Projected Coupled Diffusion for Test-Time Constrained Joint Generation

Hao Luan[1*], **Yi Xian Goh**[2†*], **See-Kiong Ng**[1,3], **Chun Kai Ling**[1]

[1]School of Computing, National University of Singapore
[2]Faculty of Computer Science and Information Technology, Universiti Malaya
[3]Institute of Data Science, National University of Singapore
haoluan@comp.nus.edu.sg, u2102756@siswa.um.edu.my, {seekiong, chunkail}@nus.edu.sg

Abstract

Modifications to test-time sampling have emerged as an important extension to diffusion algorithms, with the goal of biasing the generative process to achieve a given objective without having to retrain the entire diffusion model. However, generating jointly correlated samples from multiple pre-trained diffusion models while simultaneously enforcing task-specific constraints without costly retraining has remained challenging. To this end, we propose *Projected Coupled Diffusion* (PCD), a novel test-time framework for constrained joint generation. PCD introduces a coupled guidance term into the generative dynamics to encourage coordination between diffusion models and incorporates a projection step at each diffusion step to enforce hard constraints. Empirically, we demonstrate the effectiveness of PCD in application scenarios of image-pair generation, object manipulation, and multi-robot motion planning. Our results show improved coupling effects and guaranteed constraint satisfaction without incurring excessive computational costs.

Code — https://github.com/EdmundLuan/pcd

1 Introduction

Diffusion models have achieved remarkable success in generative modeling, with a plethora of applications ranging from image (Rombach et al. 2022), video (Ho et al. 2022), language (Li et al. 2022), graph generation (Niu et al. 2020; Madeira et al. 2024; Luan, Ng, and Ling 2025), as well as robotics (Janner et al. 2022; Chi et al. 2023; Carvalho et al. 2023). One of the crucial factors underlying these achievements is the use of *test-time* conditional sampling techniques such as classifier guidance (Dhariwal and Nichol 2021; Song et al. 2021), inpainting (Lugmayr et al. 2022; Liu et al. 2023), reward alignment (Uehara et al. 2025; Kim, Kim, and Park 2025), and projection (Christopher, Baek, and Fioretto 2024; Sharma, Kumar, and Trivedi 2024).

While these methods are primarily designed for sampling from univariate distributions, numerous real-world tasks require sampling highly correlated variables from joint distributions, *e.g.*, image pairs (Zeng et al. 2024), multimedia (Ruan et al. 2023; Tang et al. 2023; Hayakawa et al. 2025), traffic prediction (Jiang et al. 2023; Wang et al. 2024b), and multi-robot motion planning (Shaoul et al. 2025; Liang et al. 2025). Directly training a diffusion model to capture one single joint distribution is costly and inefficient. First, high-quality annotated datasets of joint behaviors are scarce, expensive, and often proprietary. One example is real-world traffic trajectory data, which are essential for prediction and planning in autonomous driving (Li et al. 2023). Second, training joint distributions becomes increasingly computationally demanding as the number of variables grows (Gu et al. 2024), and relearning the entire joint distribution becomes necessary when marginals are changed. For instance, coordinating robot teams may require *retraining the entire model* even if only one robot's behavior differs from its marginal of the pretrained joint for a new task.

Inspired by compositional modeling (Du, Li, and Mordatch 2020; Liu et al. 2022; Du and Kaelbling 2024; Wang et al. 2024a; Cao et al. 2025), we opt for a more practical approach of modeling multiple marginal distributions independently— each cheaper and simpler to train— and to couple them during *test-time* in a sensible way to obtain the required joint distribution. Unfortunately, such test-time coupling alone does not efficiently *guarantee* adherence to task-specific *hard* constraints such as safety protocols and physical limits. To address such limitation, we propose extending standard Langevin dynamics by combining projection methods with coupled dynamics. Our method cleanly integrates *multiple* pretrained diffusion models through a coupling cost while explicitly incorporating a projection step, thereby ensuring strict adherence to task-specific constraints throughout joint sampling.

Contributions. We propose *Projected Coupled Diffusion* (PCD), a novel test-time framework unifying both coupled generation leveraging multiple pre-trained diffusion models and projection-based generation to enforce hard constraints only specified at inference. PCD suitably generalizes some conditional sampling techniques including classifier guidance. We show empirically that PCD with both a coupled cost and projection is superior in *jointly* generating highly *correlated* samples with *hard constraints* compared to alternatives with the absence of either or both components.

[*]These authors contributed equally.
[†]This work was done at National University of Singapore.

2 Related Work

Diffusion Models Diffusion models conceptualize generation as a progressive denoising process (DDPM) (Ho, Jain, and Abbeel 2020) or, equivalently, as a gradient–descent-like procedure that leverages the score of the data distribution within a Langevin dynamics framework (Welling and Teh 2011; Song and Ermon 2019; Song et al. 2021). Improving DDPM, Song, Meng, and Ermon (2021) introduced DDIM to accelerate sampling, and Karras et al. (2022) systematically clarified key design choices for practitioners.

Diffusion Guidance Guidance mechanisms form an important class of conditioning techniques for diffusion sampling. Dhariwal and Nichol (2021) first introduced *classifier guidance* (CG) to steer pretrained diffusion models at inference time without retraining, while Ho and Salimans (2022) proposed *classifier-free guidance* by integrating conditioning signals directly during training. Building upon CG, subsequent work has extended guidance beyond classifiers to include analytic functions (Guo et al. 2024; Lee et al. 2025) and property predictors (Meng and Fan 2024; Feng et al. 2024) in tasks beyond image generation.

Constrained Diffusion To address requirements of constraints in real-world tasks, researchers resort to constraint-guided diffusion generation (Yang et al. 2023; Kondo et al. 2024; Feng et al. 2024). However, such paradigm falls short of *enforcing* constraint satisfaction. This incentives the introduction of projection operation at each step of diffusion (Bubeck, Eldan, and Lehec 2015; Christopher, Baek, and Fioretto 2024), later applied in other applications (Liang et al. 2025). A primal-dual LMC method by Chamon, Karimi, and Korba (2024) handles both inequality and equality constraints, and Zampini et al. (2025) proposed a Lagrangian relaxation of projection in the latent space.

Diffusion for Joint Generation Previous studies in joint generation using multiple diffusion models primarily targeted multimodal generation. Bar-Tal et al. (2023) and Lee et al. (2023) demonstrated panoramic image synthesis by synchronizing several image diffusion models. Xing et al. (2024) and Hayakawa et al. (2025) demonstrated synchronized audio–video generation, and Tang et al. (2023) introduced a framework for generating and conditioning content across arbitrary combinations of a set of modalities.

3 Preliminaries

Notation. Denote by $\mathbb{Z}^+$ the set of all positive integers, $\|\cdot\|$ the Euclidean norm and $\|\cdot\|_F$ the Frobenius norm. Let $X \in \mathbb{R}^{D_x}$ and $Y \in \mathbb{R}^{D_y}$ be random variables where $D_x, D_y \in \mathbb{Z}^+$ are their dimensionality, respectively. We denote $p_X(x)$ as the probability density function of random variable X, likewise for Y, and may omit the subscript indicating the random variable when it is clear from the context for notational brevity. Let $\mathcal{N}(\mu, \Sigma)$ be a normal distribution with mean μ and covariance Σ. Denote $\Pi_{\mathcal{K}_X}(x) : \mathbb{R}^{D_x} \to \mathbb{R}^{D_x}$ as a projection onto a *nonempty* set $\mathcal{K}_X \subseteq \mathbb{R}^{D_x}$: $\Pi_{\mathcal{K}_X}(x) \triangleq \arg\min_{z \in \mathcal{K}_X} \|z - x\|$ [1].

Diffusion and Score-based Generative Models. We examine diffusion models' inference from the perspective of Langevin dynamics. Let $E_X(x) : \mathbb{R}^{D_x} \to \mathbb{R}$ be a continuously differentiable energy function with Lipschitz-continuous gradients and $Z = \int_{\mathbb{R}^{D_x}} \exp(-E_X(x))\,\mathrm{d}x < \infty$. This energy function defines a probability density $p_X(x) = 1/Z \cdot \exp(-E_X(x))$. To sample from $p_X(x)$, one may leverage Langevin Monte Carlo (LMC) (Roberts and Tweedie 1996; Welling and Teh 2011). Given an initial sample from a prior distribution $X_0 \sim p'_X(x)$ and a fixed time step size $\delta \in \mathbb{R}^+$, the LMC iterates as follows:

$$X_{t+1} = X_t + \delta \nabla_x \log p_X(X_t) + \epsilon_t \tag{1}$$

where $\epsilon_t \in \mathcal{N}(0, 2\delta \boldsymbol{I})$ and $\nabla_x \log p_X(x)$ is called the *(Stein) score function* of $p_X(x)$. When $\delta \to 0$ and $T \to \infty$, the distribution of X_T converges to $p_X(x)$ under some regularity conditions (Welling and Teh 2011). In practice, the analytic form of $E_X(x)$ is not accessible. Instead, the score $\nabla_x \log p_X(x)$ or equivalently the gradient $\nabla E_X(x)$ is approximated by a neural network parameterized by θ and trained via denoising score matching (Song and Ermon 2019): $s^\theta_X(X_t, t) \approx \nabla_x \log p_X(X_t)$.

Classifier guidance. Classifier guidance (CG) is a "soft" way of influencing the sampling distribution. Given a desired attribute y_0 as condition, the objective of CG is to sample from a target conditional distribution $p_{X|Y}(x \mid y = y_0)$. CG achieves this by perturbing the original learned score with a likelihood term and obtain the posterior score:

$$\nabla_x \log p_{X|Y}(x \mid y_0) = \nabla_x \log p_X(x) + \nabla_x p_{Y|X}(y_0 \mid x),$$

where $\nabla_x \log p_X(x)$ is approximated by the trained score model s^θ_X, and the likelihood $p_{Y|X}(y \mid x)$ is modeled by a classifier, predictor, or a differentiable function.

Projected diffusion. To *enforce* hard constraint $X \in \mathcal{K}_X$, projected LMC performs a projection every time step:

$$X_{t+1} = \prod_{\mathcal{K}_X} (X_t + \delta \nabla_x \log p_X(x) + \epsilon_t) \tag{2}$$

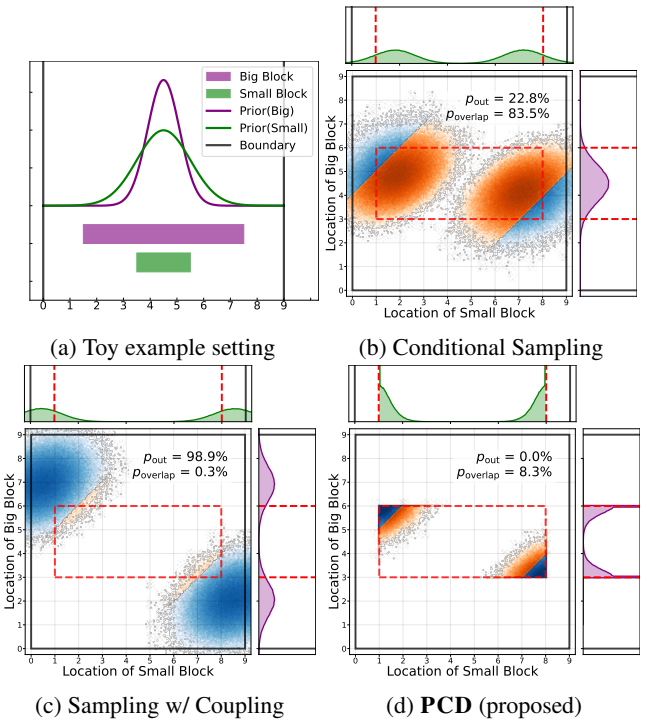
with $\epsilon_t \sim \mathcal{N}(0, 2\delta I)$. The convergence of Eq. (2) is analyzed by Bubeck, Eldan, and Lehec (2015) when $\mathcal{K}_X$ is convex and $p_X(x)$ is log-concave.

4 Projected Coupled Diffusion

We study the problem of generating *correlated* samples (X, Y) under the *test-time constraints* of $X \in \mathcal{K}_X$ and $Y \in \mathcal{K}_Y$, with two pre-trained scores or diffusion models $s^\theta_X(x, t)$ and $s^\phi_Y(y, t)$, parameterized by θ and ϕ, respectively, *without retraining* either of them.

Coupled Dynamics through Costs. To facilitate correlation between the generated X and Y, we propose using a cost function to couple their diffusion dynamics, *i.e.*, Eq. (1) for X and likewise for Y. Let the cost function $c(x, y)$:

[1] We break ties arbitrarily if $\arg\min_{z \in \mathcal{K}_X} \|z - x\|$ is not unique. Uniqueness is guaranteed when $\mathcal{K}_X$ is convex.



(a) Toy example setting (b) Conditional Sampling

(c) Sampling w/ Coupling (d) **PCD** (proposed)

Figure 1: Toy example of fitting two blocks with different sizes into a narrow corridor. (a) Problem setting: the bigger block (purple) and small block (green) are with lengths 6 and 2. The score functions have learned are two Gaussians centered at the middle of the red corridor of length 9. (b) Naive conditional approach: the big block occupies the center and the small block is highly possible to overlap if to stay within the corridor (orange for probability mass of overlapping positions, blue for non-overlapping positions). (c) The coupled dynamics can place both blocks at different sides, resulting in a much lower overlapping probability, but still can go out of the corridor. (d) PCD leads to a low overlapping probability and guarantees both blocks stay within the corridor.

$\mathbb{R}^{D_x} \times \mathbb{R}^{D_y} \to \mathbb{R}$ be continuously differentiable. Then the coupled joint dynamics of (X, Y) are

$$X_{t+1} = X_t - \gamma\delta\nabla_x c(X_t, Y_t) + \delta s_X^\theta(X_t, t) + \epsilon_{X,t} \quad \text{(3a)}$$

$$Y_{t+1} = Y_t - \gamma\delta\nabla_y c(X_t, Y_t) + \delta s_Y^\phi(Y_t, t) + \epsilon_{Y,t} \quad \text{(3b)}$$

where $\gamma \in \mathbb{R}^+$ is a coupling strength parameter, $\epsilon_{X,t} \in \mathcal{N}(0, 2\delta \boldsymbol{I}_{D_x})$, $\epsilon_{Y,t} \in \mathcal{N}(0, 2\delta \boldsymbol{I}_{D_y})$ are i.i.d. Gaussian noise drawn at each diffusion time step. The cost function $c(x, y)$ can either be an analytic function or a neural network, *e.g.*, a trained classifier or a regression model, and we demonstrate the use of both in our experiments.

As an extension, for each cost function instance $c(x, y)$, we can derive its posterior sampling (PS) variant $c_{\text{PS}}(x, y, t)$ with the DPS method (Chung et al. 2023). Concretely, by Tweedie's formula (Efron 2011) we may obtain a point estimate for each variable's denoised version through the trained scores, and then compute the cost with those estimates:

$$c_{\text{PS}}(x, y, t) = c\left(X_t + \sigma_{X,t}^2 s_X^\theta(X_t, t), Y_t + \sigma_{Y,t}^2 s_Y^\phi(Y_t, t)\right),$$

where $\sigma_{X,t}^2$ and $\sigma_{Y,t}^2$ are the corresponding noise levels at time step t associated with the score models.

Remark 1. The PS variant does not exactly match the description of a cost function in Eq. (3) due to the extra dependence on diffusion time step t. Yet empirically we find them performing well in our proposed PCD framework.

Projected Coupled Diffusion. On top of promoting correlations between the generated samples, we also aim to *enforce* the constraints given only *at test-time*. As such, we proposed to join the coupled dynamics and projection, yielding the Projected Coupled Diffusion (PCD):

$$X_{t+1} = \prod_{\mathcal{K}_X} \left(X_t - \gamma\delta\nabla_x c(X_t, Y_t) + \delta s_X^\theta(X_t, t) + \epsilon_{X,t}\right)$$

$$Y_{t+1} = \prod_{\mathcal{K}_Y} \left(Y_t - \gamma\delta\nabla_y c(X_t, Y_t) + \delta s_Y^\phi(Y_t, t) + \epsilon_{Y,t}\right)$$

where $\delta \in \mathbb{R}^+$ is the LMC step size parameter, $\gamma \in \mathbb{R}^+$ is the coupling strength parameter, $\epsilon_{X,t} \sim \mathcal{N}(0, 2\delta \boldsymbol{I}_{D_x})$, $\epsilon_{Y,t} \sim \mathcal{N}(0, 2\delta \boldsymbol{I}_{D_y})$ are i.i.d. noise drawn per step and $X_0 \sim \mathcal{N}(0, \boldsymbol{I}_{D_x})$, $Y_0 \sim \mathcal{N}(0, \boldsymbol{I}_{D_y})$. Generation algorithms of our method adopting LMC or Denoising Diffusion Probabilistic Models (DDPM) are in the appendix.

We illustrate the importance of *simultaneous* coupling and projection through a toy example (Figure 1a). Two 1D blocks of different lengths must fit within a corridor and avoid overlapping as far as possible. Each of the block's center (denoted by X and Y) are generated with a score model that has learned a Gaussian distribution centered at the midpoint of the corridor. A naive approach is to first generate X with only the learned score, and then generate Y conditioned on X via classifier guidance. However, this can lead to samples with poor mutual correlations, *e.g.*, the first generated X occupies the center of the corridor regardless of Y, leaving not enough room to fit both (Figure 1b)[2]. In contrast, coupled dynamics incorporates mutual influence into the generation process of both variables via the cost function, resulting in a much lower overlap probability, yet could violate the corridor constraint (Figure 1c). Our proposed PCD that joins coupled dynamics and projection, as shown in Figure 1d, can both avoid overlapping and enforce corridor constraint.

4.1 Relationship to Other Methods

Classifier Guidance. Interestingly, our framework can encompass the prevailing technique of classifier guidance (CG) as a special case. If we set the cost function as $c(x, y) \propto -\log p_{Y|X}(y \mid x)$, assuming a continuously differentiable density $p_{Y|X}$ exists and the (approximated) gradient $\nabla_x p_{Y|X}(y \mid x)$ is accessible, and let $\mathcal{K}_Y$ be a singleton only containing the conditioning information $\mathcal{K}_Y = \{y_0\}$ and $\mathcal{K}_X = \mathbb{R}^{D_x}$, then PCD reduces to

$$X_{t+1} = X_t + \delta\nabla_x \log\left[\left(p\left(y_0 \mid X_t\right)\right)^\gamma p(X_t)\right] + \epsilon_t \quad (5)$$

and trivially $Y_t = y_0$ with $\epsilon_t \sim \mathcal{N}(0, 2\delta \boldsymbol{I})$ and γ becoming a temperature for the likelihood; when $\gamma = 1$, the gradient term fully recovers the score of the posterior distribution

[2]In fact, if the small block is generated first, it is highly possible that it is *infeasible* to put the big block without exceeding the corridor and overlapping.

$p_{X|Y}(x \mid y = y_0)$. In that regard, CG can be seen as PCD with one variable fixed and projection removed in the other.

Projected Diffusion. We can consider projected diffusion as PCD without coupling. In PDM (Christopher, Baek, and Fioretto 2024), which only concerns a single variable X, the LMC dynamics of X is projected onto a nonempty but not necessarily convex set $\mathcal{C} \subset \mathbb{R}^{D_x}$. This work fits into PCD in the sense of (i) $\mathcal{K}_X = \mathcal{C}$, $\mathcal{K}_Y = \{y_0\}$, and (ii) the cost function $c(x, y) \equiv 0$, preserving the projection and decoupling the dynamics (turning Y_t into a dummy variable).

Compositional Diffusion. Another notable line of research is in compositional diffusion (Liu et al. 2022; Wang et al. 2024a; Xu et al. 2024; Cao et al. 2025). Similar to ours, this class of methods also aim to "combine" multiple distributions modeled by energy-based models (EBMs), *e.g.*, diffusion or score models. Unlike ours, however, they still focus on a univariate distribution (cf. a joint distribution in ours). These works attempt to sample from a target product distribution $p_X^{\text{prod}}(x) \propto p_X(x) \prod_{i=1}^N \left(p_X^i(x)/p_X(x)\right) \propto \exp(-E_X(x)) \cdot \exp\left(N \cdot E_X(x) - \sum_i^N E_{\theta^i}^i(x)\right)$ where $E_{\theta^i}^i(x)$ are the corresponding energy functions for $i = 1, \ldots, N$. In practice, the gradients of the N energy functions $\nabla E_{\theta^i}^i(x)$ are modeled by N diffusion models $s_X^{\theta^i}(x)$. PCD can conceptually cover those methods by (i) projecting X_t onto $\mathbb{R}^{D_x}$, (ii) projecting Y_t onto a singleton $\{y_0\}$ and rendering it a dummy variable, and (iii) setting the cost function as: $c(x, y) = N \cdot E_X(x) - \sum_i^N E_{\theta^i}^i(x)$, whereas only its partial gradient $\nabla_x c(x, y)$ is being used.

