \quad (22)$$

According to (22) and (20), we have

$$\frac{\mathrm{d}}{\mathrm{d}\tau}V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau)\,|\,\chi) + \alpha_a^{\mathrm{SE}}\big(V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau)\,|\,\chi)\big) \le 0. \tag{23}$$

Since α_a^{SE} is an extended class-$\mathcal{K}_\infty$ function, we have $\alpha_a^{\mathrm{SE}}(0) = 0$ and $\alpha_a^{\mathrm{SE}}(s) \in \mathbb{R}_{\ge 0}$, $\forall s \in \mathbb{R}_{\ge 0}$. Suppose $V_a^{\mathrm{SE}}(0, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(0)\,|\,\chi) \le 0$ but there exists some $\tau_1 \in (0, 1]$ such that $V_a^{\mathrm{SE}}(\tau_1, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau_1)\,|\,\chi) \in \mathbb{R}_{>0}$. By continuity, let $\tau_0 < \tau_1$ be the last time before τ_1 at which $V_a^{\mathrm{SE}}(\tau_0, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau_0)\,|\,\chi) = 0$. Then, for every $\tau \in (\tau_0, \tau_1]$, $V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau)\,|\,\chi) \in \mathbb{R}_{>0}$, $\alpha_a^{\mathrm{SE}}\big(V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau)\,|\,\chi)\big) \in \mathbb{R}_{>0}$, and $\frac{\mathrm{d}}{\mathrm{d}\tau}V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau)\,|\,\chi) < 0$ by (23), which contradicts $V_a^{\mathrm{SE}}(\tau_0, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau_0)\,|\,\chi) = 0$ and $V_a^{\mathrm{SE}}(\tau_1, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau_1)\,|\,\chi) \in \mathbb{R}_{\ge 0}$. Therefore,

$$V_a^{\mathrm{SE}}(\tau, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(\tau)\,|\,\chi) \le 0, \quad \forall \tau \in [0, 1],$$

and (12) at $\tau = 1$ gives $q_a^{\mathrm{SE}}(1, \mathbf{z}_{\mathcal{S}_a^{\mathrm{SE}}}(1)\,|\,\chi) \le \beta_a^{\mathrm{SE}}(1\,|\,\chi, \mathbf{z}(0))$.

A.2 Proof of Proposition 3.2

In this paper, we choose

$$\phi_\delta(\eta) = \begin{cases} \eta, & \eta \le -\delta, \\ \delta\,(\eta/\delta - 1)^3\,(\eta/\delta + 3)\,/16, & |\eta| < \delta, \\ 0, & \eta \ge \delta, \end{cases}$$

which satisfies that $\phi_\delta \in C^2$, $\phi_\delta(\eta) \in [-1, 0]$, and $\phi_\delta(\eta) \le \min\{\eta, 0\}$, $\forall \eta \in [-1, 1]$.

Given an agnet i, assume it is incident to two active SE factors, indexed by 1 and 2, at any $\tau \in [0, 1]$. Since $\phi_\delta(\eta) \le \eta$, then we have

$$\begin{aligned}
(\ell_{2,i}^{\mathrm{SE}})^\top d_{1,i}^{\mathrm{SE}} &= \big\|\ell_{1,i}^{\mathrm{SE}}\big\|\,\big\|\ell_{2,i}^{\mathrm{SE}}\big\|\,\big(\eta_i^{\mathrm{SE}} - \phi_\delta(\eta_i^{\mathrm{SE}})\big) \ge 0, \\
(\ell_{1,i}^{\mathrm{SE}})^\top d_{2,i}^{\mathrm{SE}} &= \big\|\ell_{1,i}^{\mathrm{SE}}\big\|\,\big\|\ell_{2,i}^{\mathrm{SE}}\big\|\,\big(\eta_i^{\mathrm{SE}} - \phi_\delta(\eta_i^{\mathrm{SE}})\big) \ge 0, \\
(\ell_{1,i}^{\mathrm{SE}})^\top d_{1,i}^{\mathrm{SE}} &= \big\|\ell_{1,i}^{\mathrm{SE}}\big\|^2\,\big(1 - \eta_i^{\mathrm{SE}}\phi_\delta(\eta_i^{\mathrm{SE}})\big) \ge 0, \\
(\ell_{2,i}^{\mathrm{SE}})^\top d_{2,i}^{\mathrm{SE}} &= \big\|\ell_{2,i}^{\mathrm{SE}}\big\|^2\,\big(1 - \eta_i^{\mathrm{SE}}\phi_\delta(\eta_i^{\mathrm{SE}})\big) \ge 0,
\end{aligned}$$

where the last two quantities are strictly positive whenever the corresponding $d_{a,i}^{\mathrm{SE}} \ne 0$ since $\eta_i^{\mathrm{SE}} \in [-1, 1]$ and $\phi_\delta(\eta_i^{\mathrm{SE}}) \in [-1, 0]$. Additionally, for every active SE factor a, (14) satisfies $w_{a,i}^{\mathrm{SE}} \ge 0$ and $\sum_{i \in \mathcal{S}_a^{\mathrm{SE}}} w_{a,i}^{\mathrm{SE}} = 1$. Also, consider the SE inequalities

$$(\ell_{a,i}^{\mathrm{SE}})^\top u_i + b_{a,i}^{\mathrm{SE}} \le 0, \quad \forall a \in \{1, 2\}. \tag{24}$$

Define $\lambda_{a,i} \coloneqq b_{a,i}^{\mathrm{SE}}/((\ell_{a,i}^{\mathrm{SE}})^\top d_{a,i}^{\mathrm{SE}})$ when $b_{a,i}^{\mathrm{SE}} > 0$, and $\lambda_{a,i} \coloneqq 0$ otherwise. One can observe that $(\ell_{a,i}^{\mathrm{SE}})^\top d_{a,i}^{\mathrm{SE}} > 0$ whenever $b_{a,i}^{\mathrm{SE}} > 0$ since $d_{a,i}^{\mathrm{SE}} \ne 0$ in this case. Now choose

$$u_i = -\lambda_{1,i} d_{1,i}^{\mathrm{SE}} - \lambda_{2,i} d_{2,i}^{\mathrm{SE}}.$$

Then, for $a = 1$,

$$(\ell_{1,i}^{\mathrm{SE}})^\top u_i + b_{1,i}^{\mathrm{SE}} = -\lambda_{1,i}(\ell_{1,i}^{\mathrm{SE}})^\top d_{1,i}^{\mathrm{SE}} - \lambda_{2,i}(\ell_{1,i}^{\mathrm{SE}})^\top d_{2,i}^{\mathrm{SE}} + b_{1,i}^{\mathrm{SE}}. \tag{25}$$

If $b_{1,i}^{\mathrm{SE}} > 0$, the first and last terms on the right-hand side of (25) cancel, and the remaining term is nonpositive. If $b_{1,i}^{\mathrm{SE}} \le 0$, then $\lambda_{1,i} = 0$ and the left-hand side of (25) is nonpositive. The same analysis holds for $a = 2$. Hence, (24) are satisfied. If only one SE factor is active, then $d_{a,i}^{\mathrm{SE}} = \ell_{a,i}^{\mathrm{SE}}$, and the same construction with a single coefficient $\lambda_{a,i}$ also gives a feasible u_i. Therefore, the feasible set of (15) is nonempty.

