Projected Coupled Diffusion for 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]Universiti Malaya [3]Institute of Data Science, National University of Singapore

*Equal contribution {haoluan@comp., seekiong@, chunkail@}nus.edu.sg u2102756@siswa.um.edu.my

ICLR NUS National University of Singapore UNIVERSITI MALAYA

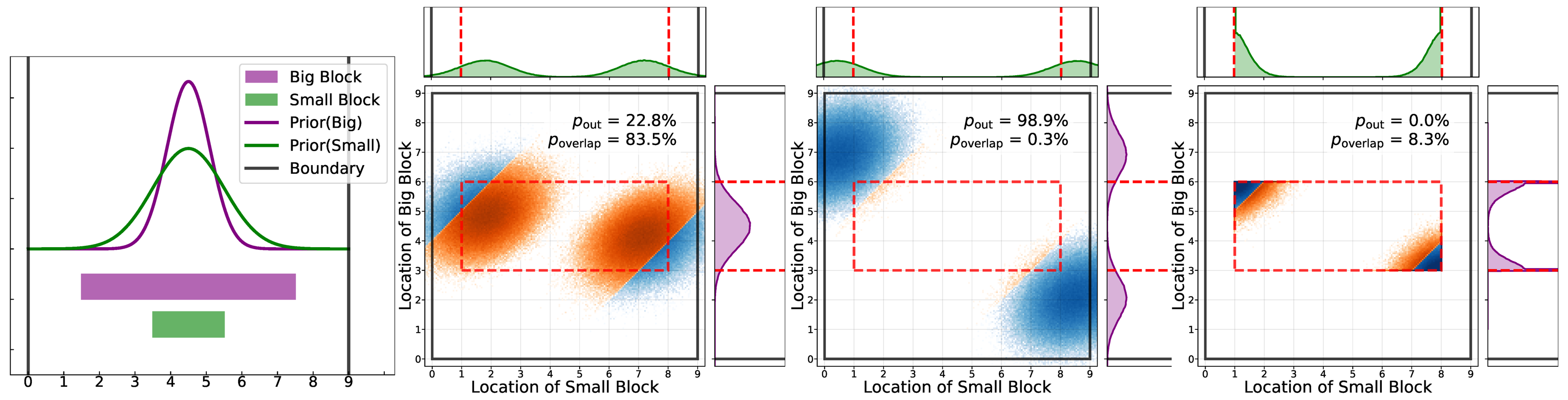
Paper

TL; DR

- Projected Coupled Diffusion (PCD) is a *training-free inference-time framework* enabling pretrained diffusion models to generate *correlated samples* subject to *hard constraints specified at test-time.*
- PCD promotes inter-sample correlations through (neural or analytical) coupling cost functions and enforces test-time hard constraints via projection.
- PCD can be broadly applied in domains including robot manipulation, multi-agent motion planning, and correlated yet constrained image generation.

Why coupling and projection combined?

Training/Retraining joint models over correlated variables is costly and inflexible. A practical alternative models marginals independently and couples them at test time, but coupling alone cannot enforce *hard constraints*. PCD joins coupling and projection: coupling promotes inter-variable correlations via cost functions, while projection enforces hard constraints.



Blocking Corridor Conditional (Big blk first) Coupling only PCD (ours)

A motivating toy example: Two 1D blocks (sizes 6 & 2) *must* fit in a corridor of length 9 minimizing overlaps. Each center is sampled from a Gaussian score model.

- Conditional generation: Generate one block first, then the other block conditioned on it via guidance. One block ignores the other, leaving high overlap probability.
- Mutual coupling: Both blocks mutually influence each other, pushing them to opposite sides with lower overlap, but the corridor boundary can be violated.