Joint Diffusion. Hayakawa et al. (2025) propose to decompose the "joint scores" $\nabla_x \log p(x, y)$ and $\nabla_y \log p(x, y)$ with Bayes rule: $\nabla_{X_t} \log p(X_t, Y_t) = \nabla_{X_t} \log p_X(X_t) + \nabla_{X_t} p_{Y|X}(Y_t|X_t)$ and likewise for $\nabla_{Y_t} \log p(X_t, Y_t)$, and train one single discriminator $\mathcal{D}_\theta(x, y)$ to approximate both conditional scores: $\nabla_{X_t} \log p(Y_t|X_t) \approx \nabla_{X_t} \log \frac{\mathcal{D}_\theta(X_t, Y_t)}{1 - \mathcal{D}_\theta(X_t, Y_t)}$, likewise for $\nabla_{Y_t} \log p(X_t|Y_t)$. PCD can cover this by setting the constraint sets to $\mathcal{K}_X = \mathbb{R}^{D_x}$, $\mathcal{K}_Y = \mathbb{R}^{D_y}$, and the cost to $c(x, y) = -\log \frac{\mathcal{D}_\theta(x,y)}{1 - \mathcal{D}_\theta(x,y)}$.

5 Experiments

We seek to address the following research question:

> *How effective is our proposed PCD method in terms of jointly generating correlated samples with test-time constraints compared to generation only with projection, coupling costs, or neither?*

We also explore through ablations tradeoffs between coupling and adherence to the trained data distribution.

5.1 Constrained Multi-Robot Navigation

We show that our method extends to more than two variables by running constrained multi-robot motion tasks in (Shaoul et al. 2025). Given a start and goal location for each robot, our objective is to use pretrained diffusion models to generate 2D path trajectories that: (i) avoid collisions with static obstacles *and* any other robots, (ii) respect velocity limits specified *at test time*, and (iii) exhibit specific motion patterns dictated by the environment.

Projection. We use projection to *enforce* max velocity constraints on each robot. Denote the velocity limit as $v_{\max}$, the (physical) time step size as Δt, and trajectory horizon as H. The projection can be written as an optimization of which the feasible set is our constraint set $\mathcal{K}_X$:

$$\min_{X \in \mathbb{R}^{H \times 2}} \ \|X - \widehat{X}\|_F^2 \tag{6a}$$

$$\text{s.t.} \quad \|x_0 - X_1\| \le v_{\max} \Delta t, \tag{6b}$$

$$\|X_h - X_{h-1}\| \le v_{\max} \Delta t, \quad h = 2, \ldots, H, \tag{6c}$$

where $\widehat{X}$ is the diffusion-predicted trajectory for one robot in matrix form, $X_h \in \mathbb{R}^2$ is the position vectors at (physical) discrete time step h and $x_0 \in \mathbb{R}^2$ is a known starting position. This convex optimization problem can be efficiently solved in parallel using the Alternating Direction Method of Multipliers (Boyd and Vandenberghe 2004). Detailed derivations are in the appendix.

Coupling Cost. We aim to avoid both collisions among robots and collisions with known static obstacles via coupling. Prior work (Carvalho et al. 2023; Shaoul et al. 2025) achieves static obstacle avoidance by CG, which we have shown is a special case of PCD coupling. Thus, our coupling cost function is a linear combination of a robot-collision and obstacle-collision costs:

$$c(X, Y) = \lambda_{\text{robo}} c_{\text{robo}}(X, Y) + \lambda_{\text{obst}} c_{\text{obst}}(X, Y),$$

where $X, Y \in \mathbb{R}^{H \times 2}$ are trajectories of 2 robots' positions. We experiment with two robot collision cost functions, (i) a log-barrier (LB) cost

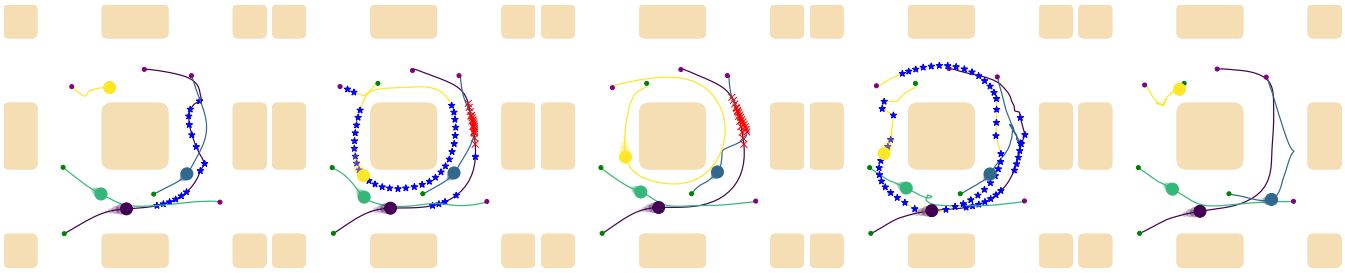
$$c_{\text{LB}}(X, Y) = -\sum\nolimits_{h=1}^{H} \log\left(\|X_h - Y_h\|_2 + \alpha\right) \tag{7}$$

where $\alpha > 0$ is a parameter, and (ii) a "squared hinge distance" (SHD):

$$c_{\text{SHD}}(X, Y) = \sum\nolimits_{h=1}^{H} \mathbf{1}[r_h \le \rho] \cdot (r_h - \rho)^2 \tag{8}$$

wherein $r_h = \|X_h - Y_h\|$ is the inter-robot distance, $\mathbf{1}[\cdot]$ is the indicator function and $\rho > 0$ is the active range parameter. For $N > 2$ robots, $X^1, \ldots, X^N$, we extend both costs to $c_\diamond(X^1, \ldots, X^N) = \sum_{1 \le i < j \le N} c_\diamond(X^i, X^j)$ with $\diamond \in \{\text{LB}, \text{SHD}\}$. We follow Carvalho et al. (2023) in designing the obstacle-avoidance cost. See details in the appendix.

Setup. We test with 2 and 4 robots on both `Highways` and `Empty` from Shaoul et al. (2025). For each task, we use the pretrained models from Shaoul et al. (2025) and choose three $v_{\max}$s. We compare our method with a vanilla diffusion model DIFFUSER (Janner et al. 2022), and MMD-CBS (Shaoul et al. 2025). We evaluate each method on 100 trials, each with an initial configuration (start and goal locations for each robot) sampled u.a.r. by rejection sampling. Except for MMD-CBS, we generate 128 i.i.d. samples ; for MMD-CBS, we also set its diffusion sampling batch size as 128. We run 25 diffusion inference steps for all methods.



(a) MMD-CBS ($N = 4$) (b) DIFFUSER ($N = 4$) (c) PD ($N = 4$) (d) CD (w/o proj., $N = 4$) (e) **PCD (Ours)** ($N = 4$)

Figure 2: Robot trajectories in `Highways` generated by the compared methods. Red crosses mark collisions; blue stars mark velocity constraint violations. The desired motion pattern is to circle the central obstacle *counterclockwise*. (a) MMD-CBS excels in collision avoidance but cannot guarantee velocity constraint satisfaction. (b) Vanilla DIFFUSER *fails* to generate both collision-free and velocity constraint-compliant trajectories. (c) Projection only enforces velocity constraint but ignores collision avoidance. (d) A coupling cost facilitates inter-robot collision avoidance but cannot guarantee velocity constraint. (e) Our method can effectively generate collision-free trajectories *and enforce* the velocity constraint.

	Task `Empty`, 4 Robots, Max Vel. **0.703**				**Task `Highways`, 4** Robots, Max Vel. **0.878**			
METHOD \Metric	SU(%)↑	RS↑	*minCS(%)↑	*minDA↑	SU(%)↑	RS↑	*minCS(%)↑	*minDA↑
DIFFUSER	65.0	0.616	62.3	**0.990**	53.0	0.208	66.7	0.979
MMD-CBS	**100**	**1.00**	11.0	**0.990**	**100**	**1.00**	63.0	0.960
DIFFUSER + projection	65.0	0.615	**100**	**0.990**	54.0	0.214	**100**	0.978
CD-LB (w/o proj.)	**100**	0.993	0.0312	0.814	**100**	0.999	0.00	**0.986**
CD-SHD (w/o proj.)	**100**	**1.00**	35.3	**0.990**	**100**	**1.00**	34.6	0.976
PCD-LB	96.0	0.916	100	0.489	**100**	0.950	100	0.957
PCD-SHD	**100**	0.993	**100**	0.960	**100**	0.996	**100**	0.963

Table 1: Performance comparison with $N = 4$ robots on both tasks. Left block: `Empty`, with constraint $v_{\max} = 0.703$; right block: `Highway`, with $v_{\max} = 0.878$. *For CS and DA, which should have been 4-tuples, we report here the *minimum of the four* due to space limit; the full version is in the appendix.

Evaluation Metrics. We evaluate performance of the methods in terms of task completion or adherence to original data distribution, constraint satisfaction, and inter-robot collision avoidance. We adopt *success rate* (SU) and *data adherence* (DA) to evaluate task completion from (Shaoul et al. 2025). SU is the average, over all initial configurations, of an indicator for whether at least one trajectory in the batch completes the task without collision. *Constraint satisfaction* (CS) is an indicator of whether a trajectory satisfies the velocity constraint at all time steps. *Inter-robot safety* (RS) is an indicator of whether a trajectory tuple is inter-robot collision-free. All metrics except SU are reported as empirical means over a batch of i.i.d. samples.

Results. Figure 2 shows sample trajectories from compared methods. Table 1 summarizes quantitative results for both environments with 4 robots under one velocity constraint. All constraint-agnostic methods (without projection) achieve low CS rates. MMD-CBS unsurprisingly achieves perfect SU score by selecting and stitching a single optimal trajectory tuple. PCD- and CD- methods approach this upper bound, while vanilla DIFFUSER and its projected variant lag behind. As before, our method shows slightly reduced data adherence (PCD-SHD), while the LB cost aggressively trades data adherence for collision avoidance due to its steep gradients. More results are in the appendix. Overall, PCD effectively promotes inter-robot collision avoidance through coupling while enforcing hard test-time velocity constraints, with a tradeoff between coupling strength and data adherence depending on the cost function.

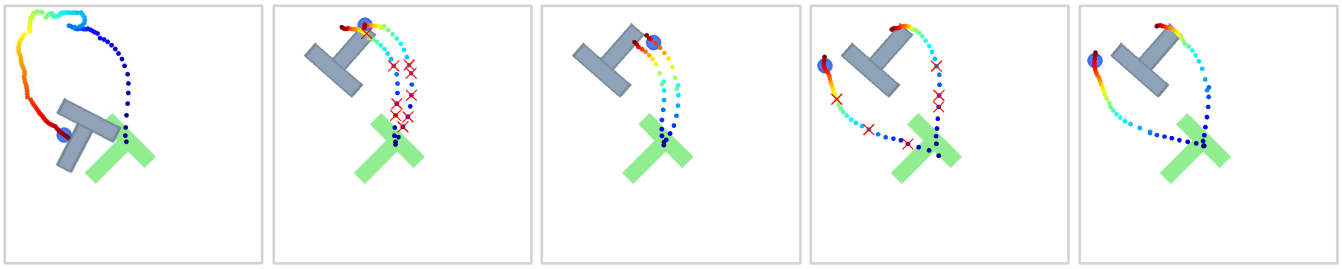
5.2 Constrained and Diverse Robot Manipulation

We evaluate our method on the `PushT` task (Florence et al. 2022; Chi et al. 2023). As shown in Figure 3a, a diffusion model is trained to generate trajectories for a robot to push the gray `T`-shaped block till it matches the green target position from different starting locations. Our objective is to utilize such pretrained models to generate a *pair* of distinct trajectories strictly satisfying a *maximum velocity constraint* imposed *at test time* and do not intersect as far as possible[3].

Projection. As in multi-robot experiment, we enforce velocity limits via projection, using the formulation in Eq. (6).

Coupling Cost. We experimented with two cost functions for encouraging trajectories from a pair to stay away from each other. The first cost builds upon the Determinantal

[3]The two trajectories in a pair *are not* pushing the block together at the same time.



(a) `PushT` Task (b) DP (c) DP + projection (d) CD (w/o projection) (e) **PCD (Ours)**

Figure 3: (a) The `PushT` task: A robot (blue circle) pushes a `T` block (gray) to a target pose (green) given different initial positions of the block and robot. Robot trajectories use a colormap (warmer colors indicate later time steps), with red crosses marking velocity violations. Only the first few dozen steps are shown for clarity. (b) Vanilla DP *fails* fails to generate trajectory pairs, each in a distinct mode, and adhering to velocity limits. (c) Projection enforces velocity limits but cannot "split" trajectories apart. (d) A coupling cost encourages non-intersecting trajectories but does not strictly enforce velocity constraints. (e) Our method generates non-intersecting trajectories *and* strictly enforces velocity constraints.

METHOD	DTW↑	DFD↑	CS(%)↑	TC↑
DP	3.16	.465	(64.8, 65.0)	(**.927**, **.931**)
DP + proj.	2.96	.428	(**100**, **100**)	(.896, .888)
CD-DPP	3.74	.544	(62.5, 62.4)	(.923, .922)
CD-DPP-PS	4.48	.648	(57.4, 57.3)	(.912, .917)
CD-LB	4.13	.596	(58.9, 58.6)	(.910, .912)
CD-LB-PS	4.50	.646	(58.8, 58.7)	(.921, .925)
PCD-DPP	4.55	.638	(**100**, **100**)	(.829, .834)
PCD-DPP-PS	4.39	.622	(**100**, **100**)	(.885, .885)
PCD-LB	**5.12**	**.708**	(**100**, **100**)	(.778, .791)
PCD-LB-PS	4.38	.618	(**100**, **100**)	(.890, .882)

Table 2: Results of `PushT` task by all compared methods with velocity limit $v_{\max} = 8.4$. CD denotes DP+coupling only; -PS denotes posterior sampling variant. Full results are in the appendix.

Point Process (DPP) guidance (Feng et al. 2025) which was designed to promote trajectory diversity:

$$c_{\text{DPP}}(X, Y) = -\log\left(\cos\angle(\tilde{X}, \tilde{Y}) + \varepsilon\right) \tag{9}$$

where $\tilde{X}, \tilde{Y} \in \mathbb{R}^{2H}$ are the flattened vectors of the trajectories, and $\varepsilon > 0$ is a small constant. The other cost is the log-barrier cost in Eq. (7). For both costs, we also devise their corresponding posterior sampling variants.

Setup. We adopt DIFFUSION POLICY (DP) of Chi et al. (2023) as our base algorithm, using pretrained weights from Feng et al. (2024)[4]. Compared methods are vanilla DP, DP with only projection, DP with only coupling and PCD. We evaluate each method on 50 uniformly random initial conditions. With each method, we generate 100 *pairs* of full trajectories, under three different max velocity limits. We use 32 diffusion steps at inference, with other settings recommended by Chi et al. (2023). Details are in the appendix.

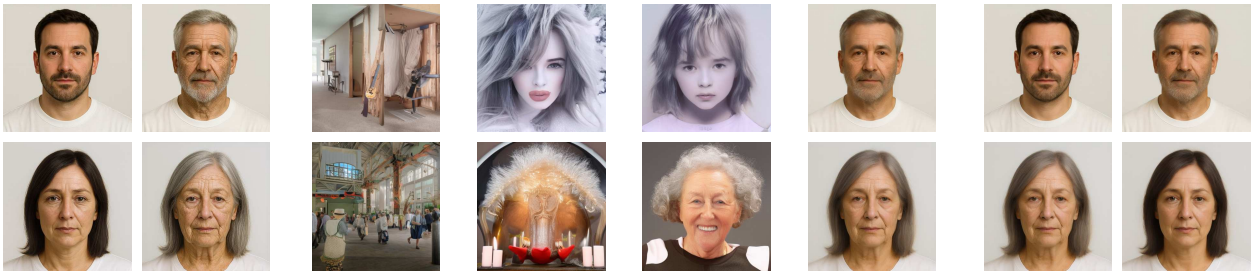
Evaluation Metrics. We use four quantitative metrics for evaluation: Dynamic Time Warping (DTW) (Berndt and Clifford 1994; Müller 2007), discrete Fréchet distance (DFD) (Alt and Godau 1995), velocity constraint satisfaction rate (CS), and task completion score (TC) (Florence et al. 2022; Chi et al. 2023). DTW and DFD quantify dissimilarity between two trajectories. CS evaluates fraction of trajectories satisfying the velocity constraint. TC measures how well the block-pushing task is accomplished, where 1.0 is the best and 0 the worst. Details regarding the metrics are in appendix. We report all metrics by their empirical means over all initial staring locations and trajectory pairs.

Results. We report results in Figure 3b–3e and Table 2. From Table 2, we see all projection-involved methods achieve perfect velocity constraint satisfaction, outperforming both the baseline and coupling-only (CD) approaches. CD and our PCD methods consistently produce higher DTW and DFD than baseline with or without projection, suggesting that coupling effectively discourages intersecting trajectories. In terms of task completion, all methods except the baseline show degraded performance, likely due to the velocity limit enforced by projection. These results show that our framework can enforce test-time velocity constraints and spatially separate generated trajectories without significantly sacrificing data adherence. Without projection, both PS variants of the coupling costs exhibit higher DTW and DFD than their respective non-PS version, and better preserves the task completion with projection. Further results including ablation study are in the appendix.

5.3 Constrained Coupled Image Pair Generation

We demonstrate a toy example of paired face generation using two latent diffusion models (LDMs) (Rombach et al. 2022) (Figure 4). Each generated pair must (i) satisfy *gender and facial attribute constraints*, and (ii) exhibit a clear *contrast between age groups*. We enforce (i) via projection and (ii) through a classifier-driven coupling loss.

[4]The model from (Feng et al. 2024) was trained on an augmented dataset with broader coverage than that of (Chi et al. 2023), yielding more diverse and feasible trajectories.



(a) Exemplar Images (b) SD (c) SD+C (d) SD+C† (e) SD+P (f) **PCD (Ours)**

Figure 4: Paired face generation with two Stable Diffusion models (SD) (row-wise). Methods: +C = coupling; +P = projection; † = with text prompt. (a) Exemplar images of each model, their latents form the convex hulls. (b) Vanilla SD often fails to produce faces. (c) Coupling alone steers samples toward the target *age-group contrast* but with attribute drift. (d) Adding text prompts yields faces, yet violates attribute constraints. (e) Projection alone enforces gender and facial attributes but not the *age-group contrast*. (f) Our method yields pairs that satisfy the *age-group contrast* plus the gender and facial-attribute constraints.

Projection. For each LDM, we generate two exemplar images of one individual at different ages (Figure 4a) using ChatGPT (OpenAI 2023), encode them via the model's VAE, and form convex hulls as feasible latent regions. At each diffusion step, we use mirror descent (Nemirovsky and Yudin 1983; Beck and Teboulle 2003) to project intermediate latents onto these hulls, enforcing strict *gender and facial attribute constraints* (see appendix for details).

Coupling Cost To induce an *age–group contrast*, we use a latent classifier that classifies age groups Y (young, < 50) and O (old, ≥ 50). Our coupling loss is $c_{\text{XOR}}(x, y) = -\sum_{a\in\{\text{Y},\text{O}\}} \text{XOR}(x, y)$, where $\text{XOR}(x, y) = p(a|x)\big(1 - p(a|y)\big) + p(a|y)\big(1 - p(a|x)\big)$ and $p(a|\cdot)$ are the probabilities of a sample belonging to age-class $a \in \{\text{Y}, \text{O}\}$, obtained from the classifier.

Stable Diffusion. We use STABLE DIFFUSION V2.1-BASE (SD-2.1) (Stability AI 2022), and disable classifier-free guidance during sampling— effectively yielding unconditional generation unless otherwise specified. For coupling, we train a latent classifier for the model using the FFHQ-Aging Dataset (Or-El et al. 2020).

Setup. We compare our method with vanilla SD-2.1, SD-2.1 with only coupling, and SD-2.1 with only projection. We also run an additional comparison against SD-2.1 with coupling plus the use of a generic text prompt (details in appendix). We use 100 DDPM diffusion steps and generate 25 *pairs* of samples using each method.

Evaluation Metrics. To evaluate projection, we report four metrics: (i) *Gender constraint satisfaction rate* (M/F); (ii) *Sample-Exemplar CLIP similarity* (SE-CLIP) (Radford et al. 2021); (iii) *Sample-Exemplar LPIPS* (SE-LPIPS) (Zhang et al. 2018); and (iv) *Intra-Sample LPIPS* (IS-LPIPS). SE-CLIP and SE-LPIPS serve as proxies for adherence to exemplar-specified facial attribute constraints (noting that satisfaction is guaranteed by design of our projection operator), while IS-LPIPS quantifies diversity across generated samples. To evaluate coupling, we evaluate the *age-group contrast satisfaction rate* (XOR) using an age-group image classifier trained on the FFHQ-Aging Dataset. We average XOR and M/F over generated pairs, SE-CLIP and SE-LPIPS over sample–exemplar pairs, IS-LPIPS over intra-model sample pairs.

METHOD	XOR%↑	M/F%↑	SE-C*↑	SE-L*↓	IS-L*↑
SD	20	71/14	.40/.41	.77/.76	**.75**/**.75**
SD+C	48	51/47	.45/.46	.76/.74	**.75**/**.75**
SD+C†	64	47/37	.55/.60	.70/.68	.69/.69
SD+P	44	**100**/**100**	**.88**/.91	.15/.16	.06/.09
PCD	**96**	**100**/**100**	**.88**/**.92**	**.11**/**.14**	.11/.13

Table 3: Paired face-generation results. Metrics: SE-C* = SE-CLIP; SE-L* = SE-LPIPS; IS-L* = IS-LPIPS. Boldface indicates the best score(s) for each metric.

