Ph. D. Thesis Defense

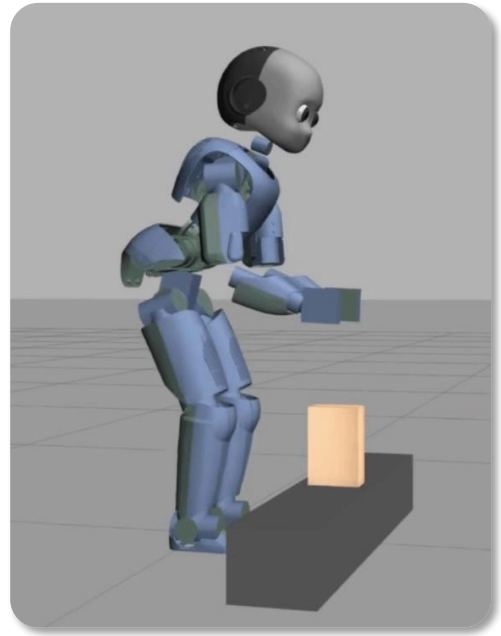
Collision Distance Estimation for High-dof Robot Systems: A Learning-based Approach

Jihwan Kim
jihwankim@robotics.snu.ac.kr

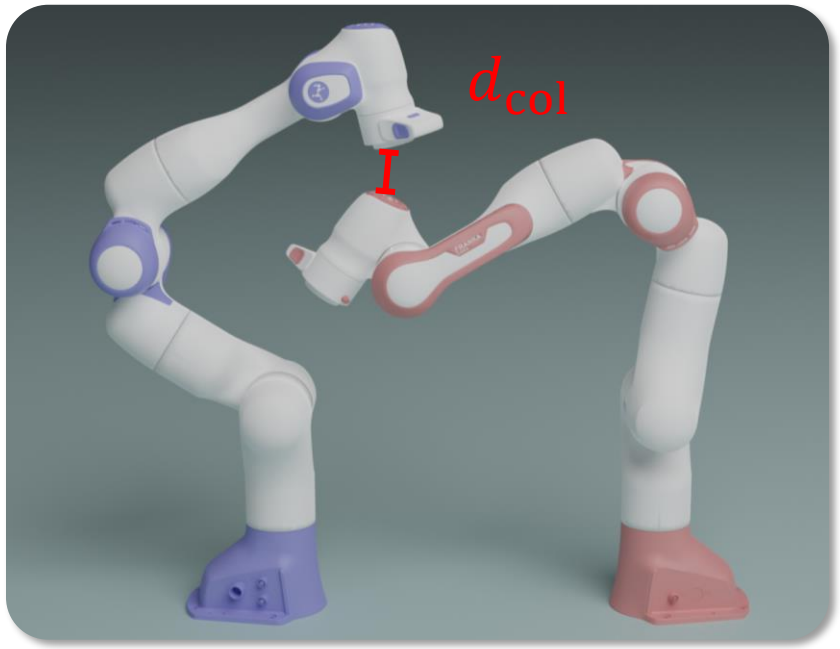
Supervisor: Prof. Frank C. Park

2024. 11. 01

Collision Checking



Collision Distance



: The minimum distance between robots and obstacles
(including other robot links for self-collision)

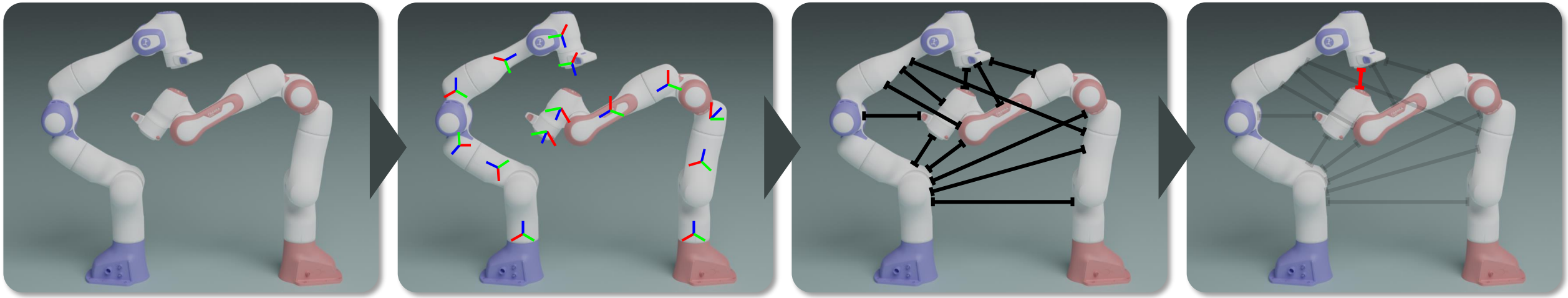
Collision Distance

The generalized coordinates
$q \in \mathbb{R}^{\mathrm{DOF}}$

Positions of elements
$\{T_i(q)\}_{i=1}^{M} = kinematics(q)$

Distances between elements
$d_{ij}(q) = dist\left(\mathcal{M}_i, \mathcal{M}_j, T_i, T_j\right)$

Collision distance
$d_{\mathrm{col}}(q) = \min_{(i,j)} d_{ij}(q)$

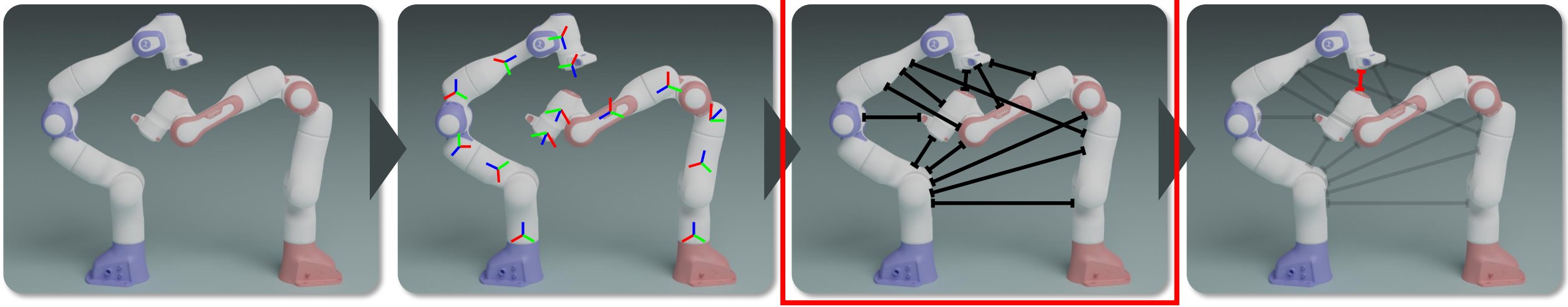


Collision Distance

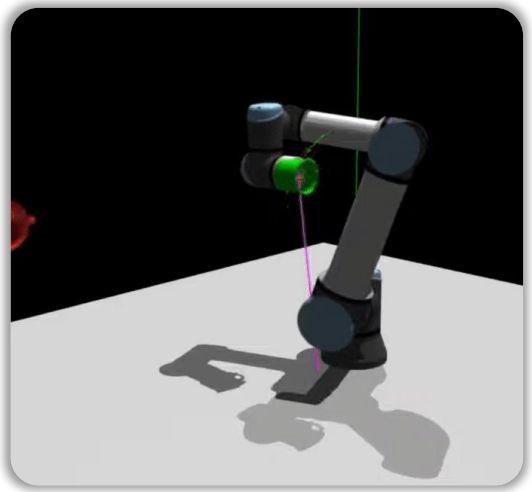
The collision distance function $d_{\mathrm{col}}(q)$

: takes q as inputs and outputs the collision distance of the system.

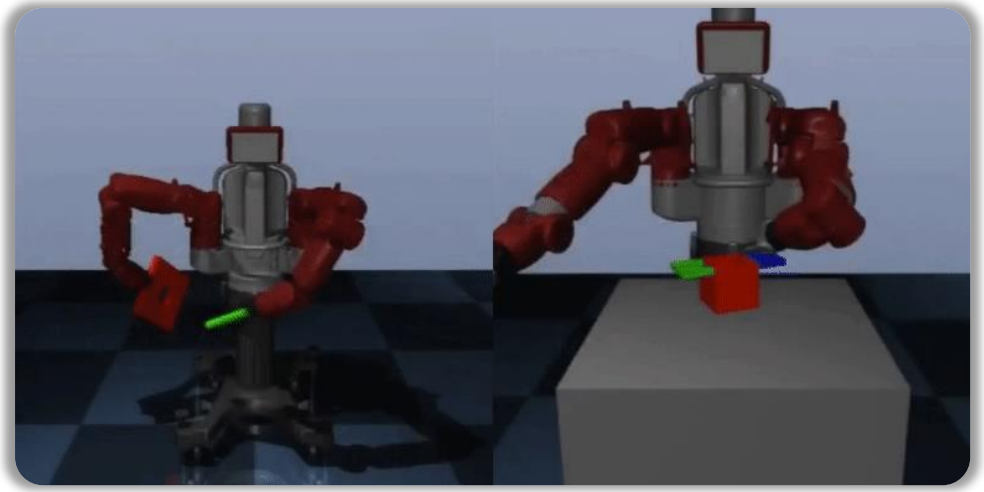
$$d_{ij} = dist(\mathcal{M}_i, \mathcal{M}_j, T_i, T_j)$$



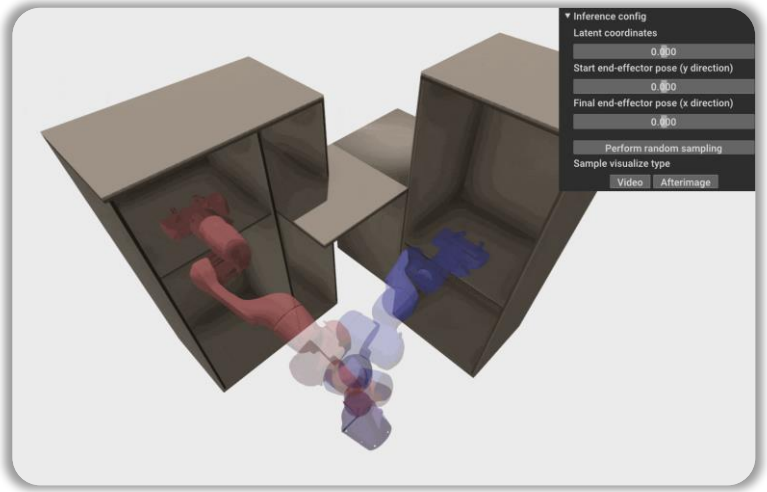
Modern Robot Planning



Sampling-based MPC [1]



Reinforcement Learning [2]



Generative model-based Planning [3]
(Path Generation)

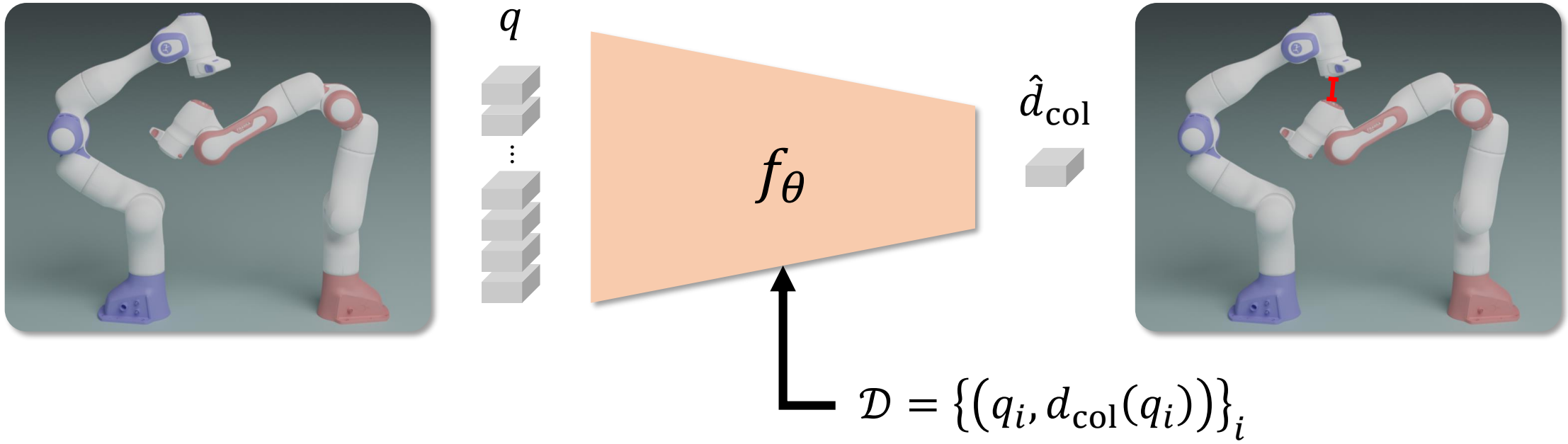
Requirements: Rapid, Batch, and Derivative Calculations

[1] Bhardwaj, Mohak, et al. "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation." Conference on Robot Learning. PMLR, 2022.
[2] Fan, Linxi, et al. "Surreal: Open-source reinforcement learning framework and robot manipulation benchmark." *Conference on robot learning*. PMLR, 2018.
[3] Lee, Yonghyeon. "MMP++: Motion Manifold Primitives With Parametric Curve Models." IEEE Transactions on Robotics (2024).

Collision Distance with Machine Learning

The **trained** collision distance function $f_\theta(q)$

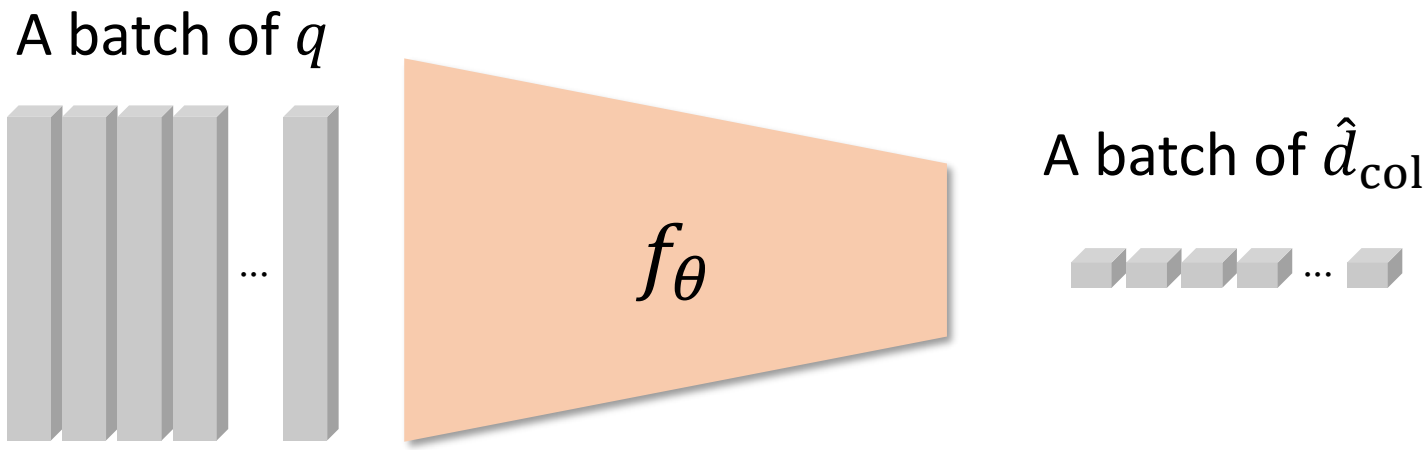
: takes q as inputs and predicts the collision distance of the system.



Collision Distance **with Machine Learning**

The **trained** collision distance function $f_\theta(q)$

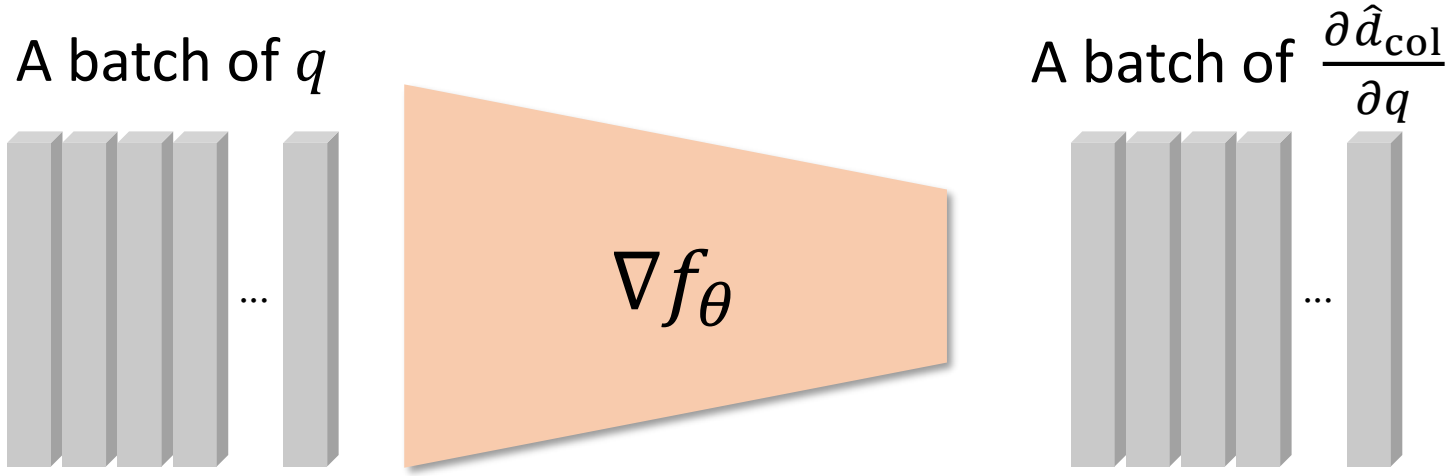
: takes q as inputs and predicts the collision distance of the system.



Collision Distance **with Machine Learning**

The **trained** collision distance function $f_\theta(q)$

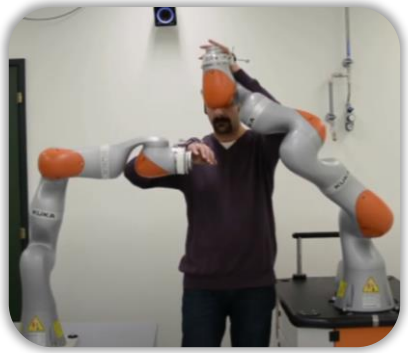
: takes q as inputs and predicts the collision distance of the system.



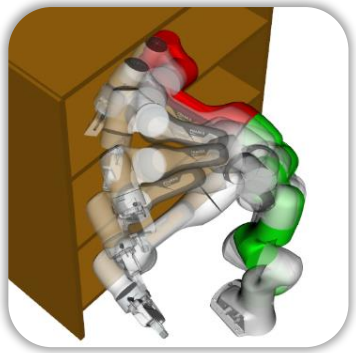
Collision Distance with Machine Learning

The **trained** collision distance function $f_\theta(q)$

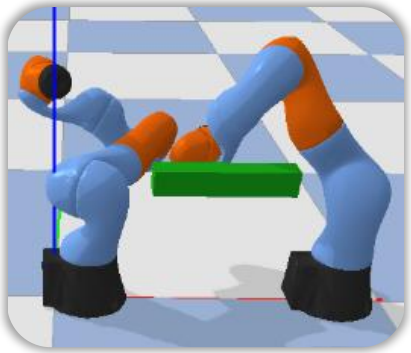
: takes q as inputs and predicts the collision distance of the system.



SCA (SVM)
(N. B. Figueroa Fernandez, et al., MLPC'18)

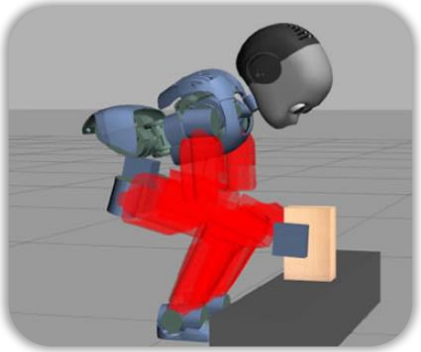


DiffCo
(Y. Zhi, et al., TRO'22)

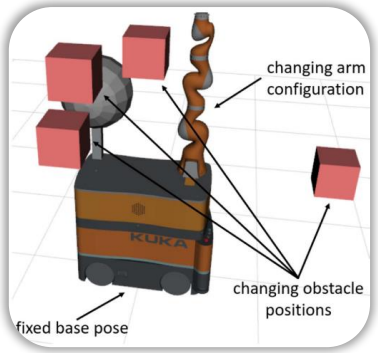


ClearanceNet
(J. Chase Kew, et al., WAFR'20)

jointNERF
(M. Bhardwaj, et al., CoRL'22)

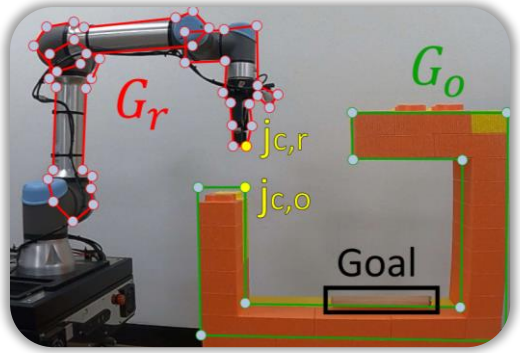


SCA (SVM, NN)
(M. Koptev, et al., RAL'21)



CollisionGP
(J. Muñoz, et al., RAL'23)

RelaxedIK
(D. Rakita, et al., RSS'18)



GraphDistNet
(Y. Kim, et al., RAL'22)

Contributions

- Dataset Construction → Active Learning of the Collision Distance Function (Kim, J., & Park, F. C., Robotica'24)
- Inherent Complexity of the Collision Distance Function
- Generalizability to Minor Environmental Changes → PairwiseNet (Kim, J., & Park, F. C., CoRL'23)

Kim, Jihwan, and Frank C. Park. "Active learning of the collision distance function for high-DOF multi-arm robot systems." Robotica (2024): 1-15.
Kim, Jihwan, and Frank C. Park. "PairwiseNet: Pairwise Collision Distance Learning for High-dof Robot Systems." Conference on Robot Learning. PMLR, 2023.