The Projected Coupled Diffusion (PCD) Framework

Diffusion sampling from an LMC perspective:

Given a learned score $s_X^\theta \approx \nabla_x \log p_X(x)$, Langevin Monte Carlo (LMC) iterates as:

$$X_{t+1} = X_t + \delta\, \mathbf{s}_X^\theta(X_t, t) + \varepsilon_{X,t}$$

Coupled LMC via a coupling cost function:

$$X_{t+1} = X_t - \gamma \cdot \delta\, \nabla_x\, c(X_t, Y_t) + \delta\, \mathbf{s}_X^\theta(X_t, t) + \varepsilon_{X,t}$$

$$Y_{t+1} = Y_t - \gamma \cdot \delta\, \nabla_y\, c(X_t, Y_t) + \delta\, \mathbf{s}_Y^\varphi(Y_t, t) + \varepsilon_{Y,t}$$

Posterior cost variant:

$$c_{\mathrm{PS}}(x, y, t) = c\left(X_t + \sigma_{X,t}^2\, \mathbf{s}_X^\theta, Y_t + \sigma_{Y,t}^2\, \mathbf{s}_Y^\varphi\right)$$

Projected Coupled Diffusion (PCD):

$$X_{t+1} = \Pi_{\mathrm{K}_X}\left(X_t - \gamma \cdot \delta\, \nabla_x\, c(X_t, Y_t) + \delta\, \mathbf{s}_X^\theta(X_t, t) + \varepsilon_{X,t}\right)$$

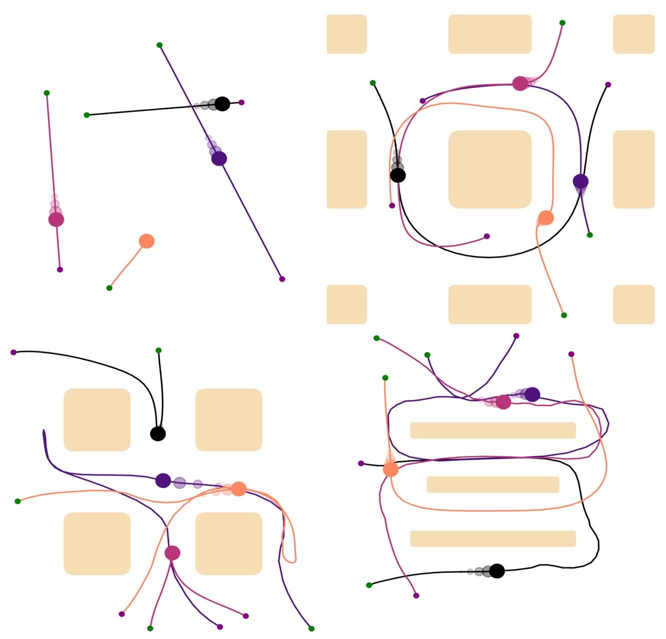
$$Y_{t+1} = \Pi_{\mathrm{K}_Y}\left(Y_t - \gamma \cdot \delta\, \nabla_y\, c(X_t, Y_t) + \delta\, \mathbf{s}_Y^\varphi(Y_t, t) + \varepsilon_{Y,t}\right)$$

where K_X, K_Y are the nonempty feasible sets and the projection is defined as

$$\Pi_{K_X}(x) \coloneqq \operatorname*{argmin}_{z \in K_X} \|z - x\|$$

Multi-Robot Motion Planning

App. #1: Generate collision-free robot trajectories with velocity constraints.



Task Demonstrations

Collision avoidance coupling costs:

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

- Obstacle cost follows MPD (Carvalho et al.) using signed distance to the closest obstacle.
- Inter-robot collision cost uses one of the following:
  - $c_{\mathrm{LB}} = -\sum_{h=1}^{H} \log(\|X_h - Y_h\| + \alpha),\ \ \alpha > 0$
  - $c_{\mathrm{SHD}} = \sum_{h=1}^{H} \left(\mathbf{1}[\|X_h - Y_h\| \le r] \cdot (r - \|X_h - Y_h\|)\right)^2,\ r > 0$

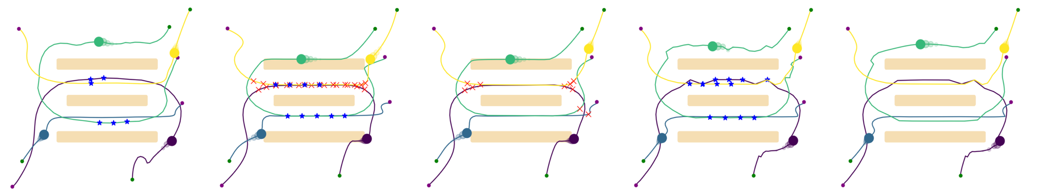
ADMM-based projector for velocity constraint:

The projection is formulated as a *convex* optimization:

$$\min_X \|X - \hat{X}\|_F^2$$

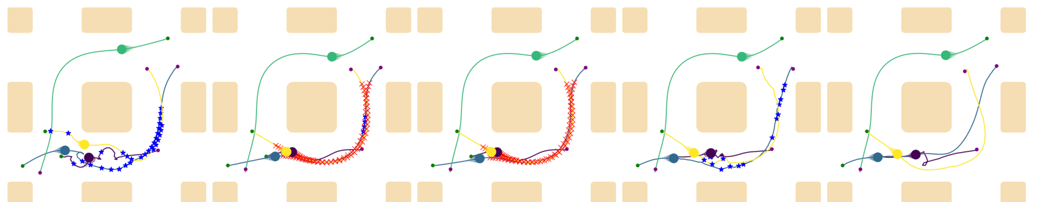
$$\text{s.t. } \|x_0 - X_1\| \le v_{\max} \Delta t$$

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



MMD-CBS Diffuser PD CD (w/o proj.) PCD (Ours)

Compared baselines incur constraint violations or inter-robot collisions.



MMD-CBS Diffuser PD CD (w/o proj.) PCD (Ours)

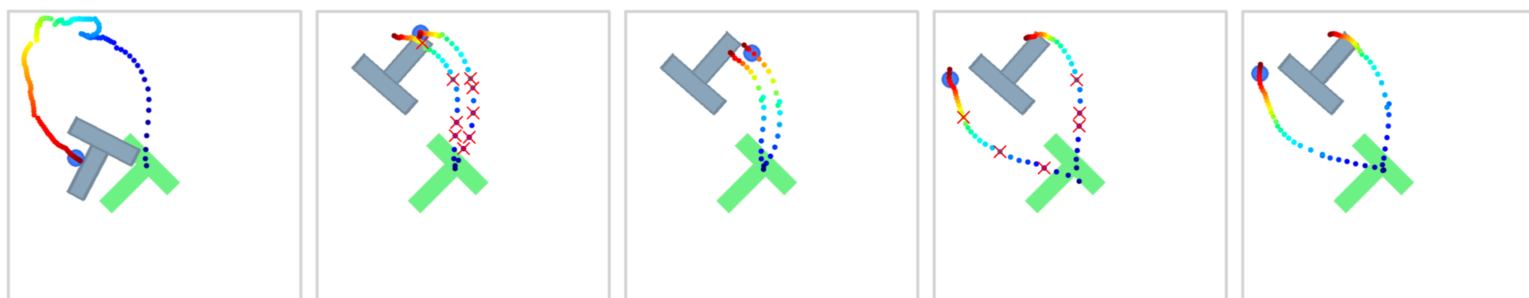
Our PCD can effectively generate collision-free trajectories that enforce velocity constraints and complete the original tasks.

Task Conveyor, 4 Robots				
Method \ Metric	SU(%)↑	RS↑	CS(%)↑	DA↑
			Max Vel. = 1.21	
vanilla Diffuser	$1._{\pm 9.9}$	$.074_{\pm .26}$	$(.53_{\pm 7.3}, .6_{\pm 7.7}, .34_{\pm 5.9}, .23_{\pm 4.8})$	$(.96_{\pm .21}, .96_{\pm .2}, .95_{\pm .21}, .97_{\pm .16})$
MMD-CBS*	$79_{\pm 41}$	$.99_{\pm .10}$	$(0._{\pm 0.}, 0._{\pm 0.}, 0._{\pm 0.}, 0._{\pm 0.})$	$(.99_{\pm .10}, .96_{\pm .20}, .99_{\pm .10}, .99_{\pm .10})$
Diffuser + projection	$0._{\pm 0.}$	$.11_{\pm .32}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.73_{\pm .44}, .72_{\pm .45}, .70_{\pm .46}, .75_{\pm .43})$
CD-LB (w/o proj.)	$100_{\pm 0}$	$.99_{\pm .10}$	$(0._{\pm 0.}, 0._{\pm 0.}, 0._{\pm 0.}, .023_{\pm 1.5})$	$(.89_{\pm .31}, .91_{\pm .28}, .90_{\pm .30}, .89_{\pm .32})$
CD-SHD (w/o proj.)	$100_{\pm 0.}$	$1._{\pm 0.}$	$(.078_{\pm 2.8}, .047_{\pm 2.2}, .039_{\pm 2.}, .023_{\pm 1.5})$	$(.98_{\pm .14}, .98_{\pm .14}, .98_{\pm .13}, .98_{\pm .13})$
PCD-LB	$95_{\pm 22}$	$.84_{\pm .36}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.80_{\pm .40}, .81_{\pm .40}, .80_{\pm .40}, .79_{\pm .41})$
PCD-SHD	$100_{\pm 0}$	$.93_{\pm .26}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.91_{\pm .28}, .90_{\pm .30}, .90_{\pm .30}, .89_{\pm .31})$