Results. We report results in Figures 4b–4f and Table 3. Projection-based methods (SD+P, PCD) achieve 100% gender satisfaction (M/F) and the strongest alignment to exemplars (higher SE-CLIP and lower SE-LPIPS) compared to vanilla SD and coupling-only variants. Projection reduces diversity (low IS-LPIPS); adding coupling partially recovers diversity relative to projection alone. Coupling-based methods improve the age-group XOR satisfaction, with PCD attaining the highest rate (96%). See appendix for additional qualitative results with larger exemplar sets and ablation.

Runtime and Memory. PCD is about 5x slower than vanilla diffusion due to per-step projection. We used a fixed but enough number of optimizer iterations without finetuning; convergence checks may decrease runtime. Memory overhead compared to vanilla diffusion is negligible.

6 Conclusion

We introduced Projected Coupled Diffusion (PCD), a test-time framework for joint generation with multiple diffusion models under hard constraints. Our method combines coupled dynamics and projection operation, generalizing exist-

ing techniques like classifier guidance and projection-based diffusion inference without requiring model retraining. Experiments on image-pair generation, object manipulation, and multi-robot motion planning show that PCD achieves better coupling and guaranteed constraint satisfaction. Future work includes exploring more sophisticated cost models and non-convex constraints.

References

Alt, H.; and Godau, M. 1995. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02): 75–91.

Arvanitidis, G.; Hansen, L. K.; and Hauberg, S. 2018. Latent Space Oddity: On the Curvature of Deep Generative Models. In *International Conference on Learning Representations (ICLR)*.

Bar-Tal, O.; Yariv, L.; Lipman, Y.; and Dekel, T. 2023. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 1737–1752.

Beck, A.; and Teboulle, M. 2003. Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization. *Operations Research Letters*, 31(3): 167–175.

Berndt, D. J.; and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 359–370. AAAI Press.

Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Bubeck, S.; Eldan, R.; and Lehec, J. 2015. Finite-time analysis of projected Langevin Monte Carlo. *Advances in Neural Information Processing Systems*, 28.

Bucker, A.; Figueredo, L.; Haddadin, S.; Kapoor, A.; Ma, S.; Vemprala, S.; and Bonatti, R. 2023. LATTE: LAnguage Trajectory TransformEr. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 7287–7294.

Cao, J.; Zhang, Q.; Guo, H.; Wang, J.; Cheng, H.; and Xu, R. 2025. Modality-Composable Diffusion Policy via Inference-Time Distribution-level Composition. *arXiv preprint arXiv:2503.12466*.

Carvalho, J.; Le, A. T.; Baierl, M.; Koert, D.; and Peters, J. 2023. Motion Planning Diffusion: Learning and Planning of Robot Motions with Diffusion Models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1916–1923.

Chamon, L. F.; Karimi, M. R.; and Korba, A. 2024. Constrained sampling with primal-dual Langevin Monte Carlo. *Advances in Neural Information Processing Systems*, 37: 29285–29323.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 02783649241273668.

Christopher, J. K.; Baek, S.; and Fioretto, N. 2024. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37: 89307–89333.

Chung, H.; Kim, J.; Mccann, M. T.; Klasky, M. L.; and Ye, J. C. 2023. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *The Eleventh International Conference on Learning Representations*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 8780–8794.

Du, Y.; and Kaelbling, L. 2024. Compositional Generative Modeling: A Single Model Is Not All You Need. arXiv:2402.01103.

Du, Y.; Li, S.; and Mordatch, I. 2020. Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems*, volume 33, 6637–6647.

Efron, B. 2011. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496): 1602–1614.

Feng, Z.; Luan, H.; Goyal, P.; and Soh, H. 2024. LTLDoG: Satisfying Temporally-Extended Symbolic Constraints for Safe Diffusion-Based Planning. *IEEE Robotics and Automation Letters*, 9(10): 8571–8578.

Feng, Z.; Luan, H.; Ma, K. Y.; and Soh, H. 2025. Diffusion Meets Options: Hierarchical Generative Skill Composition for Temporally-Extended Tasks. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*.

Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O. A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; and Tompson, J. 2022. Implicit Behavioral Cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, 158–168. PMLR.

Gu, J.; Zhai, S.; Zhang, Y.; Susskind, J.; and Jaitly, N. 2024. Matryoshka Diffusion Models. In *ICLR*.

Guo, Y.; Yuan, H.; Yang, Y.; Chen, M.; and Wang, M. 2024. Gradient guidance for diffusion models: An optimization perspective. *Advances in Neural Information Processing Systems*, 37: 90736–90770.

Hayakawa, A.; Ishii, M.; Shibuya, T.; and Mitsufuji, Y. 2025. MMDisCo: Multi-Modal Discriminator-Guided Cooperative Diffusion for Joint Audio and Video Generation. In *The Thirteenth International Conference on Learning Representations*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6840–6851. Curran Associates, Inc.

Ho, J.; and Salimans, T. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in neural information processing systems*, 35: 8633–8646.

Janner, M.; Du, Y.; Tenenbaum, J. B.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*.

Jiang, C.; Cornman, A.; Park, C.; Sapp, B.; Zhou, Y.; Anguelov, D.; et al. 2023. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9644–9653.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577.

Kim, S.; Kim, M.; and Park, D. 2025. Test-time Alignment of Diffusion Models without Reward Over-optimization. In *The Thirteenth International Conference on Learning Representations*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. ArXiv:1312.6114.

Kondo, K.; Tagliabue, A.; Cai, X.; Tewari, C.; Garcia, O.; Espitia-Alvarez, M.; and How, J. P. 2024. CGD: Constraint-Guided Diffusion Policies for UAV Trajectory Planning. *arXiv preprint arXiv:2405.01758*.

Lee, K. M.; Ye, S.; Xiao, Q.; Wu, Z.; Zaidi, Z.; D'Ambrosio, D. B.; Sanketi, P. R.; and Gombolay, M. 2025. Learning Diverse Robot Striking Motions with Diffusion Models and Kinematically Constrained Gradient Guidance. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*.

Lee, Y.; Kim, K.; Kim, H.; and Sung, M. 2023. SyncDiffusion: Coherent Montage via Synchronized Joint Diffusions. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Li, Q.; Peng, Z. M.; Feng, L.; Liu, Z.; Duan, C.; Mo, W.; and Zhou, B. 2023. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36: 3894–3920.

Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-LM improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343.

Liang, J.; Christopher, J. K.; Koenig, S.; and Fioretto, F. 2025. Simultaneous Multi-Robot Motion Planning with Projected Diffusion Models. In *Forty-second International Conference on Machine Learning*.

Liu, N.; Li, S.; Du, Y.; Torralba, A.; and Tenenbaum, J. B. 2022. Compositional Visual Generation with Composable Diffusion Models. In *Computer Vision – ECCV 2022*, volume 13677, 423–439. Cham: Springer Nature Switzerland. ISBN 978-3-031-19789-5 978-3-031-19790-1.

Liu, X.; Park, D. H.; Azadi, S.; Zhang, G.; Chopikyan, A.; Hu, Y.; Shi, H.; Rohrbach, A.; and Darrell, T. 2023. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Luan, H.; Ng, S.-K.; and Ling, C. K. 2025. DDPS: Discrete Diffusion Posterior Sampling for Paths in Layered Graphs. In *ICLR 2025 Workshop on Frontiers in Probabilistic Inference: Learning meets Sampling*.

Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Van Gool, L. 2022. RePaint: Inpainting Using Denoising Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11461–11471.

Luo, C. 2022. Understanding Diffusion Models: A Unified Perspective. Technical report, Google Research / Brown University.

Madeira, M.; Vignac, C.; Thanou, D.; and Frossard, P. 2024. Generative Modelling of Structurally Constrained Graphs. In *Advances in Neural Information Processing Systems*, volume 37, 137218–137262.

Memmel, M.; Berg, J.; Chen, B.; Gupta, A.; and Francis, J. 2025. STRAP: Robot Sub-Trajectory Retrieval for Augmented Policy Learning. In *The Thirteenth International Conference on Learning Representations*.

Meng, Y.; and Fan, C. 2024. Diverse Controllable Diffusion Policy With Signal Temporal Logic. *IEEE Robotics and Automation Letters*, 9(10): 8354–8361.

Müller, M. 2007. Dynamic time warping. *Information retrieval for music and motion*, 69–84.

Nemirovsky, A. S.; and Yudin, D. B. 1983. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics. Chichester, UK: Wiley-Interscience. ISBN 0471103454. Translated by E. R. Dawson.

Nichol, A.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In *ICML*.

Niu, C.; Song, Y.; Song, J.; Zhao, S.; Grover, A.; and Ermon, S. 2020. Permutation invariant graph generation via score-based generative modeling. In *International conference on artificial intelligence and statistics*, 4474–4484. PMLR.

OpenAI. 2023. ChatGPT: Optimizing Language Models for Dialogue. https://chat.openai.com. Accessed: 2025-07-29.

Or-El, R.; Sengupta, S.; Fried, O.; Shechtman, E.; and Kemelmacher-Shlizerman, I. 2020. Lifespan Age Transformation Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Radford, A.; Kim, J. W.; Hallacy, L.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Rana, M. A.; Li, A.; Fox, D.; Boots, B.; Ramos, F.; and Ratliff, N. 2020. Euclideanizing Flows: Diffeomorphic Reduction for Learning Stable Dynamical Systems. In *Pro-*

ceedings of the 2nd Conference on Learning for Dynamics and Control, volume 120, 630–639. PMLR.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32 of *Proceedings of Machine Learning Research*, 1278–1286. PMLR.

Roberts, G. O.; and Tweedie, R. L. 1996. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4): 341 – 363.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ruan, L.; Ma, Y.; Yang, H.; He, H.; Liu, B.; Fu, J.; Yuan, N. J.; Jin, Q.; and Guo, B. 2023. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10219–10228.

Shaoul, Y.; Mishani, I.; Vats, S.; Li, J.; and Likhachev, M. 2025. Multi-Robot Motion Planning with Diffusion Models. In *The Thirteenth International Conference on Learning Representations*.

Sharma, K.; Kumar, S.; and Trivedi, R. 2024. Diffuse, Sample, Project: Plug-And-Play Controllable Graph Generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 44545–44564.

Shoemake, K. 1985. Animating Rotation with Quaternion Curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*, 245–254. ACM.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Stability AI. 2022. Stable Diffusion v2.1 Base — Model Card. https://huggingface.co/stabilityai/stable-diffusion-2-1-base. Hugging Face model card. Contributors: Robin Rombach, Patrick Esser, David Ha. Accessed: 2025-07-29.

Tang, Z.; Yang, Z.; Zhu, C.; Zeng, M.; and Bansal, M. 2023. Any-to-Any Generation via Composable Diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Uehara, M.; Zhao, Y.; Wang, C.; Li, X.; Regev, A.; Levine, S.; and Biancalani, T. 2025. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*.

von Platen, P.; Patil, S.; Lozhkov, A.; Cuenca, P.; Lambert, N.; Rasul, K.; Davaadorj, M.; Nair, D.; Paul, S.; Berman, W.; Xu, Y.; Liu, S.; and Wolf, T. 2022. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Wang, L.; Zhao, J.; Du, Y.; Adelson, E.; and Tedrake, R. 2024a. PoCo: Policy Composition from and for Heterogeneous Robot Learning. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation.

Wang, Y.; Tang, C.; Sun, L.; Rossi, S.; Xie, Y.; Peng, C.; Hannagan, T.; Sabatini, S.; Poerio, N.; Tomizuka, M.; et al. 2024b. Optimizing diffusion models for joint trajectory prediction and controllable generation. In *European Conference on Computer Vision*, 324–341. Springer.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. Citeseer.

Xing, Y.; He, Y.; Tian, Z.; Wang, X.; and Chen, Q. 2024. Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7151–7161.

Xu, Y.; Mao, J.; Du, Y.; Lozano-Pérez, T.; Kaelbling, L. P.; and Hsu, D. 2024. Set It Up!: Functional Object Arrangement with Compositional Generative Models. In *Proceedings of Robotics: Science and Systems*. Delft, Netherlands.

Yang, Z.; Mao, J.; Du, Y.; Wu, J.; Tenenbaum, J. B.; Lozano-Pérez, T.; and Kaelbling, L. P. 2023. Compositional Diffusion-Based Continuous Constraint Solvers. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, 3242–3265.

Zampini, S.; Christopher, J.; Oneto, L.; Anguita, D.; and Fioretto, F. 2025. Training-Free Constrained Generation With Stable Diffusion Models. *arXiv preprint arXiv:2502.05625*.

Zeng, Y.; Patel, V. M.; Wang, H.; Huang, X.; Wang, T.-C.; Liu, M.-Y.; and Balaji, Y. 2024. Jedi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6786–6795.

Zhang, J.; Mohammadi, H. B.; and Rozo, L. 2022. Learning Riemannian Stable Dynamical Systems via Diffeomorphisms. In *6th Annual Conference on Robot Learning*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–595.

A Method Details

Our proposed PCD framework can perform inference with LMC and Denoising Diffusion Probablistic Models (DDPM) (Ho, Jain, and Abbeel 2020). Moreover, it is also easy to apply Diffusion Posterior Sampling (DPS) (Chung et al. 2023) within our framework. We present here three algorithms under PCD framework: PCD-LMC, PCD-DDPM, and PCD-DPS, detailed in Algorithm 1, Algorithm 2, and Algorithm 3, respectively.

Algorithm 1: Projected Coupled Diffusion with LMC

Require: Score models s_X^θ, s_Y^ϕ; projectors $\Pi_{\mathcal{K}_X}$, $\Pi_{\mathcal{K}_Y}$; coupling strength γ; LMC step size δ; max iteration T.

$X_0 \sim \mathcal{N}(0, \boldsymbol{I}_{D_x})$, $Y_0 \sim \mathcal{N}(0, \boldsymbol{I}_{D_y})$ ▷ *Initialize from std. Gaussian*

for $t = 1$ to $T - 1$ **do**

  ▷ *Coupled LMC dynamics* ◁

  $X_{t+1} \leftarrow X_{t+1} - \delta s_X^\theta(X, t) - \gamma\delta\nabla_X c(X_t, Y_t)$

  $Y_{t+1} \leftarrow Y_{t+1} - \delta s_Y^\phi(Y, t) - \gamma\delta\nabla_Y c(X_t, Y_t)$

  $\epsilon_X \sim \mathcal{N}(0, \boldsymbol{I}_{D_x})$, $\epsilon_Y \sim \mathcal{N}(0, \boldsymbol{I}_{D_y})$ ▷ *i.i.d. noise*

  $X_{t+1} \leftarrow X_{t+1} + \sqrt{2\delta}\epsilon_X$

  $Y_{t+1} \leftarrow X_{t+1} + \sqrt{2\delta}\epsilon_Y$

  ▷ *Projection step* ◁

  $X_{t+1} \leftarrow \Pi_{\mathcal{K}_X}(X_{t+1})$

  $Y_{t+1} \leftarrow \Pi_{\mathcal{K}_Y}(Y_{t+1})$

return (X_T, Y_T) ▷ *Return joint samples*

Algorithm 2: Projected Coupled Diffusion with DDPM

Require: Score models s_X^θ, s_Y^ϕ; projectors $\Pi_{\mathcal{K}_X}$, $\Pi_{\mathcal{K}_Y}$; coupling strength γ; DDPM noise schedule $\{\alpha_t\}_{t=1}^T$; DDPM inference step T.

$X_T \sim \mathcal{N}(0, \boldsymbol{I}_{D_x})$, $Y_T \sim \mathcal{N}(0, \boldsymbol{I}_{D_y})$ ▷ *Initialize from std. Gaussian*

for $t = T$ to 1 **do**

  ▷ *Normal diffusion* ◁

  $\epsilon_X \sim \mathcal{N}(0, \boldsymbol{I}_{D_x})$, $\epsilon_Y \sim \mathcal{N}(0, \boldsymbol{I}_{D_y})$

  $s_X \leftarrow s_X^\theta(X_t, t)$, $s_Y \leftarrow s_Y^\phi(Y_t, t)$

  $X_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(X_t + (1 - \alpha_t)s_X) + \sqrt{1 - \alpha_t}\epsilon_X$

  $Y_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(Y_t + (1 - \alpha_t)s_Y) + \sqrt{1 - \alpha_t}\epsilon_Y$

  ▷ *Coupling step* ◁

  $X_{t-1} \leftarrow X_{t-1} - \gamma\nabla_{X_t} c(X_t, Y_t)$

  $Y_{t-1} \leftarrow Y_{t-1} - \gamma\nabla_{Y_t} c(X_t, Y_t)$

  ▷ *Projection step* ◁

  $X_{t-1} \leftarrow \Pi_{\mathcal{K}_X}(X_{t-1})$

  $Y_{t-1} \leftarrow \Pi_{\mathcal{K}_Y}(Y_{t-1})$

return (X_0, Y_0) ▷ *Return joint samples*

B Implementation Details

B.1 General

Computational Hardware. All experiments were run on a workstation with 1 AMD Ryzen Threadripper PRO 5995WX 64-Core CPU, 504 GB RAM, and 2 NVIDIA RTX A6000 GPUs each with 48GB GPU memory.

Software and Code Bases All experiments were run using PyTorch (Paszke et al. 2019). Image experiments were also run with Diffusers (von Platen et al. 2022). The `PushT` experiment builds upon LTLDoG (Feng et al. 2024) and DIFFUSION POLICY (Chi et al. 2023). The multi-robot experiment builds upon MMD (Shaoul et al. 2025).

B.2 Constrained Coupled Image Pair Generation Experiment

Text Prompt for Stable Diffusion As shown in both Figure 4d and Table 3, we also run an additional comparison against SD-2.1 with coupling plus the use of a generic text prompt. For all text-prompted runs, we use *"High-resolution passport photo of a person, facing forward with a neutral expression. Wearing a plain white t-shirt, with a clean white background and even,*

Algorithm 3: Projected Coupled Diffusion with DPS

Require: Score models s_X^θ, s_Y^ϕ; projectors $\mathrm{Proj}_{\mathcal{K}_X}$, $\mathrm{Proj}_{\mathcal{K}_Y}$; coupling strength γ; DDPM noise schedule $\{\alpha_t\}_{t=1}^T$; DDPM inference step T.

Pre-compute $\bar{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau$ for $t = 1, \dots, T$
$X_T, Y_T \sim \mathcal{N}(0, \boldsymbol{I})$ ▷ *Initialize from std. Gaussian*
for $t = T$ to 1 **do**
  ▷ *Normal diffusion* ◁
  $\epsilon_X \sim \mathcal{N}(0, \boldsymbol{I}_{D_x})$, $\epsilon_Y \sim \mathcal{N}(0, \boldsymbol{I}_{D_y})$
  $s_X \leftarrow s_\theta(X_t, t)$, $s_Y \leftarrow s_\phi(Y_t, t)$
  $X_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} (X_t + (1-\alpha_t) s_X) + \sqrt{1-\alpha_t}\epsilon_X$
  $Y_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} (Y_t + (1-\alpha_t) s_Y) + \sqrt{1-\alpha_t}\epsilon_Y$
  ▷ *Coupling with posterior sampling* ◁
  $\hat{X}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} (X_t + (1-\bar{\alpha}_t) s_X)$ ▷ *Tweedie's formula*
  $\hat{Y}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} (Y_t + (1-\bar{\alpha}_t) s_Y)$
  $X_{t-1} \leftarrow X_{t-1} - \gamma \nabla_{X_t} c(\hat{X}_0, \hat{Y}_0)$
  $Y_{t-1} \leftarrow Y_{t-1} - \gamma \nabla_{Y_t} c(\hat{X}_0, \hat{Y}_0)$
  ▷ *Projection step* ◁
  $X_{t-1} \leftarrow \Pi_{\mathcal{K}_X}(X_{t-1})$
  $Y_{t-1} \leftarrow \Pi_{\mathcal{K}_Y}(Y_{t-1})$
return (X_0, Y_0) ▷ *Return joint samples*

soft lighting. The composition is centered and symmetrical, with the head at the center of the frame." and set classifier-free guidance scale s=25.

Projection via Mirror Descent Given two exemplar sets of sizes M_x and M_y, we encode them via VAE encoders and obtain the latents $X^{(e)} = [\, X_1^{(e)} \cdots X_{M_x}^{(e)} \,] \in \mathbb{R}^{d \times M_x}$ and $Y^{(e)} = [\, Y_1^{(e)} \cdots Y_{M_y}^{(e)} \,] \in \mathbb{R}^{d \times M_y}$, where d is the flattened latent dimension. Define $\mathcal{K}_X = \{\, X^{(e)}\lambda \mid \lambda \in \Delta_{M_x} \}$ the constraint set for X, where Δ_{M_x} is an M_x-simplex, and define $\mathcal{K}_Y$ likewise. At each diffusion step t, we project the current latent X_t onto $\mathcal{K}_X$ by solving the simplex-constrained problem:

$$\lambda_{X,t}^\star = \underset{\lambda \in \Delta_{M_x}}{\arg\min} \, \|X^{(e)}\lambda - X_t\|_2^2 \tag{10}$$

via Mirror Descent (MD) using the negative-entropy mirror map, which yields exponentiated-gradient updates that remain on Δ_{M_x} by construction (Nemirovsky and Yudin 1983; Beck and Teboulle 2003). For the MD updates, define

$$G_X := {X^{(e)}}^\top X^{(e)} \in \mathbb{R}^{M_x \times M_x}, \tag{11a}$$

$$b_{X,t} := {X^{(e)}}^\top X_t \in \mathbb{R}^{M_x}. \tag{11b}$$

and let $f_{X,t}(\lambda) = \|X^{(e)}\lambda - X_t\|_2^2$. Its gradient is

$$\nabla f_{X,t}(\lambda) = 2\,(G_X\lambda - b_{X,t}). \tag{12}$$

Starting from $\lambda_{X,t}^{(0)} = \frac{1}{M_x}\mathbf{1}$, each MD step performs

$$\log \lambda_{X,t}^{(k+1)} = \log \lambda_{X,t}^{(k)} - \eta\, \nabla f_{X,t}(\lambda_{X,t}^{(k)}), \tag{13a}$$

$$\lambda_{X,t}^{(k+1)} = \mathrm{softmax}\big(\log \lambda_{X,t}^{(k+1)}\big), \tag{13b}$$

with learning rate $\eta > 0$, which is equivalent to

$$\lambda_{X,t}^{(k+1)} \propto \lambda_{X,t}^{(k)} \odot \exp\left\{-\eta\, \nabla f_{X,t}\left(\lambda_{X,t}^{(k)}\right)\right\} \tag{14}$$

followed by normalization. After $K_{\max}$ steps, we obtain $\lambda_{X,t}^\star$ and compute the final projected latent $\hat{X}_t = X^{(e)}\lambda_{X,t}^\star$. $\hat{Y}_t$ is computed likewise. We run MD for $K_{\max} = 10,000$ steps and set its learning rate $\eta = 10^{-5}$ for convergence.