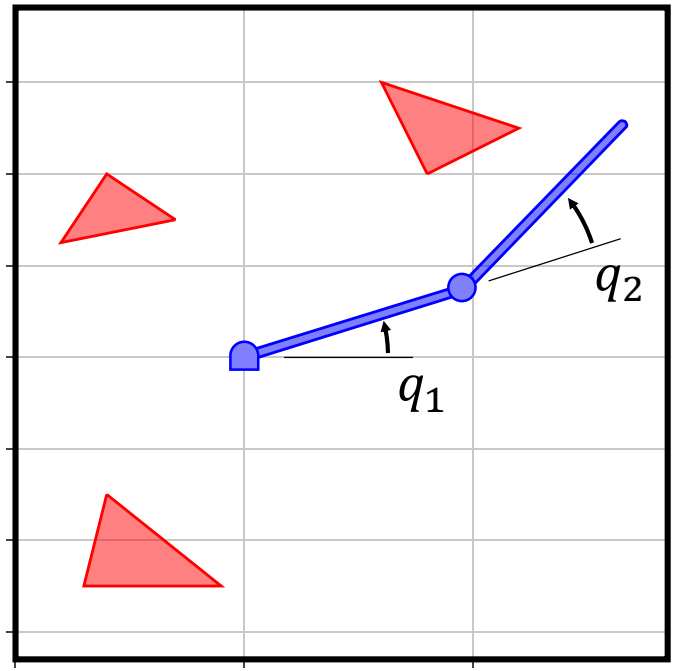
Chapter

Active Learning of the Collision Distance Function

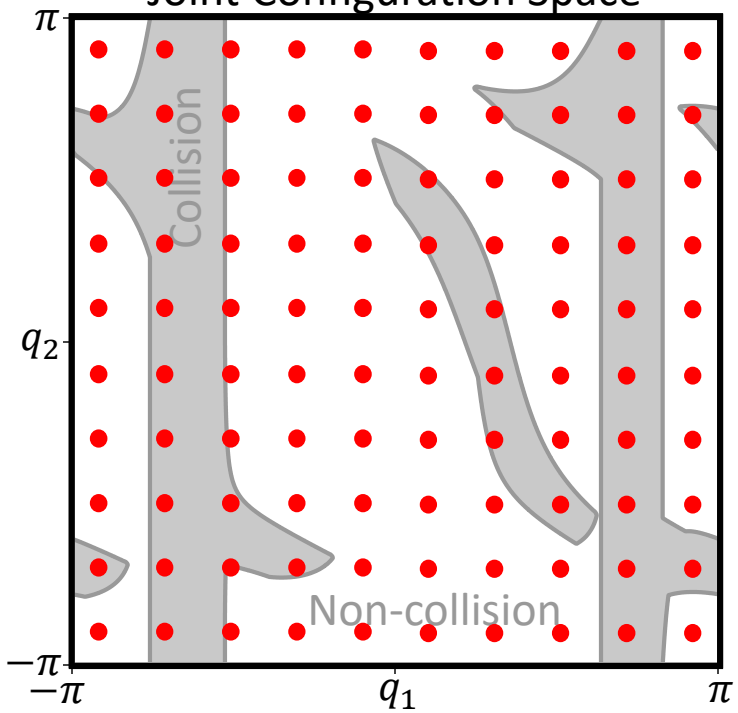
Dataset Construction

: Most methods rely on uniform sampling in the joint configuration space for dataset construction

ex) A 2R planar robot system

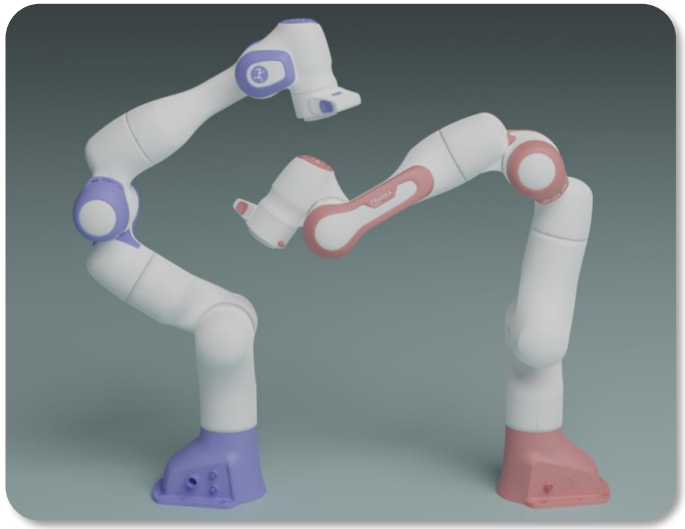


Joint Configuration Space



10 data points per joint
= 10^2 data points

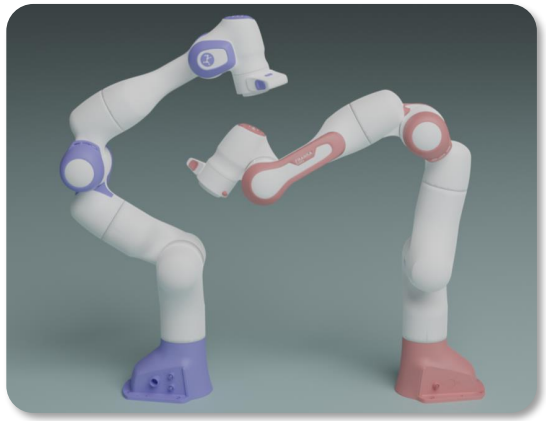
For 14-dof, 10^{14} points?



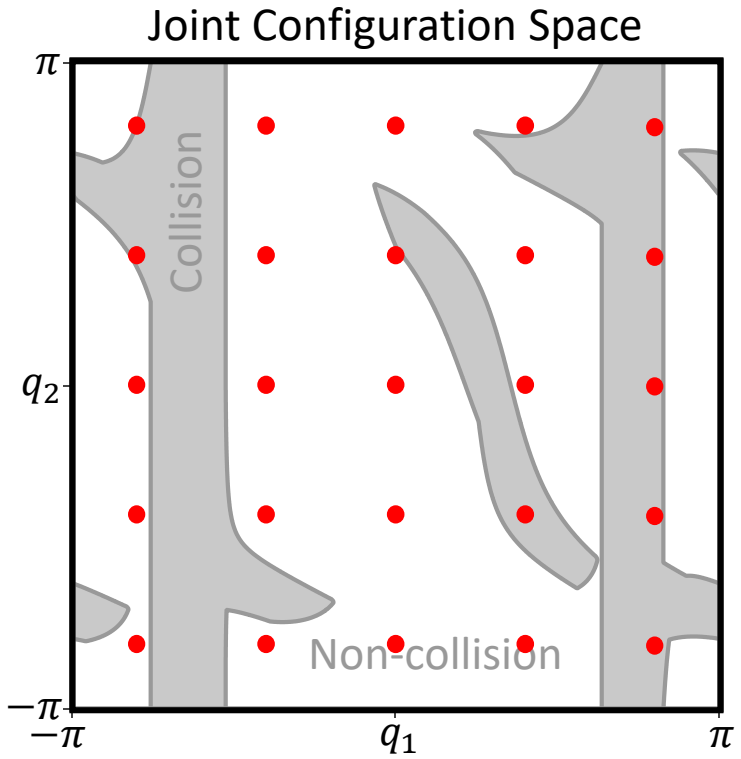
Dataset Construction

Most methods utilize datasets of less than a million samples.

→ For a 14-dof two-arm system, a million samples only provide approximately $\sqrt[14]{10^6} \approx 2.68$ samples per joint.