A.3 Proof of Theorem 3.3

Since $\partial_\tau V_i^{\mathrm{PE}}(\tau, \mathbf{z} \,|\, \chi) = 0, \forall i \in \mathcal{N}$, we have

$$\frac{\mathrm{d}}{\mathrm{d}\tau} V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) = \sum_{i \in \mathcal{N}} \nabla_{z_i} V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi)^\top \Big(f_i^\theta(\tau, \mathbf{z}(\tau) \,|\, \chi) + g_i(\tau, \mathbf{z}(\tau) \,|\, \chi) u_i(\tau) \Big). \tag{26}$$

Using (16) in (10), summing (9) over $i \in \mathcal{N}$, and applying (26) yield

$$\frac{\mathrm{d}}{\mathrm{d}\tau} V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) + c^{\mathrm{PE}} \omega^{\mathrm{PE}}(\tau) \sum_{i \in \mathcal{N}} \big(V_i^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big)^{\rho^{\mathrm{PE}}} \leq 0.$$

Since $\rho^{\mathrm{PE}} \in (0, 1)$, $\alpha^{\mathrm{PE}}(s) = c^{\mathrm{PE}} s^{\rho^{\mathrm{PE}}}$ is subadditive on $\mathbb{R}_{\geq 0}$. Hence,

$$\alpha^{\mathrm{PE}}\big(V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big) \leq \sum_{i \in \mathcal{N}} \alpha^{\mathrm{PE}}\big(V_i^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big),$$

and thus

$$\frac{\mathrm{d}}{\mathrm{d}\tau} V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \leq -c^{\mathrm{PE}} \omega^{\mathrm{PE}}(\tau) \big(V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big)^{\rho^{\mathrm{PE}}}, \tag{27}$$

which adapts the finite-time Lyapunov inequality (Bhat & Bernstein, 2000) to the generative process using the time-varying weight $\omega^{\mathrm{PE}}(\tau)$. On any interval over which $V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) > 0$, multiplying (27) by $(1 - \rho^{\mathrm{PE}}) \big(V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big)^{-\rho^{\mathrm{PE}}}$ leads to

$$\frac{\mathrm{d}}{\mathrm{d}\tau} \big(V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big)^{1-\rho^{\mathrm{PE}}} \leq -c^{\mathrm{PE}}(1 - \rho^{\mathrm{PE}}) \omega^{\mathrm{PE}}(\tau).$$

Integrating from 0 to τ while $V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) > 0$ results in

$$\big(V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \big)^{1-\rho^{\mathrm{PE}}} \leq \big(V^{\mathrm{PE}}(\mathbf{z}(0) \,|\, \chi) \big)^{1-\rho^{\mathrm{PE}}} - c^{\mathrm{PE}}(1 - \rho^{\mathrm{PE}}) \int_0^\tau \omega^{\mathrm{PE}}(s) \, \mathrm{d}s.$$

Additionally, (27) and $V^{\mathrm{PE}} \geq 0$ imply that, for any $\bar{\tau} \in [0, 1]$,

$$V^{\mathrm{PE}}(\mathbf{z}(\bar{\tau}) \,|\, \chi) = 0 \implies V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) = 0, \ \forall \tau \in [\bar{\tau}, 1].$$

Therefore,

$$V^{\mathrm{PE}}(\mathbf{z}(\tau) \,|\, \chi) \leq \max \left\{ 0, \ \big(V^{\mathrm{PE}}(\mathbf{z}(0) \,|\, \chi) \big)^{1-\rho^{\mathrm{PE}}} - c^{\mathrm{PE}}(1 - \rho^{\mathrm{PE}}) \int_0^\tau \omega^{\mathrm{PE}}(s) \, \mathrm{d}s \right\}^{\frac{1}{1-\rho^{\mathrm{PE}}}},$$

and evaluating the upper bound at $\tau = 1$ implies $V^{\mathrm{PE}}(\mathbf{z}(1) \,|\, \chi) = 0$.

A.4 Proof of Corollary 3.4

If $b_i^{\mathrm{PE}} \leq 0$, then $u_i = 0$ satisfies the inequality constraint in (17). If $b_i^{\mathrm{PE}} > 0$ and $\ell_i^{\mathrm{PE}} \neq 0$, choose

$$u_i = -\frac{b_i^{\mathrm{PE}}}{(\ell_i^{\mathrm{PE}})^\top H_i^{-1} \ell_i^{\mathrm{PE}}} H_i^{-1} \ell_i^{\mathrm{PE}}.$$

Since $H_i \succ 0$ and $\ell_i^{\mathrm{PE}} \neq 0$, we have $(\ell_i^{\mathrm{PE}})^\top H_i^{-1} \ell_i^{\mathrm{PE}} > 0$, and thus $(\ell_i^{\mathrm{PE}})^\top u_i + b_i^{\mathrm{PE}} = 0$. Therefore, the feasible set of (17) is nonempty.

A.5 Proof of Theorem 3.5

For an initial sample $\xi \sim p_0$, define $\Delta(\tau \,|\, \xi) \coloneqq \mathbf{X}^{\mathrm{gui}}(\tau \,|\, \xi) - \mathbf{X}^{\mathrm{nom}}(\tau \,|\, \xi)$. Since the nominal and guided trajectories start from the same initial sample, $\Delta(0 \,|\, \xi) = 0$. Then, we can obtain

$$\Delta(\tau \,|\, \xi) = \int_0^\tau \big(f^\theta(s, \mathbf{X}^{\mathrm{gui}}(s \,|\, \xi) \,|\, \chi) - f^\theta(s, \mathbf{X}^{\mathrm{nom}}(s \,|\, \xi) \,|\, \chi) + \Gamma(s \,|\, \xi) \big) \, \mathrm{d}s.$$

Since $\left\| f^\theta(\tau, \mathbf{y} \,|\, \chi) - f^\theta(\tau, \mathbf{z} \,|\, \chi) \right\| \leq L(\tau) \left\| \mathbf{y} - \mathbf{z} \right\|$, we have

$$\left\| \Delta(\tau \,|\, \xi) \right\| \leq \int_0^\tau L(s) \left\| \Delta(s \,|\, \xi) \right\| \, \mathrm{d}s + \int_0^\tau \left\| \Gamma(s \,|\, \xi) \right\| \, \mathrm{d}s.$$

Then, applying Grönwall's inequality leads to

$$\left\| \Delta(1 \,|\, \xi) \right\| \leq \int_0^1 \exp\left(\int_s^1 L(r) \, \mathrm{d}r \right) \left\| \Gamma(s \,|\, \xi) \right\| \, \mathrm{d}s.$$

Thus, using Minkowski's inequality, we can obtain

$$\sqrt{\int_{\mathcal{Z}^N} \left\| \Delta(1 \,|\, \xi) \right\|^2 p_0(\xi) \, \mathrm{d}\xi} \leq \int_0^1 \exp\left(\int_s^1 L(r) \, \mathrm{d}r \right) \sqrt{\int_{\mathcal{Z}^N} \left\| \Gamma(s \,|\, \xi) \right\|^2 p_0(\xi) \, \mathrm{d}\xi} \, \mathrm{d}s. \tag{28}$$

Since the Wasserstein distance between μ and ν with finite second moments is

$$W_2^2(\mu, \nu) \coloneqq \inf_{\pi \in \Pi(\mu, \nu)} \int \left\| \mathbf{x} - \mathbf{y} \right\|^2 \, \mathrm{d}\pi(\mathbf{x}, \mathbf{y}), \tag{29}$$

where $\Pi(\mu, \nu)$ denotes the set of probability measures π on $\mathcal{Z}^N \times \mathcal{Z}^N$ whose first and second marginal distributions are μ and ν, respectively, we have

$$W_2^2(\mu_1, \nu_1) \leq \int_{\mathcal{Z}^N} \left\| \mathbf{X}^{\mathrm{gui}}(1 \,|\, \xi) - \mathbf{X}^{\mathrm{nom}}(1 \,|\, \xi) \right\|^2 p_0(\xi) \, \mathrm{d}\xi = \int_{\mathcal{Z}^N} \left\| \Delta(1 \,|\, \xi) \right\|^2 p_0(\xi) \, \mathrm{d}\xi.$$

Using(28), (18) can be obtained.