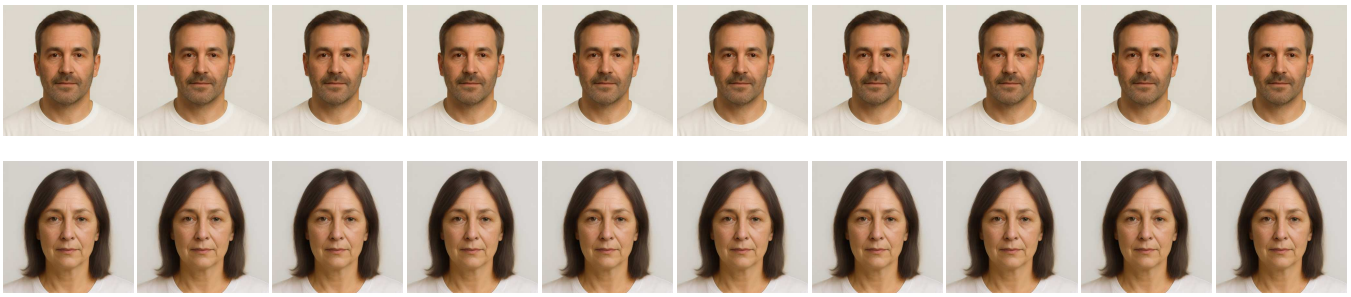


Figure 5: Generated samples obtained by projecting intermediate DDPM latents (Eq. (18)) onto the exemplar convex hulls at every step. Top: projection using the exemplar pair from the *top* row of Figure 4a; bottom: using the *bottom*-row pair. Within each row, the samples collapse to a narrow mode, illustrating significantly reduced diversity induced by per-step projection onto a fixed exemplar set.

Mode Collapse. We follow the denoising diffusion probabilistic model (DDPM) formulation of Ho, Jain, and Abbeel (2020). Let $\{\beta_t\}_{t=1}^{T} \subset (0,1)$ be a predefined noise-variance schedule for the forward process $q(z_t \mid z_{t-1}) = \mathcal{N}\big(\sqrt{1-\beta_t}\, z_{t-1},\, \beta_t I\big)$. Define

$$\alpha_t := 1 - \beta_t, \tag{15}$$

$$\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s \quad \text{(cumulative product)}, \tag{16}$$

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t \quad \text{(posterior variance)}. \tag{17}$$

At inference, the standard DDPM sampling update with a learned noise predictor $\varepsilon_\theta(x_t, t)$ is

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(z_t,t)\right) + \sqrt{\tilde{\beta}_t}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,I) \text{ if } t>1, \text{ else } 0. \tag{18}$$

Recall $\mathcal{K}_X = \{\, X^{(e)}\lambda : \lambda \in \Delta_{M_x} \,\}$ (and $\mathcal{K}_Y$ analogously for Y). When sampling with Eq. (18), we observe that projecting the latents at every step — i.e., $\hat{X}_t = \Pi_{\mathcal{K}_X}(X_t)$ and $\hat{Y}_t = \Pi_{\mathcal{K}_Y}(Y_t)$ — leads to pronounced mode collapse in the generated samples as shown in Figure 5.

As shown in both Song et al. (2021) and Luo (2022), using Tweedie's formula, the score predicted by the model at timestep t can be expressed from the model's noise prediction $\varepsilon_\theta(z_t, t)$ as

$$s_\theta(z_t,t) \;\equiv\; \nabla_{z_t} \log p_t(z_t) \;\approx\; -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(z_t,t). \tag{19}$$

In Figure 6, we analyze the score field along the 1-D subspace spanned by two exemplar latents from the second row of Figure 4a. Concretely, we visualize the signed projection of the score onto the line segment (the convex hull of two points), i.e., the component of $\nabla_{z_t} \log p_t(z_t)$ parallel to the line segment connecting the two exemplar latents. With projection enabled, this projected score points almost exclusively toward the same endpoint across timesteps, yielding a nearly deterministic path toward the final projected latent $\hat{z}_0$. This concentration substantially reduces sample diversity for any given exemplar set.

Following Nichol and Dhariwal (2021), which highlights the impact of reverse-process variance on sampling, we additionally adopt a DDIM-style stochasticity control and scale the noise term by a factor of k during sampling to increase output diversity. The updated DDPM sampling update is hence

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(z_t,t)\right) + \sqrt{\tilde{\beta}_t}\,k\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,I) \text{ if } t>1, \text{ else } 0. \tag{20}$$

where $k \in \mathbb{R}_{\geq 1}$ scales the stochastic term ($k=1$ recovers standard DDPM; $k>1$ increases the noise standard deviation by k, variance by k^2). To set k, we compute at each diffusion step the average ratio $r_t = \big\|\nabla_{z_t} \log p_t(z_t)\big\| \big/ \big\|\Pi\big(\nabla_{z_t} \log p_t(z_t)\big)\big\|$ between the model score magnitude and the magnitude of its projected component (see Figure 7). We aggregate r_t over timesteps and exemplar sets and choose k near this summary; empirically, $k=20$ provided a good diversity-stability trade-off for all experiment runs involving projection. See Section C for an ablation on k.

Coupling Guidance Strength γ. We observe that projection dampens the gradients supplied by the coupling loss, which calls for the need of γ to be scaled well beyond the values typical for classifier guidance (Dhariwal and Nichol 2021) in order to take effect. We find that empirically, recognizable young–old contrast appears only when $\gamma \geq 200$. This is consistent with our earlier finding that the score component aligned with the line segment connecting the exemplar latents is roughly 13–$20\times$ weaker than the full score (as illustrated in Figure 7), suggesting similar trends in the gradients provided by the coupling loss. Consequently, a simple estimate of the effective guidance strength gives

$$\gamma_{\text{eff}} \approx \gamma / r_t \tag{21}$$

which means compensating for a $r_t \approx 20$ reduction requires γ to increase by roughly $50\times$. We therefore set $\gamma = 450$ (yielding $\gamma_{\text{eff}} \approx 9.0$) for all experiment runs involving both coupling and projection if not otherwise specified. See Section C for an ablation on γ.

Latent Space Interpolatability. VAEs are trained to encourage a smooth, approximately Euclidean latent space by regularizing posteriors toward a standard Gaussian prior, making nearby latent codes decode to similar images and thereby support interpolation (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014). However, we find empirically that *linear* interpolation is reliable only when exemplars are both structurally and spatially aligned. Concretely, if spatial layouts differ, straight-line paths (and thus convex-hull projections) tend to leave the data manifold and decode implausibly. See examples of interpolation between latents of samples from FFHQ-Aging Dataset (Or-El et al. 2020) in Figure 8. This observation accords with geometric analyses showing that semantically consistent transitions follow *curved* geodesics under the decoder-induced Riemannian metric, rather than straight lines in Euclidean latent coordinates (Arvanitidis, Hansen, and Hauberg 2018). Hence, we also experimented with spherical linear interpolations (SLERP) (Shoemake 1985) to maintain constant-norm paths under an isotropic Gaussian prior. In practice, SLERP still required close structural alignment of exemplars, and it only interpolates between two endpoints, whereas our convex-hull projection must accommodate multiple exemplars. Moreover, adopting SLERP consistently would call for a manifold-aware projection, which is nontrivial and beyond the scope of this work. Consequently, we use simple linear interpolation but restrict convex sets to closely related exemplars to preserve visual coherence.

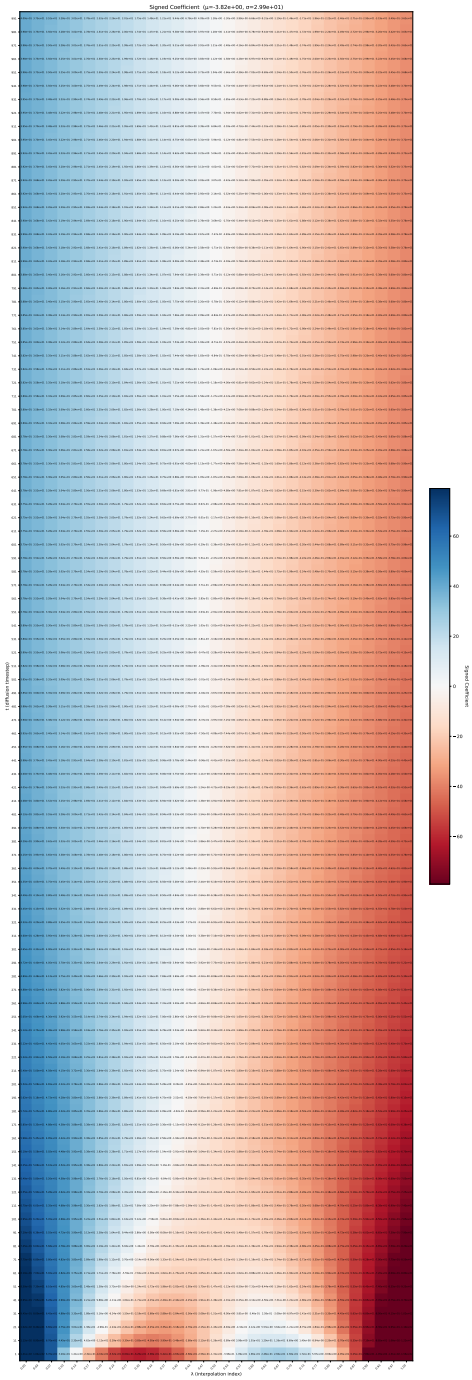


Figure 6: Signed component of the model score $\nabla_{z_t} \log p_t(z_t)$ projected onto the line segment connecting two exemplar latents (the convex hull of two exemplars). The x-axis represents interpolation between the two exemplar latents (from left to right), and the y-axis denotes the diffusion timestep t. *Color indicates the direction and strength of the projected score: blue values push towards the right exemplar, while red values push towards the left.* This visualization, based on two exemplars from row 2 of Figure 4a, reveals that the projected score components consistently point towards one side (left), creating a narrow "white" transition band that funnels every sample to the left exemplar — hence a nearly deterministic path and little diversity. Such behavior is observed consistently in all exemplar sets experimented and likely stems from the exemplars being relatively similar, which induces a narrow feasible region for the diffused latents. However, the exemplars cannot differ too much in spatial structure — an empirical limitation. When exemplars differ significantly, interpolations between them often fail to represent coherent or meaningful images (see discussion in later sections).

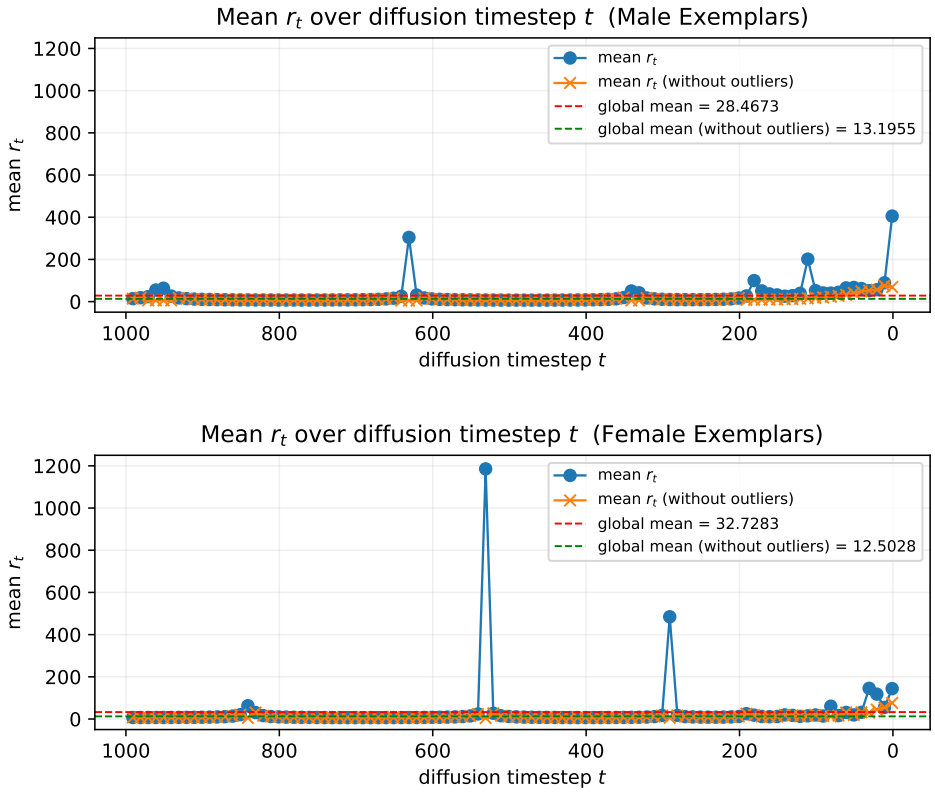


Figure 7: Mean ratio $r_t = \left\|\nabla_{z_t} \log p_t(z_t)\right\| / \left\|\Pi\big(\nabla_{z_t} \log p_t(z_t)\big)\right\|$ over diffusion timesteps t, where $\Pi(\cdot)$ projects onto the convex hull formed by exemplar latents. Top: computed with male exemplar pair (top row of Fig. 4a); bottom: female exemplar pair (bottom row). Large outlier r_t values indicate the predicted score is nearly orthogonal to the chord connecting the two exemplar latents. Outliers are defined as $r_t \geq \mu + 1.5\sigma$, where μ is the global mean and σ the global standard deviation across t.

Figure 8: Linear latent interpolations between pairs of FFHQ-Aging Dataset images (Or-El et al. 2020). Rows 1–3 use exemplars that differ in pose or spatial layout; midway latents leave the data manifold and decode to implausible faces. Rows 4–6 use structurally aligned exemplars; the entire interpolation produces coherent and plausible images. The contrast illustrates that latent interpolations are reliable only for closely aligned exemplars.

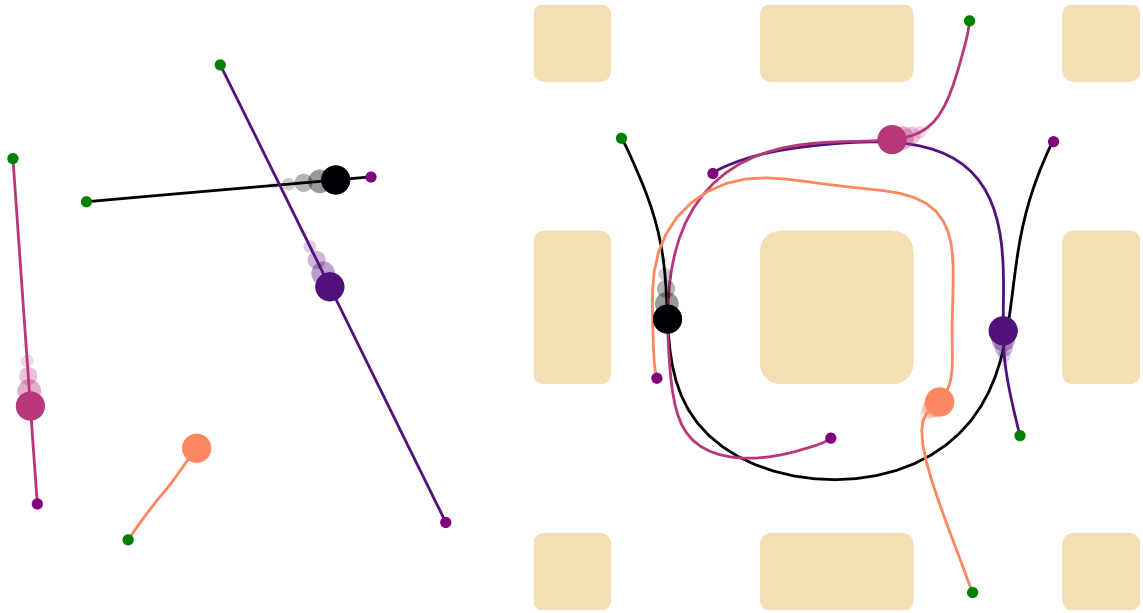


Figure 9: Multi-robot motion planning tasks with 4 robots and simulation environments `Empty` (left) and `Highways` (right), with straight-line and counterclockwise-roundabout motion patterns, respectively. Each of the trajectories' colors denotes an individual robot; green and purple small dots represents start and goal positions.

B.3 Multi-Robot Experiment

Simulation Tasks and Environments Two simulated environments used in our experiments are from the MMD benchmark (Shaoul et al. 2025), `Empty` and `Highways`, as shown in Figure 9. Each environment comes with a trajectory distribution through a trajectory dataset collected under a specific motion pattern. In each environment, given a start and a goal position $(s, g) \in \mathbb{R}^2 \times \mathbb{R}^2$ of a single robot, a diffusion model is trained to generate 2D trajectories $X \in \mathbb{R}^{H\times 2}$, where H is the trajectory length. In `Empty`, the motion pattern is straight-line movements from start to goal; in `Highways`, the pattern is circling around a central block in *counterclockwise* direction when moving from start to goal, resembling a traffic roundabout. The task in both environments is to generate collision-free trajectories for N robots, given an initial configuration consisting of start and goal positions for all robots:

$$\mathcal{P} \triangleq \{(s_1, g_1), \ldots, (s_N, g_N)\} \in (\mathcal{W}_{\text{free}} \times \mathcal{W}_{\text{free}})^N ,$$

where s_i, g_i are the start and goal positions for robot i, and $\mathcal{W}_{\text{free}} \subset \mathbb{R}^2$ denotes the obstacle-free environment space.

ADMM for Projection We repeat Eq. (6) here for reader's convenience:

$$\min_{X\in\mathbb{R}^{H\times 2}} \ \|X - \widehat{X}\|_F^2 \tag{6a}$$

$$\text{s.t.} \quad \|x_0 - X_1\| \le v_{\max}\,\Delta t, \tag{6b}$$

$$\|X_h - X_{h-1}\| \le v_{\max}\,\Delta t, \quad h = 2, \ldots, H, \tag{6c}$$

where $\widehat{X}$ is the diffusion-predicted trajectory for one robot in matrix form, $X_h \in \mathbb{R}^2$ is the position vectors at (physical) discrete time step h and $x_0 \in \mathbb{R}^2$ is a known starting position.

A direct approach for solving the optimization in Eq. (6) is to leverage off-the-shelf solvers to optimize *each trajectory*. However, this incurs significant computation overheads upon large batch of trajectories. Alternatively, we can reformulate Eq. (6) and efficiently solve a batch of such problem instances *in parallel* using Alternating Direction Method of Multipliers (ADMM) (Boyd and Vandenberghe 2004).

To apply ADMM, we need to introduce auxiliary variables $Z_h \in \mathbb{R}^2$ representing the positional displacements:

$$Z_1 = X_1 - x_0, \tag{22a}$$

$$Z_h = X_h - X_{h-1} \quad \text{for } h = 2, \ldots, H. \tag{22b}$$

The constraints Eq. (6b) (6c) then become:

$$\|Z_h\| \le v_{\max}\Delta t, \quad h = 1, \ldots, H.$$

Let $Z = [Z_1, \cdots, Z_H]^\top \in \mathbb{R}^{H\times 2}$ and define the constraint set for Z:

$$\mathcal{K}_Z = \left\{ Z \in \mathbb{R}^{H\times 2} \mid \|Z_h\| \le v_{\max}\Delta t, h = 1, \ldots, H \right\}, \tag{23}$$

and the indicator function of $\mathcal{K}_Z$:

$$\mathbb{I}_{\mathcal{K}_Z}(Z) = \begin{cases} 0 & \text{if } Z \in \mathcal{K}_Z, \\ \infty & \text{otherwise.} \end{cases} \tag{24}$$

Let $A \in \mathbb{R}^{H\times H}$ be a matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \tag{25}$$

and an offset matrix $b \in \mathbb{R}^{H\times 2}$

$$b = \begin{bmatrix} x_0 & \mathbf{0}_2 & \cdots & \mathbf{0}_2 \end{bmatrix}^\top \tag{26}$$

where $\mathbf{0}_2 \in \mathbb{R}^2$ is a zero column vector.