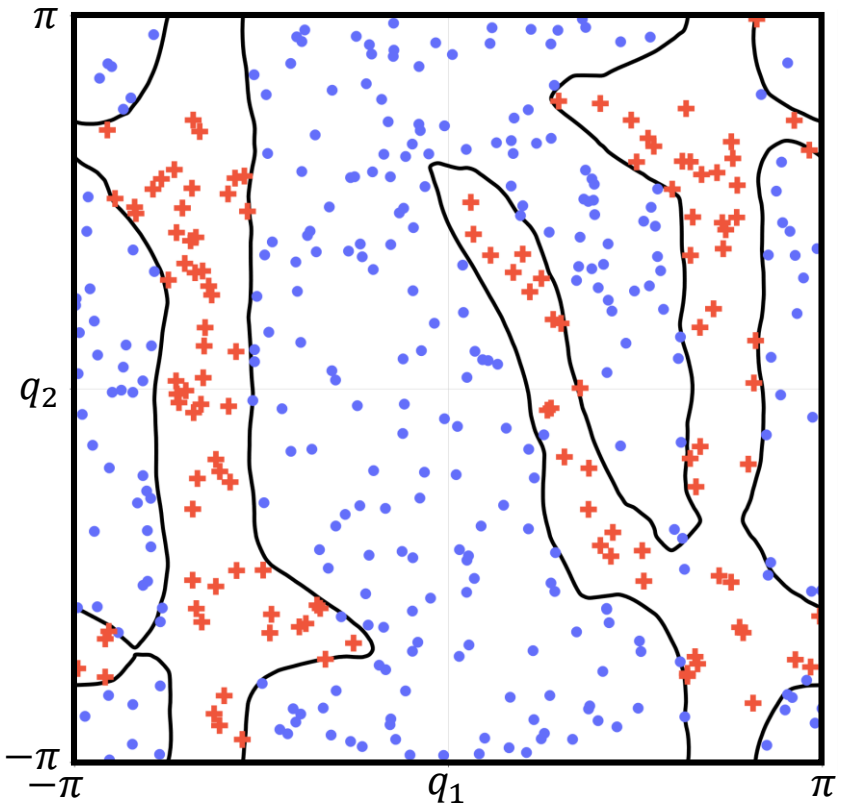


Same interval length



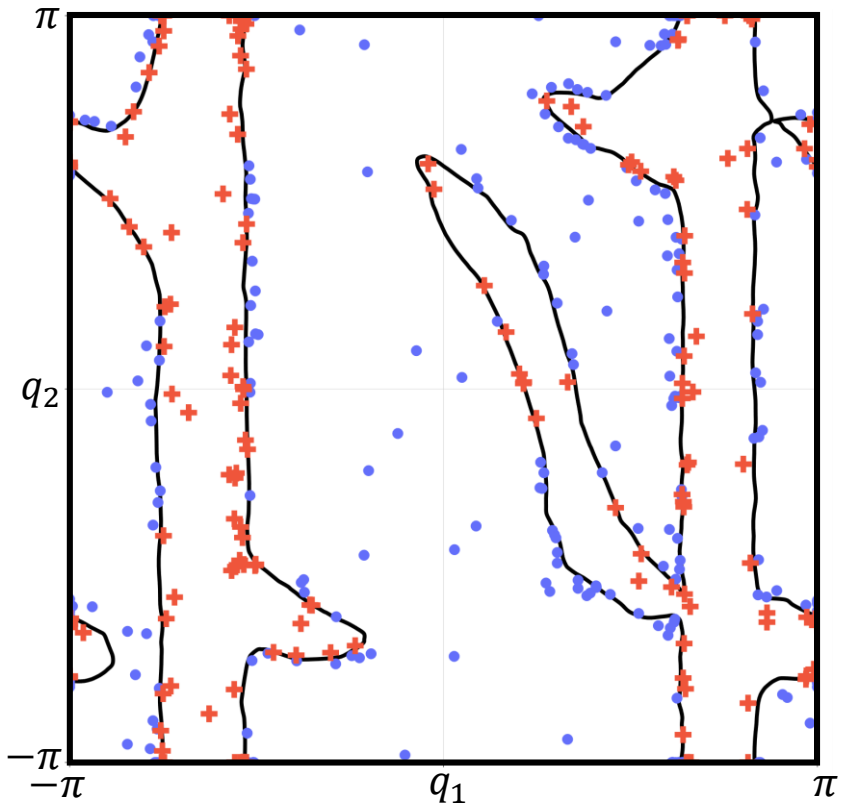
Key insight

Uniform Sampling

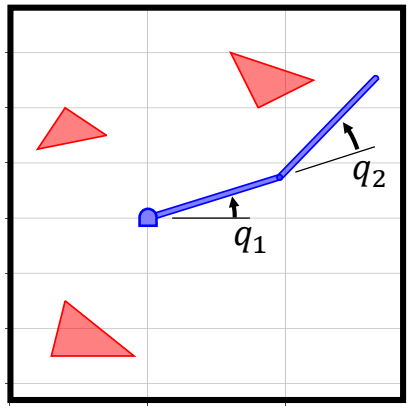


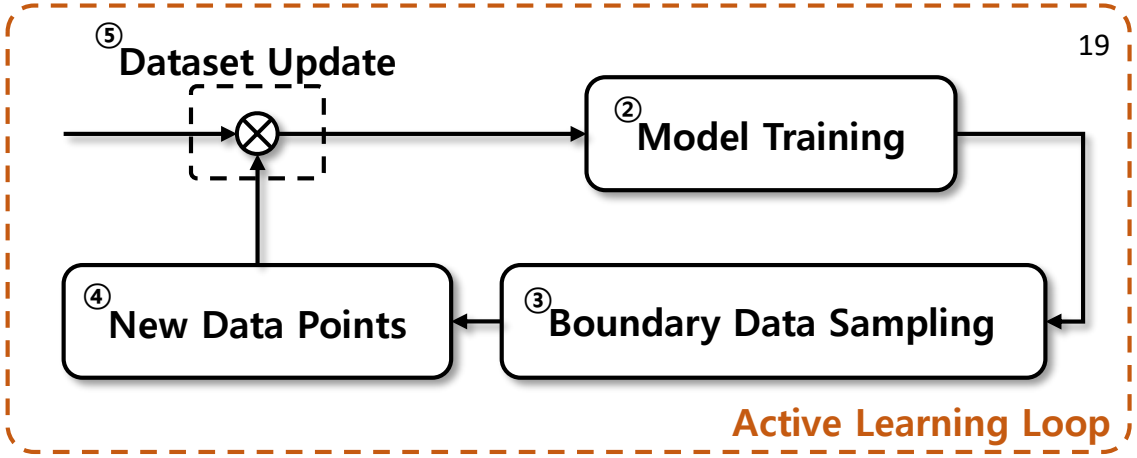
<

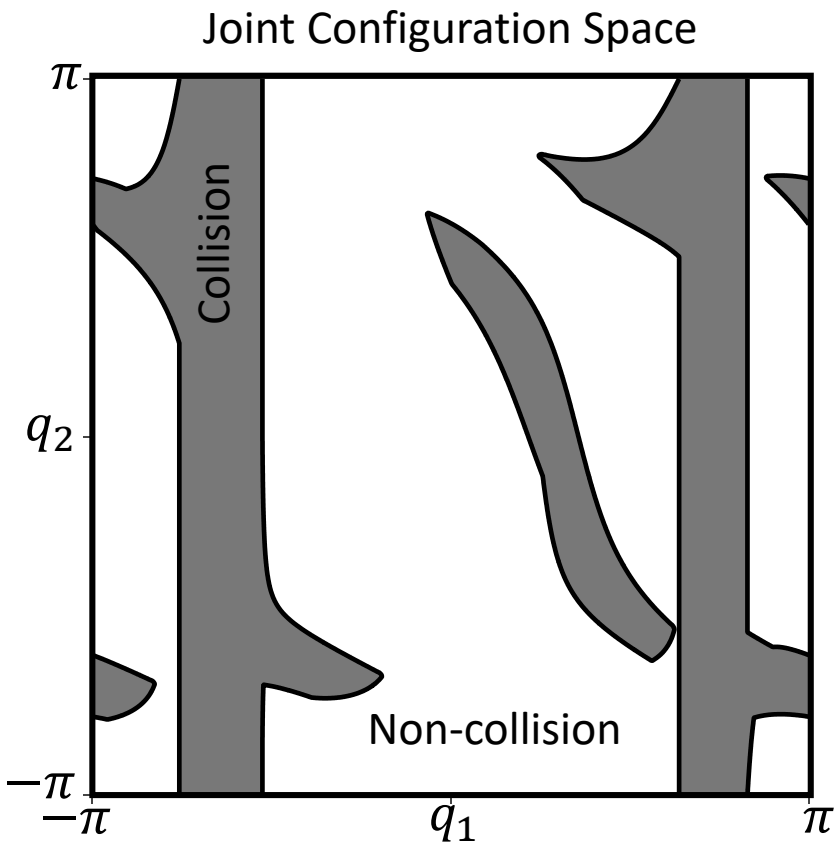
Near-boundary Sampling

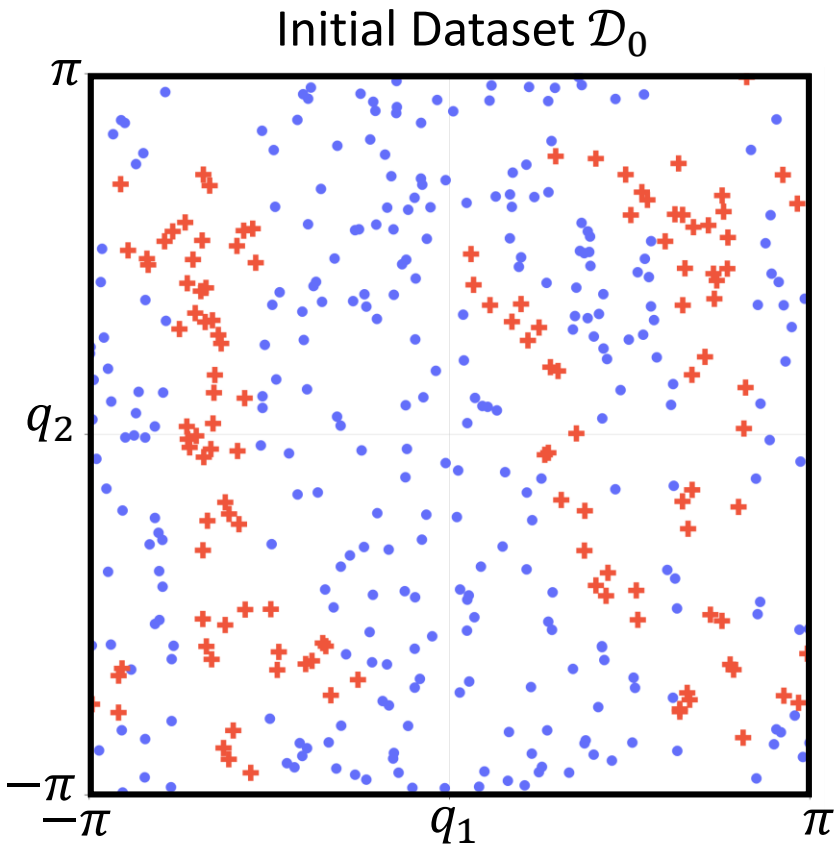


2R Planar Robot System

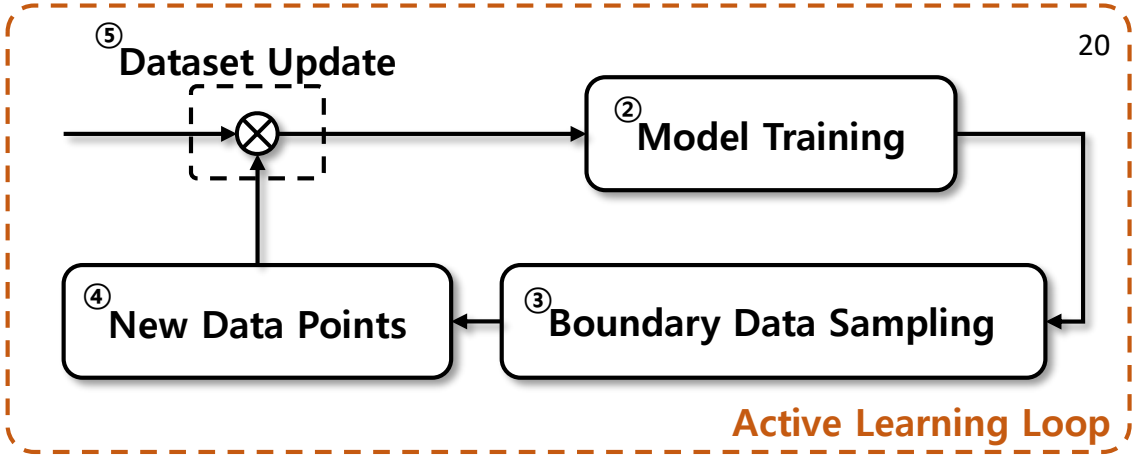




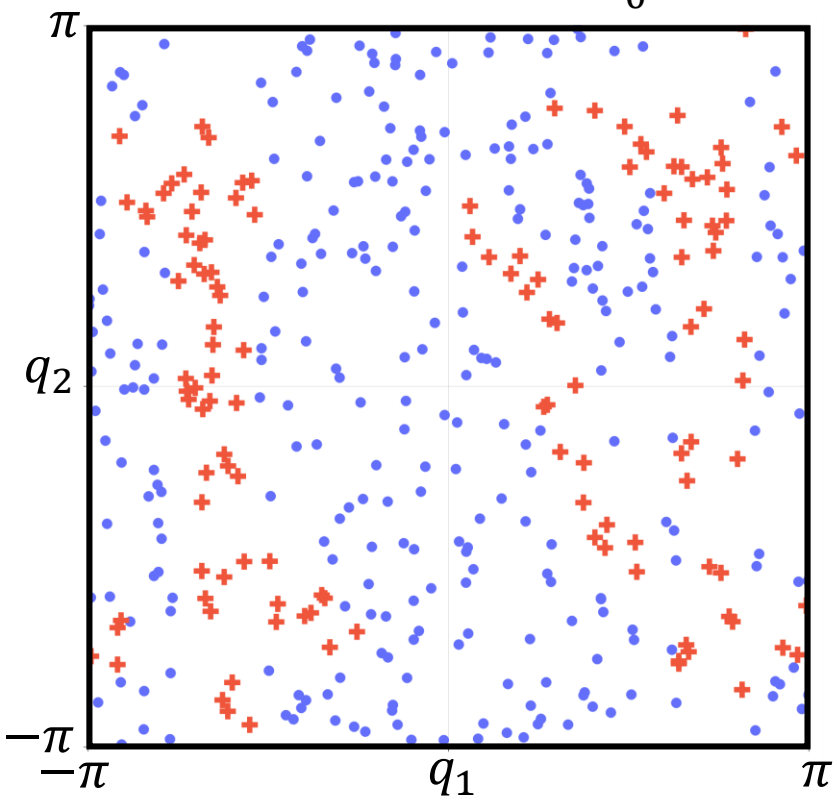




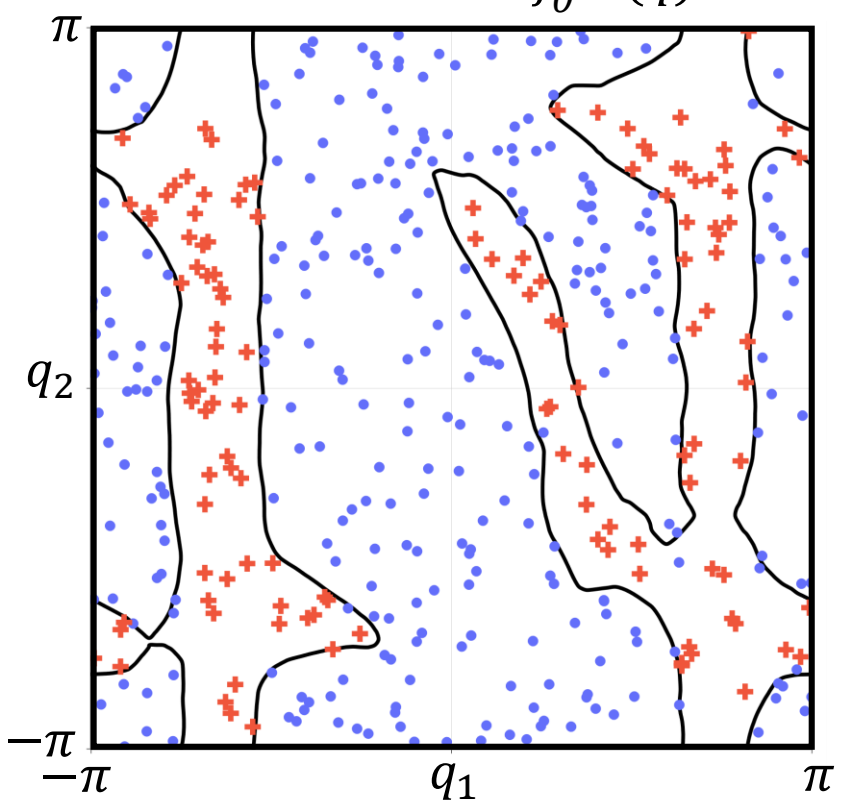
2R Planar Robot System



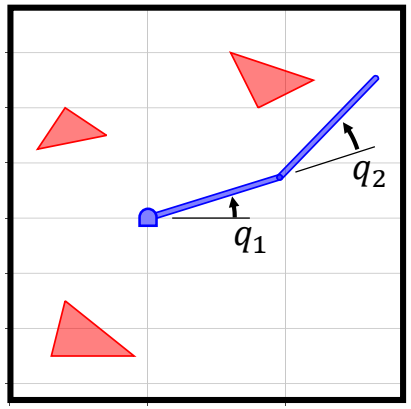
Initial Dataset $\mathcal{D}_0$

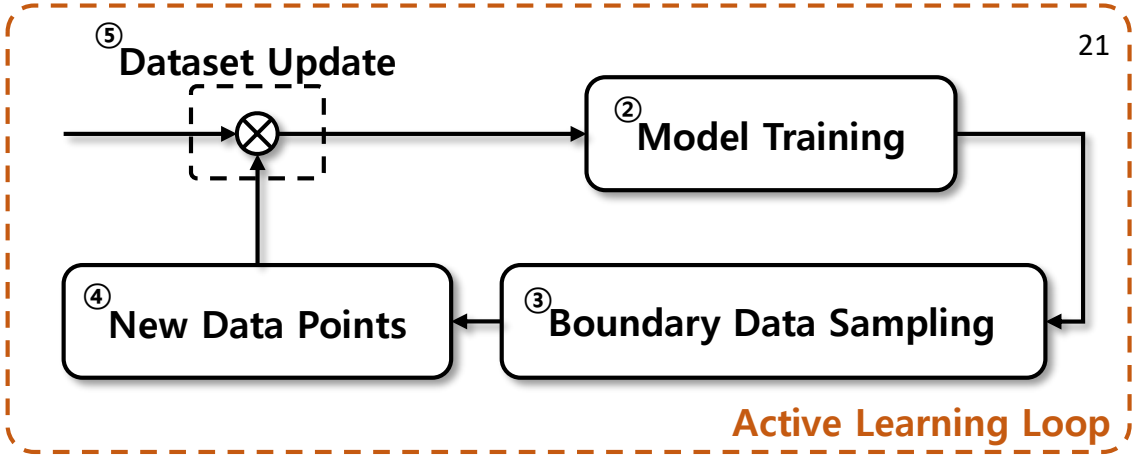


Trained Model $f_\theta^{(1)}(q)$

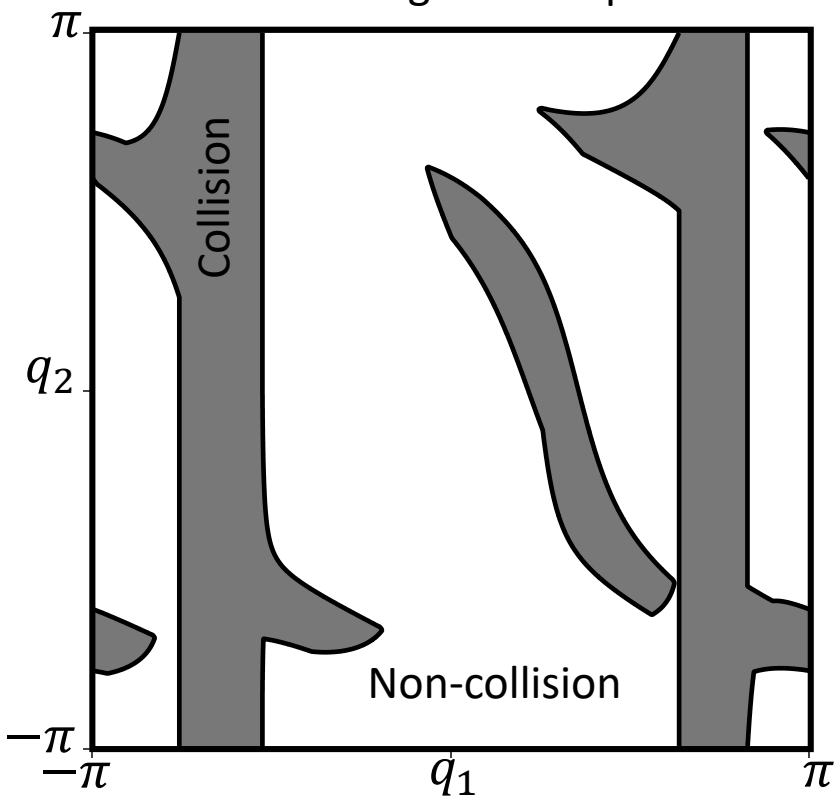


2R Planar Robot System

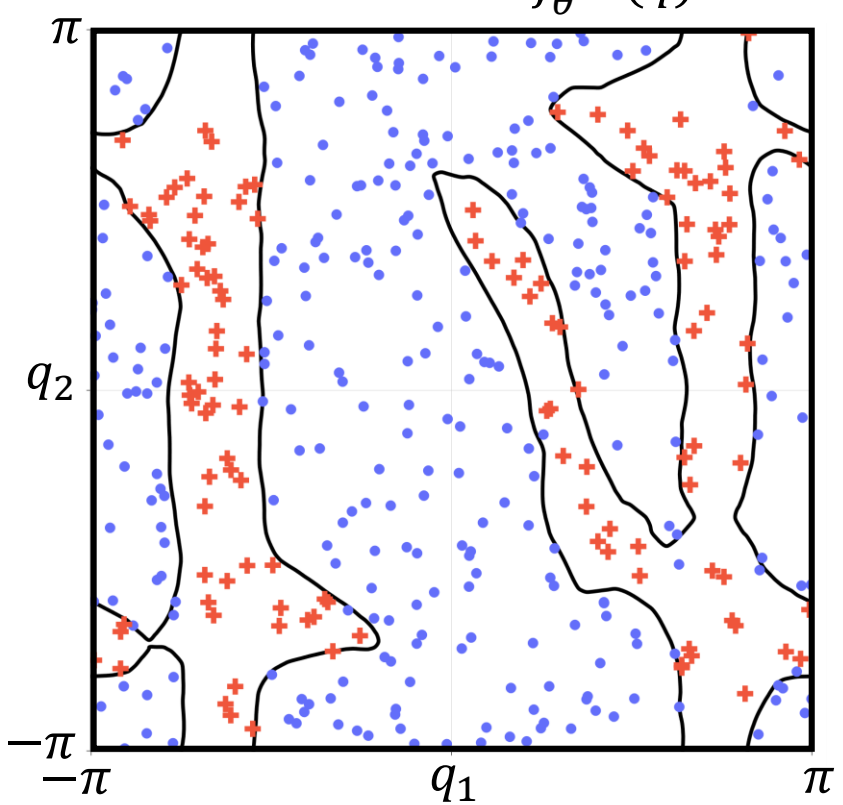




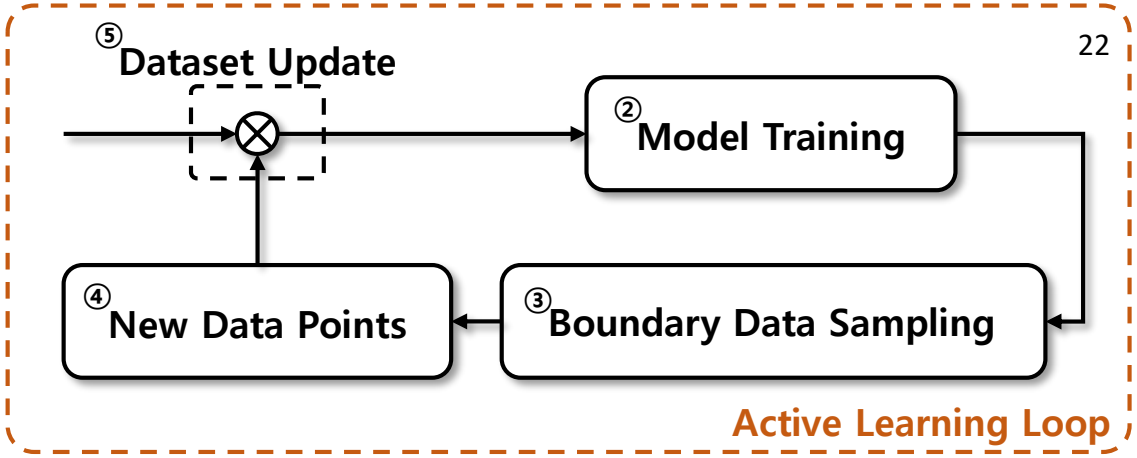
Joint Configuration Space



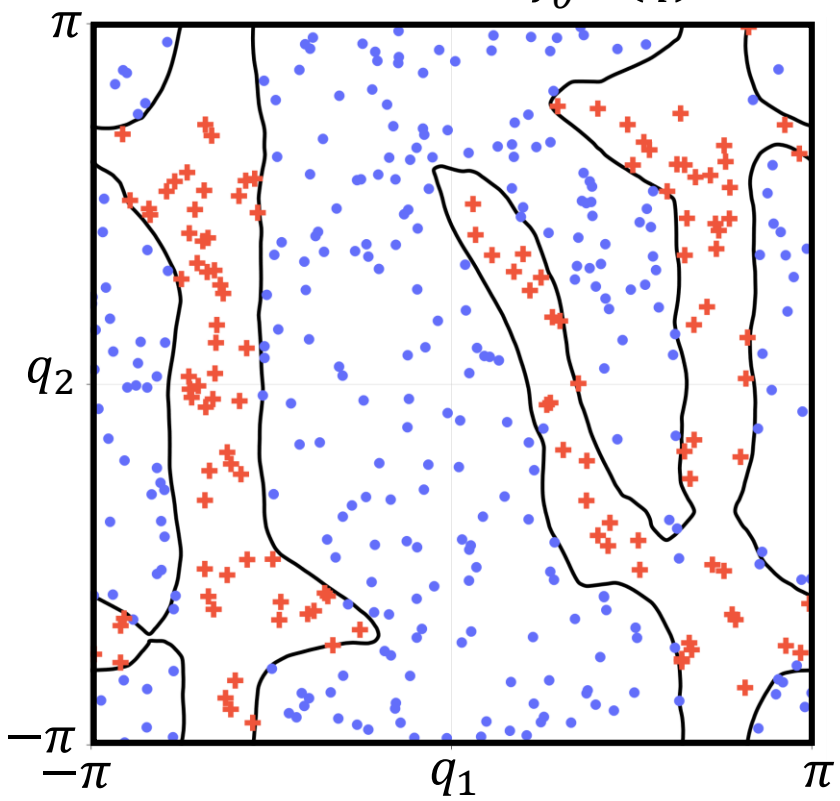
Trained Model $f_\theta^{(1)}(q)$



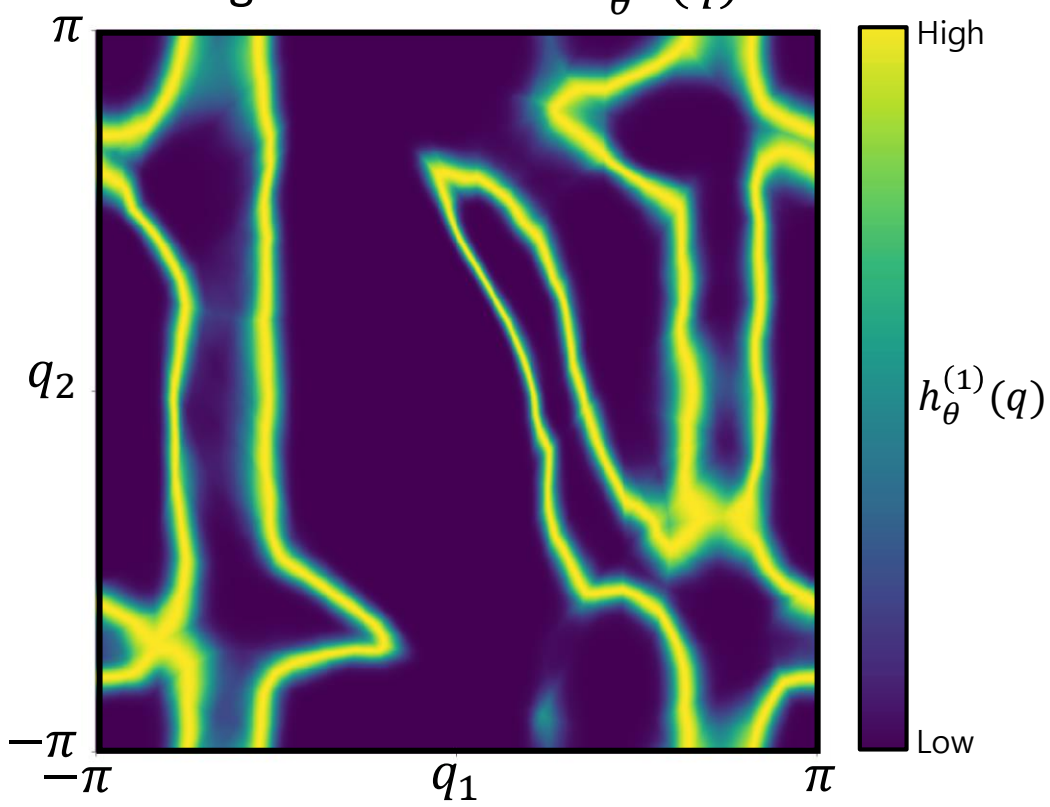
2R Planar Robot System



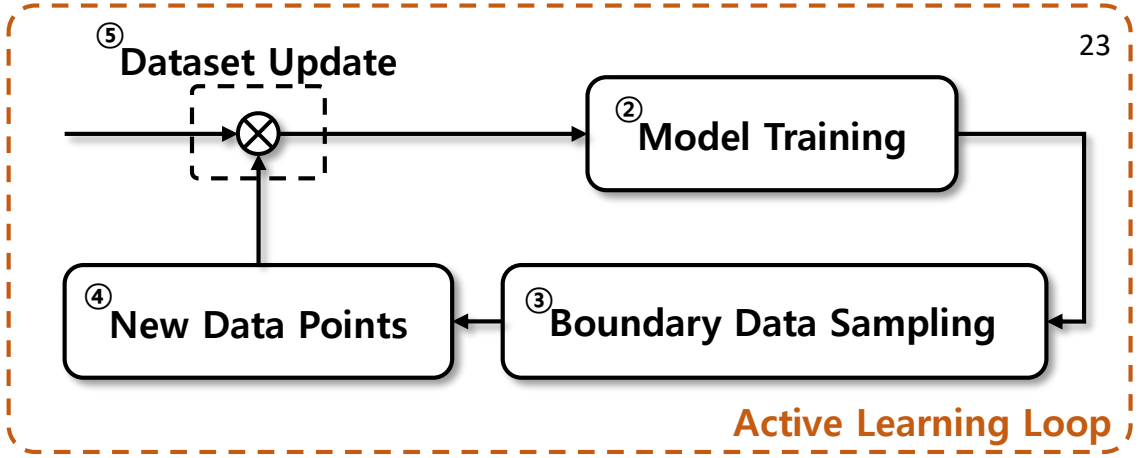
Trained Model $f_\theta^{(1)}(q)$



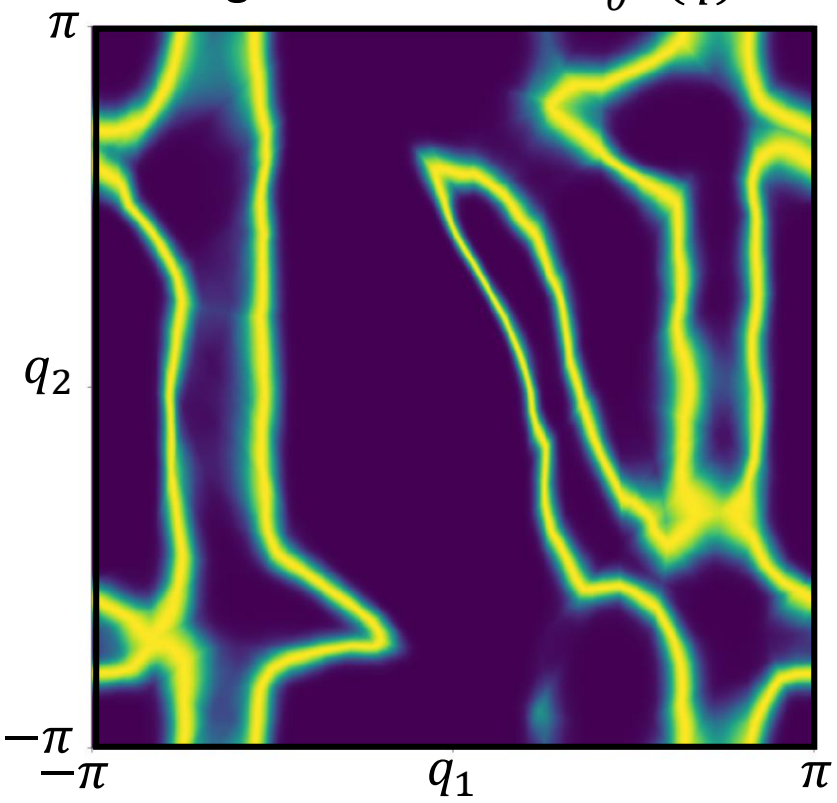
Target Distribution $h_\theta^{(1)}(q)$



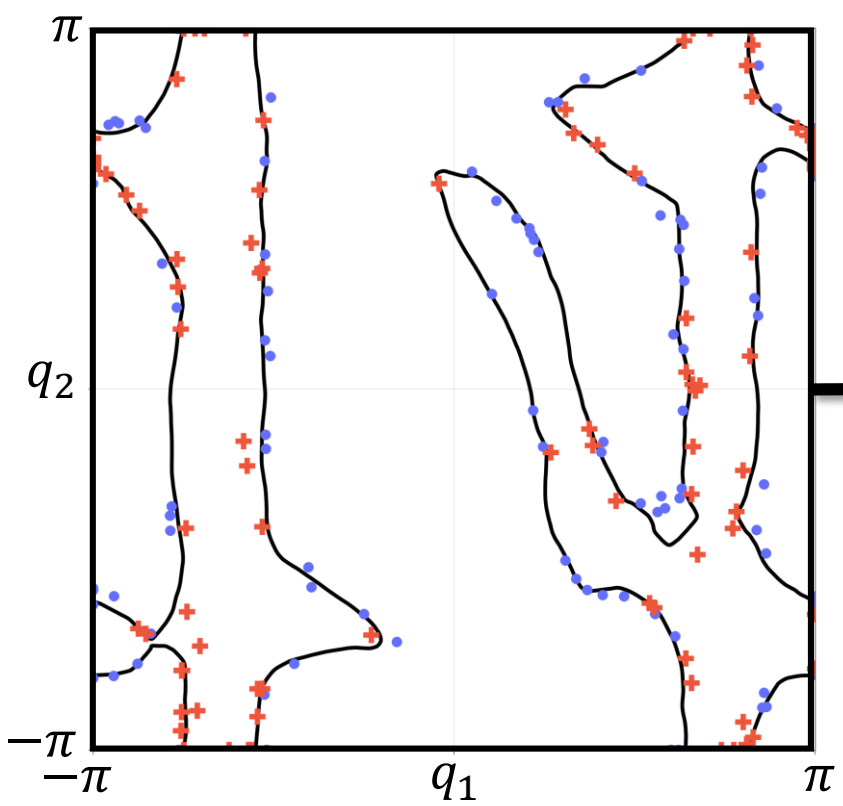
2R Planar Robot System



Target Distribution $h_\theta^{(1)}(q)$

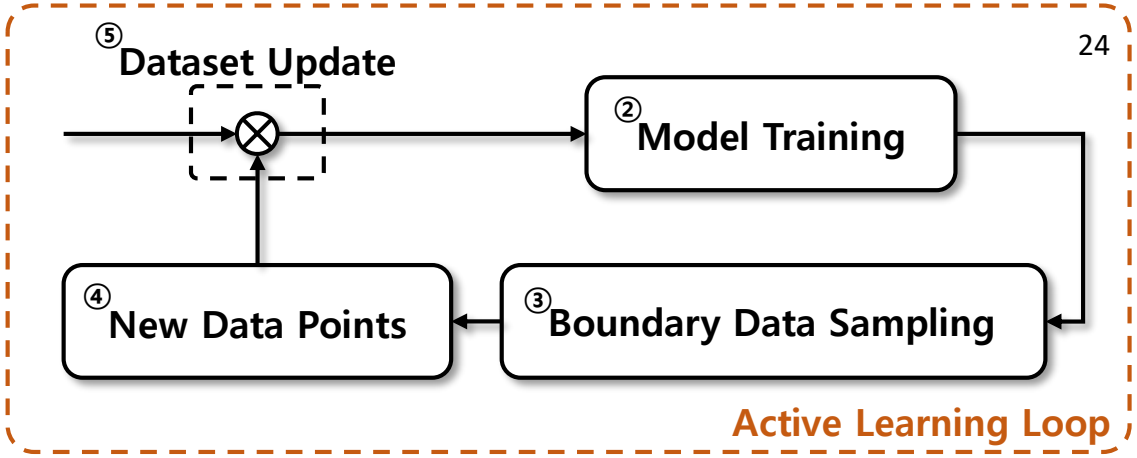


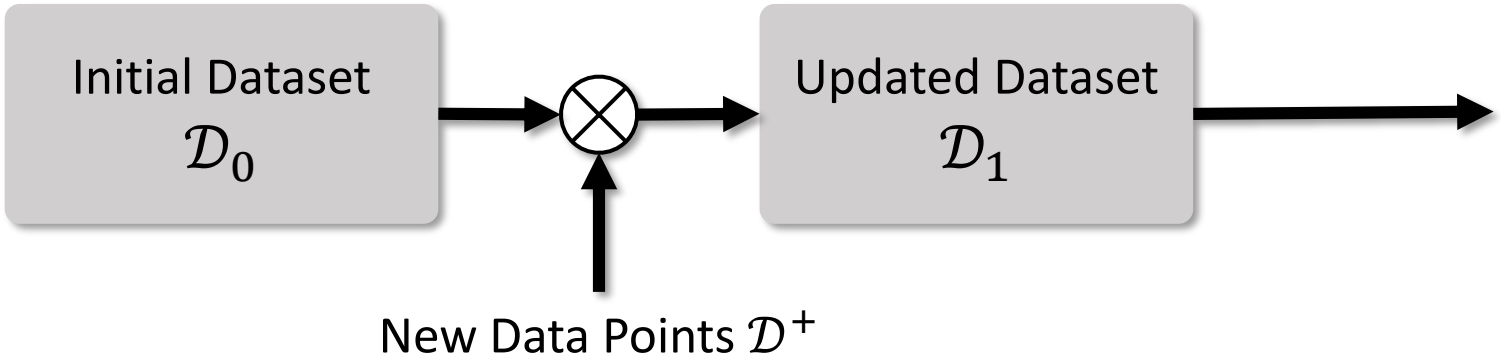
New Data Points $\mathcal{D}^+$

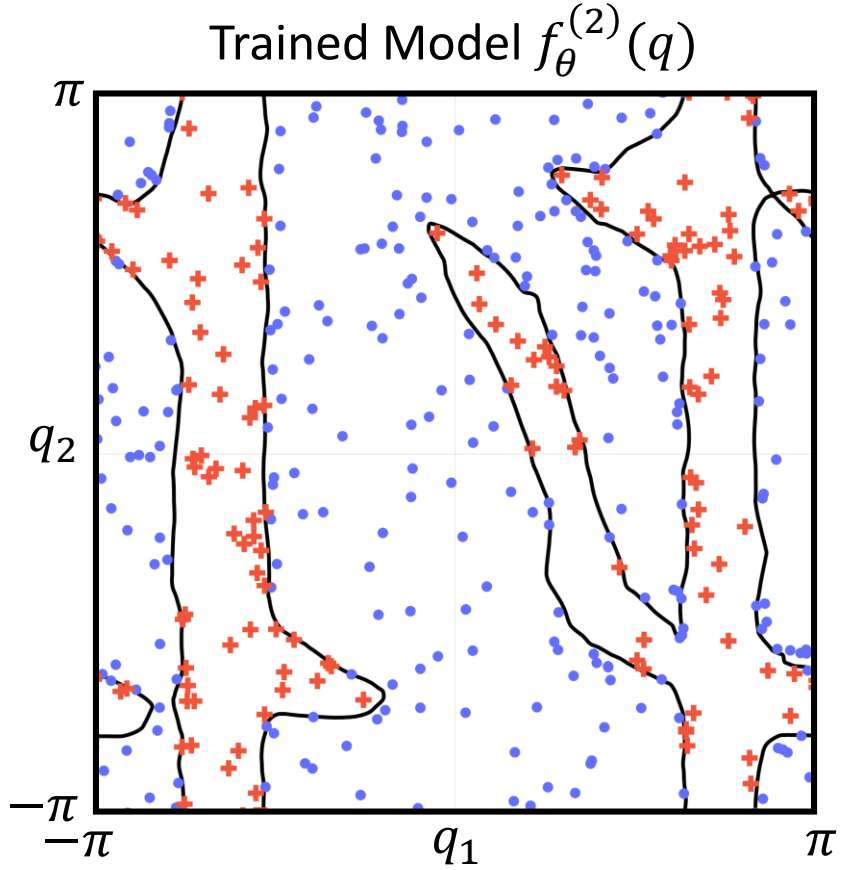


Initial Dataset $\mathcal{D}_0$

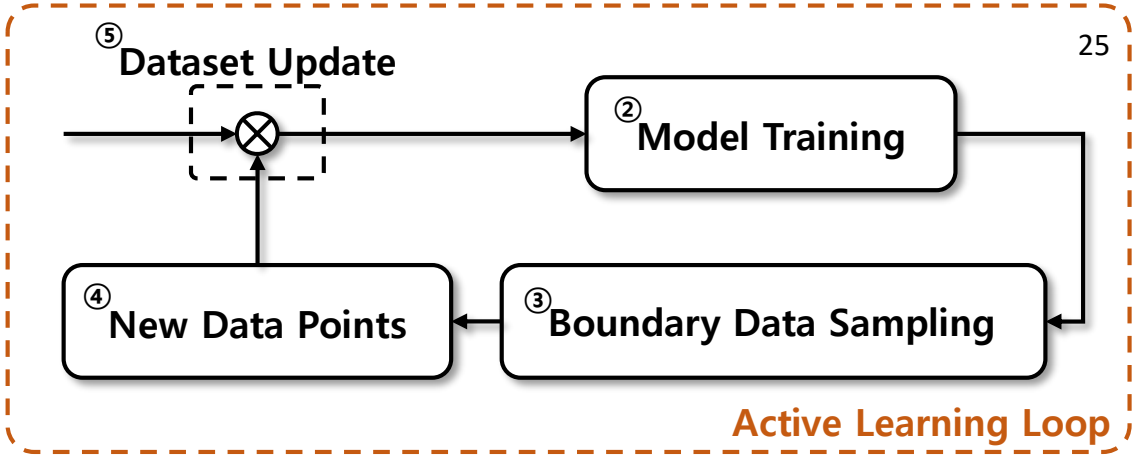
2R Planar Robot System

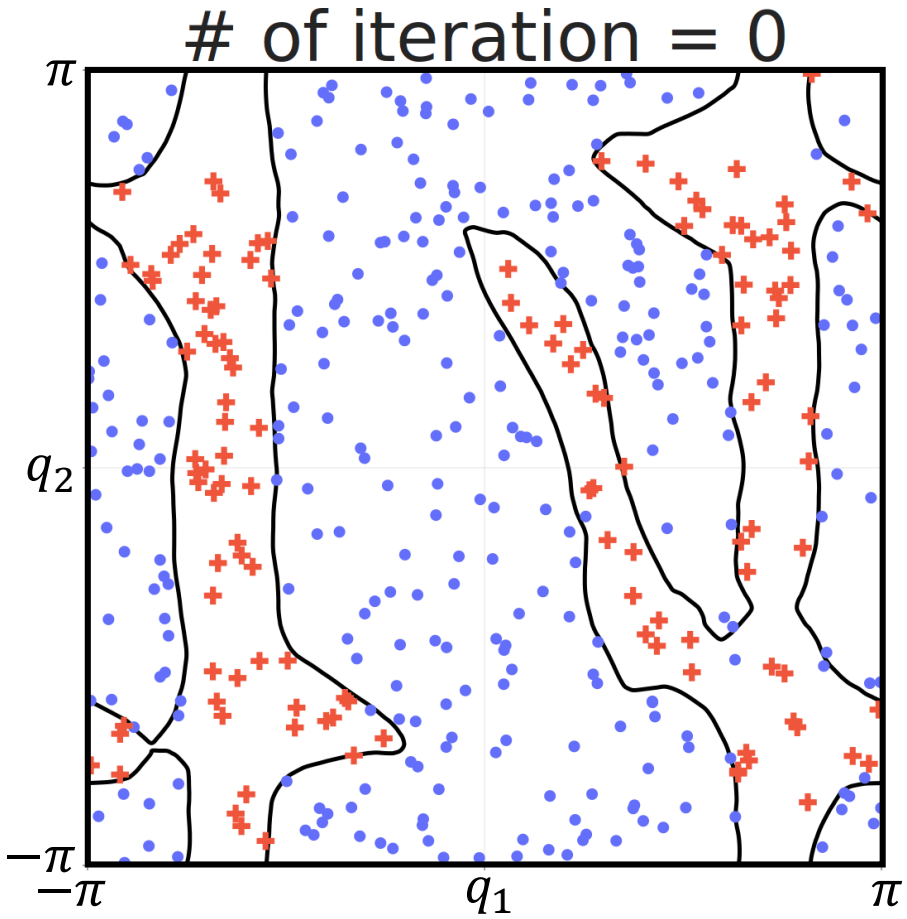


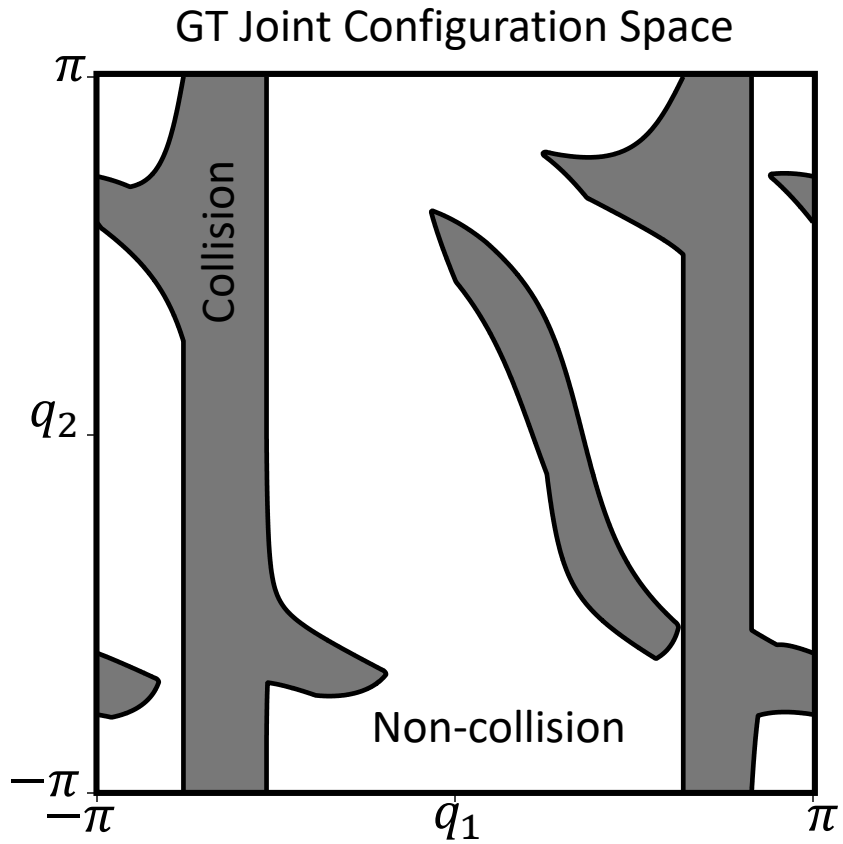




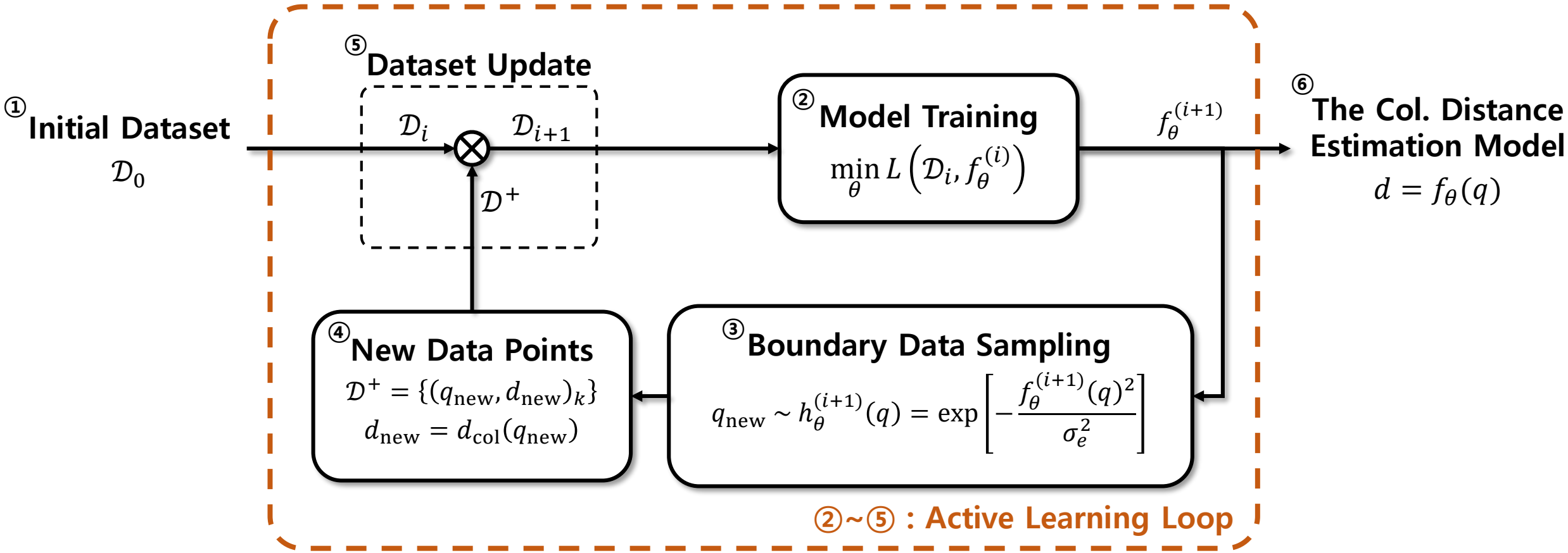
2R Planar Robot System



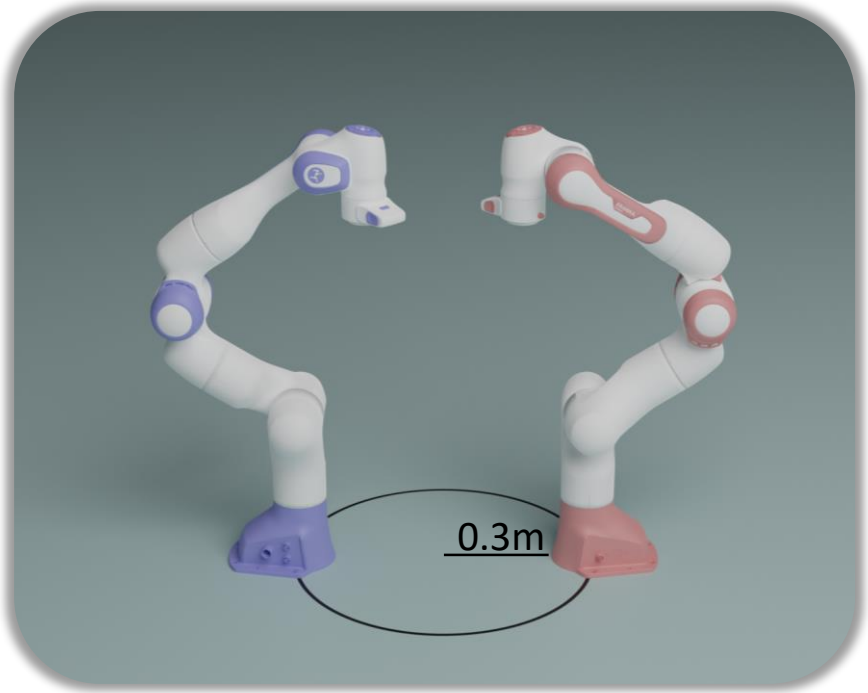




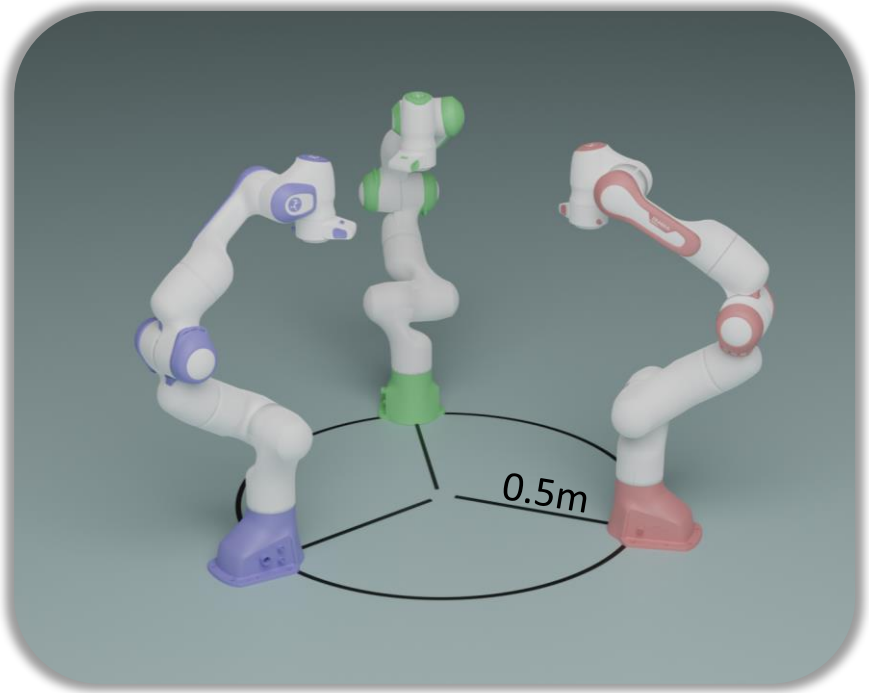
Active Learning-based Training



High-dof Multi-arm Robot Systems



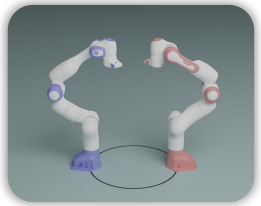
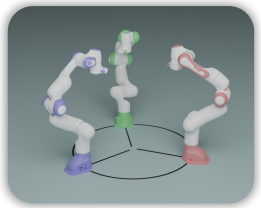
A two-arm system (14-dof)



A three-arm system (21-dof)

Results