In addition, for a $\pi \in \Pi(\mu_1, \nu_1)$, let $(\mathbf{Z}^{\mathrm{nom}}, \mathbf{Z}^{\mathrm{gui}})$ have joint probability distribution π. Then $\mathcal{D}_N(\mathbf{Z}^{\mathrm{nom}} \,|\, \chi)$ and $\mathcal{D}_N(\mathbf{Z}^{\mathrm{gui}} \,|\, \chi)$ have probability distributions $\bar{\mu}_1$ and $\bar{\nu}_1$, respectively. Thus, according to (29) and $\left\| \mathcal{D}_N(\mathbf{y} \,|\, \chi) - \mathcal{D}_N(\mathbf{z} \,|\, \chi) \right\| \leq L_D \left\| \mathbf{y} - \mathbf{z} \right\|$, we can obtain

$$W_2^2(\bar{\mu}_1, \bar{\nu}_1) \leq \int \left\| \mathcal{D}_N(\mathbf{x} \,|\, \chi) - \mathcal{D}_N(\mathbf{y} \,|\, \chi) \right\|^2 \, \mathrm{d}\pi(\mathbf{x}, \mathbf{y}) \leq L_D^2 \int \left\| \mathbf{x} - \mathbf{y} \right\|^2 \, \mathrm{d}\pi(\mathbf{x}, \mathbf{y}).$$

Since $\pi \in \Pi(\mu_1, \nu_1)$ is arbitrary, we can have

$$W_2^2(\bar{\mu}_1, \bar{\nu}_1) \leq L_D^2 W_2^2(\mu_1, \nu_1),$$

which gives (19).

B Task 1: Detailed Settings and Additional Results

B.1 Representation, conditioning, network structure, and training

Each Task 1 trial consists of three stages. The plank pose, robot initial and goal positions, and robot-plank contact points are randomly sampled subject to workspace constraints. First, the robots move to the contact points. Then, the learned multi-agent flow policy generates the collaborative motions for the robots to manipulate the plank to form a bridge over the spatial gap, with or without SE guidance. The success rates in Table 1 evaluate only this manipulation stage. After the bridge is successfully constructed, the robots move to the goals through the bridge. The first and last stages are shown only for full process visualization.

Denote the plank pose as $g_B \in \mathbb{SE}(2)$, with orientation $R_B \in \mathbb{SO}(2)$ and position $p_B \in \mathbb{R}^2$. Its body twist is

$$\xi_B^b = (g_B^{-1} \dot{g}_B)^\vee = [(v_B^b)^\top, \omega_B^b]^\top \in \mathbb{R}^3,$$

where $(\cdot)^\vee : \mathfrak{se}(2) \to \mathbb{R}^3$. For robot i, let $r_i = [r_{i,x}, r_{i,y}]^\top \in \mathbb{R}^2$ denote its contact point in the body frame of the plank and define $r_i^\perp \coloneqq [-r_{i,y}, \, r_{i,x}]^\top$. The velocity of the contact point is $c_i^b = J_B(r_i) \xi_B^b$, with $J_B(r_i) \coloneqq [I_2, r_i^\perp] \in \mathbb{R}^{2 \times 3}$.

The object generated in Task 1 is a motion policy over a finite physical horizon. Conditioned on χ, the decoder maps the joint generative state to a sequence of robot contact-point velocities,

$$\mathcal{D}_N(\mathbf{z} \,|\, \chi) = \{c_{i,t}^b\}_{i \in \mathcal{N}, \, t \in \mathcal{T}},$$

where $\mathcal{T} \coloneqq \{1, \ldots, T\}$ and $c^b_{i,t} = J_B(r_i)\xi^b_{B,t}(\mathbf{z} \,|\, \chi)$. Only the first T_{exec} actions of the generated policy are executed before a new policy is generated.

Let $\mathcal{K} \coloneqq \{1, \ldots, K\}$. For each robot, the generative state $z_i \in \mathbb{R}^{24}$ is decoded into $K = 8$ three-dimensional vectors $\eta_{i,k} \in \mathbb{R}^3, \forall k \in \mathcal{K}$. Define $M_i \coloneqq J_B(r_i)^\top J_B(r_i)$ and $G \coloneqq \sum_{i=1}^N M_i$, with $G \succ 0$. Let $\mathcal{T} \coloneqq \{1, \ldots, T\}$ with $T = 20$, and let $B_{\text{spl}} \in \mathbb{R}^{T \times K}$ be the B-spline matrix. Before a new policy is generated, a T-step body twist sequence $\{\bar{\xi}^b_{B,t}\}_{t \in \mathcal{T}}$ is constructed with the previously generated policy and the current motion. Then, the decoder $\mathcal{D}_N(\mathbf{z} \,|\, \chi) = \{c^b_{i,t}\}_{i \in \mathcal{N}, t \in \mathcal{T}}$ gives $\xi^b_{B,t} = \bar{\xi}^b_{B,t} + \sum_{k=1}^K (B_{\text{spl}})_{t,k}(G^{-1} \sum_{i=1}^N M_i \eta_{i,k})$, and $c^b_{i,t} = J_B(r_i)\xi^b_{B,t}, \forall i \in \mathcal{N}, t \in \mathcal{T}$. Only the first $T_{\text{exec}} = 4$ actions of the generated T-step policy are executed before a new policy is generated. Before the next initial sample is drawn from a Gaussian distribution, the previous policy is shifted forward by these four actions, with its final value repeated to keep length T, and its first five values interpolated between the current motion and the shifted sequence. The resulting sequence $\{\bar{\xi}^b_{B,t}\}_{t \in \mathcal{T}}$ is included in the next condition χ.

For Task 1, the condition χ contains only quantities known before a new policy is generated and is characterized by three vectors for each robot i. Specifically, the vector $\lambda_i \in \mathbb{R}^{23}$ represents robot i, the plank, and their locations relative to the workspace geometry. For each other robot $j \neq i$, the vector $\pi_{ij} \in \mathbb{R}^8$ represents the position and velocity of robot j relative to robot i. The vector $c_i \in \mathbb{R}^{45}$ represents robot i's contact location on the plank, the desired plank pose, and the continuation of its contact point velocities from the previously generated policy. In particular, let $p^w_i, v^w_i \in \mathbb{R}^2$ be the current position and velocity of robot i in the world frame. Denote the current plank pose as $g_B = (R_B, p_B)$ with $p_B = [x_B, y_B]^\top$, and let θ_B denote the angle associated with R_B. Let $v^w_B \in \mathbb{R}^2$ and $\omega^b_B \in \mathbb{R}$ be the current translational and angular velocities of the plank. The corresponding quantities in the body frame of the plank are $\Delta p^b_i \coloneqq R_B^\top(p^w_i - p_B)$, $v^b_i \coloneqq R_B^\top v^w_i$, and $v^b_B \coloneqq R_B^\top v^w_B$. Let $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}]$ quantify the workspace, define $X \coloneqq (X_{\max} - X_{\min})/2$ and $Y \coloneqq (Y_{\max} - Y_{\min})/2$, and let x^L_{gap} and x^R_{gap} be the two x-coordinates of the gap edges. We denote by $x^{\min}_{\text{goal}}$ the minimum x-coordinate of the region from which robot goals are sampled. The vector $e_B \in \mathbb{R}^{11}$ contains the plank orientation and its position relative to the workspace, i.e., it consists of $\cos\theta_B$, $-\sin\theta_B$, $\sin\theta_B$, $\cos\theta_B$, $(X_{\min} - x_B)/X$, $(x^L_{\text{gap}} - x_B)/X$, $(x^R_{\text{gap}} - x_B)/X$, $(x^{\min}_{\text{goal}} - x_B)/X$, $(X_{\max} - x_B)/X$, $(Y_{\min} - y_B)/Y$ and $(Y_{\max} - y_B)/Y$, Denote $o_i \in \{0, 1\}^3$, which indicates if robot i is on the initial side ground, on the plank, or on the opposite side ground. Let $a_i \in \{0, 1\}$ indicate whether robot i contacts with the plank, and let $b_B \in \{0, 1\}$ indicate whether bridge construction has succeeded. Additionally,

$$\lambda_i = \left[\left(\text{diag}\left(\frac{1}{X}, \frac{1}{Y} \right) \Delta p^b_i \right)^\top, \left(\frac{v^b_i}{0.86} \right)^\top, o_i^\top, \left(\frac{v^b_B}{0.44} \right)^\top, \frac{\omega^b_B}{0.32}, b_B, a_i, e_B^\top \right]^\top \in \mathbb{R}^{23}.$$