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 + projection	$54_{\pm 50}$	$.21_{\pm .41}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(1._{\pm .052}, .98_{\pm .15}, 1._{\pm .07}, 1._{\pm .047})$
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	$100_{\pm 0}$	$1._{\pm .063}$	$(100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0}, 100_{\pm 0})$	$(.98_{\pm .13}, .96_{\pm .19}, .99_{\pm .12}, .99_{\pm .076})$

Diversifying Object Manipulation Solutions

App. #2: Generate diverse motions for robotic object manipulation.



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

Spatial-temporal diversity coupling costs:

- Log-barrier (LB): same as App #1.
- Determinantal Point Process (DPP) cost:

$$c_{\mathrm{DPP}}(X, Y) = \log(\cos\angle(\tilde{X}, \tilde{Y}) + \varepsilon)$$

where $\tilde{X}, \tilde{Y}$ are flattened trajectories and $\varepsilon > 1$.

Constraint:

Velocity constraint on the end-effector (EF) of a robot arm, enforced by ADMM projector (same as App #1).

Method	DTW↑	DFD↑	CS(%)↑	TC↑
DP	3.2	.47	(65, 65)	(**.93**, **.93**)
DP + proj.	3.0	.43	(**100**, **100**)	(.90, .89)
CD-DPP	3.7	.54	(63, 62)	(.92, .92)
CD-DPP-PS	4.5	.65	(57, 57)	(.91, .92)
CD-LB	4.1	.60	(59, 59)	(.91, .91)
CD-LB-PS	4.5	.65	(59, 59)	(.92, .93)
PCD-DPP	4.6	.64	(**100**, **100**)	(.83, .83)
PCD-DPP-PS	4.4	.62	(**100**, **100**)	(.89, .89)
PCD-LB	**5.1**	**.71**	(**100**, **100**)	(.78, .79)
PCD-LB-PS	4.4	.62	(**100**, **100**)	(.89, .88)

PCD is effective in both promoting diversity and enforcing test-time EF velocity constraints.

Image Pair Generation

App. #3: Faces w/ age-group contrast & facial attribute constraints.

(a) Exemplars

Age-group Contrastive Coupling cost:

A *latent* age-group classifier predicts $p(a|X)$ for $a \in \{\text{Young}, \text{Old}\}$. The XOR cost encourages one to be young and the other old:

$$c_{\mathrm{XOR}}(X, Y) = -\sum_{a \in \{\mathrm{Y}, \mathrm{O}\}} p(a|X)(1 - p(a|Y)) + p(a|Y)(1 - p(a|X))$$

Minimizing c_{XOR} pushes the pair toward *opposite* age groups.

Exemplar-based Projection with Mirror Descent:

For each model, encode structurally similar exemplar images via VAE and form a convex hull as the feasible region in the latent space. At each diffusion step, project latents onto this convex hull via mirror descent.