With X and Z, optimization problem Eq. (6) can be reformulated in an ADMM fashion as:

$$\min_{X\in\mathbb{R}^{H\times 2}, Z\in\mathbb{R}^{H\times 2}} \|X - \hat{X}\|_F^2 + \mathbb{I}_{\mathcal{K}_Z}(Z) \tag{27a}$$

$$\text{s.t.} \quad AX - Z = b. \tag{27b}$$

The augmented Lagrangian of Eq. (27) is:

$$\mathcal{L}_\xi(X, Z, \Lambda) = \|X - \hat{X}\|_F^2 + \mathbb{I}_{\mathcal{K}_Z}(Z) + \text{Tr}\left(\Lambda^\top (AX - Z - b)\right) + \frac{\xi}{2}\|AX - Z - b\|_F^2 \tag{28}$$

where $\Lambda \in \mathbb{R}^{H\times 2}$ is the dual variable, $\xi > 0$ is a penalty parameter, and $\text{Tr}(\cdot)$ denotes the trace of a matrix.

The update rules of X, Z and the dual Λ is dervied as follows by ADMM:

- X-update:

  $$X^{k+1} = \arg\min_X \mathcal{L}_\xi(X, Z^k, \Lambda^k).$$

  It has a closed-form solution given by taking the gradient w.r.t. X and setting it to zero:

  $$2(X - \hat{X}) + A^\top\Lambda^k + \xi A^\top(AX - Z^k - b) = 0,$$

  yielding

  $$X^{k+1} = (2\boldsymbol{I}_H + \xi A^\top A)^{-1}(2\hat{X} + \xi A^\top Z^k + \xi A^\top b - A^\top\Lambda^k). \tag{29}$$

- Z-update:

  $$Z^{k+1} = \arg\min_Z \mathcal{L}_\xi(X^{k+1}, Z, \Lambda^k),$$

  which is

  $$Z^{k+1} = \arg\min_{Z\in\mathcal{K}_Z} \left(\text{Tr}\left(-Z^\top\Lambda^k\right) + \frac{\xi}{2}\|Z - (AX^{k+1} - b)\|_F^2 \right).$$

  The solution to the above is

  $$Z^{k+1} = \Pi_{\mathcal{K}_Z}\left(AX^{k+1} - b + \frac{1}{\xi}\Lambda^k \right), \tag{30}$$

  where the projection operation is applied row-wise

  $$Z_h^{k+1} = \begin{cases} (v_{\max}\Delta t)\frac{w_h}{\|w_h\|} & \text{if } \|w_h\|_2 > v_{\max}\Delta t, \\ w_h & \text{otherwise,} \end{cases}$$

  with w_h being the h-th row of $\left(AX^{k+1} - b + \frac{1}{\xi}\Lambda^k\right)$.

- Λ-update:

  $$\Lambda^{k+1} = \Lambda^k + \xi(AX^{k+1} - Z^{k+1} - b). \tag{31}$$

Remark 2. Note that the matrix $\left(2\boldsymbol{I}_H + \xi A^\top A\right)$ in Eq. (29) is symmetric positive definite and constant across iterations, which allows for caching its inverse. Furthermore, one may utilize LU or Cholesky decomposition on $(2\boldsymbol{I}_H + \xi A^\top A)$ and solve linear system $(2\boldsymbol{I}_H + \xi A^\top A)X^{k+1} = 2\hat{X} + \xi A^\top Z^k + \xi A^\top b - A^\top\Lambda^k$ with cached decomposition matrices at each iteration.

The above derivations lead to Algorithm 4.

Algorithm 4: Batched ADMM Projection for Velocity-Constrained Trajectories

Require: Predicted trajectory batch $\hat{\mathbf{X}}$, x_0, $v_{\max} > 0,\ \Delta t > 0$, penalty $\xi > 0$, max iteration $K_{\max}$, tolerance ε.

Pre-compute A, b matrices ▷ *same for all batches*
Pre-compute $M \leftarrow 2I_H + \xi\, A^\top A$ ▷ *same for all batches*
Caching inverse or decomposition of M ▷ *same for all batches*
Initialize $\mathbf{Z}^0 \leftarrow \mathbf{0},\ \mathbf{\Lambda}^0 \leftarrow \mathbf{0}$ ▷ *zero tensors of shape* $B \times H \times 2$
for $k \leftarrow 0$ **to** $K_{\max} - 1$ **do**
  ▷ *X-update, Eq. (29)* ◁
  $\mathbf{V} = 2\hat{\mathbf{X}} + \xi A^\top \left(\mathbf{Z}^k + b - \frac{1}{\xi}\mathbf{\Lambda}^k\right)$ ▷ *A, b broadcast across batches*
  $\mathbf{X}^{k+1} \leftarrow \texttt{SolveLinearSystBatch}\big(M\mathbf{X}^{k+1} = \mathbf{V}\big)$ ▷ *M broadcasts across batches*
  ▷ *Z-update, Eq. (30)* ◁
  $\mathbf{W} \leftarrow A\,\mathbf{X}^{k+1} - b + \frac{1}{\xi}\mathbf{\Lambda}^k$
  for all $(\beta, h) \in \{1, \dots, B\} \times \{1, \dots, H\}$ **in parallel do** ▷ *Vectorization and broadcasting by PyTorch*
    $\mathbf{Z}^{k+1}_{\beta,h} \leftarrow \max\left\{(v_{\max}\Delta t), \|\mathbf{W}_{\beta,h}\|\right\} \frac{\mathbf{W}_{\beta,h}}{\|\mathbf{W}_{\beta,h}\|}$
  ▷ *Dual-update, Eq. (31)* ◁
  $\mathbf{\Lambda}^{k+1} \leftarrow \mathbf{\Lambda}^k + \xi\left(A\,\mathbf{X}^{k+1} - \mathbf{Z}^{k+1} - b\right)$
  ▷ *Optional: Convergence check* ◁
  if check convergence **then**
    $\mathbf{R} \leftarrow A\,\mathbf{X}^{k+1} - \mathbf{Z}^{k+1} - b$ ▷ *Primal residuals*
    $r_{\max} \leftarrow \max_{\beta=1,\dots,B} \|\mathbf{R}_\beta\|_F$
    if $r_{\max} \le \varepsilon$ **then**
      break
return $\mathbf{X}^{k+1}$

Obstacle Cost For static obstacle avoidance, we follow Carvalho et al. (2023) using a cost based on signed distance to a static obstacle. Specifically, let $\varphi(x) : \mathbb{R}^2 \to \mathbb{R}$ be a *differentiable* signed distance from a robot to its *closest* obstacle, and then the obstacle cost term reads

$$c_{\text{obst}}(X^1, \dots, X^N) = \sum_{h=1}^{H} \sum_{i=1}^{N} \mathbf{1}\left[\varphi(X^i_h) \le r'\right] \cdot \left(r' - \varphi(X^i_h)\right) \tag{32}$$

where $r' > 0$ is also a parameter.

Hyperparameters Let all robots share the same radius R. We set $\lambda_{\text{robo}} = 1.0$ and $\lambda_{\text{obst}} = 0.1/\gamma$. For SHD cost, we set $\rho = 6R$ and typically $\gamma \in [0.6, 3.0]$. For LB cost, we set $\alpha = 1.9R$ and typically $\gamma \in [0.01, 0.06]$. Regarding projection, typically we set the penalty $\xi = 10$, max iteration $K_{\max} = 700$, and tolerance $\varepsilon = 2 \times 10^{-5}$.

B.4 Diverse Robot Manipulation Experiment

Diffusion Policy. We adopt DIFFUSION POLICY (DP) (Chi et al. 2023) as our base algorithm, using pretrained weights from (Feng et al. 2024)[5]. DP is a conditional diffusion model operating in a receding-horizon manner. Conditioned on an observation $O \in \mathbb{R}^{H_o \times 5}$— a trajectory of H_o steps where each step is a 5D state vector capturing the planar position of the robot end effector and the T block's center and orientation — it generates an action segment $X \in \mathbb{R}^{H \times 2}$ representing future end-effector positions. Only the first $H_a \le H$ steps are executed (in simulation), after which a new observation O' is obtained and the process repeats, until a total of $H_{\max}$ execution steps is reached.

Detailed Setup. We compare our method with the vanilla DIFFUSION POLICY (DP) as baseline, DP with only projection, and DP with only coupling. We evaluate each method on 50 uniformly random initial observations. With each method, we generate 100 *pairs* of full trajectories $A \in \mathbb{R}^{H_{\max} \times 2}$ (concatenated by the *executed portion* of each action segments) conditioned on every initial observation. For projection, we choose three different max velocity limits, corresponding to the 80%, 90% and 95% quantiles of the velocities[6] generated by the baseline across the initial observations. We use 32 diffusion steps at inference, and take 1 gradient descent step for coupling. We adopt the setting of prediction horizon $H = 16$, action horizon $H_a = 8$, and observation horizon $H_o = 2$ as recommended in (Chi et al. 2023). The maximal action steps $H_{\max}$ is set to 360.

Remark 3. We only take the executed part of the generated action segments and concatenate them along time dimension to form a *full* trajectory $A \in \mathbb{R}^{H_{\max} \times 2}$ for evaluation.

[5]The model from (Feng et al. 2024) was trained on an augmented dataset with broader coverage than that of (Chi et al. 2023), yielding more diverse and feasible trajectories.

[6]Calculated by forward difference of positions of the robot.

Projection Details. We use the same formulation and implementation of projection as in the multi-robot experiments. The penalty parameter used is $\xi = 6.0$ and the max iteration $K_{\text{max}} = 250$.

Coupling Costs. For all methods at all velocity limits we use the same cost-dependent γ value. Concretely, for DPP and DPP-PS costs we set $\gamma = 0.2$; for LB and LB-PS costs we set $\gamma = 0.02$. These parameters are chosen based on a coarse parameter scan and selecting one among the Pareto front of the TC-DTW and TC-DFD relations.

Details in Evaluation Metrics. We use four quantitative metrics for evaluation: Dynamic Time Warping (DTW) (Berndt and Clifford 1994; Müller 2007), discrete Fréchet distance (DFD) (Alt and Godau 1995), velocity constraint satisfaction rate (CS), and task completion score (TC) (Florence et al. 2022; Chi et al. 2023). DTW and DFD quantifies dissimilarity between two trajectories and have been widely used in robotics (Bucker et al. 2023; Memmel et al. 2025) and dynamical systems learning (Rana et al. 2020; Zhang, Mohammadi, and Rozo 2022). For each pair of *full* trajectories, we report the DTW and DFD between the two corresponding segments, and average them over number of segments within each full trajectory. The velocity constraint satisfaction rate for each full trajectory is defined as the fraction of action segments respecting the constraint within the full trajectory. The task completion score measures how well the manipulation task is accomplished by a *full* trajectory given an initial observation, where 1.0 is the best, and 0 the worst. We report all metrics by their empirical mean over all initial observations and full trajectory pairs.

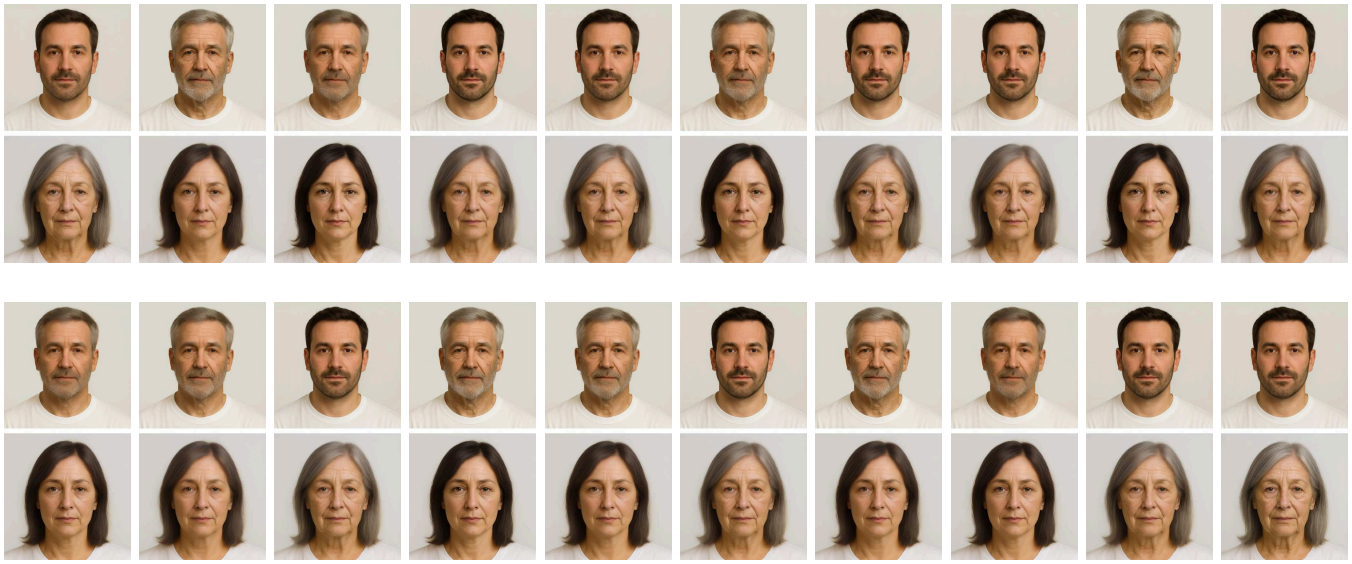


Figure 10: Sample pairs generated with both projection and coupling applied. The two exemplars used for each model (each row) are as per Figure 4a.

METHOD	XOR%↑	M/F%↑	SE-CLIP*↑	SE-LPIPS*↓	IS-LPIPS*↑
SD	20	71/14	$.40_{\pm 0.0463}/.41_{\pm 0.0347}$	$.77_{\pm 0.0426}/.76_{\pm 0.0257}$	$\mathbf{.75}_{\pm 0.0598}/\mathbf{.75}_{\pm 0.0357}$
SD+C	48	51/47	$.45_{\pm 0.0572}/.46_{\pm 0.0669}$	$.76_{\pm 0.0421}/.74_{\pm 0.03315}$	$\mathbf{.75}_{\pm 0.0523}/\mathbf{.75}_{\pm 0.0514}$
SD+C†	64	47/37	$.55_{\pm 0.0637}/.60_{\pm 0.0970}$	$.70_{\pm 0.0535}/.68_{\pm 0.0438}$	$.69_{\pm 0.0659}/.69_{\pm 0.0635}$
SD+P	44	**100**/**100**	$\mathbf{.88}_{\pm 0.0015}/.91_{\pm 0.0027}$	$.15_{\pm 0.0223}/.16_{\pm 0.0384}$	$.06_{\pm 0.0434}/.09_{\pm 0.0639}$
PCD	**96**	**100**/**100**	$\mathbf{.88}_{\pm 0.0028}/\mathbf{.92}_{\pm 0.0041}$	$\mathbf{.11}_{\pm 0.0191}/\mathbf{.14}_{\pm 0.0316}$	$.11_{\pm 0.0876}/.13_{\pm 0.0899}$

Table 4: Paired face-generation results with projection and coupling applied. Exemplars used are as per Figure 4a. Boldface indicates the best score(s) for each metric.

C Additional Results

C.1 Constrained Coupled Image Pair Generation

Additional Qualitative Results Figure 10 presents additional pairs generated by PCD when *two* exemplar images (Figure 4f) define the convex hull. We repeat the experiment with *six* exemplars per model (Figure 11); the corresponding samples are shown in Figure 12.

Additional Quantitative Results Table 4 reports the full set of metrics including the standard deviations for the samples generated using *two* exemplars per model, while Table 5 for the samples generated with *six* exemplars per model. Quantitatively, both setups show the same trend throughout, that is: Projection-based methods (SD+P, PCD) achieve 100% gender satisfaction (M/F) and the strongest alignment to exemplars (higher SE-CLIP and lower SE-LPIPS) compared to vanilla SD and coupling-only variants. Projection reduces diversity (low IS-LPIPS); adding coupling partially recovers diversity relative to projection alone. Coupling-based methods improve the age-group XOR satisfaction, with PCD attaining the highest rate (96% for the setup using *two* exemplars and 76% for the setup using *six* exemplars).

Ablation Study We ablate both the coupling strength γ and the noise-scaling factor k. For **PCD**, we vary $\gamma \in \{50, 150, 250, 350, 450, 500\}$ with $M \in \{2, 6\}$ exemplar images per model. Separately, for **SD+P** we vary $k \in \{1, 2, 5, 10, 20\}$ (again with $M \in \{2, 6\}$) to examine how amplifying the noise standard deviation impacts diversity and other metrics. Figures 13 and 14 summarize the trends for XOR%, M/F%, SE-CLIP, SE-LPIPS, and IS-LPIPS as γ and k increase, respectively.

As γ increases, we observe that XOR% increases monotonically, M/F% (gender constraint satisfaction) is maintained perfect by design, SE-CLIP maintains in place, SE-LPIPS drops monotonically, while IS-LPIPS increases. This suggests that (i) a larger γ strengthens the coupling signal, yielding a clearer young/old contrast (higher XOR%), (ii) generated sample pairs resemble more of the exemplars provided (denoted by gradually decreasing SE-LPIPS) due to gradients from the coupling loss, (iii) stronger coupling also injects modest additional diversity (higher IS-LPIPS), and (iv) projection ensures the gender constraint is satisfied irrespective of γ.

Figure 11: *Six* exemplar images for each model (one row per model), generated via ChatGPT (OpenAI 2023). The exemplars differ only by minor visual details, ensuring close structural alignment.

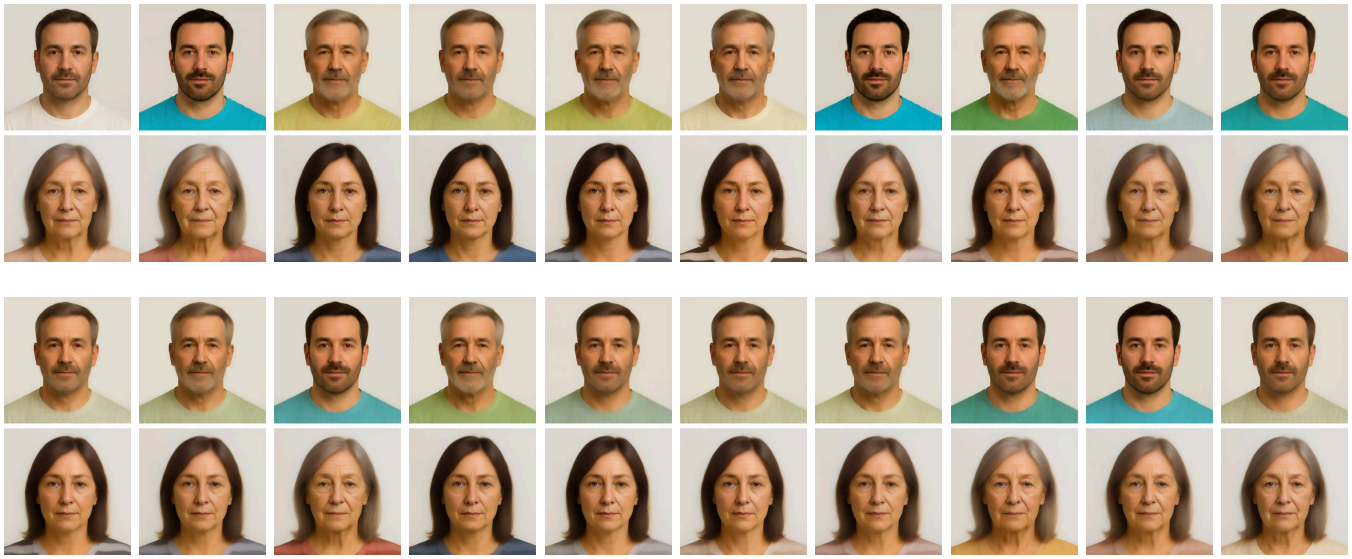


Figure 12: Sample pairs generated with both projection and coupling applied. The *six* exemplars used for each model (each row) are as per Figure 11.

In the independent study of k, we observe that as k increases, the trend across all metrics are very much similar to that of γ, with one notable difference: IS-LPIPS rises sharply and monotonically as k grows. This suggests that increasing the noise scaling factor boosts sample variation, and yet does not violate the attribute constraints provided by the exemplars (perfect M/F%, high SE-CLIP and low SE-LPIPS).

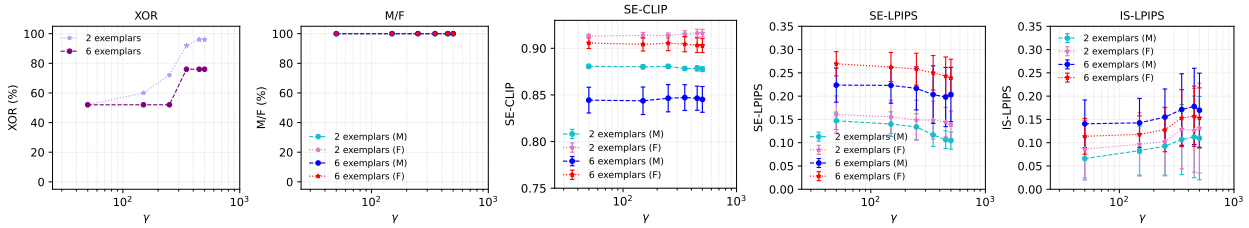
C.2 Multi-Robot Navigation

Additional Qualitative Results Figure 15 and Figure 16 exhibit more trajectory samples generated by the compared methods on both tasks with $N = 2$ and $N = 4$ robots. These results demonstrate the effectiveness of PCD in generating highly correlated trajectories and simultaneously enforcing hard constraints compared to methods with the absence of either coupling or projection, or both.