For every $j \neq i$, denote the relative position and velocity $\Delta p^b_{ij} \coloneqq R_B^\top(p^w_j - p^w_i)$ and $\Delta v^b_{ij} \coloneqq R_B^\top(v^w_j - v^w_i)$ in the body frame of the plank, then

$$\pi_{ij} = \left[\left(\text{diag}\left(\frac{1}{X}, \frac{1}{Y} \right) \Delta p^b_{ij} \right)^\top, \left(\frac{\Delta v^b_{ij}}{0.86} \right)^\top, o_j^\top, a_j \right]^\top \in \mathbb{R}^8.$$

Let L_B be the plank length, and the desired plank pose be $g^{\text{goal}}_B = (R_g(\theta_g), p_g)$. Denote $R_B^\top(p_g - p_B) = [\Delta x^b_g, \Delta y^b_g]^\top$, and let $\Delta\theta_g \in (-\pi, \pi]$ be the shortest signed angular difference from θ_B to θ_g. Let $C^{\text{cont}}_i = [(J_B(r_i)\bar{\xi}^b_{B,1})^\top, \ldots, (J_B(r_i)\bar{\xi}^b_{B,T})^\top]^\top \in \mathbb{R}^{T \times 2}$. For each of the 40 entries of C^{cont}_i, the corresponding training set mean is subtracted and the result is divided by the corresponding training set standard deviation. Let $\widetilde{C}^{\text{cont}}_i \in \mathbb{R}^{T \times 2}$ denote the resulting matrix, then

$$c_i = \left[\frac{r_{i,x}}{L_B/2}, \text{sgn}(r_{i,y}), \frac{\Delta x^b_g}{L_B}, \frac{\Delta y^b_g}{Y}, \frac{\Delta\theta_g}{\pi/3}, \text{vec}(\widetilde{C}^{\text{cont}}_i)^\top \right]^\top \in \mathbb{R}^{45},$$

where $\text{sgn}(s) = 1$ for $s > 0$ and $\text{sgn}(s) = -1$ for $s < 0$, and $\text{vec}(\cdot)$ stacks the entries of a matrix into a vector.

For each $j \neq i$, the multilayer perceptron (MLP) ψ_{msg}, with layer dimensions $77 \to 160 \to 160$, is applied to z_j, c_j, and π_{ij} to form the message vector

$$m_{ij} = \psi_{\text{msg}}\big([z_j^\top, c_j^\top, \pi_{ij}^\top]^\top\big) \in \mathbb{R}^{160}.$$

The message vectors from all other robots are averaged as $\bar{m}_i \coloneqq (N-1)^{-1} \sum_{j \neq i} m_{ij}$. The MLP ψ_{ag} shared by each agent, with layer dimensions $258 \to 256 \to 256 \to 256 \to 24$, is used to represent

$$f_i^\theta(\tau, \mathbf{z} \,|\, \chi) = \psi_{\text{ag}}\big([z_i^\top, c_i^\top, \lambda_i^\top, \bar{m}_i^\top, (N-1)/7, \varphi(\tau)^\top]^\top\big) \in \mathbb{R}^{24},$$

where $\varphi(\tau) = [\tau, \sin(\pi\tau), \cos(\pi\tau), \sin(2\pi\tau), \cos(2\pi\tau)]^\top \in \mathbb{R}^5$.

The nominal vector field is trained and validated on $N \in \{4, 5, 6\}$. For Task 1, we instantiate the conditional probability path by choosing $p_0 = \mathcal{N}(0, I)$. For an encoded data sample $\mathbf{Z} \sim p_{\text{data}}(\mathbf{z} \,|\, \chi)$, we draw $\mathbf{X}_0 \sim p_0$ and choose $\mathbf{X}(\tau) = (1-\tau)\mathbf{X}_0 + \tau\mathbf{Z}$, $\forall \tau \sim \text{Unif}[0, 1]$. Then, the corresponding conditional vector field is $v(\tau, \mathbf{X}(\tau) \,|\, \mathbf{Z}, \chi) = \mathbf{Z} - \mathbf{X}_0$. At τ in the generative process, the estimated endpoint is $\widehat{\mathbf{Z}} = \mathbf{X}(\tau) + (1-\tau) f^\theta(\tau, \mathbf{X}(\tau) \,|\, \chi)$. Let $\widehat{c}_{i,t}^{\,b}$ and $\widehat{\xi}_{B,t}^{\,b}$ denote the velocity of the contact point and the body twist of the plank obtained by decoding $\widehat{\mathbf{Z}}$, and let $c_{i,t}^{b,*}$ and $\xi_{B,t}^{b,*}$ denote the corresponding quantities from the trajectory used to construct $\mathbf{Z}$. Let $\sigma_c \in \mathbb{R}^2$ contain the training set scales for the two components of the contact point velocity, and let $\sigma_\xi \in \mathbb{R}^3$ contain the training set scales for the three components of the body twist of the plank. Define $\rho_t = 2.5$ for $t = 1, \ldots, 4$ and $\rho_t = 1$ otherwise. The loss function $\mathcal{L}_{\text{MAC}}$ for Task 1 is augmented as

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{MAC}} + 0.16\, \mathcal{L}_{\text{motion}} + 0.45\, \mathcal{L}_{\text{twist}},$$

in which

$$\mathcal{L}_{\text{motion}} = \frac{1}{2N \sum_{t=1}^{T} \rho_t} \sum_{i=1}^{N} \sum_{t=1}^{T} \rho_t \left\| \text{diag}(\sigma_c)^{-1} \left(\widehat{c}_{i,t}^{\,b} - c_{i,t}^{b,*} \right) \right\|^2,$$

and

$$\mathcal{L}_{\text{twist}} = \frac{1}{3 \sum_{t=1}^{T} \rho_t} \sum_{t=1}^{T} \rho_t \left\| \text{diag}(\sigma_\xi)^{-1} \left(\widehat{\xi}_{B,t}^{\,b} - \xi_{B,t}^{b,*} \right) \right\|^2.$$

In addition, training and validation curves over three independent seeds are provided in Appendix B.3.

B.2 Task and safety factors

The task and safety factors are constructed from geometric margins. Each margin is positive when its corresponding requirement is satisfied. Specifically, the workspace is $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}] = [-7, 6.8] \times [-3, 3]$, and the spatial gap takes $x \in [x_{\text{gap}}^{\min}, x_{\text{gap}}^{\max}] = [0, 2]$. The plank length and width are $L_B = 5.8$ and $W_B = 0.72$, each robot is a disk with radius $r_R = 0.16$, the required plank support on each side of the gap is $d_{\text{sup}} = 0.55$, and the traversable region for the robot centers must extend at least $d_{\text{cor}} = 0.05$ beyond each gap edge. Let $g_{B,t} = (R_{B,t}, p_{B,t})$, $t = 1, \ldots, T_{\text{exec}}$, denote the plank poses predicted after the $T_{\text{exec}} = 4$ actions that will be executed, with $p_{B,t} = [x_t, y_t]^\top$ and orientation angle θ_t. Define $|\cos\theta|_\varepsilon \coloneqq \sqrt{\cos^2\theta + \varepsilon^2}$ and $|\sin\theta|_\varepsilon \coloneqq \sqrt{\sin^2\theta + \varepsilon^2}$ with $\varepsilon = 10^{-6}$. The distance from the plank center to its outermost point along the x-axis of the world frame is $h_x(\theta) \coloneqq \frac{1}{2}(L_B |\cos\theta|_\varepsilon + W_B |\sin\theta|_\varepsilon)$. The projection onto the same axis of the usable half-length of the plank for a robot center is $h_c(\theta) \coloneqq (L_B/2 - r_R - 0.075)\,|\cos\theta|_\varepsilon$. At predicted step t, the four margins are $m_{1,t} = x_{\text{gap}}^{\min} - x_t + h_x(\theta_t) - d_{\text{sup}}$, $m_{2,t} = x_t + h_x(\theta_t) - x_{\text{gap}}^{\max} - d_{\text{sup}}$, $m_{3,t} = x_{\text{gap}}^{\min} - x_t + h_c(\theta_t) - d_{\text{cor}}$, and $m_{4,t} = x_t + h_c(\theta_t) - x_{\text{gap}}^{\max} - d_{\text{cor}}$. Let $m_{1,t}^{\text{task}} = m_{1,t}, \ldots, m_{4,t}^{\text{task}} = m_{4,t}$, and let $m_{5,t}^{\text{task}}, \ldots, m_{M_{\text{task}},t}^{\text{task}}$ denote the remaining geometric margins included in the task factor, where M_{task} is the total number of task margins. The task factor evaluates all of these margins at the final predicted plank pose as