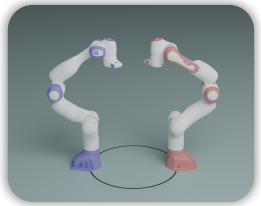
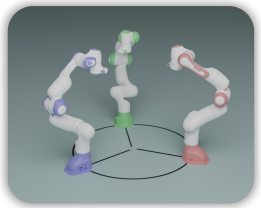
near-metrics (*near-Accuracy*, *near-AUROC*)
: metrics tested with near-boundary dataset ($-0.1\text{m} < d_{col} < 0.1\text{m}$)

Training Method		*none*			
Env.	Model	Accuracy (ϵ=0.0)	AUROC	near-Accuracy (ϵ=0.0)	near-AUROC
	JointNN	0.9765	0.9942	0.8799	0.9491
	PosNN	0.9810	0.9965	0.9025	0.9681
	ClearanceNet	0.9627	0.9855	0.8128	0.8845
	SE3NN	0.9862	0.9981	0.9293	0.9830
	JointNN	0.9691	0.9868	0.8189	0.8875
	PosNN	0.9799	0.9943	0.8808	0.9485
	ClearanceNet	0.9443	0.9507	0.6944	0.7202
	SE3NN	0.9826	0.9958	0.8969	0.9619

- JointNN [M. Koptev, et al., RAL'21]
- PosNN [D. Rakita, et al., RSS'18]
- ClearanceNet [J. Chase Kew, et al., WAFR'20]
- SE3NN [Kim, J., & Park, F. C., Robotica'24]

Results

near-metrics (*near-Accuracy*, *near-AUROC*)
: metrics tested with near-boundary dataset ($-0.1\mathrm{m} < d_{col} < 0.1\mathrm{m}$)

Training Method		*none*				*active*			
Env.	Model	Accuracy (ϵ=0.0)	AUROC	near-Accuracy (ϵ=0.0)	near-AUROC	Accuracy (ϵ=0.0)	AUROC	near-Accuracy (ϵ=0.0)	near-AUROC
	JointNN	0.9765	0.9942	0.8799	0.9491	0.9809	0.9960	0.9018	0.9643
	PosNN	0.9810	0.9965	0.9025	0.9681	0.9828	0.9971	0.9117	0.9735
	ClearanceNet	0.9627	0.9855	0.8128	0.8845	0.9707	0.9900	0.8516	0.9186
	SE3NN	0.9862	0.9981	0.9293	0.9830	0.9886	0.9987	0.9416	0.9877
	JointNN	0.9691	0.9868	0.8189	0.8875	0.9765	0.9918	0.8609	0.9280
	PosNN	0.9799	0.9943	0.8808	0.9485	0.9821	0.9955	0.8937	0.9587
	ClearanceNet	0.9443	0.9507	0.6944	0.7202	0.9532	0.9619	0.7377	0.7746
	SE3NN	0.9826	0.9958	0.8969	0.9619	0.9858	0.9971	0.9157	0.9738

Summary

- Existing methods fail to address the insufficiency (sparsity) of uniformly-sampled datasets.
- Our active learning-based training procedure iteratively generates near-boundary datasets, resulting in more accurate collision distance estimation models.
- For high-dof multi-arm robot systems, our method successfully enhances the performance of existing baseline models.