Additional Quantitative Results Table 6 and Table 7 summarize quantitative evaluation on the both environments under one maximum velocity constraint with 2 robots. Other results are in Table 8–Table 9. In terms of constraint satisfaction, constraint-agnostic methods (vanilla DIFFUSER, MMD-CBS, and all coupling-only CD- variants) achieve similar rates: as low as 8–22% in `Empty` and between around 28% and 62% in `Highways`. In contrast, every projection-based variant (DIFFUSER with projection and our PCD-LB/SHD) enforces the constraint in all cases, confirming the effectiveness of projection. Inter-agent safety scores show the similar trend. Because MMD-CBS repeatedly samples and then stitches together *one* optimal trajectory tuple, it unsurprisingly attains perfect score. Among the one-shot methods, PCD- and CD- methods can almost match this upper bound, while vanilla DIFFUSER and and its projected variant performs much worse. Slightly degraded data adherence performance is again observed in our method: the LB cost function gets affected more due to its steeper gradients by design. Overall, the results show that PCD effectively promotes inter-robot collision avoidance through coupling costs, while enforcing hard test-time velocity constraints. A tradeoff exists between coupling strength and data adherence, depending on the coupling cost.

METHOD	XOR%↑	M/F%↑	SE-CLIP*↑	SE-LPIPS*↓	IS-LPIPS*↑
SD	20	71/14	$.40_{\pm 0.0463}/.41_{\pm 0.0347}$	$.77_{\pm 0.0426}/.76_{\pm 0.0257}$	$\mathbf{.75}_{\pm 0.0598}/\mathbf{.75}_{\pm 0.0357}$
SD+C	48	51/46	$.44_{\pm 0.0553}/.44_{\pm 0.0646}$	$.76_{\pm 0.0421}/.73_{\pm 0.0297}$	$\mathbf{.75}_{\pm 0.0524}/\mathbf{.75}_{\pm 0.0514}$
SD+C†	64	47/38	$.54_{\pm 0.0640}/.59_{\pm 0.1044}$	$.70_{\pm 0.05347}/.68_{\pm 0.0441}$	$.69_{\pm 0.0659}/.69_{\pm 0.0635}$
SD+P	44	**100**/**100**	$.84_{\pm 0.0140}/\mathbf{.91}_{\pm 0.0056}$	$.22_{\pm 0.0364}/.27_{\pm 0.0232}$	$.14_{\pm 0.0511}/.11_{\pm 0.0387}$
PCD	**76**	**100**/**100**	$\mathbf{.85}_{\pm 0.0129}/.90_{\pm 0.0078}$	$\mathbf{.19}_{\pm 0.0635}/\mathbf{.24}_{\pm 0.0410}$	$.18_{\pm 0.0819}/.16_{\pm 0.0646}$

Table 5: Paired face-generation results with projection and coupling applied. Exemplars used are as per Figure 4a. Boldface indicates the best score(s) for each metric.



(a) XOR% ↑ (b) M/F% ↑ (c) SE-CLIP ↑ (d) SE-LPIPS ↓ (e) IS-LPIPS ↑

Figure 13: Ablation of coupling strength γ for **PCD**, using $M \in \{2, 6\}$ exemplars per model. We evaluate $\gamma \in \{50, 150, 250, 350, 450, 500\}$.

Remark 4. A direct comparison between MMD-CBS and our method is not straightforward due to fundamental differences in their sampling procedures. MMD-CBS is a search-based approach that generates a batch of trajectories using diffusion models, *selects the best one* based on a cost function, *adopts only a partial segment* of the selected trajectory to resolve collisions, and repeats this process iteratively. As a result, it produces only *one trajectory* per robot per forward pass, making it inefficient for generating multiple i.i.d. samples. In contrast, DIFFUSER and our method produce a full batch of i.i.d. trajectories in a single pass. Thus, directly comparing a 128-sample batch from DIFFUSER or ours against a single output from MMD-CBS, or vice versa, would not be very meaningful.

Ablation Study We perform ablation study on the coupling strength γ for both tasks `Empty` and `Highways` with the velocity limits reported in Table 1, but with both 2 and 4 robots. Results are presented in Figure 17 and Figure 18. In task `Empty`, as the coupling strength γ increases, SU and RS increases and saturated near 1.0, with an exception in $N = 4$ where RS drops a bit when γ is too high; CS remains at 100% by projection, and DA in general gradually drops. Results of `Highways` demonstrates the similar trends, with the difference where SU drops significantly for the LB cost when γ is large. This is because SU also takes into accounts collisions with *static obstacles*. When γ is high, the gradient of the robot-collision cost overwhelms that of the obstacle-avoidance term[7], resulting in the robots bumping into obstacles. This is supported by Figure 19 in which we report the decreasing obstacle safe rate, suggesting more trajectories are hitting static obstacles as γ increases. The obstacle safe rate is defined as the average of an indicator of whether all trajectories are *not* colliding into static obstacles.

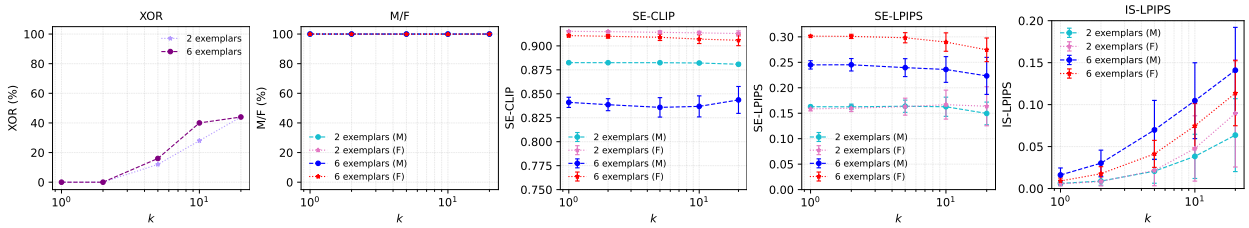
C.3 Constrained Diverse Robotic Manipulation

Qualitative Results We present in Figure 20 more comparative results on the trajectories by all compared methods. Sheer contrast between our PCD method and others highlights the efficacy of our framework in jointly generating correlated samples while enforcing hard constraints.

Quantitative Results at Different Velocity Limits Table 10 summarizes quantitative results of all compared methods with all three velocity limits. We also report standard deviations in the tables.

Ablation Study We perform ablation study investigating how the coupling strength γ affects the performance of our PCD method. Figure 21 shows the trends of evaluation metrics' (DTW, DFD, CS, TC) change as the coupling strength parameter γ increase, with all three velocity limits we experimented. Above all, the general trends look expected and similar across the three velocity limits: as we increase γ, the DTW and DFD monotonically increase, velocity constraint satisfaction (CS) is maintained perfect by design, and task completion score (TC) drops monotonically. This suggests that (i) the projection can guarantee velocity constraint regardless of coupling strength; (ii) as γ increases, the correlations between the variables get stronger, with

[7]The coefficient for the gradient of the obstacle-avoidance cost is *fixed* in our experiments. See Appendix B.3 for details.

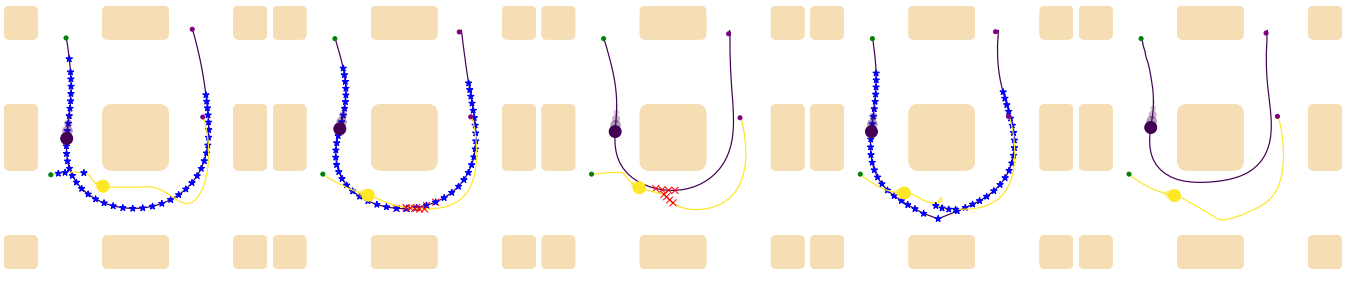


(a) XOR% ↑ (b) M/F% ↑ (c) SE-CLIP ↑ (d) SE-LPIPS ↓ (e) IS-LPIPS ↑

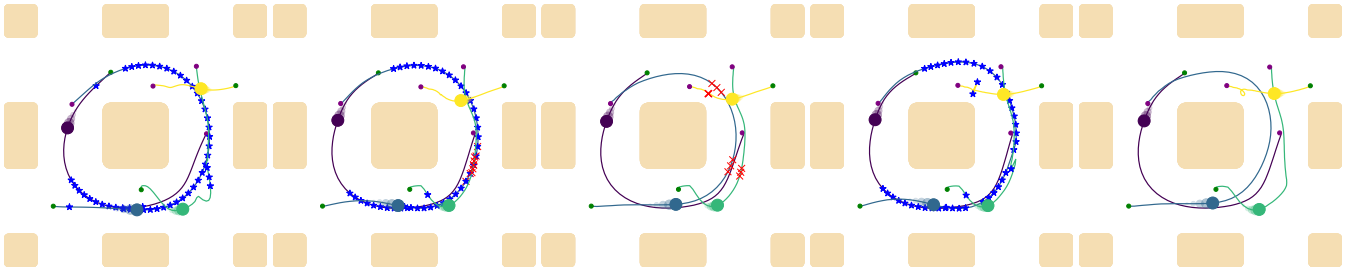
Figure 14: Ablation of noise scaling factor k for **SD+P**, using $M \in \{2, 6\}$ exemplars per model. We evaluate $k \in \{1, 2, 5, 10, 20\}$.

the cost gradients gradually overwhelming the learned score and thus deviating from the original data distribution. A tradeoff exists between data adherence and correlation strength.

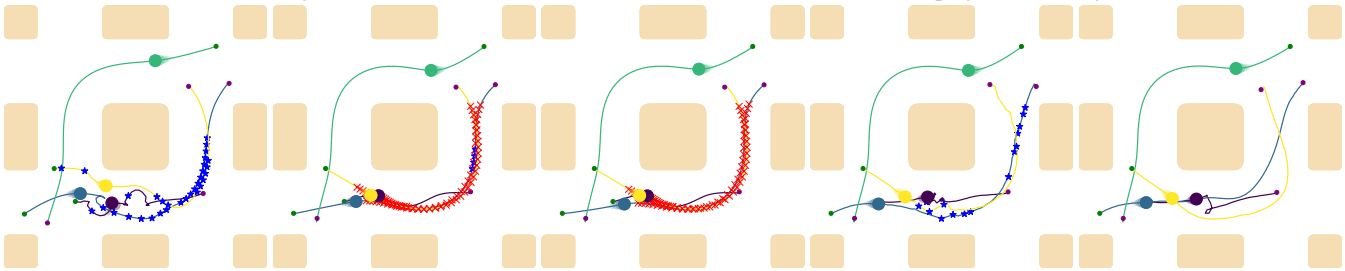
A similar ablation is also conducted on coupling-only method (CD) by removing the projection; results are in Figure 22. The same tradeoff between correlations and data adherence also exists. Unsurprisingly the velocity constraint satisfaction rates drop as γ increases, and the LB cost function is more sensitive.



(a) MMD-CBS($N = 2$) (b) DIFFUSER($N = 2$) (c) PD ($N = 2$) (d) CD (w/o proj., $N = 2$) (e) **PCD (Ours)**($N = 2$)

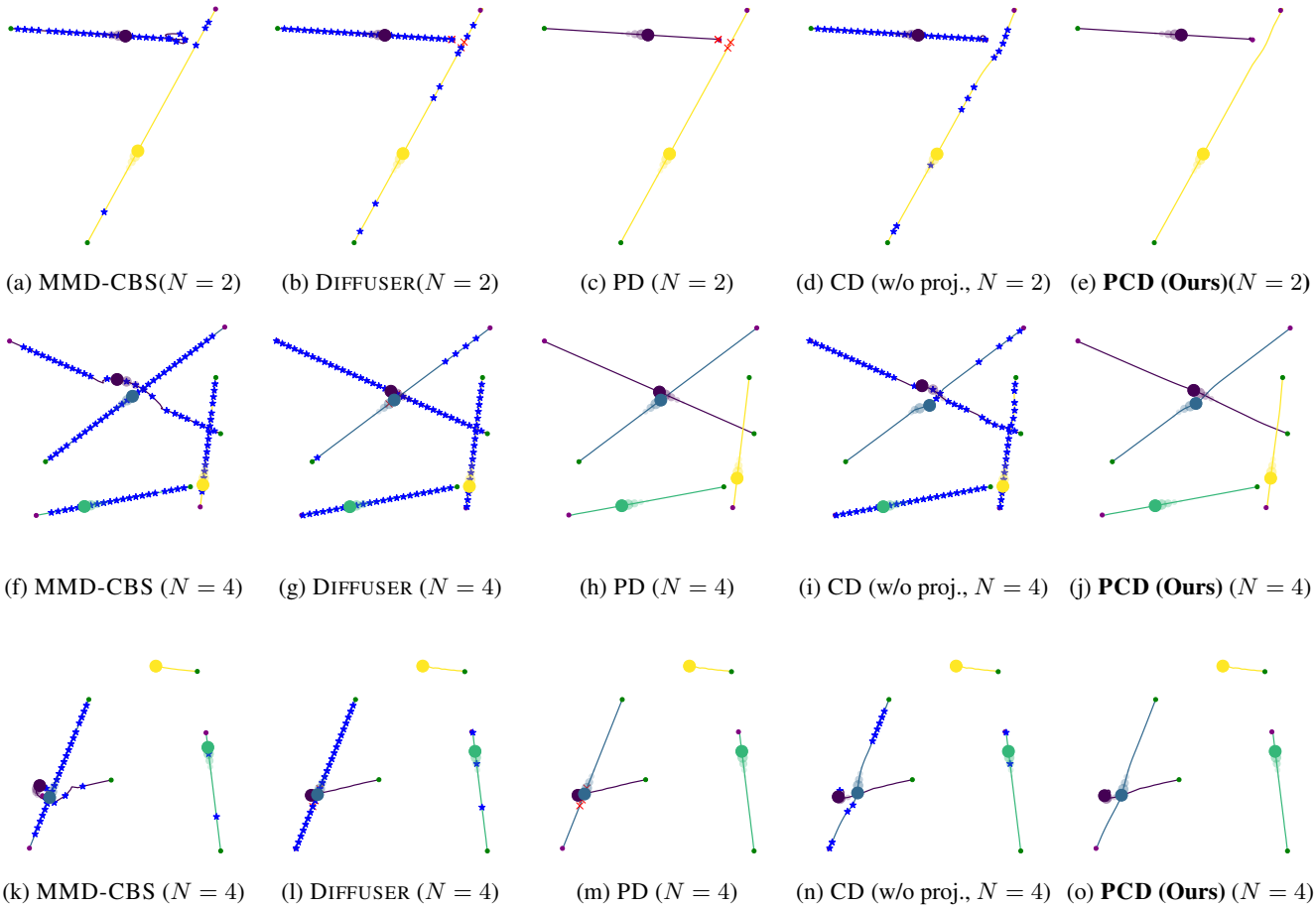


(f) MMD-CBS ($N = 4$) (g) DIFFUSER ($N = 4$) (h) PD ($N = 4$) (i) CD (w/o proj., $N = 4$) (j) **PCD (Ours)** ($N = 4$)



(k) MMD-CBS ($N = 4$) (l) DIFFUSER ($N = 4$) (m) PD ($N = 4$) (n) CD (w/o proj., $N = 4$) (o) **PCD (Ours)** ($N = 4$)

Figure 15: Robot trajectories in environment `Highways` generated by the compared methods with $N = 2$ and $N = 4$ robots running. Red crosses mark collisions and blue stars mark velocity constraint violations. Each row corresponds to one initial configuration (start and goal positions for each robot).



(a) MMD-CBS($N = 2$) (b) DIFFUSER($N = 2$) (c) PD ($N = 2$) (d) CD (w/o proj., $N = 2$) (e) **PCD (Ours)**($N = 2$)

(f) MMD-CBS ($N = 4$) (g) DIFFUSER ($N = 4$) (h) PD ($N = 4$) (i) CD (w/o proj., $N = 4$) (j) **PCD (Ours)** ($N = 4$)

(k) MMD-CBS ($N = 4$) (l) DIFFUSER ($N = 4$) (m) PD ($N = 4$) (n) CD (w/o proj., $N = 4$) (o) **PCD (Ours)** ($N = 4$)

Figure 16: Robot trajectories in environment `Empty` generated by the compared methods with $N = 2$ and $N = 4$ robots running. Red crosses mark collisions and blue stars mark velocity constraint violations. Each row corresponds to one initial configuration (start and goal positions for each robot).

Task `Empty`, 2 Robots				
METHOD \ Metric	SU(%)↑	RS↑	CS(%)↑	DA↑
	Max Vel. = 0.703			
vanilla DIFFUSER	$92_{\pm 27}$	$0.92_{\pm 0.27}$	$(59_{\pm 49}, 65_{\pm 48})$	$(0.99_{\pm 0.085}, 1.0_{\pm 0.0})$
MMD-CBS	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(8.0_{\pm 27}, 12_{\pm 32})$	$(0.99_{\pm 0.073}, 1.0_{\pm 0.0})$
DIFFUSER + projection	$92_{\pm 27}$	$0.92_{\pm 0.27}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.99_{\pm 0.085}, 1.0_{\pm 0.0})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.065}$	$(7.3_{\pm 26}, 5.5_{\pm 23})$	$(0.92_{\pm 0.2}, 0.93_{\pm 0.18})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(45_{\pm 50}, 52_{\pm 50})$	$(0.99_{\pm 0.09}, 1.0_{\pm 0.0062})$
PCD-LB	$95_{\pm 22}$	$0.95_{\pm 0.22}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.12}, 1.0_{\pm 0.0087})$
PCD-SHD	$100_{\pm 0}$	$1.0_{\pm 0.061}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.99_{\pm 0.089}, 1.0_{\pm 0.0092})$
	Max Vel. = 0.692			
vanilla DIFFUSER	$92_{\pm 27}$	$0.92_{\pm 0.27}$	$(40_{\pm 49}, 44_{\pm 50})$	$(0.99_{\pm 0.085}, 1.0_{\pm 0.0})$
MMD-CBS	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(8.0_{\pm 27}, 12_{\pm 32})$	$(0.99_{\pm 0.073}, 1.0_{\pm 0.0})$
DIFFUSER + projection	$93_{\pm 26}$	$0.92_{\pm 0.27}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.99_{\pm 0.085}, 1.0_{\pm 0.0})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.065}$	$(6.9_{\pm 25}, 4.6_{\pm 21})$	$(0.92_{\pm 0.2}, 0.93_{\pm 0.18})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(31_{\pm 46}, 37_{\pm 48})$	$(0.99_{\pm 0.09}, 1.0_{\pm 0.0062})$
PCD-LB	$95_{\pm 22}$	$0.95_{\pm 0.22}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.12}, 1.0_{\pm 0.0088})$
PCD-SHD	$100_{\pm 0}$	$1_{\pm 0.065}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.088}, 1.0_{\pm 0.012})$
	Max Vel. = 0.675			
vanilla DIFFUSER	$92_{\pm 27}$	$0.92_{\pm 0.27}$	$(19_{\pm 40}, 22_{\pm 41})$	$(0.99_{\pm 0.085}, 1.0_{\pm 0.0})$
MMD-CBS	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(8.0_{\pm 27}, 12_{\pm 32})$	$(0.99_{\pm 0.073}, 1.0_{\pm 0.0})$
DIFFUSER + projection	$93_{\pm 26}$	$0.92_{\pm 0.27}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.99_{\pm 0.085}, 1.0_{\pm 0})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1_{\pm 0.065}$	$(6.45_{\pm 25}, 3.6_{\pm 19})$	$(0.92_{\pm 0.2}, 0.93_{\pm 0.18})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(18.4_{\pm 39}, 21.3_{\pm 41})$	$(0.99_{\pm 0.09}, 1.0_{\pm 0.0062})$
PCD-LB	$95_{\pm 22}$	$0.95_{\pm 0.22}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.92_{\pm 0.2}, 0.93_{\pm 0.17})$
PCD-SHD	$100_{\pm 0}$	$1.0_{\pm 0.069}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.99_{\pm 0.088}, 1.0_{\pm 0.012})$

Table 6: Task `Empty`, 2 robots, 100 random tests, sample size 128 except MMD-CBS.