$$q_{\text{task}}(\mathbf{z} \,|\, \chi) = \frac{1}{80} \log\left(\sum_{k=1}^{2} \exp\left[-80\, \frac{m_{k,T_{\text{exec}}}^{\text{task}}}{0.55} \right] + \sum_{k=3}^{M_{\text{task}}} \exp\left[-80\, \frac{m_{k,T_{\text{exec}}}^{\text{task}}}{0.2} \right] \right).$$

Let $m_{t,1}^{\text{safe}}, \ldots, m_{t,M_{\text{safe}}}^{\text{safe}}$ denote the geometric margins at predicted step $t \in \{1, \ldots, T_{\text{exec}}\}$, where M_{safe} is the number of safety margins evaluated at each step. These margins measure the distances

of the plank from the upper and lower workspace boundaries using a 0.2 buffer, the distances of the robot disks from the gap and from the upper and lower workspace boundaries using a 0.1 buffer, and whether the plank has sufficient support on the side of the workspace where robots are initiated or has reached sufficient support on the opposite side. The safety factor is

$$q_{\text{safe}}(\mathbf{z} \mid \chi) = \frac{1}{60} \log \left(\sum_{t=1}^{T_{\text{exec}}} \sum_{k=1}^{M_{\text{safe}}} \exp \left[-60 \, \frac{m_{t,k}^{\text{safe}}}{0.1} \right] \right).$$

The success and failure are evaluated using the exact geometry (without buffers) instead of the factors above. A trial is successful if the robots can move the plank such that it provides at least $d_{\text{sup}} = 0.55$ of support on both sides of the gap and the region traversable by the robot centers extends beyond both gap edges as required. A failure is counted if the plank loses support on the initial side before reaching the opposite side, leaves the workspace, or causes any robot to cross the workspace boundary or enter the spatial gap.

We choose $g_i = I_{24}$, $H_i = I_{24}$, and $\alpha_a^{\text{SE}}(s) = s$. For either factor, $V_a^{\text{SE}}(\tau, \mathbf{z} \mid \chi) = q_a(\mathbf{z} \mid \chi) - \beta_a(\tau)$. Thus, $\beta_a(\tau)$ is the upper bound imposed on q_a during the generative process, and $V_a^{\text{SE}} \leq 0$ is equivalent to $q_a(\mathbf{z} \mid \chi) \leq \beta_a(\tau)$. Choose

$$\beta_a(\tau) = \beta_{a,\text{end}} + (\beta_{a,\text{start}} - \beta_{a,\text{end}}) \, (1 - 6\tau^5 + 15\tau^4 - 10\tau^3).$$

At the beginning of each policy generation, let g_B^{cur} denote the plank pose and let $q_{\text{task}}^{\text{cur}}$ denote the task factor evaluated at g_B^{cur} instead of the predicted terminal plank pose. Let $\mathbf{z}_0 \coloneqq \mathbf{z}(0)$ denote the Gaussian initial state during the generative process. Specifically, $\beta_{\text{task,end}} = q_{\text{task}}^{\text{cur}} - 0.05$, $\beta_{\text{task,start}} = \max\{q_{\text{task}}(\mathbf{z}_0 \mid \chi) + 0.02, \, \beta_{\text{task,end}} + 0.02\}$, $\beta_{\text{safe,end}} = 0$, and $\beta_{\text{safe,start}} = q_{\text{safe}}(\mathbf{z}_0 \mid \chi) + 0.02$. Thus, at the end of each generative process, the task factor is required to satisfy $q_{\text{task}} \leq q_{\text{task}}^{\text{cur}} - 0.05$, while the safety factor is required to satisfy $q_{\text{safe}} \leq 0$. Note that $\beta_{task,\text{start}}$ is calculated once before the generative process and is fixed throughout the generative process so $\beta_{\text{task}}(\tau)$ is C^1 in τ. In addition, the adaptive constraint allocation coefficients $w_{a,i}^{\text{SE}}$ are calculated using (14) with $\delta = 0.02$. Each robot then solves the decoupled QP (15) with one constraint corresponding to the task factor and one corresponding to the safety factor.

B.3 Additional results for Task 1

Figure 3 shows $q_{\text{task}}(\mathbf{z}(\tau) \mid \chi)$ and $\beta_{\text{task}}(\tau)$ along τ in the generative process for policies generated at early, middle, and late physical times in the $N = 5$ Guided trial corresponding to Figure 1. For visualization only, both curves in each plot are shifted by the same constant $\beta_{\text{task}}(1)$, and $q_{\text{task}} - \beta_{\text{task}}$ is unchanged. The shaded region corresponds to $q_{\text{task}}(\mathbf{z}(\tau) \mid \chi) \leq \beta_{\text{task}}(\tau)$. As shown in Figure 3, early in physical time, q_{task} starts close to its upper bound β_{task} and then moves farther below it during the generative process. This is because $\beta_{\text{task,start}}$ is initialized just above the task factor value of the initial sample, and $\beta_{\text{task,end}} = q_{\text{task}}^{\text{cur}} - 0.05$ requires the generated policy to reduce the task factor by at least 0.05 relative to its value at the current plank pose. The SE guidance therefore progressively separates q_{task} from the contracting upper bound while improving the task accomplishment. At the middle physical time, q_{task} stays below β_{task} even though it increases during the generative process. This is because the SE guidance does not require q_{task} to decrease monotonically. Instead, it only requires $q_{\text{task}} \leq \beta_{\text{task}}$, allowing the nominal vector field to dominate the generated policy. Across the three physical times, the observed relation $q_{\text{task}}(\mathbf{z}(\tau) \mid \chi) \leq \beta_{\text{task}}(\tau)$ is consistent with Theorem 3.1.

Figure 4 shows an $N = 7$ trial in which the two policies start from the same physical configuration and initial sample. The policy generated by the nominal vector field drives the plank out of the workspace, whereas the policy generated with SE guidance avoids this failure and successfully constructs the bridge. Figure 5 shows the evolution of q_{safe} and its time-varying upper bound β_{safe} during a generative process with SE guidance right before the policies generated by the nominal vector field drive the plank out of the workspace. During the generative process, $q_{\text{safe}}(\mathbf{z}(\tau) \mid \chi)$ stays below $\beta_{\text{safe}}(\tau)$ as the latter goes to $\beta_{\text{safe}}(1) = 0$, with the generated policy satisfying $q_{\text{safe}}(\mathbf{z}(1) \mid \chi) < 0$.

Figure 6 shows the training and validation losses over three independent training seeds. The lines show the mean across seeds, and the shaded regions show one sample standard deviation.