Chapter

PairwiseNet: Pairwise Collision Distance Learning

Contributions

1. Dataset Construction → Active Learning of the Collision Distance Function (Kim, J., & Park, F. C., Robotica'24)

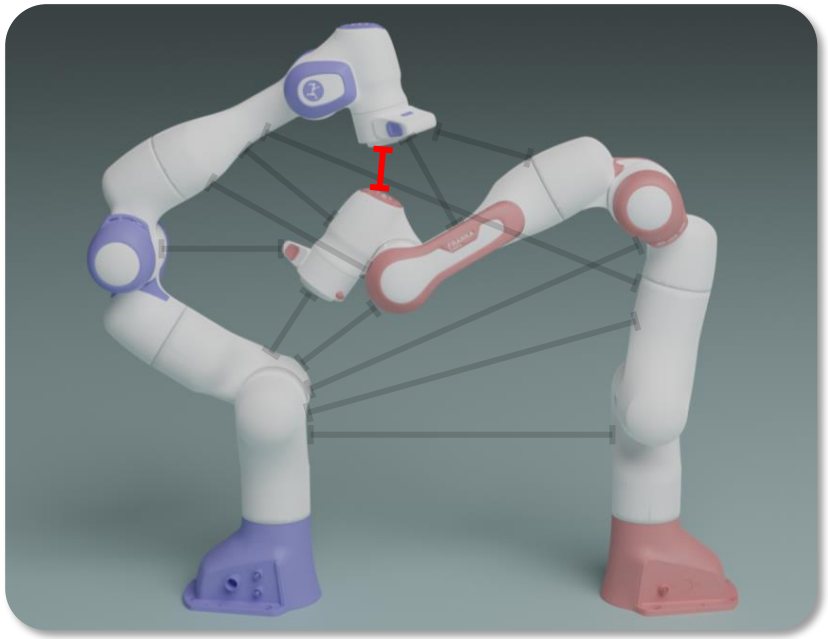
2. Inherent Complexity of the Collision Distance Function

3. Generalizability to Minor Environmental Changes → PairwiseNet (Kim, J., & Park, F. C., CoRL'23)

Kim, Jihwan, and Frank C. Park. "Active learning of the collision distance function for high-DOF multi-arm robot systems." Robotica (2024): 1-15.
Kim, Jihwan, and Frank C. Park. "PairwiseNet: Pairwise Collision Distance Learning for High-dof Robot Systems." Conference on Robot Learning. PMLR, 2023.

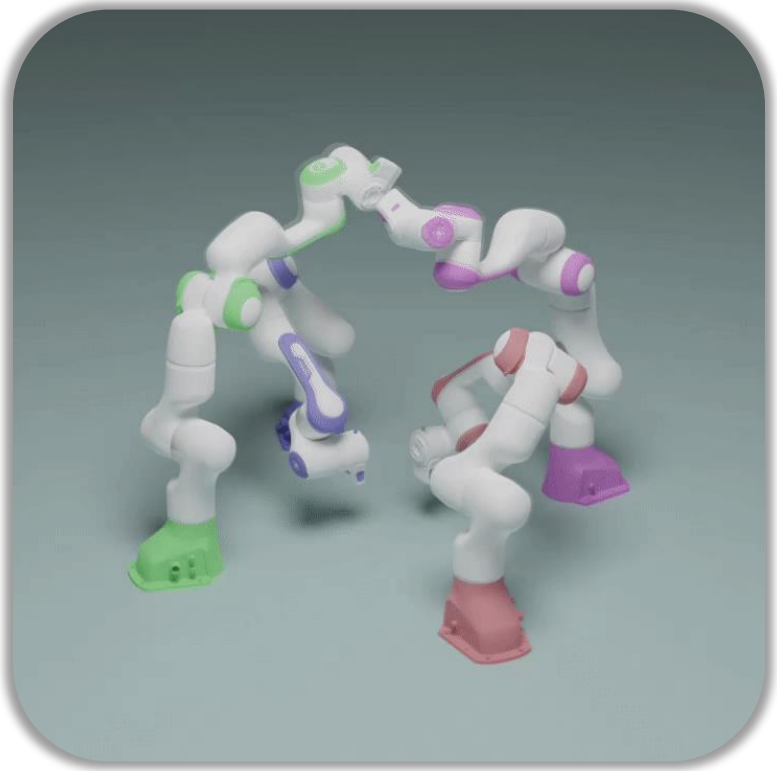
Inherent Complexity of Collision Distance

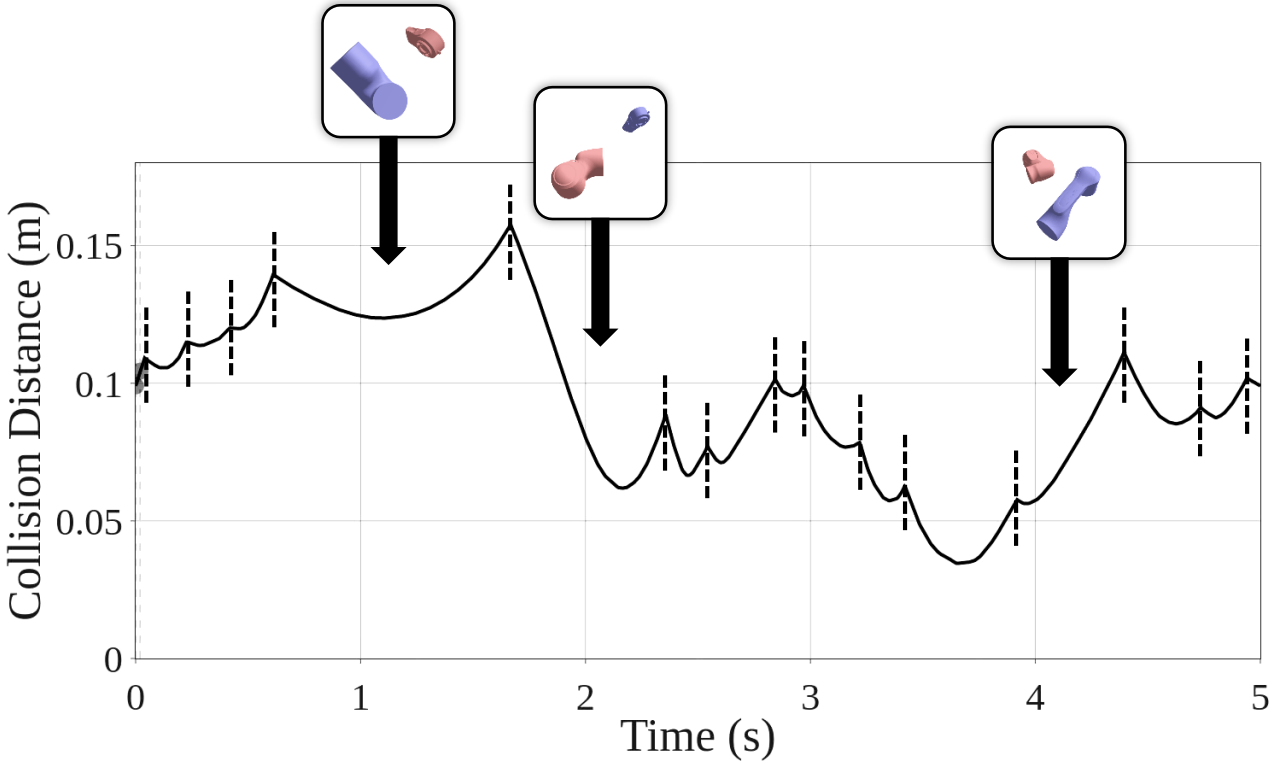
$$d_{\text{col}}(q) = \min_{(i,j)} d_{ij}(q)$$



The closest pair of elements continuously changes
: leads to **non-smoothness** and **abrupt variations**

Inherent Complexity of Collision Distance

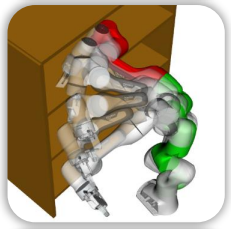




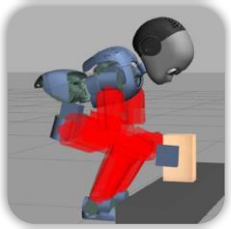
Inherent Complexity of Collision Distance

- Kernel-based models

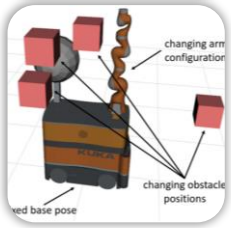
SCA [N. B. Figueroa Fernandez, et al., MLPC'18]
: Support Vector Machine (SVM)



DiffCo [Y. Zhi, et al., TRO'22]
: Kernel perceptron



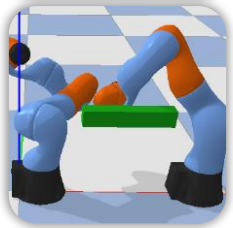
SCA [M. Koptev, et al., RAL'21]
: Support Vector Machine (SVM)
: Fully-connected neural networks



CollisionGP [Y. Zhi, et al., TRO'22]
: Gaussian Process (GP)

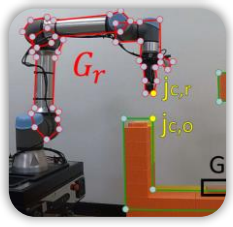
- Neural network-based models

RelaxedIK [D. Rakita, et al., RSS'18]
: Fully-connected neural networks



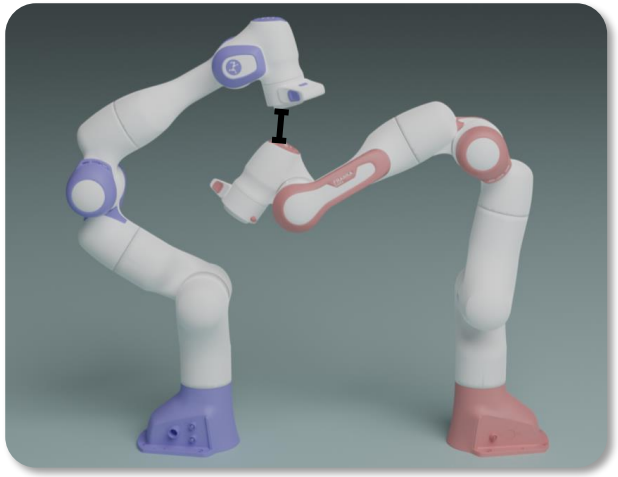
ClearanceNet [J. Chase Kew, et al., WAFR'20]
: Fully-connected neural networks