Task **Highways**, **2** Robots				
METHOD \ Metric	SU(%)↑	RS↑	CS(%)↑	DA↑
	Max Vel. = 0.878			
vanilla DIFFUSER	$93_{\pm 26}$	$0.81_{\pm 0.39}$	$(72_{\pm 45}, 76_{\pm 43})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.13})$
MMD-CBS	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(73_{\pm 44}, 79_{\pm 41})$	$(0.99_{\pm 0.099}, 0.97_{\pm 0.17})$
DIFFUSER + projection	$92_{\pm 27}$	$0.81_{\pm 0.39}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.99_{\pm 0.12}, 0.98_{\pm 0.14})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.033}$	$(22_{\pm 42}, 18_{\pm 38})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.15})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.029}$	$(65_{\pm 48}, 66_{\pm 47})$	$(0.99_{\pm 0.12}, 0.98_{\pm 0.13})$
PCD-LB	$100_{\pm 0}$	$0.99_{\pm 0.1}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.14}, 0.97_{\pm 0.18})$
PCD-SHD	$100_{\pm 0}$	$1.0_{\pm 0.012}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.14}, 0.97_{\pm 0.16})$
	Max Vel. = 0.781			
vanilla DIFFUSER	$93_{\pm 26}$	$0.81_{\pm 0.39}$	$(63_{\pm 48}, 65_{\pm 48})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.13})$
MMD-CBS	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(67_{\pm 47}, 67_{\pm 47})$	$(0.99_{\pm 0.099}, 0.97_{\pm 0.17})$
DIFFUSER + projection	$95_{\pm 22}$	$0.81_{\pm 0.39}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.12}, 0.98_{\pm 0.14})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.033}$	$(14_{\pm 35}, 12_{\pm 32})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.15})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(55_{\pm 50}, 53_{\pm 50})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.13})$
PCD-LB	$100_{\pm 0}$	$0.99_{\pm 0.11}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.14}, 0.96_{\pm 0.19})$
PCD-SHD	$100_{\pm 0}$	$1.0_{\pm 0.041}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.14}, 0.97_{\pm 0.16})$
	Max Vel. = 0.647			
vanilla DIFFUSER	$93_{\pm 26}$	$0.81_{\pm 0.39}$	$(52_{\pm 50}, 50_{\pm 50})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.13})$
MMD-CBS	$100_{\pm 0}$	$1.0_{\pm 0.0}$	$(62_{\pm 49}, 54_{\pm 50})$	$(0.99_{\pm 0.099}, 0.97_{\pm 0.17})$
DIFFUSER + projection	$91_{\pm 29}$	$0.84_{\pm 0.36}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.13}, 0.98_{\pm 0.14})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$0.99_{\pm 0.12}$	$(29_{\pm 45}, 22_{\pm 41})$	$(0.99_{\pm 0.11}, 0.98_{\pm 0.16})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1.0_{\pm 0.029}$	$(49_{\pm 50}, 46_{\pm 50})$	$(0.99_{\pm 0.12}, 0.98_{\pm 0.13})$
PCD-LB	$99_{\pm 9.9}$	$0.98_{\pm 0.14}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.15}, 0.96_{\pm 0.19})$
PCD-SHD	$100_{\pm 0}$	$1.0_{\pm 0.022}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.98_{\pm 0.15}, 0.97_{\pm 0.17})$

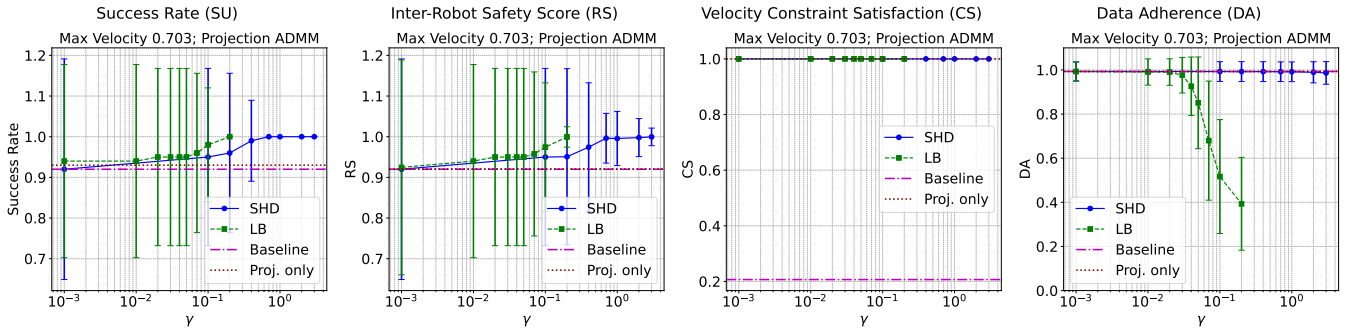
Table 7: Task `Highways`, 2 robots, 100 random tests, sample size 128 except MMD-CBS.

Task `Empty`, 4 Robots				
METHOD \ Metric	SU(%)↑	RS↑	CS(%)↑	DA↑
			Max Vel. = 0.703	
vanilla DIFFUSER	$65_{\pm 48}$	$.62_{\pm .49}$	$(69_{\pm 46}, 65_{\pm 48}, 64_{\pm 48}, 62_{\pm 48})$	$(.99_{\pm .073}, 1._{\pm .00055}, 1._{\pm .0014}, 1._{\pm .0016})$
MMD-CBS	$100_{\pm 0}$	$1._{\pm 0.}$	$(16_{\pm 36}, 17_{\pm 38}, 11_{\pm 31}, 17_{\pm 38})$	$(.99_{\pm .044}, 1._{\pm .0035}, 1._{\pm .019}, 1._{\pm 0.})$
DIFFUSER + proj.	$65_{\pm 48}$	$.61_{\pm .49}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.99_{\pm .073}, 1_{\pm .00055}, 1_{\pm .0014}, 1_{\pm .0016})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$.99_{\pm .085}$	$(.031_{\pm 1.8}, 1_{\pm 9.9}, 3_{\pm 17}, 3.1_{\pm 17})$	$(.81_{\pm .33}, .88_{\pm .26}, .86_{\pm .26}, .85_{\pm .28})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1_{\pm 0}$	$(38_{\pm 49}, 38_{\pm 48}, 35_{\pm 48}, 37_{\pm 48})$	$(.99_{\pm .076}, 1_{\pm .001}, 1_{\pm .0085}, 1_{\pm .003})$
PCD-LB	$96_{\pm 20}$	$.92 \pm .28$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.57_{\pm .38}, .51_{\pm .37}, .52_{\pm .37}, .49_{\pm .37})$
PCD-SHD	$100_{\pm 0}$	$.99_{\pm .084}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.96_{\pm .1}, .98_{\pm .046}, .97_{\pm .069}, .96_{\pm .078})$
			Max Vel. = 0.692	
vanilla DIFFUSER	$65_{\pm 48}$	$.62_{\pm .49}$	$(42_{\pm 49}, 42_{\pm 49}, 43_{\pm 49}, 41_{\pm 49})$	$(.99_{\pm .073}, 1._{\pm .00055}, 1._{\pm .0014}, 1._{\pm .0016})$
MMD-CBS	$100_{\pm 0}$	$1._{\pm 0.}$	$(15_{\pm 36}, 17_{\pm 38}, 11_{\pm 31}, 16_{\pm 37})$	$(.99_{\pm .044}, 1._{\pm .0035}, 1._{\pm .019}, 1._{\pm 0})$
DIFFUSER + proj.	$64_{\pm 48}$	$.61_{\pm .49}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.99_{\pm .073}, 1._{\pm .00053}, 1._{\pm .0014}, 1._{\pm .0016})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$.99_{\pm .085}$	$(0_{\pm 0}, 1_{\pm 9.9}, 2.5_{\pm 16}, 3.1_{\pm 17})$	$(.81_{\pm .33}, .88_{\pm .26}, .86_{\pm .26}, .85_{\pm .28})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1._{\pm 0.}$	$(28_{\pm 45}, 29_{\pm 45}, 27_{\pm 44}, 27_{\pm 45})$	$(.99_{\pm .076}, 1._{\pm .001}, 1._{\pm .0085}, 1._{\pm .003})$
PCD-LB	$94_{\pm 24}$	$.92_{\pm .28}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.6_{\pm .39}, .56_{\pm .37}, .58_{\pm .38}, .55_{\pm .38})$
PCD-SHD	$100_{\pm 0}$	$.99_{\pm .095}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.96_{\pm .1}, .98_{\pm .045}, .97_{\pm .07}, .96_{\pm .078})$
			Max Vel. = 0.675	
vanilla DIFFUSER	$65_{\pm 48}$	$.62_{\pm .49}$	$(24_{\pm 43}, 22_{\pm 42}, 24_{\pm 42}, 26_{\pm 44})$	$(.99_{\pm .073}, 1._{\pm .00055}, 1._{\pm .0014}, 1._{\pm .0016})$
MMD-CBS	$100_{\pm 0}$	$1._{\pm 0.}$	$(15_{\pm 36}, 17_{\pm 38}, 11_{\pm 31}, 14_{\pm 35})$	$(.99_{\pm .044}, 1._{\pm .0035}, 1._{\pm .019}, 1._{\pm 0.})$
DIFFUSER + proj.	$64_{\pm 48}$	$.63_{\pm .48}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.99_{\pm .073}, 1._{\pm .00055}, 1._{\pm .0014}, 1._{\pm .0016})$
CD-LB (w/o proj.)	$100_{\pm 0.}$	$.99_{\pm .085}$	$(0._{\pm 0.}, 1._{\pm 9.9}, 1.7_{\pm 13}, 3._{\pm 17})$	$(.81_{\pm .33}, .88_{\pm .26}, .86_{\pm .26}, .85_{\pm .28})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1._{\pm 0.}$	$(21_{\pm 41}, 21_{\pm 41}, 19_{\pm 39}, 21_{\pm 40})$	$(.99_{\pm .076}, 1._{\pm .001}, 1._{\pm .0085}, 1._{\pm .003})$
PCD-LB	$92_{\pm 27}$	$.88_{\pm .33}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.73_{\pm .38}, .79_{\pm .33}, .77_{\pm .34}, .75_{\pm .34})$
PCD-SHD	$100_{\pm 0}$	$.99_{\pm .099}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.96_{\pm .1}, .98_{\pm .045}, .97_{\pm .069}, .96_{\pm .077})$

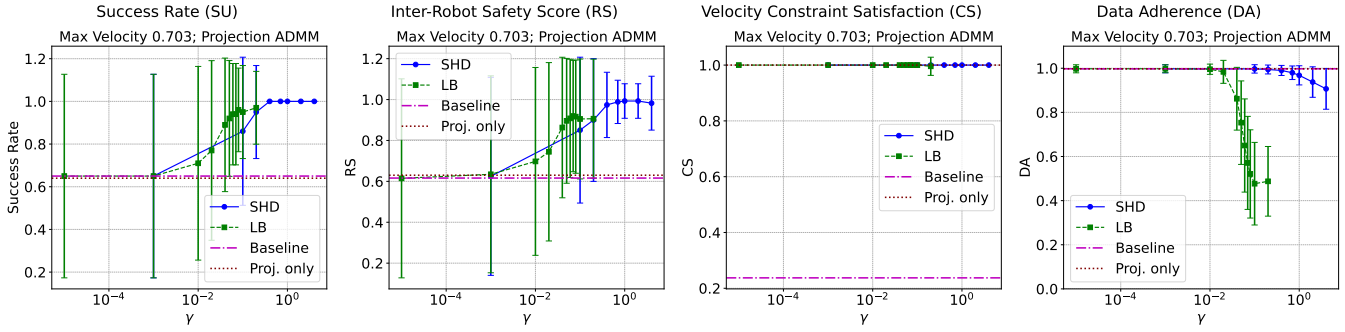
Table 8: Task `Empty`, 4 robots, 100 random tests, sample size 128 except MMD-CBS.

Task **Highways**, **4** Robots				
METHOD \ Metric	SU(%)↑	RS↑	CS(%)↑	DA↑
			Max Vel. = 0.878	
vanilla DIFFUSER	$53_{\pm 50}$	$.21_{\pm .41}$	$(73_{\pm 44}, 68_{\pm 47}, 68_{\pm 47}, 67_{\pm 47})$	$(1._{\pm .04}, .98_{\pm .14}, 1._{\pm .04}, 1._{\pm .028})$
MMD-CBS	$100_{\pm 0}$	$1._{\pm 0.}$	$(68_{\pm 47}, 64_{\pm 48}, 63_{\pm 48}, 65_{\pm 48})$	$(.98_{\pm .14}, .96_{\pm .20}, .97_{\pm .17}, .99_{\pm .10})$
DIFFUSER + proj.	$54_{\pm 50}$	$.21_{\pm .41}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(1._{\pm .052}, .98_{\pm .15}, 1._{\pm .07}, 1._{\pm .047})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1._{\pm .032}$	$(0_{\pm 0}, .11_{\pm 3.3}, 0._{\pm 0.}, 0._{\pm 0.})$	$(.99_{\pm .12}, .99_{\pm .11}, 1._{\pm .067}, 1._{\pm .066})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1._{\pm .012}$	$(35_{\pm 48}, 35_{\pm 48}, 36_{\pm 48}, 35_{\pm 48})$	$(1._{\pm .07}, .98_{\pm .15}, 1._{\pm .07}, 1._{\pm .044})$
PCD-LB	$100_{\pm 0}$	$.95_{\pm .22}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.97_{\pm .16}, .96_{\pm .2}, .99_{\pm .12}, .99_{\pm .098})$
PCD-SHD	$100_{\pm 0}$	$1._{\pm .063}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.98_{\pm .13}, .96_{\pm .19}, .99_{\pm .12}, .99_{\pm .076})$
			Max Vel. = 0.781	
vanilla DIFFUSER	$53_{\pm 50}$	$.21_{\pm .41}$	$(62_{\pm 49}, 57_{\pm 49}, 58_{\pm 49}, 57_{\pm 50})$	$(1._{\pm .04}, .98_{\pm .14}, 1._{\pm .04}, 1._{\pm .028})$
MMD-CBS	$100_{\pm 0}$	$1._{\pm 0.}$	$(58_{\pm 49}, 51_{\pm 50}, 58_{\pm 49}, 59_{\pm 49})$	$(.98_{\pm .14}, .96_{\pm .20}, .97_{\pm .17}, .99_{\pm .10})$
DIFFUSER + proj.	$53_{\pm 50}$	$.22_{\pm .42}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(1._{\pm .054}, .98_{\pm .15}, .99_{\pm .075}, 1._{\pm .048})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$1._{\pm .069}$	$(0._{\pm 0.}, .031_{\pm 1.8}, .07_{\pm 2.7}, .0078_{\pm .88})$	$(.99_{\pm .11}, .98_{\pm .13}, 1._{\pm .067}, 1._{\pm .062})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1._{\pm .051}$	$(49_{\pm 50}, 42_{\pm 49}, 45_{\pm 50}, 43_{\pm 50})$	$(.99_{\pm .072}, .98_{\pm .15}, 1._{\pm .071}, 1._{\pm .048})$
PCD-LB	$100_{\pm 0}$	$.91_{\pm .28}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.98_{\pm .15}, .95_{\pm .21}, .99_{\pm .12}, .99_{\pm .093})$
PCD-SHD	$100_{\pm 0}$	$1_{\pm .064}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.98_{\pm .14}, .96_{\pm .19}, .98_{\pm .12}, .99_{\pm .082})$
			Max Vel. = 0.647	
vanilla DIFFUSER	$53_{\pm 50}$	$.21_{\pm .41}$	$(46_{\pm 50}, 43.5_{\pm 50}, 39.2_{\pm 49}, 45.9_{\pm 50})$	$(1._{\pm .04}, .98_{\pm .14}, 1._{\pm .04}, 1._{\pm .028})$
MMD-CBS	$100_{\pm 0}$	$1._{\pm 0.}$	$(48_{\pm 50}, 41_{\pm 49}, 46_{\pm 50}, 51_{\pm 50})$	$(.98_{\pm .14}, .96_{\pm .20}, .97_{\pm .17}, .99_{\pm .10})$
DIFFUSER + proj.	$49_{\pm 50}$	$.28_{\pm .45}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(1._{\pm .059}, .98_{\pm .15}, .99_{\pm .084}, 1._{\pm .052})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$.97_{\pm .17}$	$(.07_{\pm 2.7}, .21_{\pm 4.6}, .27_{\pm 5.1}, .055_{\pm 2.3})$	$(.99_{\pm .11}, .97_{\pm .16}, 1._{\pm .068}, 1._{\pm .061})$
CD-SHD (w/o proj.)	$100_{\pm 0}$	$1._{\pm .012}$	$(18_{\pm 39}, 16_{\pm 37}, 22_{\pm 41}, 21_{\pm 41})$	$(1._{\pm .07}, .98_{\pm .15}, 1._{\pm .07}, 1._{\pm .044})$
PCD-LB	$87_{\pm 34}$	$.92_{\pm .27}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.98_{\pm .15}, .95_{\pm .22}, .98_{\pm .13}, .99_{\pm .099})$
PCD-SHD	$100_{\pm 0}$	$.99_{\pm .1}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.96_{\pm .2}, .93_{\pm .25}, .97_{\pm .18}, .97_{\pm .17})$

Table 9: Task `Highways`, 4 robots, 100 random tests, sample size 128 except MMD-CBS.

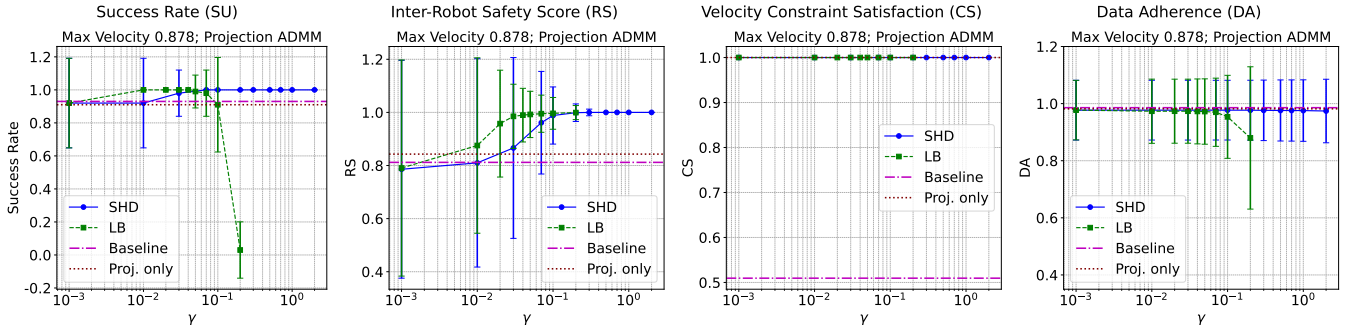


(a) SU ↑ ($N = 2$) (b) RS ↑ ($N = 2$) (c) CS ↑ ($N = 2$) (d) DA ↑ ($N = 2$)

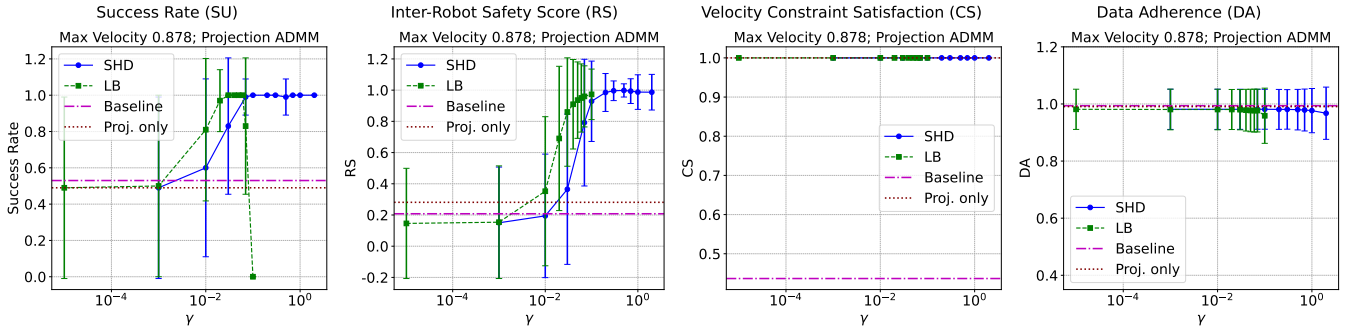


(e) SU ↑ ($N = 4$) (f) RS ↑ ($N = 4$) (g) CS ↑ ($N = 4$) (h) DA ↑ ($N = 4$)

Figure 17: Coupling Strength Ablation of **PCD** on task **`Empty`** with velocity limit $v_{\max} = 0.703$. (a,b,c,d) $N = 2$ robots; (e,f,g,h) $N = 4$ robots.