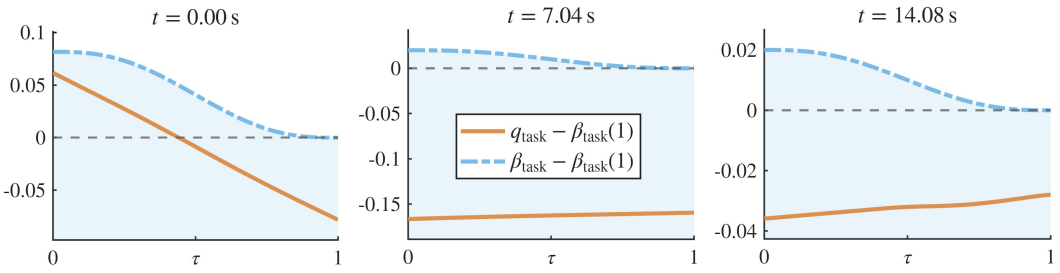

Figure 3: Evolution of q_{task} and its time-varying upper bound β_{task} during generative processes for policies generated at early, middle, and late physical times in the trial corresponding to Figure 1.

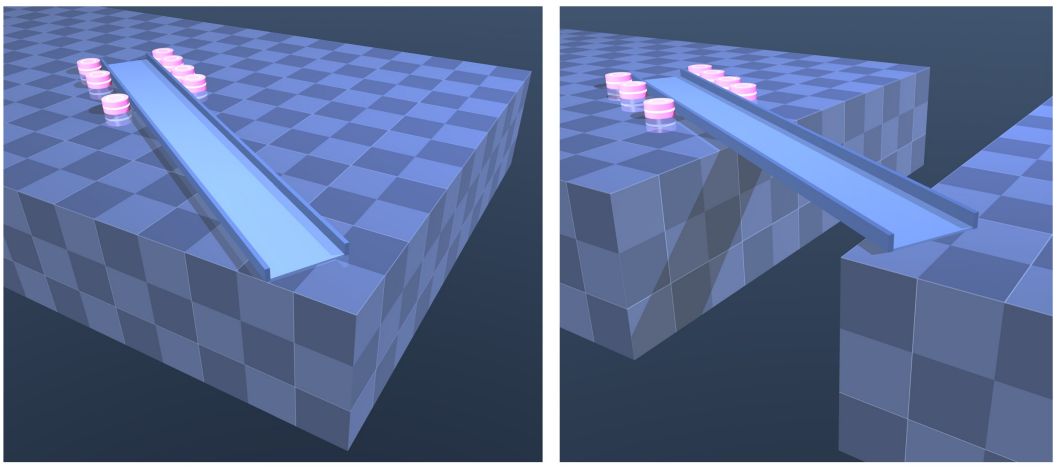

Figure 4: An $N = 7$ trial of Task 1. Left: the policies generated by the nominal vector field are driving the plank out of the workspace. Right: the policies with SE guidance lead to successful bridge construction.

C Task 2: Detailed Settings and Additional Results

C.1 Detailed settings for Task 2

Task 2 considers tabletop scenes with a library of 12 object types. For each scene, N different objects are randomly selected from the library. Each object has a fixed geometric shape and one or more nearby regions that should remain unobstructed when the object is used, which are referred to as affordance regions. A user-specified affordance requirement can change the location, size, or direction of these regions relative to the object. Thus, an object can be required to remain accessible from a different side, to have more free space around it, or to satisfy combinations of these changes. The complete object library, geometric representations, affordance-region definitions, and ranges used to generate modified requirements are given below.

For Task 2, we choose $g_i = I_3$, $\delta^{\text{PE}} = 0.02$, $\rho^{\text{PE}} = 0.75$, $\omega^{\text{PE}}(\tau) = 0.1 + 1.8\tau$, and $c^{\text{PE}} = 1.15 \max\{V^{\text{PE}}(0), 10^{-30}\}^{1-\rho^{\text{PE}}}/(1-\rho^{\text{PE}})$. The tabletop is centered at the origin and has size 1.40×0.94. For each object i, $z_i \in \mathbb{R}^3$ is mapped to the position and orientation by $[p_i^\top, \theta_i]^\top = \text{diag}(0.70, 0.47, 1.35)\, z_i$, with $p_i \in \mathbb{R}^2$ and $R_i(\theta_i) \in \mathbb{SO}(2)$. Each type object also has a fixed geometric shape defined in its body frame, represented as a disk, a rounded rectangle, or a combination of these shapes. In addition, one or more nearby regions are specified for each object and should be unobstructed when the object is used. We refer to these regions as *affordance regions*. Table 3 summarizes the 12 object categories, with each category showing at most once in a scene. Denote $C(r)$ for a disk of radius r, $R(a, b, r_c)$ for a rounded rectangle with half width a, half height b, and rounded corner radius r_c, and $A(x, y, r)$ for a circular affordance region of radius r centered at (x, y)

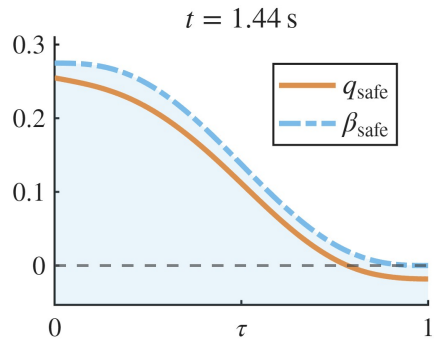

Figure 5: Evolution of q_{safe} and its time-varying upper bound β_{safe} during a generative process with SE guidance at a physical time right before the policies generated by the nominal vector field drive the plank out of the workspace in an $N = 7$ trial.

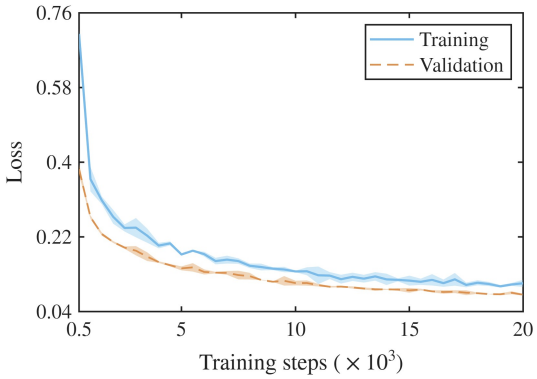

Figure 6: Training and validation losses over three independent seeds for the GNN-represented nominal vector field in Task 1. Shaded regions represent $\pm$ one sample standard deviation.

in the object's body frame. For multiple affordance regions with the same y-coordinate and radius, we list their x-coordinates as a set. For example, $A(\{x_1, \ldots, x_m\}, y, r)$ denotes circles centered at $(x_1, y), \ldots, (x_m, y)$, each with radius r. For the qth affordance region of object i, let $a_{iq} \in \mathbb{R}^2$ denote the center of the region in the body frame of object i, and let $r_{iq} > 0$ denote its radius. A user-specified requirement is parameterized by $(s_i^o, s_i^r, \varphi_i)$, where s_i^o changes the distance of the affordance region from the object, s_i^r changes its radius, and φ_i changes its direction relative to the object. Then, the center and radius expressed in the world frame are $a_{iq}^W = p_i + R(\theta_i + \varphi_i)s_i^o a_{iq}$ and $r_{iq}^W = s_i^r r_{iq}$, respectively. The default requirement corresponds to $(s_i^o, s_i^r, \varphi_i) = (1, 1, 0)$. A user may instead require an object to remain accessible from another side, require more free space around the object, or combine these changes. We sample $s_i^o \in [0.98, 1.12]$ and $s_i^r \in [1.04, 1.24]$. For a change in access direction, φ_i is selected from the allowed directions for that object type, with an additional perturbation in $[-\pi/22.5, \pi/22.5]$. The alternative directions are $-\pi/6$ and $\pi/6$ for drawer organizer and drawing board; $-\pi/2$ and $\pi/2$ for desk lamp, handled mug, and tissue box; π, $-\pi/2$, and $\pi/2$ for teacup, laptop, scissors, pencil case, and pen cup; and $-\pi/2$, $\pi/2$, and π for potted plant and compact printer.