jointNERF [M. Bhardwaj, et al., CoRL'22]
: Fully-connected neural networks

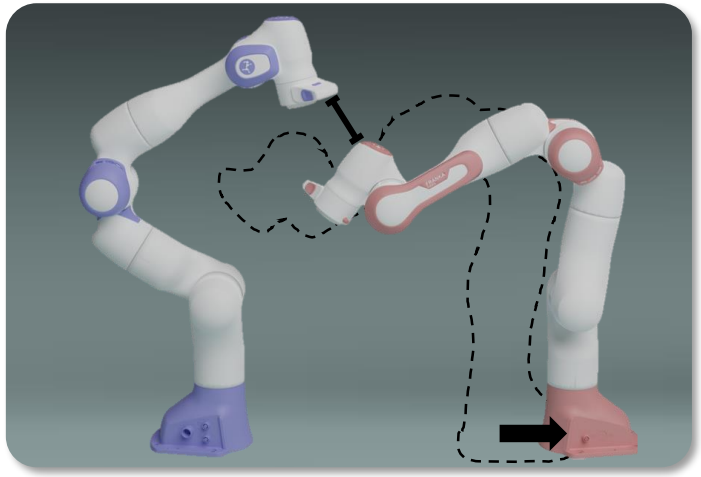


GraphDistNet [Y. Kim, et al., RAL'22]
: Graph neural networks

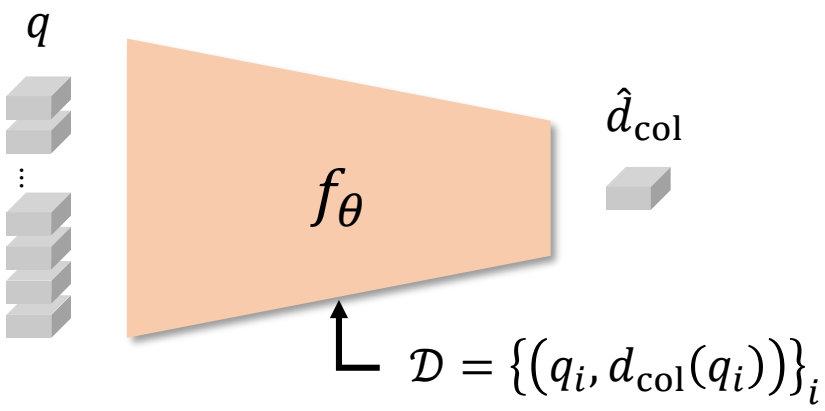
Generalizability to Minor Env. Changes

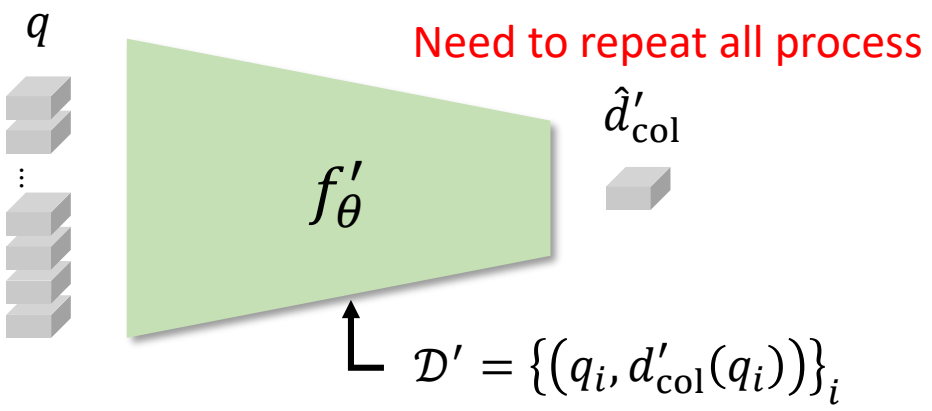


Collsion Distance $d_{\text{col}}(q)$

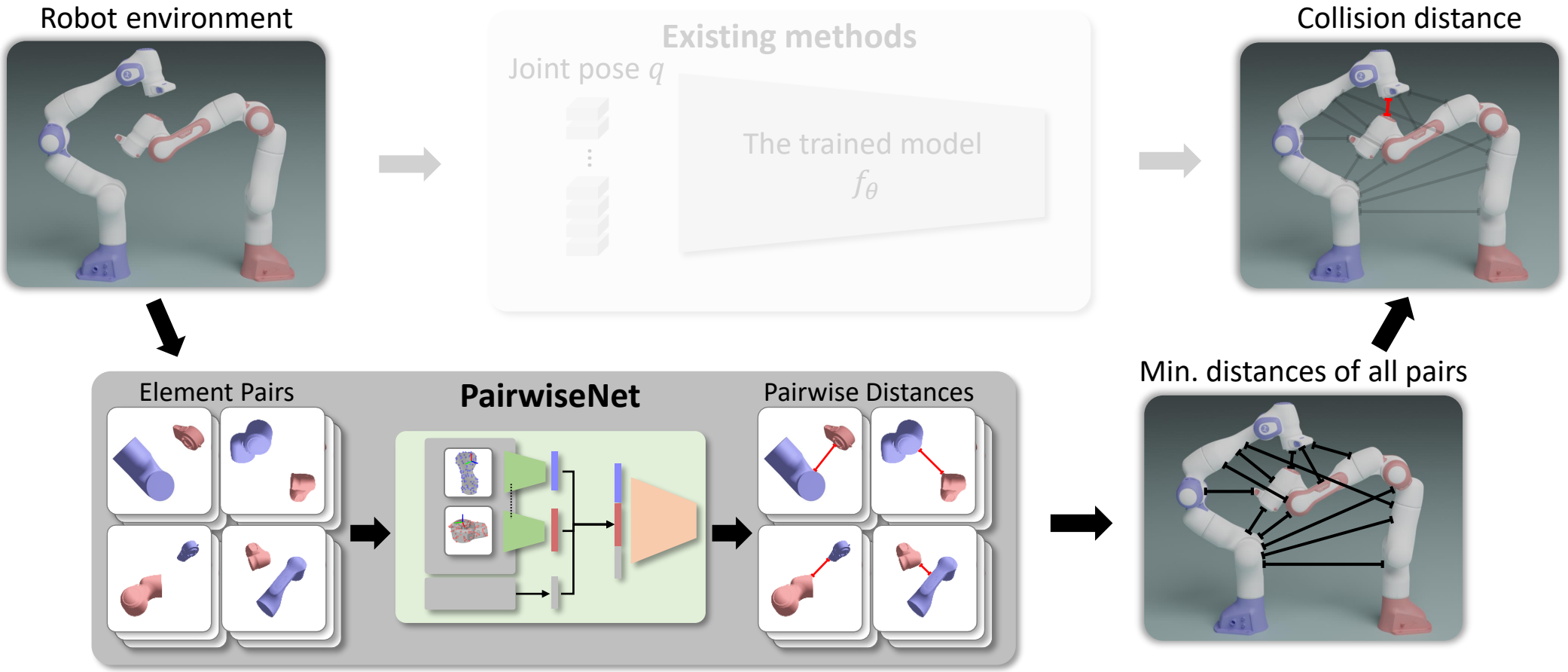


Collsion Distance $d'_{\text{col}}(q)$



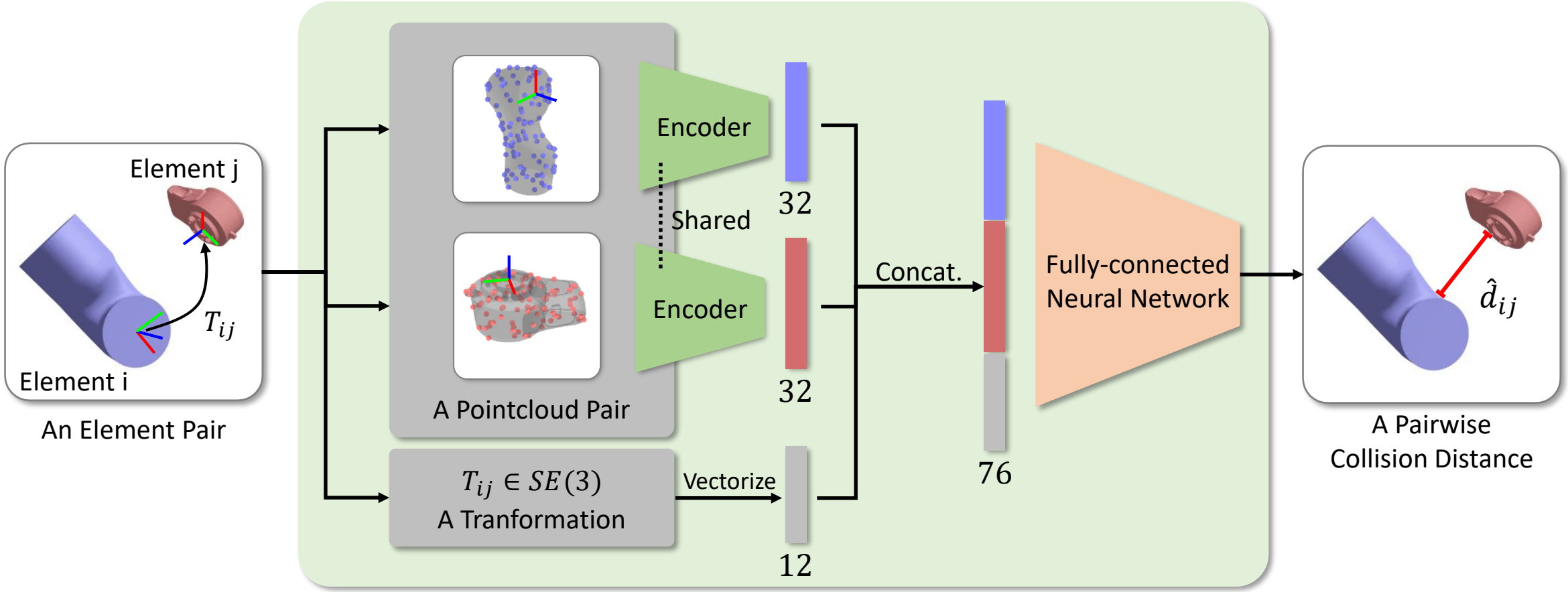


PairwiseNet: Pairwise Collision Distance Learning

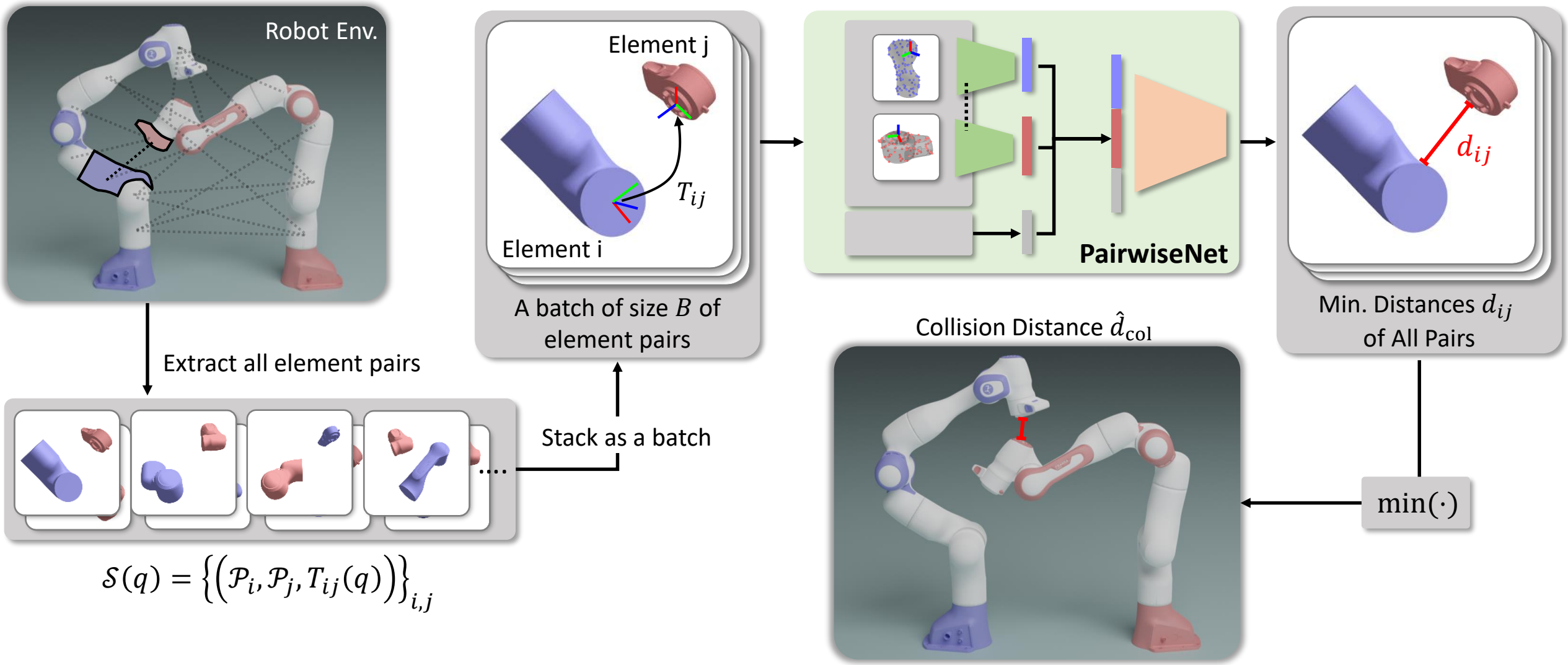


PairwiseNet: Model Architecture

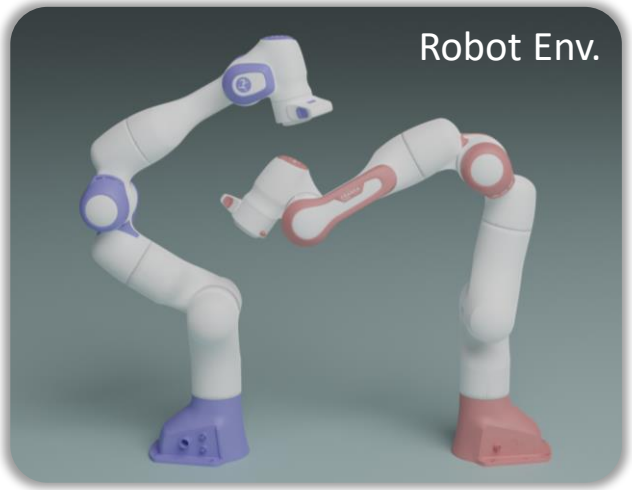
- Pairwise Collision Distance $\hat{d}_{ij} = f_\psi(\mathcal{P}_i, \mathcal{P}_j, T_{ij})$



PairwiseNet: Overall Process



PairwiseNet: Overall Process



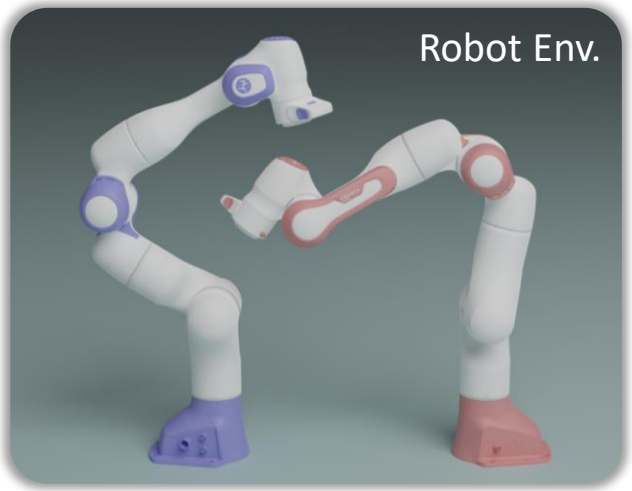
Given a set of element pairs $\mathcal{S}(q) = \left\{\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)\right)\right\}_{i,j}$,

$$\hat{d}_{\text{col}} = F_\psi\left(q; f_\psi, \mathcal{S}\right)$$

$$= \min_{\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)\right) \in \mathcal{S}(q)} f_\psi\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)\right)$$

PairwiseNet

PairwiseNet: Overall Process



Collision Distance

$$\hat{d}_{\text{col}}(q)$$

>>

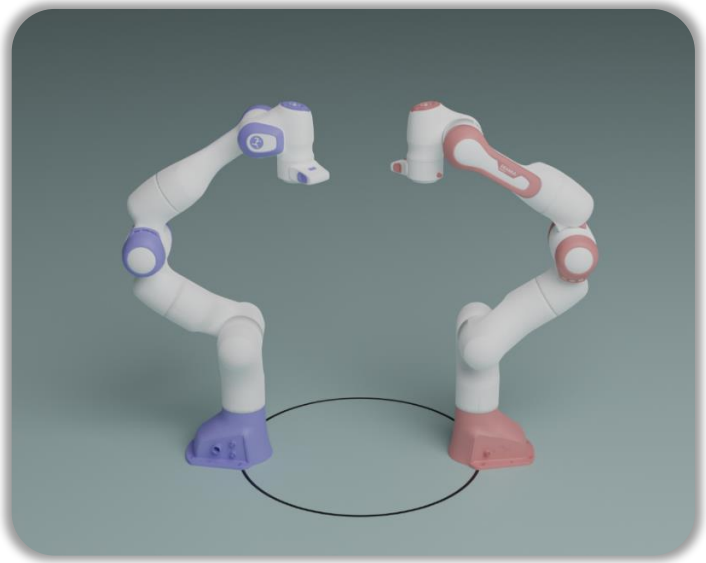
Training Difficulty

Pairwise

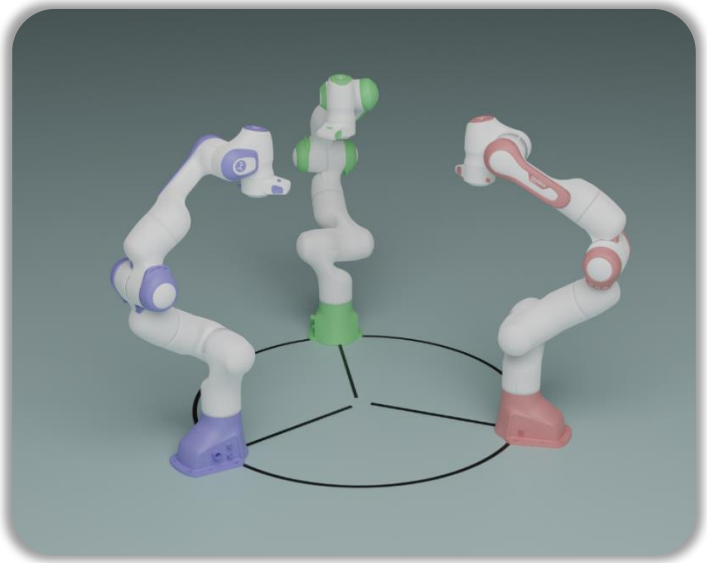
Collision Distance

$$f_{\psi}\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}\right)$$

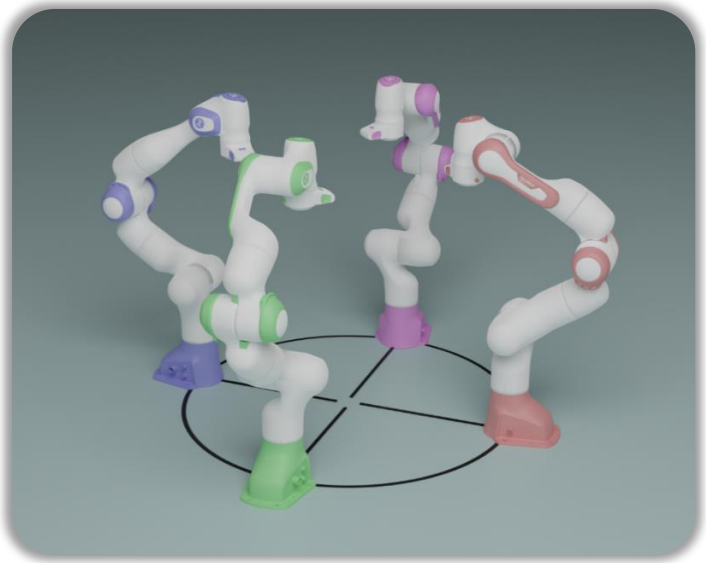
PairwiseNet: Estimation Performance



Two arms (14-dof)

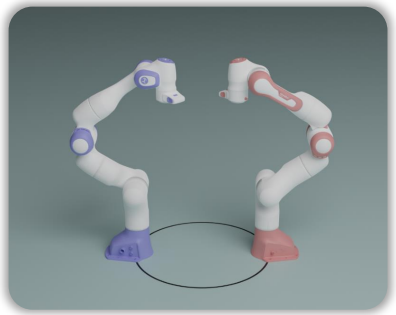


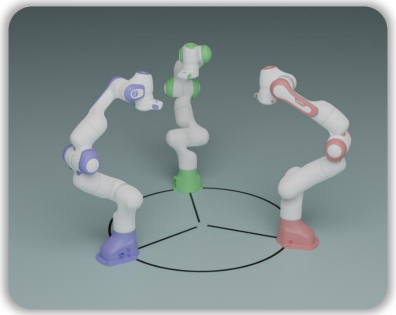
Three arms (21-dof)

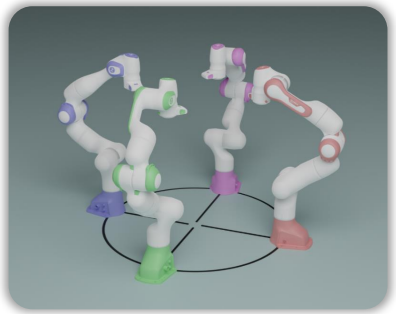


Four arms (28-dof)

PairwiseNet: Estimation Performance



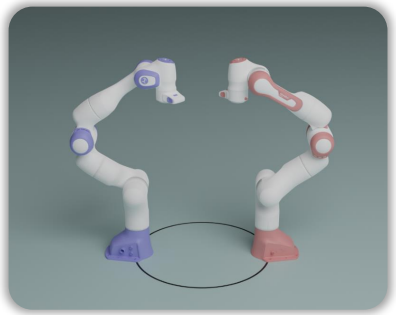


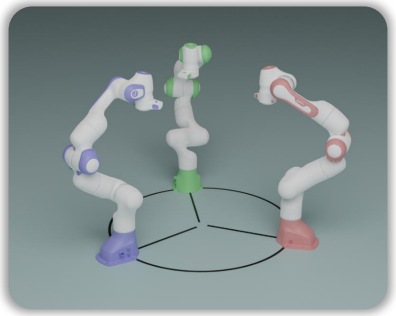


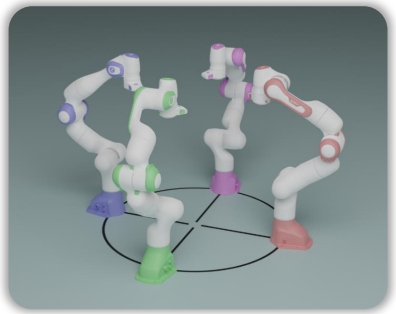
Methods	MSE	AUROC	Accuracy ($\epsilon = 0$)	safe-FPR
Capsule	5.47e-4	0.9995	0.9776	0.0247
JointNN	3.63e-4	0.9955	0.9794	0.3200
PosNN	2.71e-4	0.9970	0.9823	0.1476
jointNERF	2.98e-4	0.9962	0.9808	0.2371
ClearanceNet	1.11e-3	0.9853	0.9621	0.4570
DiffCo*	-	0.9824	0.9818	0.3141
PairwiseNet (ours)	**0.24e-4**	**0.9998**	**0.9941**	**0.0200**
Capsule	5.96e-4	0.9993	0.9775	0.0241
JointNN	9.69e-4	0.9902	0.9721	0.2679
PosNN	5.41e-4	0.9951	0.9801	0.1336
jointNERF	8.22e-4	0.9920	0.9747	0.2213
ClearanceNet	4.63e-3	0.9499	0.9395	0.6067
DiffCo*	-	0.9603	0.9453	0.5858
PairwiseNet (ours)	**0.24e-4**	**0.9997**	**0.9944**	**0.0189**
Capsule	6.66e-4	0.9986	0.9468	0.0694
JointNN	1.59e-3	0.9718	0.9183	0.6988
PosNN	7.18e-4	0.9885	0.9478	0.5371
jointNERF	1.30e-3	0.9778	0.9280	0.6260
ClearanceNet	6.67e-3	0.8738	0.8202	0.9965
DiffCo*	-	0.8811	0.8306	0.9874
PairwiseNet (ours)	**0.46e-4**	**0.9994**	**0.9858**	**0.0650**

- Capsule [A. El Khoury, et al., ICRA'13]
- JointNN [M. Koptev, et al., RAL'21]
- PosNN [D. Rakita, et al., RSS'18]
- jointNERF [M. Bhardwaj, et al., CoRL'22]
- ClearanceNet [J. Chase Kew, et al., WAFR'20]
- DiffCo [Y. Zhi, et al., TRO'22]

PairwiseNet: Estimation Performance







Methods	MSE	AUROC	Accuracy ($\epsilon = 0$)	safe-FPR
Capsule	5.47e-4	0.9995	0.9776	0.0247
JointNN	3.63e-4	0.9955	0.9794	0.3200
PosNN	2.71e-4	0.9970	0.9823	0.1476
jointNERF	2.98e-4	0.9962	0.9808	0.2371
ClearanceNet	1.11e-3	0.9853	0.9621	0.4570
DiffCo*	-	0.9824	0.9818	0.3141
PairwiseNet (ours)	**0.24e-4**	**0.9998**	**0.9941**	**0.0200**
Capsule	5.96e-4	0.9993	0.9775	0.0241
JointNN	9.69e-4	0.9902	0.9721	0.2679
PosNN	5.41e-4	0.9951	0.9801	0.1336
jointNERF	8.22e-4	0.9920	0.9747	0.2213
ClearanceNet	4.63e-3	0.9499	0.9395	0.6067
DiffCo*	-	0.9603	0.9453	0.5858
PairwiseNet (ours)	**0.24e-4**	**0.9997**	**0.9944**	**0.0189**
Capsule	6.66e-4	0.9986	0.9468	0.0694
JointNN	1.59e-3	0.9718	0.9183	0.6988
PosNN	7.18e-4	0.9885	0.9478	0.5371
jointNERF	1.30e-3	0.9778	0.9280	0.6260
ClearanceNet	6.67e-3	0.8738	0.8202	0.9965
DiffCo*	-	0.8811	0.8306	0.9874
PairwiseNet (ours)	**0.46e-4**	**0.9994**	**0.9858**	**0.0650**

Safe-FPR

: False Positive Rate with **safe threshold**

: Worst-case evaluation metric

: Higher safe-FPR

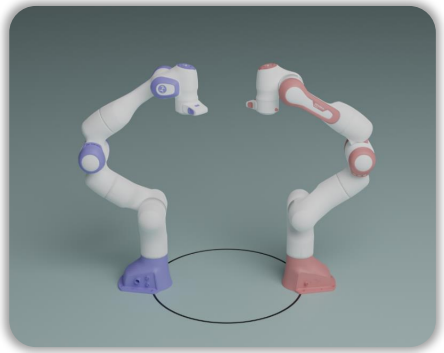
→ more false alarms

→ overly conservative planning

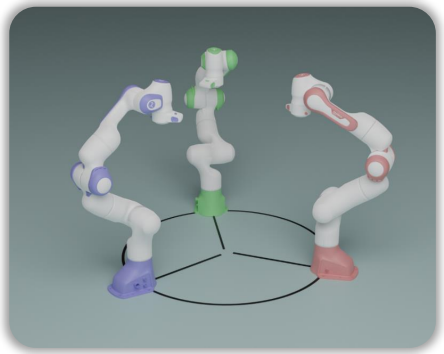
Safe threshold

: The least conservative threshold that that still detects all actual collisions in test data (ensuring zero false negative)

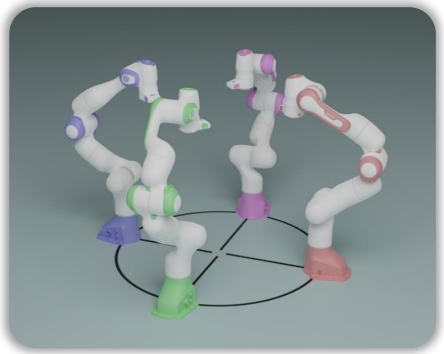
PairwiseNet: Generalizability



$$d_{\text{col}}^{(1)}(q_1) \rightarrow \hat{d}_{\text{col}}^{(1)}(q_1) \quad \mathcal{D}^{(1)}$$

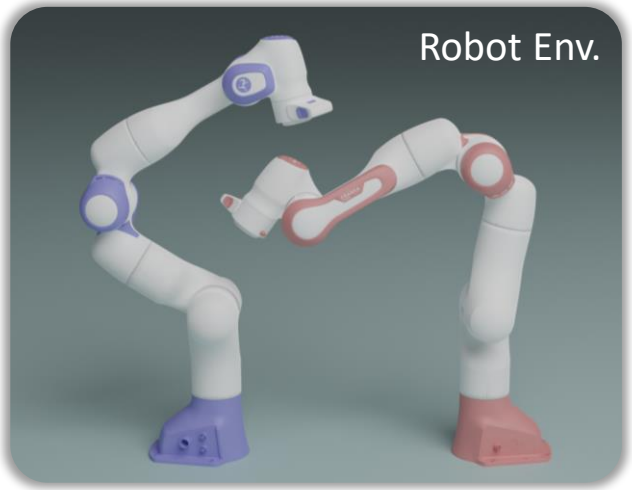


$$d_{\text{col}}^{(2)}(q_2) \rightarrow \hat{d}_{\text{col}}^{(2)}(q_2) \quad \mathcal{D}^{(2)}$$



$$d_{\text{col}}^{(3)}(q_3) \rightarrow \hat{d}_{\text{col}}^{(3)}(q_3) \quad \mathcal{D}^{(3)}$$

PairwiseNet: Generalizability

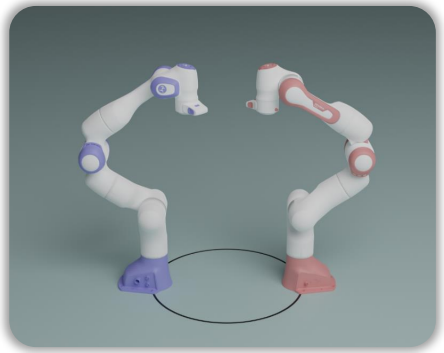


Given a set of element pairs $\mathcal{S}(q) = \left\{\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)\right)\right\}_{i,j}$,

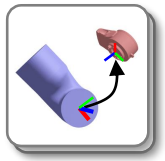
$$\hat{d}_{\text{col}} = F_\psi\left(q; f_\psi, \mathcal{S}\right)$$

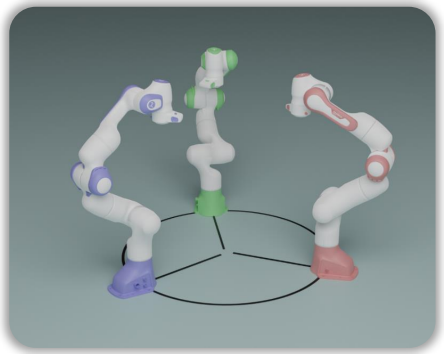
$$= \min_{\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)\right) \in \mathcal{S}(q)} f_\psi\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)\right)$$

PairwiseNet: Generalizability

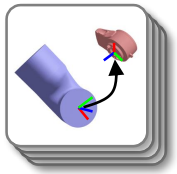


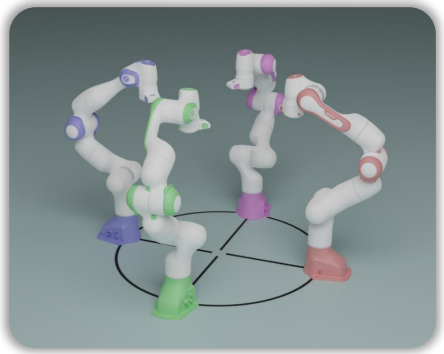
$\mathcal{S}^{(1)}(q_1)$



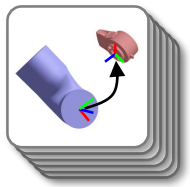


$\mathcal{S}^{(2)}(q_2)$





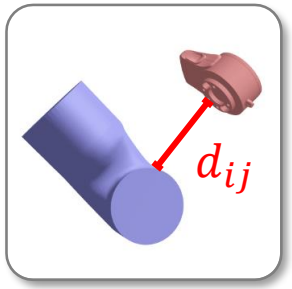
$\mathcal{S}^{(3)}(q_3)$



Given a set of element pairs $\mathcal{S}^{(k)}(q_k) = \left\{\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q_k)\right)\right\}_{i,j}$,

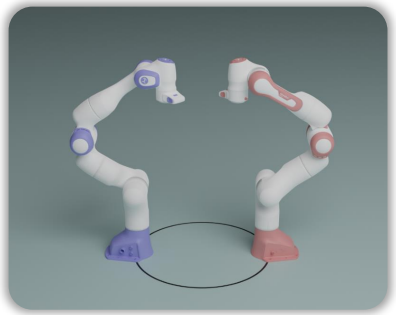
$$\hat{d}_{\text{col}} = F_\psi\left(q; f_\psi, \mathcal{S}^{(k)}\right)$$

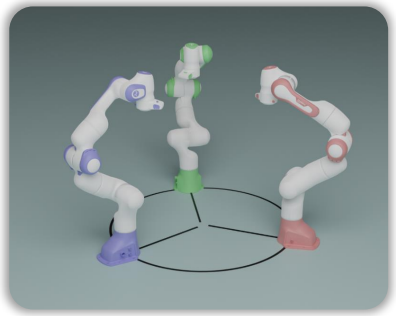
$$= \min_{\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q_k)\right) \in \mathcal{S}^{(k)}(q_k)} f_\psi\left(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q_k)\right)$$

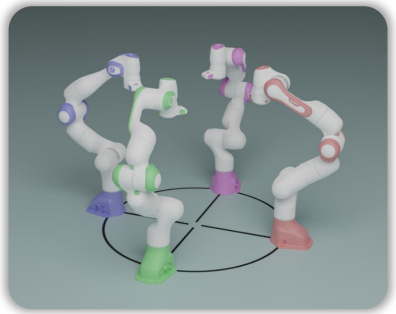


can be utilized for
all three environments!