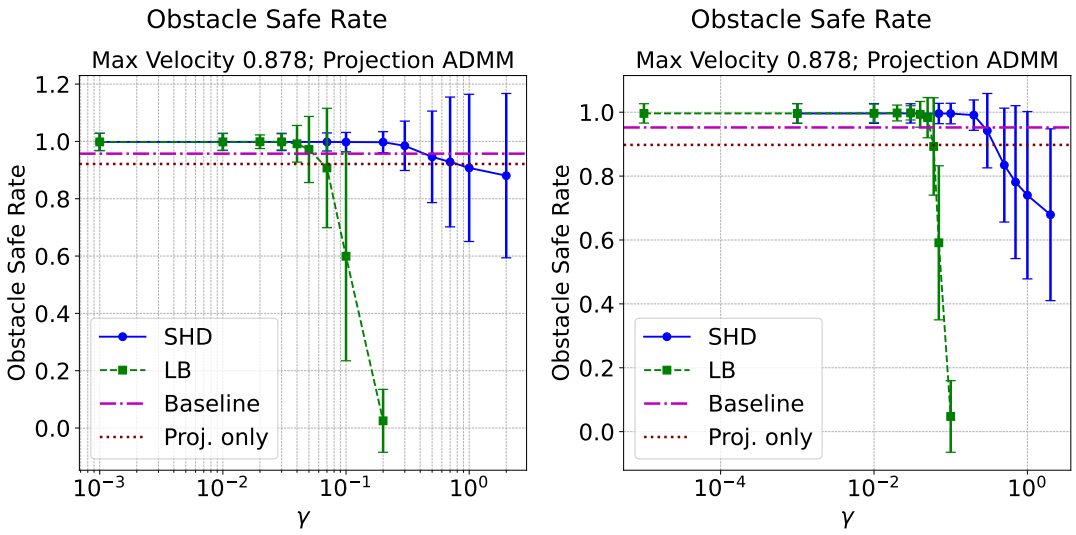


(a) SU ↑ ($N = 2$) (b) RS ↑ ($N = 2$) (c) CS ↑ ($N = 2$) (d) DA ↑ ($N = 2$)



(e) SU ↑ ($N = 4$) (f) RS ↑ ($N = 4$) (g) CS ↑ ($N = 4$) (h) DA ↑ ($N = 4$)

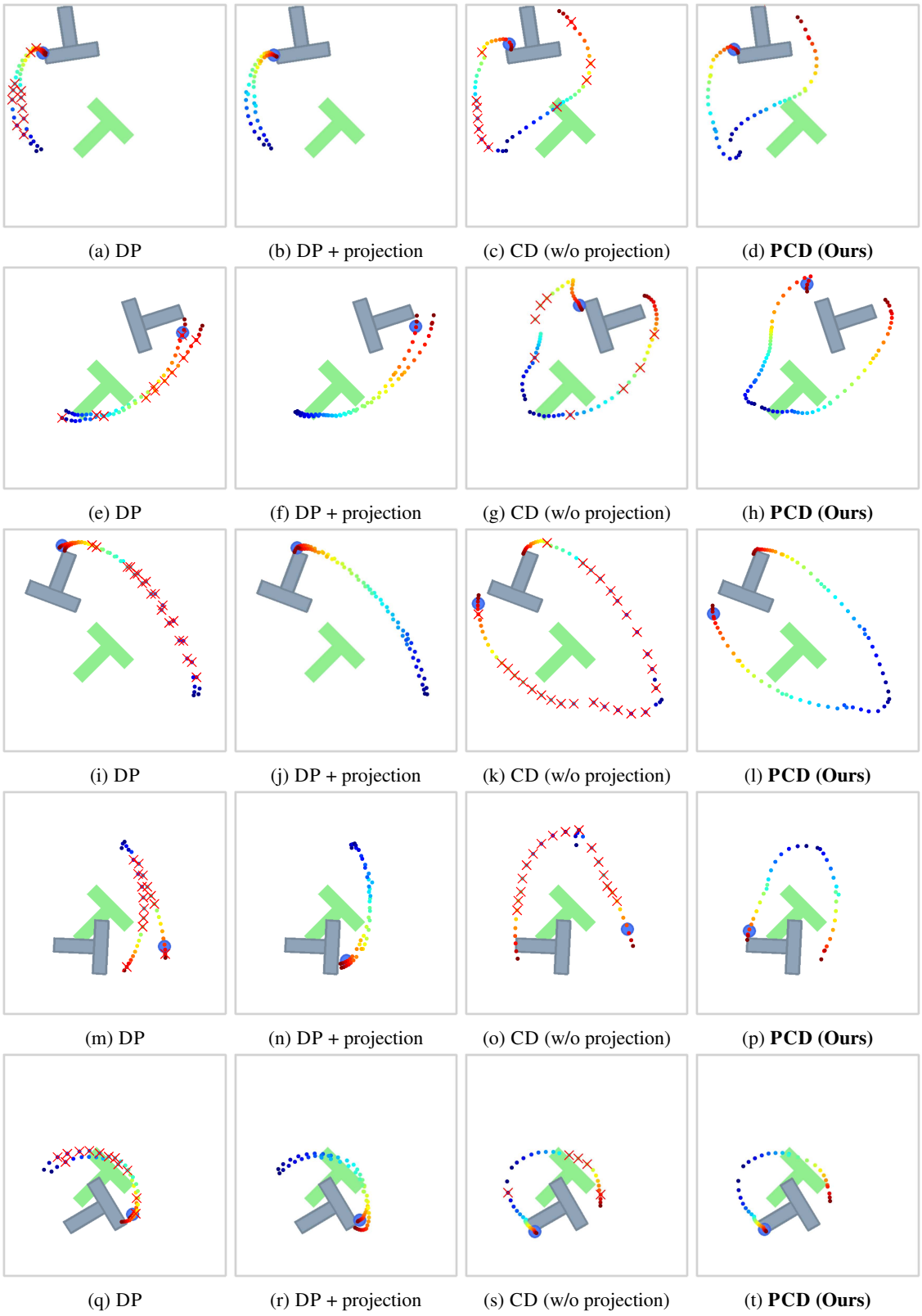
Figure 18: Coupling Strength Ablation of **PCD** on task **`Highways`** with velocity limit $v_{\max} = 0.878$. (a,b,c,d) $N = 2$ robots; (e,f,g,h) $N = 4$ robots.



(a) Obstacle Safe Rate ↑ ($N = 2$)

(b) Obstacle Safe Rate ↑ ($N = 4$)

Figure 19: Obstacle safe rates for the coupling strength ablation of **PCD** on task **`Highways`** with velocity limit $v_{\max} = 0.878$.



(a) DP (b) DP + projection (c) CD (w/o projection) (d) **PCD (Ours)**

(e) DP (f) DP + projection (g) CD (w/o projection) (h) **PCD (Ours)**

(i) DP (j) DP + projection (k) CD (w/o projection) (l) **PCD (Ours)**

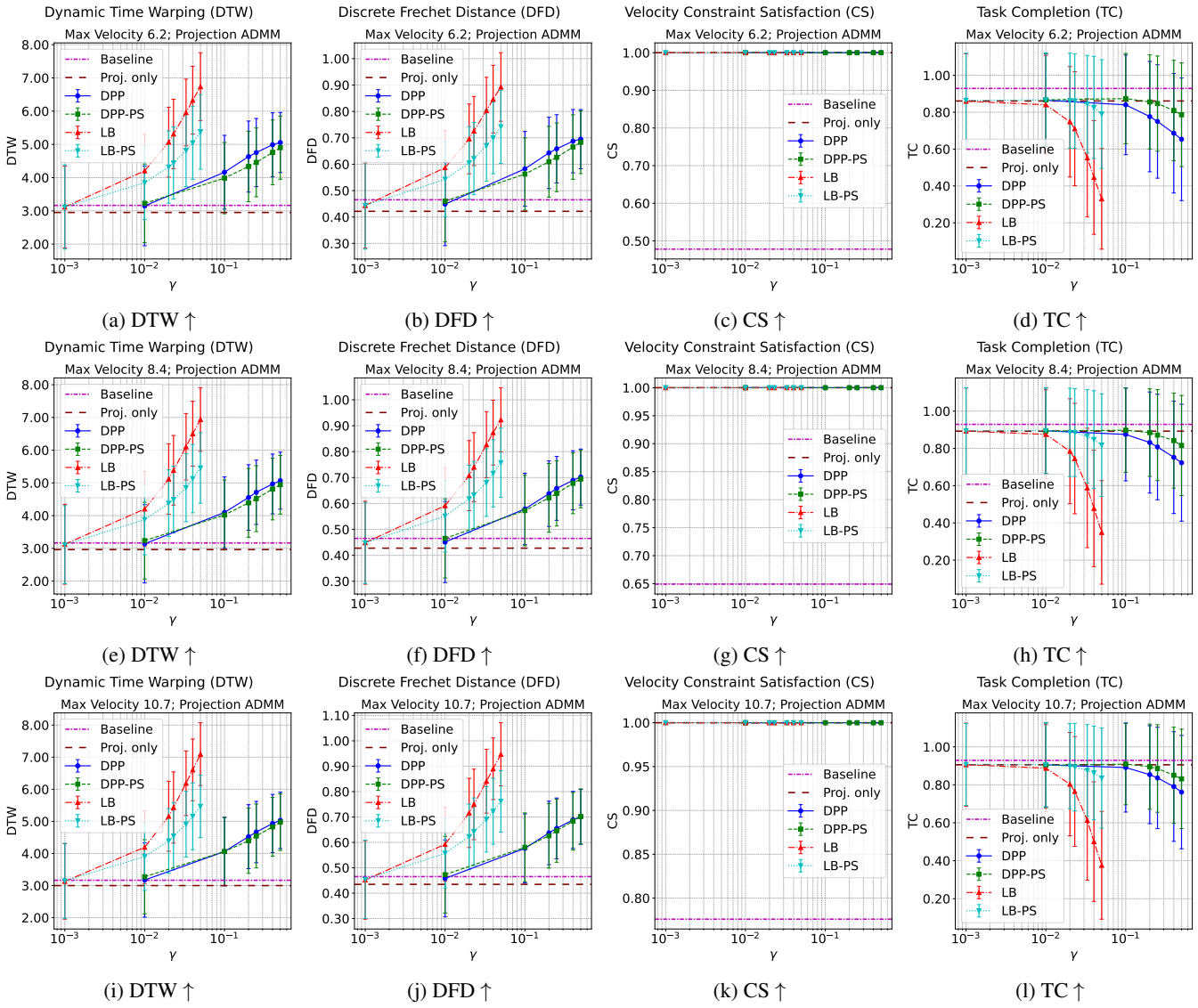
(m) DP (n) DP + projection (o) CD (w/o projection) (p) **PCD (Ours)**

(q) DP (r) DP + projection (s) CD (w/o projection) (t) **PCD (Ours)**

Figure 20: Additional qualitative results of the `PushT` experiment. Rows correspond to initial observations; columns correspond to methods. Robot trajectories use a colormap (warmer colors indicate later time steps); red crosses mark velocity violations. Only the first few dozen steps are shown for clarity.

PushT Experiment Results				
METHOD\Metric	DTW↑	DFD↑	CS(%)↑	TC↑
		Max Vel. = 6.2		
DP	$3.16_{\pm 1.13}$	$0.465_{\pm 0.149}$	$(47.6_{\pm 17.1}, 48_{\pm 17.0})$	$(0.927_{\pm 0.195}, 0.931_{\pm 0.188})$
DP + proj.	$2.95_{\pm 1.26}$	$0.422_{\pm 0.166}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.862_{\pm 0.253}, 0.860_{\pm 0.255})$
CD-DPP	$3.74_{\pm 1.04}$	$0.544_{\pm 0.135}$	$(44.4_{\pm 16.6}, 44.5_{\pm 16.3})$	$(0.923_{\pm 0.206}, 0.922_{\pm 0.204})$
CD-DPP-PS	$4.48_{\pm 0.911}$	$0.648_{\pm 0.114}$	$(39.6_{\pm 15.5}, 39.7_{\pm 15.5})$	$(0.912_{\pm 0.208}, 0.917_{\pm 0.199})$
CD-LB	$4.13_{\pm 1.17}$	$0.596_{\pm 0.151}$	$(41.6_{\pm 15.4}, 41.3_{\pm 15.5})$	$(0.910_{\pm 0.208}, 0.912_{\pm 0.201})$
CD-LB-PS	$4.50_{\pm 1.01}$	$0.646_{\pm 0.129}$	$(41.4_{\pm 16.4}, 41.4_{\pm 16.3})$	$(0.921_{\pm 0.199}, 0.925_{\pm 0.189})$
PCD-DPP	$4.63_{\pm 1.07}$	$0.643_{\pm 0.135}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.776_{\pm 0.301}, 0.777_{\pm 0.298})$
PCD-DPP-PS	$4.34_{\pm 1.06}$	$0.610_{\pm 0.134}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.856_{\pm 0.258}, 0.855_{\pm 0.253})$
PCD-LB	$5.07_{\pm 1.05}$	$0.696_{\pm 0.132}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.747_{\pm 0.3}, 0.750_{\pm 0.3})$
PCD-LB-PS	$4.32_{\pm 1.09}$	$0.605_{\pm 0.138}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.865_{\pm 0.257}, 0.861_{\pm 0.259})$
		Max Vel. = 8.4		
DP	$3.16_{\pm 1.13}$	$0.465_{\pm 0.149}$	$(64.8_{\pm 13.8}, 65_{\pm 13.7})$	$(0.927_{\pm 0.195}, 0.931_{\pm 0.188})$
DP + proj.	$2.96_{\pm 1.21}$	$0.428_{\pm 0.159}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.896_{\pm 0.227}, 0.888_{\pm 0.237})$
CD-DPP	$3.74_{\pm 1.04}$	$0.544_{\pm 0.135}$	$(62.5_{\pm 13.9}, 62.4_{\pm 13.6})$	$(0.923_{\pm 0.206}, 0.922_{\pm 0.204})$
CD-DPP-PS	$4.48_{\pm 0.911}$	$0.648_{\pm 0.114}$	$(57.4_{\pm 14.0}, 57.3_{\pm 13.9})$	$(0.912_{\pm 0.208}, 0.917_{\pm 0.199})$
CD-LB	$4.13_{\pm 1.17}$	$0.596_{\pm 0.151}$	$(58.9_{\pm 13.8}, 58.6_{\pm 14.1})$	$(0.910_{\pm 0.208}, 0.912_{\pm 0.201})$
CD-LB-PS	$4.50_{\pm 1.01}$	$0.646_{\pm 0.129}$	$(58.8_{\pm 14.4}, 58.7_{\pm 14.3})$	$(0.921_{\pm 0.199}, 0.925_{\pm 0.189})$
PCD-DPP	$4.55_{\pm 1.00}$	$0.638_{\pm 0.126}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.829_{\pm 0.272}, 0.834_{\pm 0.271})$
PCD-DPP-PS	$4.39_{\pm 1.05}$	$0.622_{\pm 0.133}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.885_{\pm 0.236}, 0.885_{\pm 0.233})$
PCD-LB	$5.12_{\pm 1.08}$	$0.708_{\pm 0.135}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.778_{\pm 0.288}, 0.791_{\pm 0.275})$
PCD-LB-PS	$4.38_{\pm 1.02}$	$0.618_{\pm 0.129}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.890_{\pm 0.234}, 0.882_{\pm 0.240})$
		Max Vel. = 10.7		
DP	$3.16_{\pm 1.13}$	$0.465_{\pm 0.149}$	$(77.6_{\pm 9.97}, 77.6_{\pm 9.81})$	$(0.927_{\pm 0.195}, 0.931_{\pm 0.188})$
DP + proj.	$3.00_{\pm 1.18}$	$0.435_{\pm 0.155}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.905_{\pm 0.222}, 0.906_{\pm 0.216})$
CD-DPP	$3.74_{\pm 1.04}$	$0.544_{\pm 0.135}$	$(76.2_{\pm 10.3}, 76.2_{\pm 10.1})$	$(0.923_{\pm 0.206}, 0.922_{\pm 0.204})$
CD-DPP-PS	$4.48_{\pm 0.911}$	$0.648_{\pm 0.114}$	$(71.8_{\pm 10.8}, 71.7_{\pm 10.8})$	$(0.912_{\pm 0.208}, 0.917_{\pm 0.199})$
CD-LB	$4.13_{\pm 1.17}$	$0.596_{\pm 0.151}$	$(72.7_{\pm 10.9}, 72.3_{\pm 11.0})$	$(0.910_{\pm 0.208}, 0.912_{\pm 0.201})$
CD-LB-PS	$4.50_{\pm 1.01}$	$0.646_{\pm 0.129}$	$(72.6_{\pm 11.1}, 72.6_{\pm 11.1})$	$(0.921_{\pm 0.199}, 0.925_{\pm 0.189})$
PCD-DPP	$4.52_{\pm 0.996}$	$0.637_{\pm 0.125}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.852_{\pm 0.26}, 0.855_{\pm 0.257})$
PCD-DPP-PS	$4.40_{\pm 1.02}$	$0.626_{\pm 0.127}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.892_{\pm 0.228}, 0.900_{\pm 0.220})$
PCD-LB	$5.16_{\pm 1.09}$	$0.716_{\pm 0.136}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.804_{\pm 0.271}, 0.803_{\pm 0.273})$
PCD-LB-PS	$4.39_{\pm 1.01}$	$0.622_{\pm 0.129}$	$(100_{\pm 0}, 100_{\pm 0})$	$(0.896_{\pm 0.227}, 0.900_{\pm 0.222})$

Table 10: Results of `PushT` task by all compared methods with all three velocity limits. DP refers to DIFFUSION POLICY; CD denotes DP+coupling only; PS denotes posterior sampling variants of cost functions.

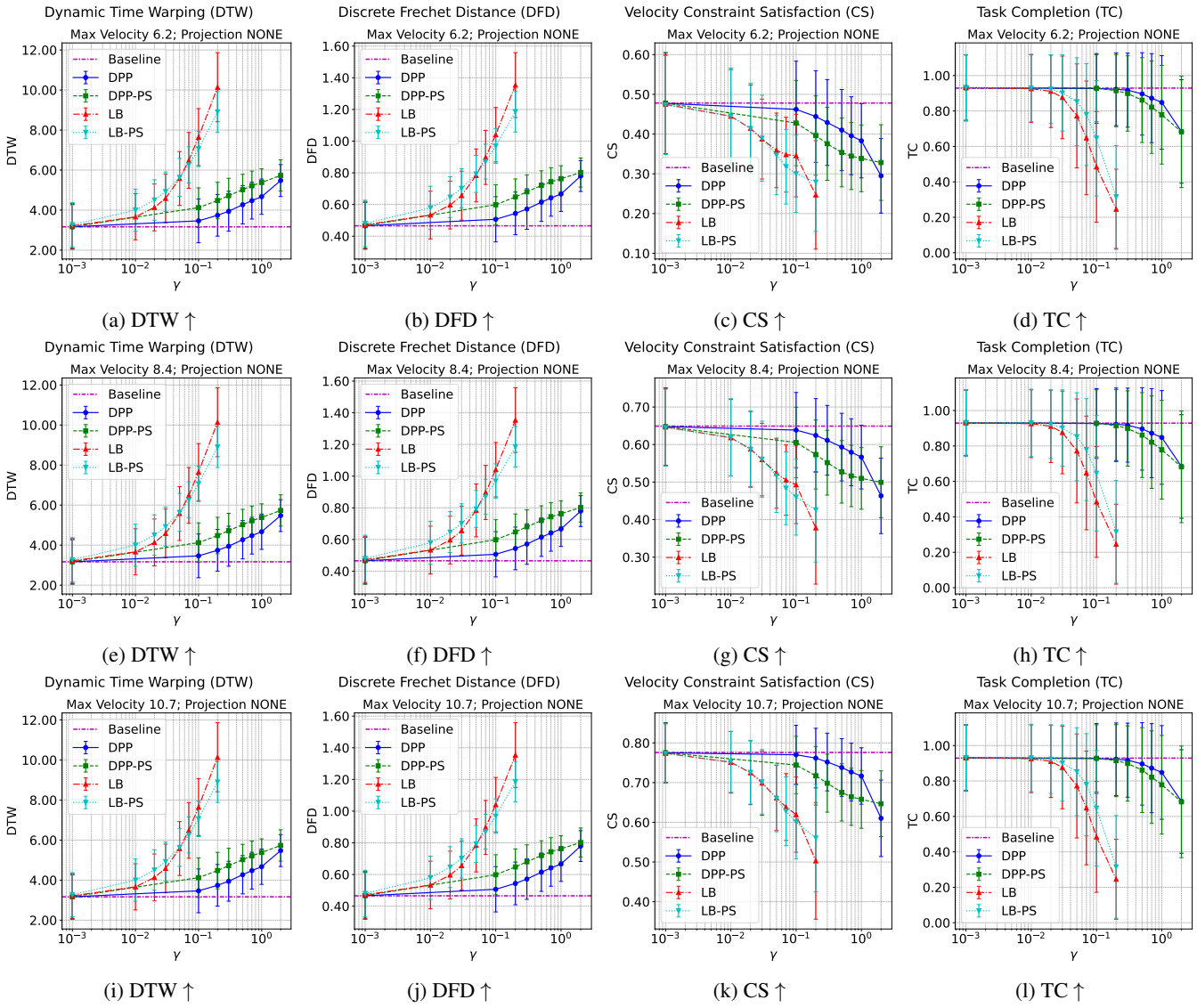


(a) DTW ↑ (b) DFD ↑ (c) CS ↑ (d) TC ↑

(e) DTW ↑ (f) DFD ↑ (g) CS ↑ (h) TC ↑

(i) DTW ↑ (j) DFD ↑ (k) CS ↑ (l) TC ↑

Figure 21: Coupling Strength Ablation of **PCD**: Evaluation metrics results at different $v_{\max}$s as coupling stength γ and coupling function vary. (a,b,c,d) $v_{\max} = 6.2$; (e,f,g,h) $v_{\max} = 8.4$; (i,j,k,l) $v_{\max} = 10.7$.



(a) DTW ↑ (b) DFD ↑ (c) CS ↑ (d) TC ↑

(e) DTW ↑ (f) DFD ↑ (g) CS ↑ (h) TC ↑

(i) DTW ↑ (j) DFD ↑ (k) CS ↑ (l) TC ↑

Figure 22: Coupling Strength Ablation of **CD** (coupling-only, without projection): Evaluation metrics results at different $v_{\max}$s as coupling stength γ and coupling function vary. (a,b,c,d) $v_{\max} = 6.2$; (e,f,g,h) $v_{\max} = 8.4$; (i,j,k,l) $v_{\max} = 10.7$.