In addition, training contains 4,000 scenes for each $N \in \{4, 5, 6\}$ and 400 validation scenes for each N. For every scene, N different types of objects are sampled uniformly without replacement from the library. Thus, all object categories appear during training, while $N = 7, 8$ evaluate generalization to scenes containing more objects than those used for training. The nominal vector field is represented by a GNN shared by every agent. It is trained using (5) without augmentation.

Table 3: Object geometry and default affordance regions for Task 2

Object	Geometry	Affordance region(s)
Drawer organizer	$R(0.156, 0.090, 0.012)$	$A(\{-0.100, 0, 0.100\}, -0.205, 0.108)$
Teacup	$C(0.046) \cup (C(0.019) + (0.061, 0))$	$A(0.125, 0, 0.055)$
Desk lamp	$C(0.060)$	$A(0, -0.147, 0.079)$
Laptop	$R(0.085, 0.065, 0.006)$	$A(0.155, 0, 0.070)$
Potted plant	$C(0.078)$	$A(0, -0.123, 0.062)$
Handled mug	$C(0.067)$	$A(0, -0.141, 0.054)$
Drawing board	$R(0.135, 0.048, 0.008)$	$A(\{-0.065, 0.065\}, -0.108, 0.050)$
Scissors	$R(0.039, 0.026, 0.016)$	$A(0.083, 0, 0.043)$
Compact printer	$R(0.099, 0.070, 0.009)$	$A(0, -0.133, 0.057)$
Pencil case	$R(0.037, 0.072, 0.007)$	$A(0.087, 0, 0.046)$
Tissue box	$R(0.049, 0.040, 0.009)$	$A(0, -0.103, 0.047)$
Pen cup	$C(0.034)$	$A(0.083, 0, 0.039)$

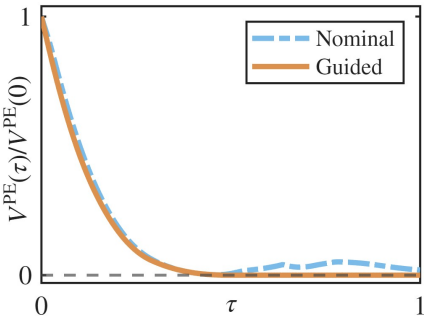

Figure 7: Evolution of V^{PE} during the generative process for the $N = 7$ example in Figure 2.

C.2 Additional results

Figure 7 shows the evolution of V^{PE} during the generative process for the same $N = 7$ example corresponding to Figure 2. Both trajectories start from the same initial Gaussian sample and are evaluated under the same left-handed laptop use requirement, which requires the mouse operating region on the left side of the laptop to be unobstructed. Without guidance, this requirement is violated at the end of the generative process, whereas PE guidance guarantees that V^{PE} converges to zero and generates a scene satisfying the new requirement.

In addition, we evaluate five types of requirement: the default requirements, opposite-side access, side-direction access, a larger affordance region, and combinations of these changes. Opposite-side access selects the direction farthest from the default one, and side-direction access selects one of the other non-default directions. Each type contains 50 cases, with 10 cases for each $N \in \{4, 5, 6, 7, 8\}$. As shown in Table 4, Guided directly generates the scenes satisfying all 50 cases in every type of requirement without retraining the nominal vector field.

Figure 8 shows a complementary $N = 8$ example without any user-specified change to the default requirements. The scene generated by the nominal vector field contains several violations: the compact printer and scissors are too close to each other, the desk lamp and pen cup are also too close to each other, and the drawer organizer obstructs the affordance region of the handled mug. Starting from the same initial Gaussian sample, PE guidance directly generates a scene satisfying all of these requirements, without any model retraining.

Figure 9 shows the training and validation flow matching losses over three independent training seeds. The lines show the mean across seeds, and the shaded regions show one sample standard deviation.

We further evaluate how much PE guidance changes the distribution characterized by the nominal flow. Task 2 provides a natural setting for this evaluation as each generative process directly produces one complete scene consisting of the poses of all N objects. For each fixed $N \in \{4, 5, 6, 7, 8\}$,

Table 4: Task 2 results for different affordance requirements.

Requirement	Natural Nominal	Private Nominal	Guided
Default	27/50 (54%)	27/50 (54%)	50/50 (100%)
Opposite-side access	27/50 (54%)	23/50 (46%)	50/50 (100%)
Side-direction access	31/50 (62%)	23/50 (46%)	50/50 (100%)
Larger affordance region	23/50 (46%)	22/50 (44%)	50/50 (100%)
Combined	27/50 (54%)	17/50 (34%)	50/50 (100%)

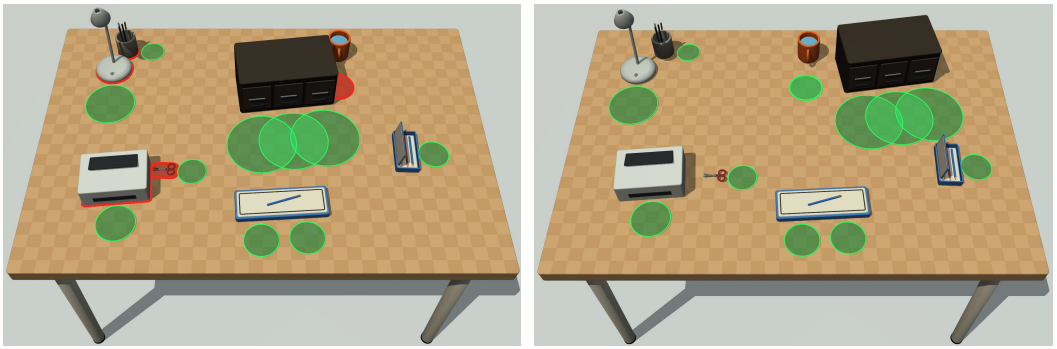

Figure 8: An $N = 8$ example of Task 2. Green circles denote unobstructed affordance regions, a red circle denotes an obstructed affordance region, and red object outlines indicate pairs of objects that are too close to each other. Left: the compact printer and scissors, and the desk lamp and pen cup, are too close to each other, and the drawer organizer obstructs the affordance region of the handled mug. Right: PE guidance resolves all three violations from the same initial sample without any model retraining.

let $M = 50$ denote the number of evaluated scenes. For scene $m \in \{1, \dots, M\}$ and object $i \in \{1, \dots, N\}$, denote the final generated states by Nominal and Guided by $z_{m,i}^{\mathrm{Nom}}$ and $z_{m,i}^{\mathrm{Gui}}$, respectively. The corresponding joint generative states of the complete scenes are then denoted by $\mathbf{z}_m^{\mathrm{Nom}} = [(z_{m,1}^{\mathrm{Nom}})^\top, \dots, (z_{m,N}^{\mathrm{Nom}})^\top]^\top$ and $\mathbf{z}_m^{\mathrm{Gui}} = [(z_{m,1}^{\mathrm{Gui}})^\top, \dots, (z_{m,N}^{\mathrm{Gui}})^\top]^\top$, respectively. Thus, the two sets $\{\mathbf{z}_m^{\mathrm{Nom}}\}_{m=1}^M$ and $\{\mathbf{z}_m^{\mathrm{Gui}}\}_{m=1}^M$ approximate the corresponding Nominal and Guided scene distributions. First, we evaluate the distance directly in the generative state space. For two scene states $\mathbf{z} = [z_1^\top, \dots, z_N^\top]^\top$ and $\mathbf{z}' = [(z_1')^\top, \dots, (z_N')^\top]^\top$, denote the normalized Euclidean distance as $d_z^2(\mathbf{z}, \mathbf{z}') = \frac{1}{N}\sum_{i=1}^N \|z_i - z_i'\|_2^2$. Using the M generated scenes, we approximate the Wasserstein distance between the Nominal and Guided joint distributions as

$$\widehat{W}_{2,z}^2(N) = \min_{\sigma \in \mathfrak{S}_M} \frac{1}{M} \sum_{m=1}^{M} d_z^2 \left(\mathbf{z}_m^{\mathrm{Nom}}, \mathbf{z}_{\sigma(m)}^{\mathrm{Gui}} \right),$$

where $\mathfrak{S}_M$ denotes the set of all permutations of $\{1, \dots, M\}$. Thus, σ specifies a optimal transport assignment between the M Nominal scenes and the M Guided scenes.