PairwiseNet: Generalizability





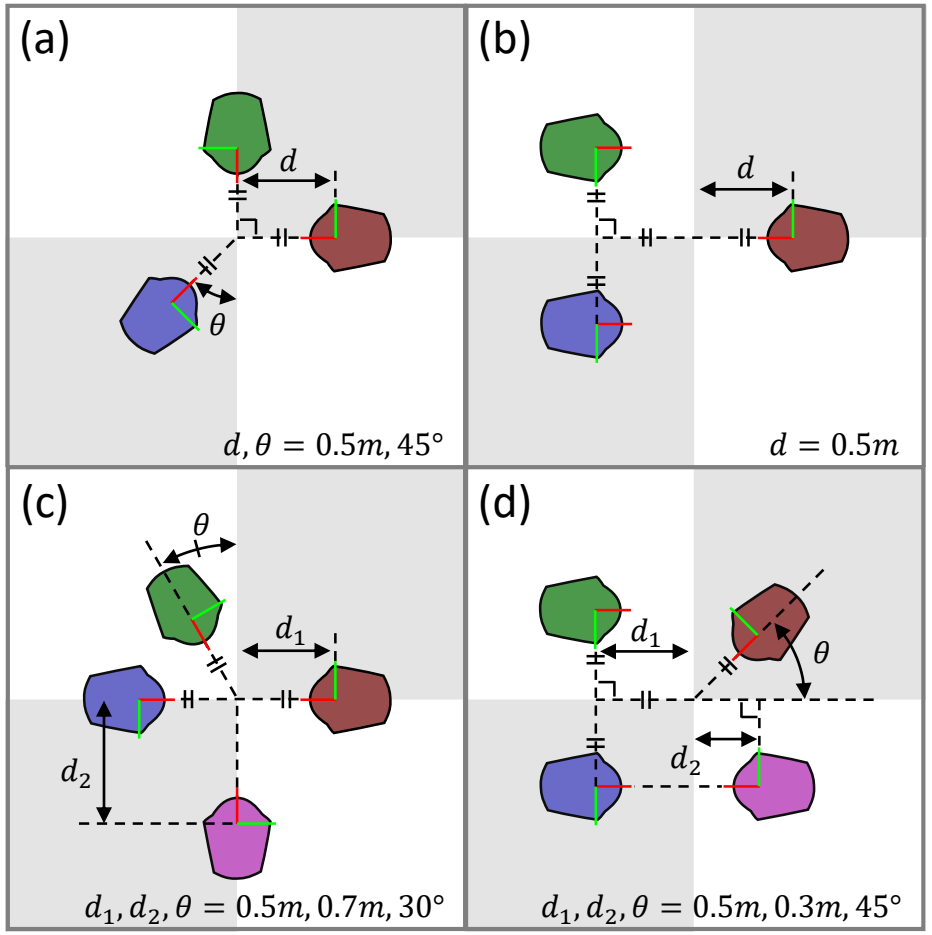


Methods	MSE	AUROC	Accuracy ($\epsilon = 0$)	safe-FPR
Capsule	5.47e-4	0.9995	0.9776	0.0247
JointNN	3.63e-4	0.9955	0.9794	0.3200
PosNN	2.71e-4	0.9970	0.9823	0.1476
jointNERF	2.98e-4	0.9962	0.9808	0.2371
ClearanceNet	1.11e-3	0.9853	0.9621	0.4570
DiffCo*	-	0.9824	0.9818	0.3141
PairwiseNet (ours)	**0.24e-4**	**0.9998**	**0.9941**	**0.0200**
Capsule	5.96e-4	0.9993	0.9775	0.0241
JointNN	9.69e-4	0.9902	0.9721	0.2679
PosNN	5.41e-4	0.9951	0.9801	0.1336
jointNERF	8.22e-4	0.9920	0.9747	0.2213
ClearanceNet	4.63e-3	0.9499	0.9395	0.6067
DiffCo*	-	0.9603	0.9453	0.5858
PairwiseNet (ours)	**0.24e-4**	**0.9997**	**0.9944**	**0.0189**
Capsule	6.66e-4	0.9986	0.9468	0.0694
JointNN	1.59e-3	0.9718	0.9183	0.6988
PosNN	7.18e-4	0.9885	0.9478	0.5371
jointNERF	1.30e-3	0.9778	0.9280	0.6260
ClearanceNet	6.67e-3	0.8738	0.8202	0.9965
DiffCo*	-	0.8811	0.8306	0.9874
PairwiseNet (ours)	**0.46e-4**	**0.9994**	**0.9858**	**0.0650**

→ Utilize a single PairwiseNet model

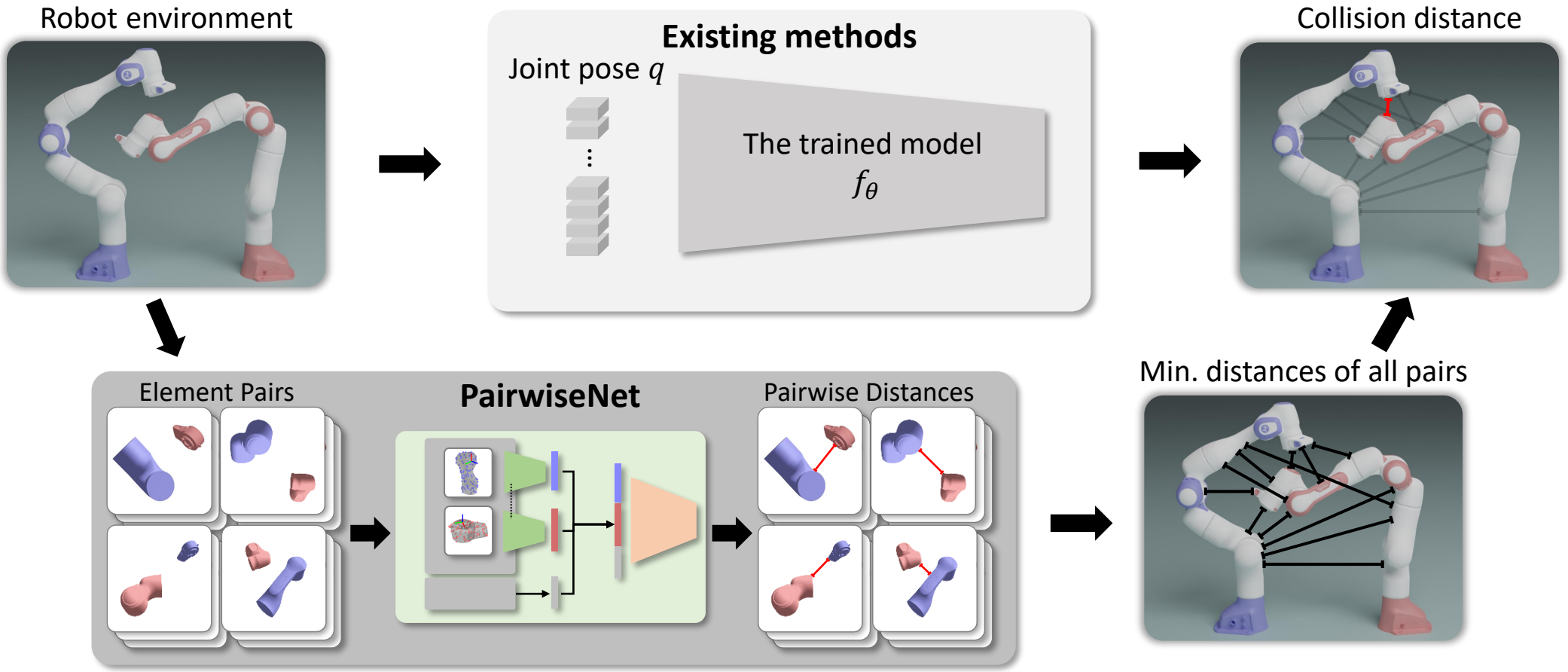
PairwiseNet: Generalizability

- Top views of four multi-arm systems



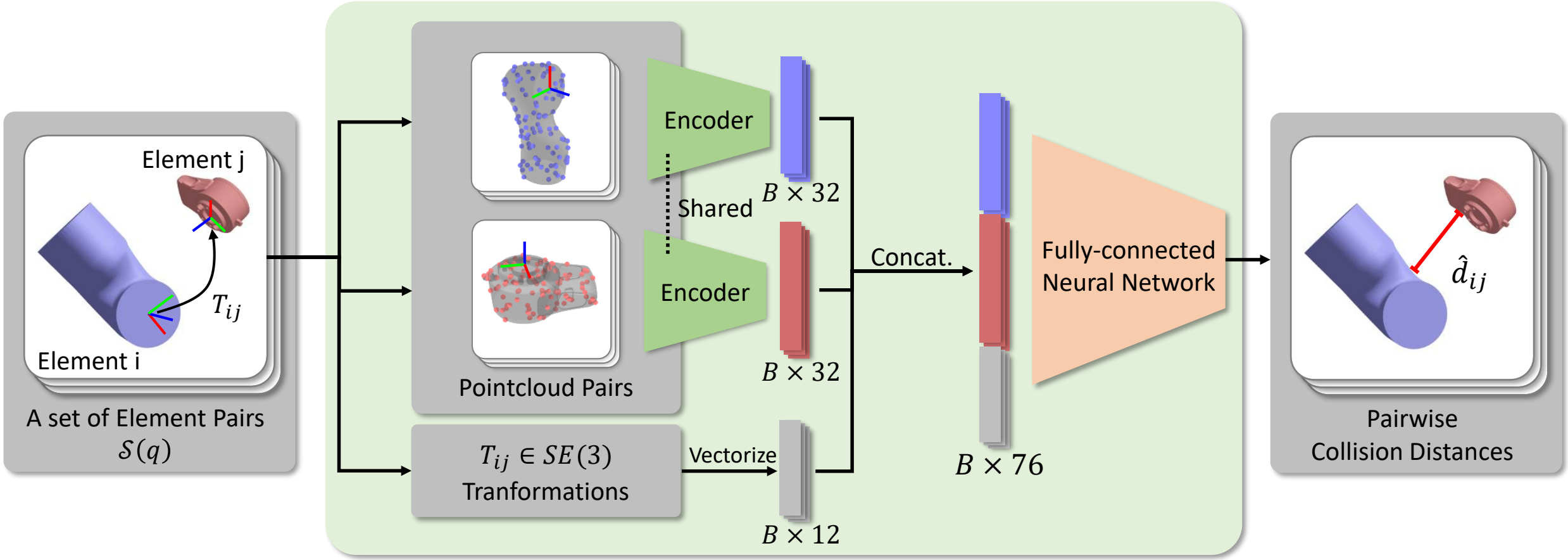
	Env.	MSE	AUROC	Accuracy ($\epsilon = 0$)	safe-FPR
Fig. 4.6	(a) (Three arms)	0.24e-4	0.9997	0.9943	0.0341
Fig. 4.6	(b) (Three arms)	0.17e-4	0.9999	0.9987	0.0048
Fig. 4.6	(c) (Four arms)	0.43e-4	0.9991	0.9859	0.0611
Fig. 4.6	(d) (Four arms)	0.27e-4	0.9995	0.9931	0.0292

PairwiseNet: Inference Time

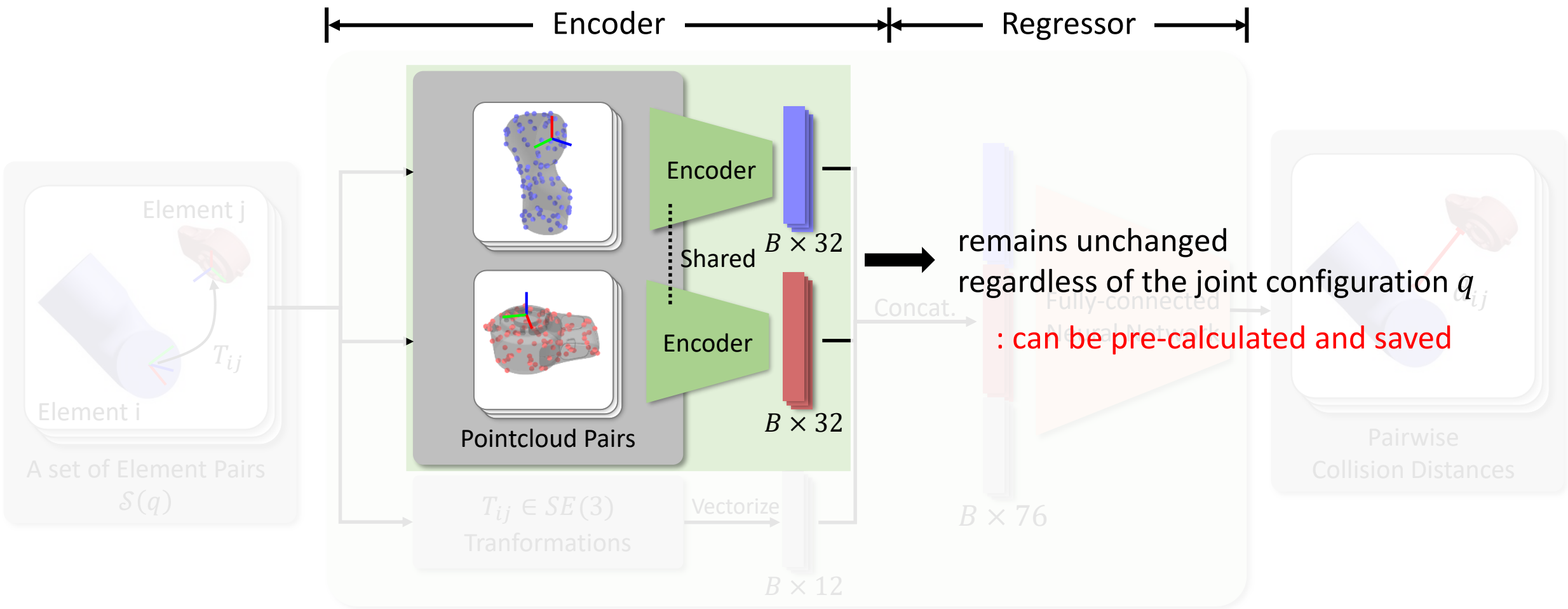


PairwiseNet: Inference Time

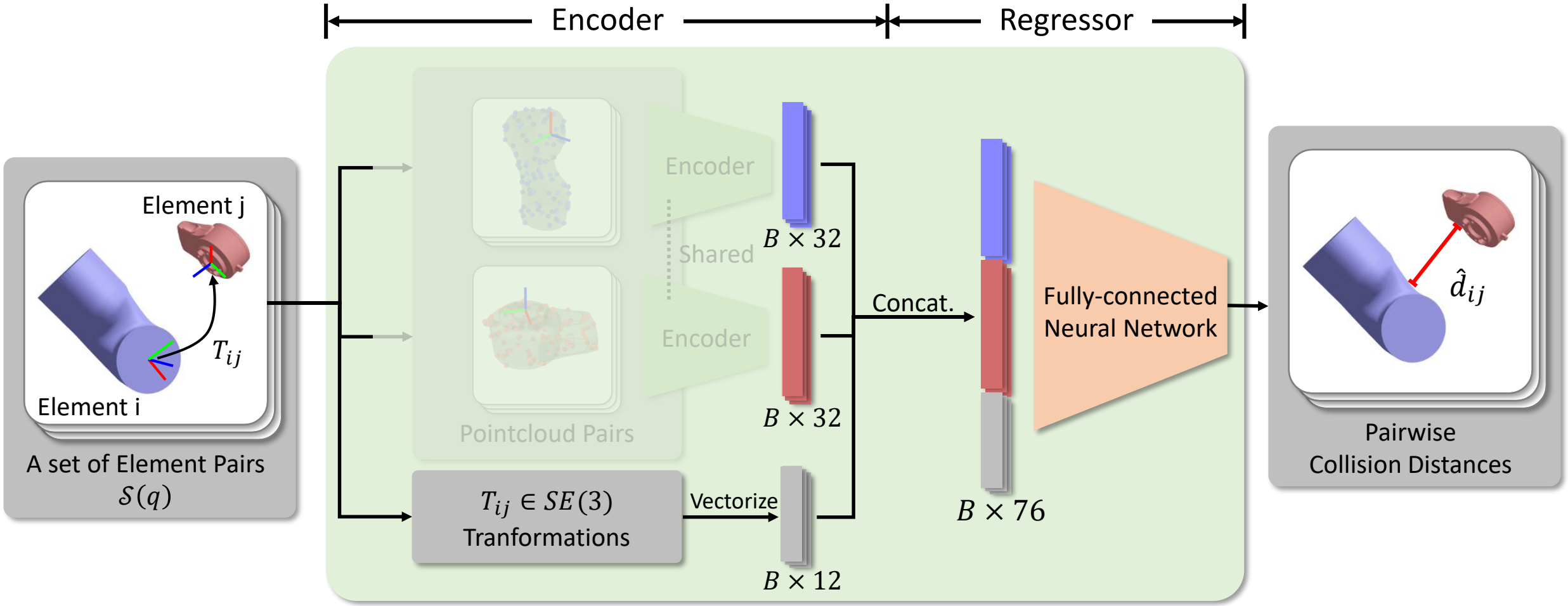
- Simultaneous calculation of all pairwise distances via batch calculation of neural networks



PairwiseNet: Inference Time



PairwiseNet: Inference Time



PairwiseNet: Inference Time

Methods	Inference time for 1000 joint poses (s)					
	Two arms (64 pairs)		Three arms (192 pairs)		Four arms (384 pairs)	
FCL w/ original mesh	13.87		43.97		92.63	
FCL w/ simplified mesh	2.479		9.079		17.25	
HPP-FCL w/ simplified mesh	0.3920		0.8560		1.9490	
	CPU	GPU	CPU	GPU	CPU	GPU
ClearanceNet	0.1450	0.0919	0.1484	0.0818	0.1493	0.0825
ClearanceNet (Batch)	0.0299	0.0001	0.0250	0.0001	0.0222	0.0001
PairwiseNet	0.1054	0.0988	0.1590	0.0998	0.2025	0.1045
PairwiseNet (Batch)	0.0362	0.0207	0.1070	0.0224	0.2121	0.0253

Summary

- Existing methods fail to deal with the inherent complexity of the collision distance function, and needs to retrain the model even for the minor environmental changes.
- We propose PairwiseNet, a novel collision distance estimation method that estimates the pairwise collision distances instead of directly predicting the collision distance.
- By simplifying the problem into smaller sub-problems, our approach achieves significant performance improvements for high-dof systems, while maintaining its rapid inference capability.
- PairwiseNet is capable of handling minor environmental changes without additional training or fine-tuning.