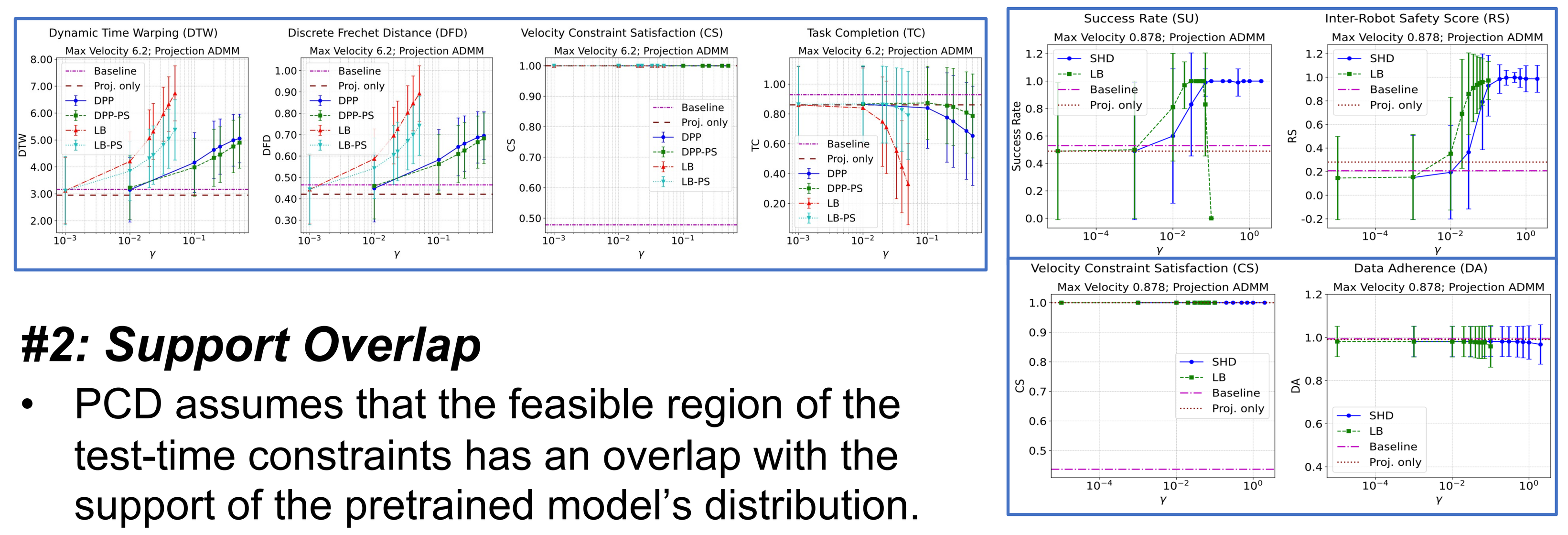
(b) SD (c) Sync[†] (d) SD+C (e) SD+C[†] (f) CNet[†] (g) SD+P (h) **PCD (Ours)**

Compared Methods:
- SD: Stable Diffusion
- Sync: SyncDiffusion
- CNet: ControlNet-XS
- +C: w/ Coupling
- +P: w/ Projection
- †: w/ text prompt

Ablations & Discussions

#1: Sensitivity to Coupling Strength γ

- Higher γ in general improves correlation metrics such as robot safety (RS) or trajectory diversity (DTW/DFD) but degrades data adherence to the original pretrained distribution (DA/TC). Sensitivity to values depends on cost function instances; PS variants appear to offer a better tradeoff.
- Bottomline is that γ never affects constraint satisfaction (CS).

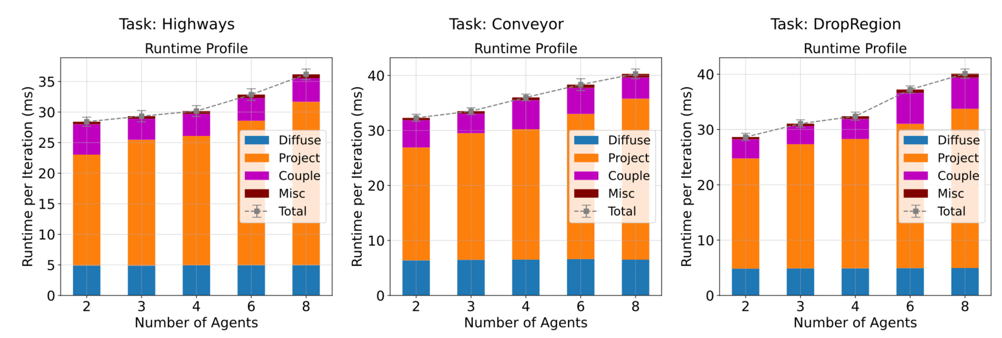


#2: Support Overlap

- PCD assumes that the feasible region of the test-time constraints has an overlap with the support of the pretrained model's distribution.

#3: Runtime Scalability

- Time complexity: Runtime scales *linearly* with number of agents N, number of projection iterations, and number diffusion steps T.
- Projection takes up a major part of the runtime. Efficient projectors are crucial.



Discuss further?

haoluan@ comp.nus.edu.sg

Code