We also evaluate the same distributional deviation after decoding into object poses using $[(p_{m,i}^b)^\top, \theta_{m,i}^b]^\top = \mathrm{diag}(0.70, 0.47, 1.35)\, z_{m,i}^b$, with $b \in \{\mathrm{Nom}, \mathrm{Gui}\}$ representing Nominal and Guided, and $g_{m,i}^b \in \mathbb{SE}(2)$. Denote the complete decoded scene as $\mathbf{g}_m^b = \left(g_{m,1}^b, \dots, g_{m,N}^b\right)$. For two object poses g_i and g_i', we use the geodesic (Park, 1995)

$$d_{\mathbb{SE}(2)}^2(g_i, g_i') = \|p_i - p_i'\|_2^2 + \lambda_\theta^2 \Delta\theta_i^2,$$

where $\Delta\theta_i = \mathrm{atan2}\left(\sin(\theta_i - \theta_i'), \cos(\theta_i - \theta_i')\right) \in [-\pi, \pi]$, and we choose $\lambda_\theta = 0.10\,\mathrm{m}$, which places translational and rotational differences on a common length scale comparable to the sizes of the objects in Table 3. For two decoded scenes $\mathbf{g} = (g_1, \dots, g_N)$ and $\mathbf{g}' = (g_1', \dots, g_N')$, define the scene distance by

$$d_{\mathrm{scene}}^2(\mathbf{g}, \mathbf{g}') = \frac{1}{N} \sum_{i=1}^{N} d_{\mathbb{SE}(2)}^2(g_i, g_i').$$

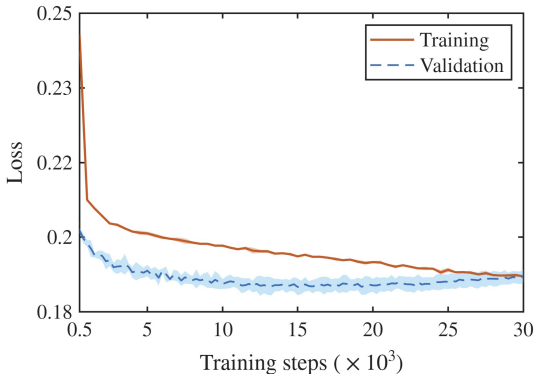

Figure 9: Training and validation flow matching losses over three independent seeds for the nominal vector field in Task 2. Shaded regions represent ± one sample standard deviation.

Then the corresponding approximation of the Wasserstein distance is

$$\widehat{W}_{2,g}^2(N) = \min_{\sigma \in \mathfrak{S}_M} \frac{1}{M} \sum_{m=1}^{M} d_{\text{scene}}^2 \left(\mathbf{g}_m^{\text{Nom}}, \mathbf{g}_{\sigma(m)}^{\text{Gui}} \right).$$

The magnitude of a Wasserstein distance is difficult to interpret without a reference scale, so we compare it with the root mean square (RMS) pairwise distance among the M complete scenes generated by the nominal vector field. In the generative state space, define

$$S_{\text{Nom},z}(N) = \left[\frac{2}{M(M-1)} \sum_{1 \le m < n \le M} d_z^2 \left(\mathbf{z}_m^{\text{Nom}}, \mathbf{z}_n^{\text{Nom}} \right) \right]^{1/2}.$$

Similarly, in the decoded physical space, define

$$S_{\text{Nom},g}(N) = \left[\frac{2}{M(M-1)} \sum_{1 \le m < n \le M} d_{\text{scene}}^2 \left(\mathbf{g}_m^{\text{Nom}}, \mathbf{g}_n^{\text{Nom}} \right) \right]^{1/2}.$$

We then plot the two ratios $\frac{\widehat{W}_{2,z}(N)}{S_{\text{Nom},z}(N)}$ and $\frac{\widehat{W}_{2,g}(N)}{S_{\text{Nom},g}(N)}$ in Figure 10, where the first ratio measures the Wasserstein distance between the Nominal and Guided distributions in the generative state space relative to the RMS pairwise distance among the M complete scenes generated by the nominal vector field, and the second ratio measures the one in the decoded pose space. For example, a value of 0.3 means that the Wasserstein distance between the Nominal and Guided distributions is 0.3 times the RMS pairwise distance among the M complete scenes generated by the nominal vector field. As the number of objects increases, both curves increase gradually, meaning that guidance makes larger modifications when there are more objects and thus more coupled requirements. As shown in Figure 10, in the generative state space, $\widehat{W}_{2,z}(N)/S_{\text{Nom},z}(N)$ increases from 20.6% for $N = 4$ to 34.5% for $N = 8$. In the decoded pose space, $\widehat{W}_{2,g}(N)/S_{\text{Nom},g}(N)$ increases from 20.5% to 33.7%. Thus, even for $N = 8$, the Wasserstein distance between the Nominal and Guided distributions is only about one third of the RMS pairwise distance among the complete scenes generated by the nominal vector field that measures how spread out the generated scenes are. The results show that guidance introduce a distributional deviation that remains much smaller than the variation among scenes generated by the nominal vector field.

We additionally examine whether guidance reduces how spread the generated objects are. Define

$$S_{\text{Gui},z}(N) = \left[\frac{2}{M(M-1)} \sum_{1 \le m < n \le M} d_z^2 \left(\mathbf{z}_m^{\text{Gui}}, \mathbf{z}_n^{\text{Gui}} \right) \right]^{1/2},$$

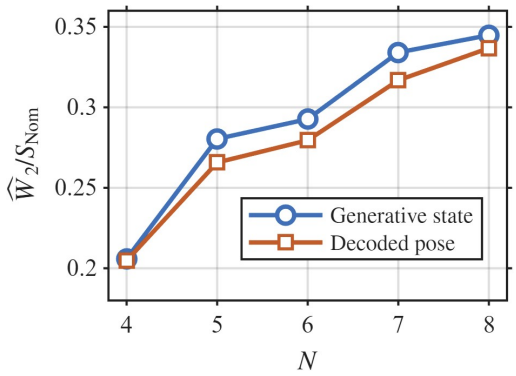

Figure 10: Approximation of the Wasserstein distance between the Nominal and Guided distributions in Task 2 with respect the total number of generated objects, normalized by the RMS pairwise distance among the $M = 50$ scenes generated by the nominal vector field. The blue and orange curves use the distance metrics in the generative state space and the decoded pose space, respectively.

and

$$S_{\text{Gui},g}(N) = \left[\frac{2}{M(M-1)} \sum_{1 \le m < n \le M} d^2_{\text{scene}} \left(\mathbf{g}^{\text{Gui}}_m, \mathbf{g}^{\text{Gui}}_n \right) \right]^{1/2} .$$

These quantities are defined in exactly the same way as $S_{\text{Nom},z}(N)$ and $S_{\text{Nom},g}(N)$, but using the M complete scenes generated with guidance. Across $N = 4, \ldots, 8$, $S_{\text{Gui},z}(N)/S_{\text{Nom},z}(N)$ ranges from 98.3% to 99.9%, and $S_{\text{Gui},g}(N)/S_{\text{Nom},g}(N)$ ranges from 99.1% to 100.5%, meaning that the RMS pairwise distance among Guided scenes is almost the same as that among scenes generated by the nominal vector field.

Together with Figure 10, these results show that PE guidance satisfies the desired task requirements or hard constraints while resulting in only a moderate deviation of the distribution, and approximately preserving the spread of the generated scenes.