Chapter

Planning with Collision Distance Estimator

Offline Trajectory Optimization

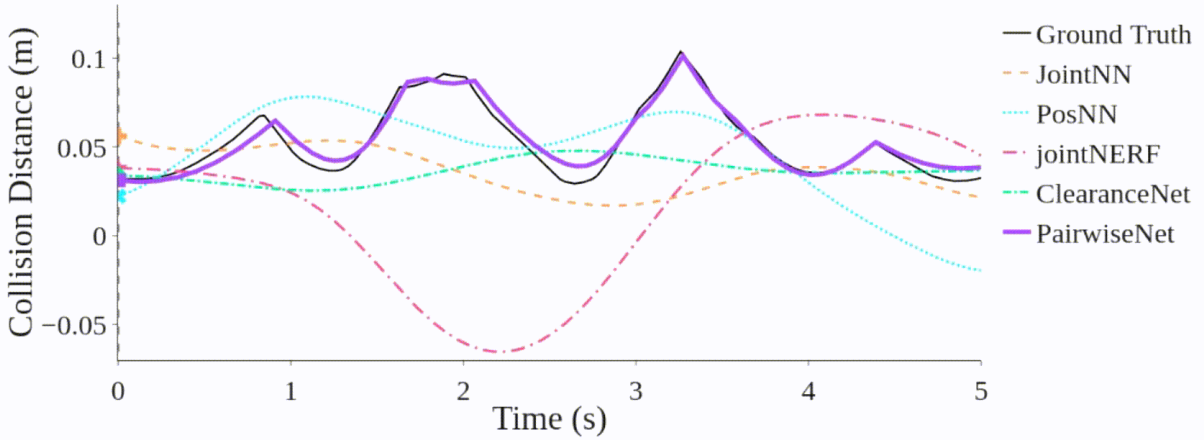
- Given initial and final joint poses q_i and q_f, find the trajectory $q(t)$

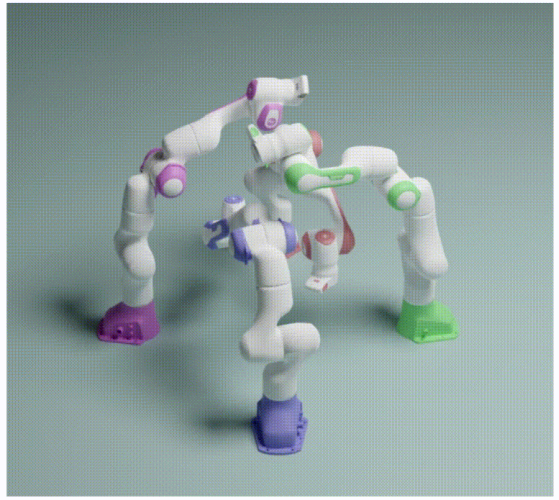
$$\underset{q(t)}{\text{minimize}} \quad \texttt{length}[q(t)]$$

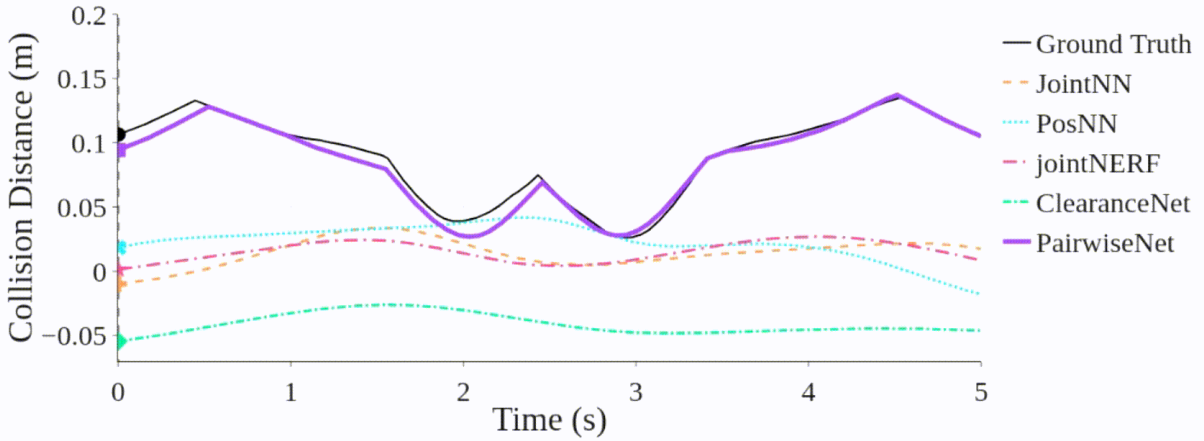
$$\text{subject to} \quad \begin{cases} q(0) = q_i,\ q(T) = q_f \\ q_{\min} \leq q(t) \leq q_{\max} \\ \dot{q}_{\min} \leq \dot{q}(t) \leq \dot{q}_{\max} \\ F_\psi(q(t); f_\psi, \mathcal{S}) \geq \epsilon \end{cases}$$

PairwiseNet : Analytically differentiable

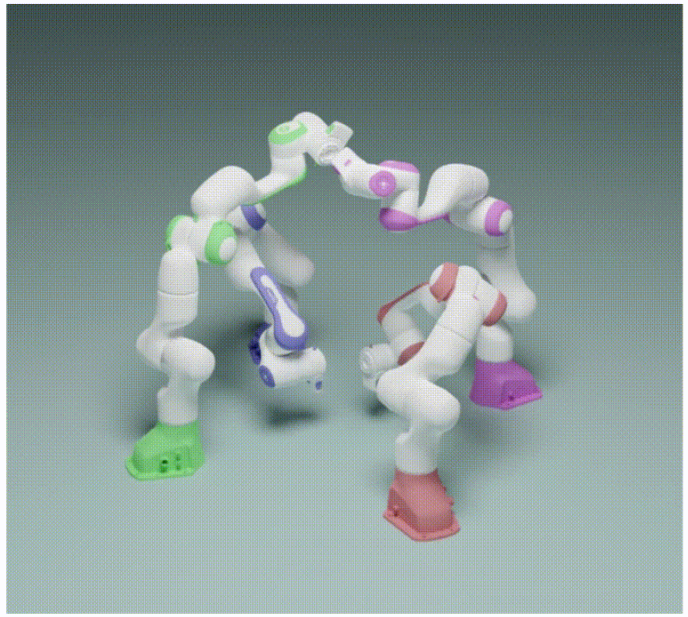
Offline Trajectory Optimization

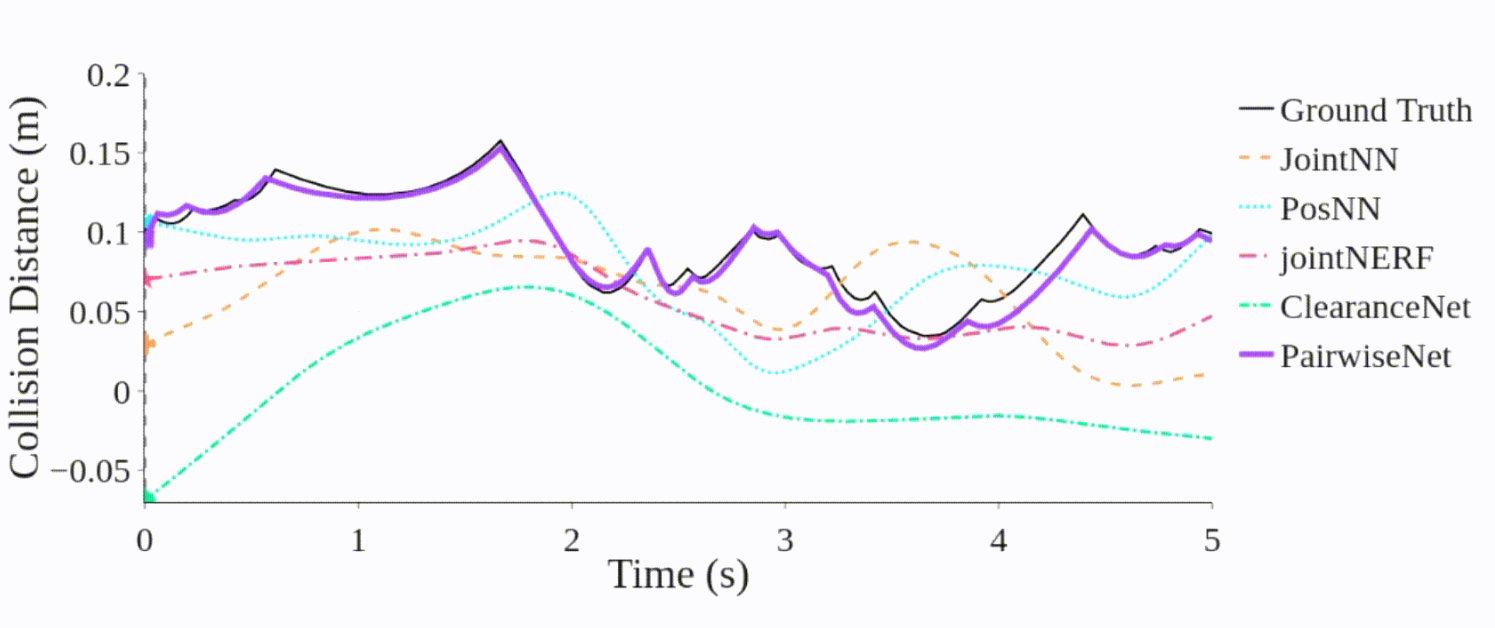






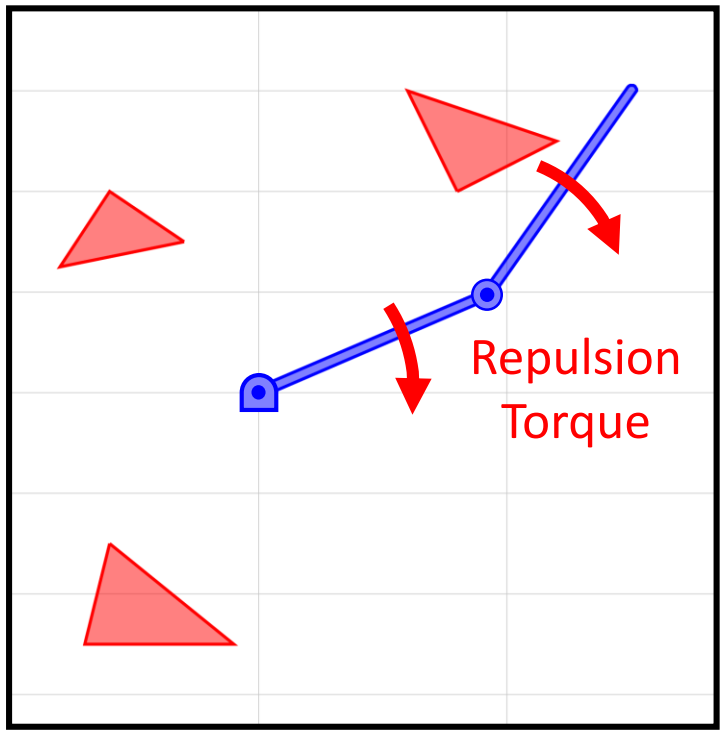
Offline Trajectory Optimization



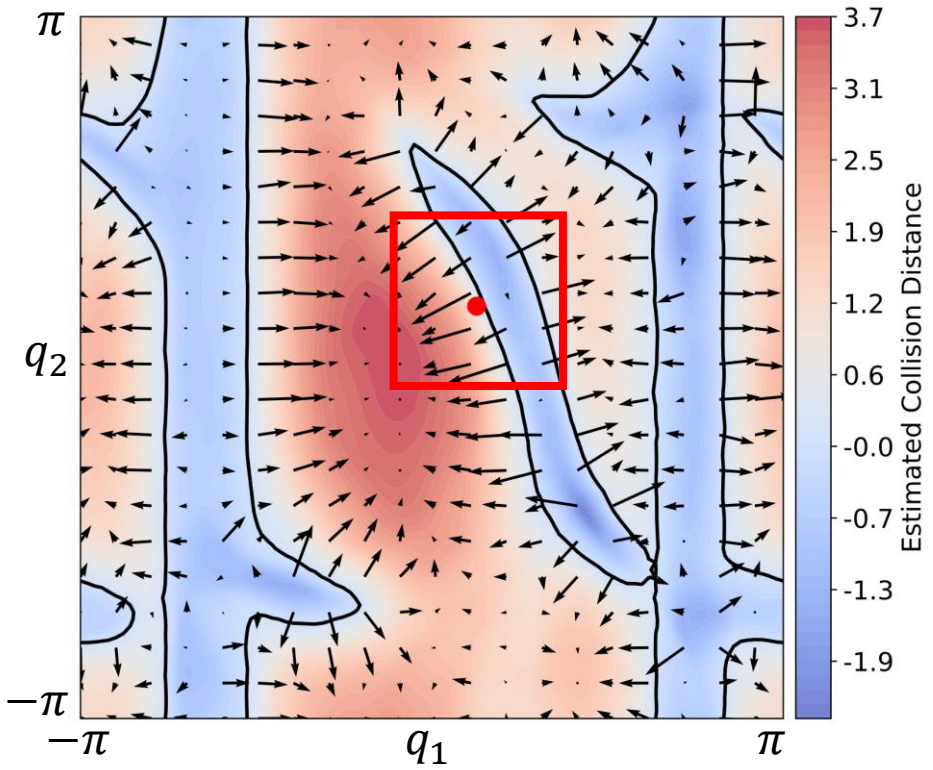


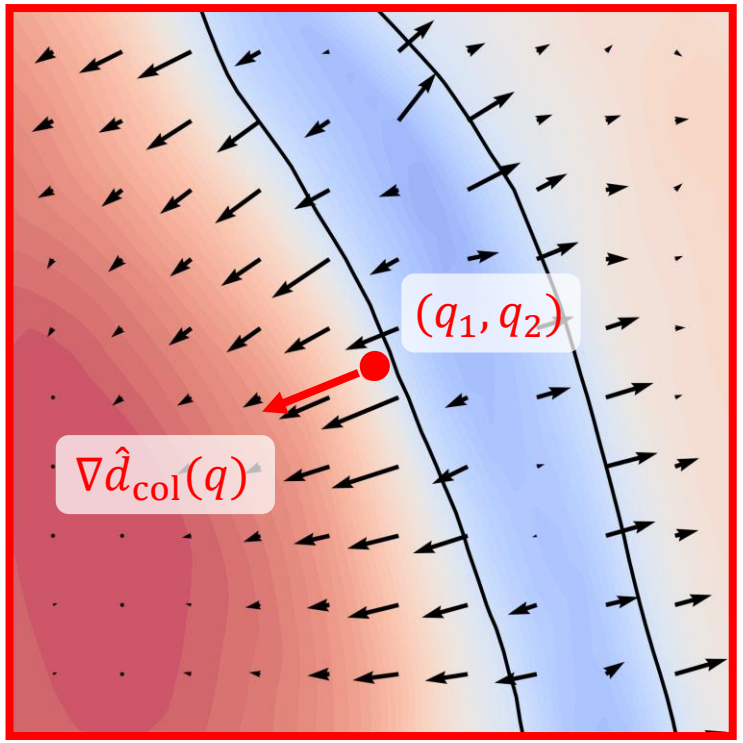
Collision Repulsion

ex) A 2R planar robot system



Collision Distance estimator
$\hat{d}_{\mathrm{col}}(q)$ and $\nabla\hat{d}_{\mathrm{col}}(q)$





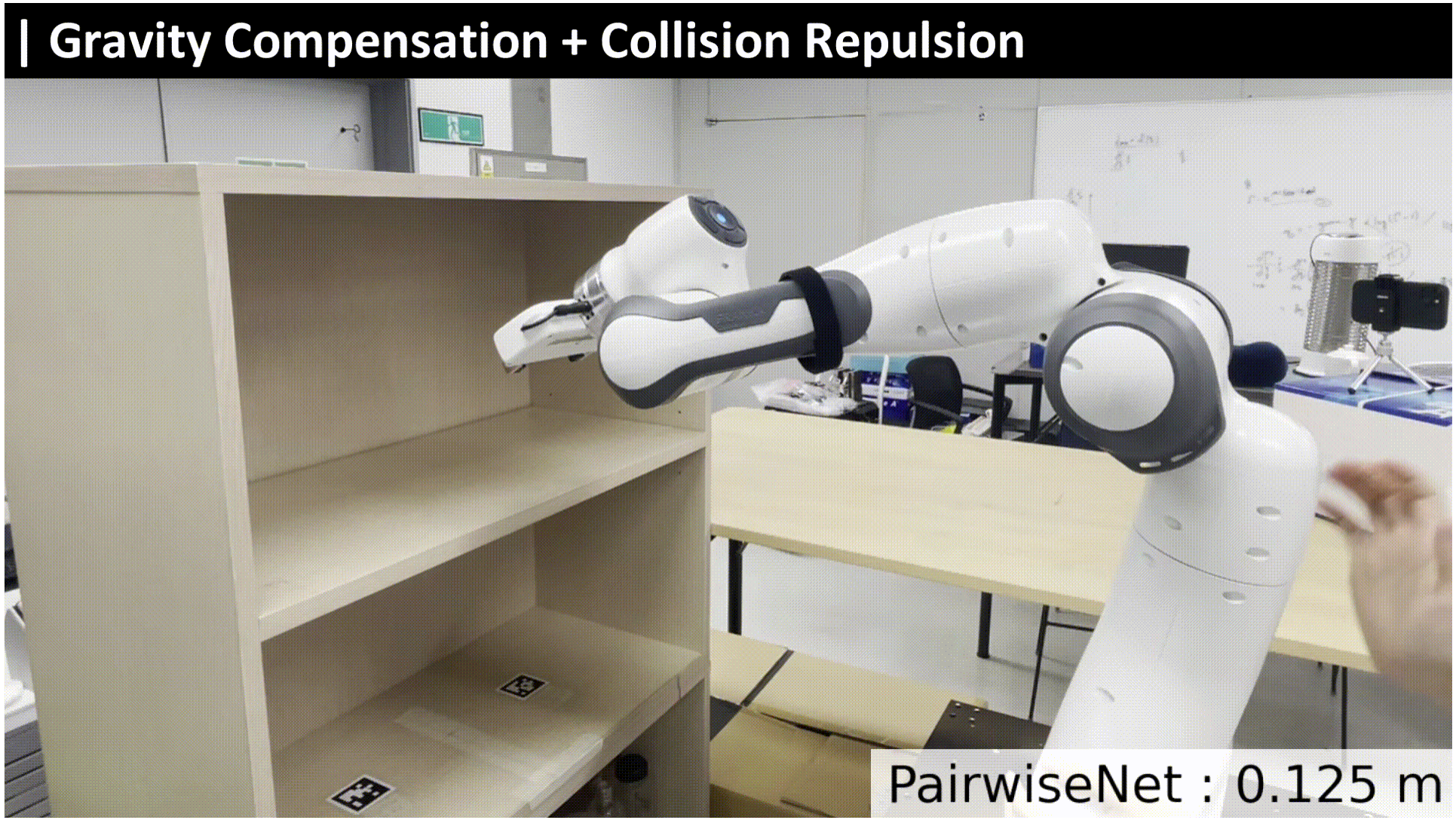
PairwiseNet: A System with Obstacles

Methods	MSE	AUROC	Accuracy ($\epsilon = 0$)	safe-FPR
JointNN	2.48e-4	0.9955	0.9738	0.1603
PosNN	1.10e-3	0.9859	0.9535	0.3075
jointNERF	2.10e-4	0.9963	0.9761	0.1481
ClearanceNet	5.63e-4	0.9877	0.9560	0.3978
DiffCo	-	0.9416	0.9132	0.9750
PairwiseNet (ours)	**6.04e-5**	**0.9983**	**0.9842**	**0.0758**

Collision Repulsion

| Gravity Compensation + Collision Repulsion

Collision Repulsion



Conclusion

- Dataset Construction → **Active Learning of the Collision Distance Function**
  - We have proposed an active learning-based training procedure.
  - This procedure iteratively generates near-boundary datasets and refines the model, improving the accuracy of collision distance estimation models.
- Inherent Complexity of the Collision Distance Function
- Generalizability to Minor Environmental Changes

→ **PairwiseNet**

  - We have proposed PairwiseNet, a novel method for collision distance learning.
  - Instead of directly predicting global collision distance, PairwiseNet estimates pairwise distances between elements.
  - Experimental results demonstrate superior performance and generalization capability.

References

Active Learning of the Collision Distance Function

- Kim, Jihwan, and Frank Chongwoo Park. *Active learning of the collision distance function for high-DOF multi-arm robot systems.* Robotica (2024): 1-15.

PairwiseNet

- Kim, Jihwan, and Frank C. Park. *PairwiseNet: Pairwise Collision Distance Learning for High-dof Robot Systems.* Conference on Robot Learning. PMLR, 2023.

Thank you for listening!

Ph. D. Thesis Defense
Collision Distance Estimation for High-dof Robot Systems: A Learning-based Approach

Jihwan Kim
jihwankim@robotics.snu.ac.kr

Chapter

Appendix

Balanced Dataset Generation

- Balanced dataset [N. B. Figueroa Fernandez, et al., MLPC'18][M. Koptev, et al., RAL'21]

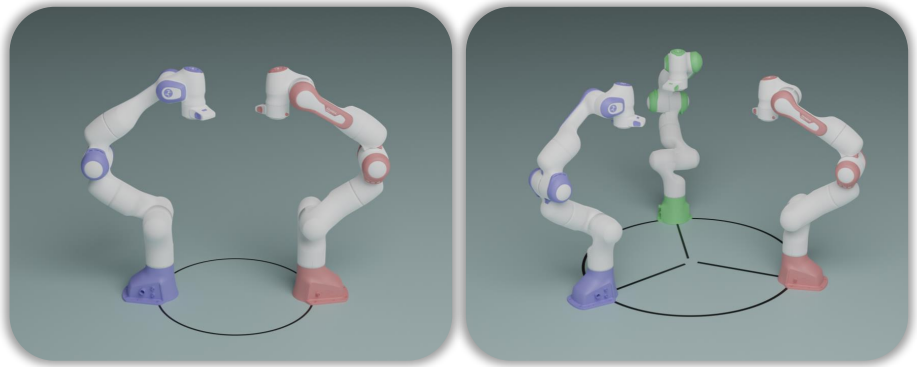
: contains 50% collision data points $d_{\text{col}} < 1\text{cm}$

35% near-collision data points $1\text{cm} < d_{\text{col}} < 5\text{cm}$

15% non-collision data points $5\text{cm} < d_{\text{col}}$

$d_{\text{col}}(q)$: Slow, no batch calculation, not differentiable

: Generating balanced datasets only relies on Rejection Sampling



Need to calculate collision distances
more than 9 times the desired # of data points

Balanced Dataset Generation

- Balanced dataset [N. B. Figueroa Fernandez, et al., MLPC'18][M. Koptev, et al., RAL'21]

: contains 50% collision data points $d_{\text{col}} < 1\text{cm}$

35% near-collision data points $1\text{cm} < d_{\text{col}} < 5\text{cm}$

15% non-collision data points $5\text{cm} < d_{\text{col}}$

$d_{\text{col}}(q)$: Slow, no batch calculation, not differentiable

: Generating balanced datasets only relies on Rejection Sampling

$\hat{d}_{\text{col}}(q)$: Fast, batch calculation, differentiable

: practically feasible to employ advanced sampling methods (e.g., MCMC, LMC)

Balanced Dataset Generation

- Balanced dataset [N. B. Figueroa Fernandez, et al., MLPC'18][M. Koptev, et al., RAL'21]

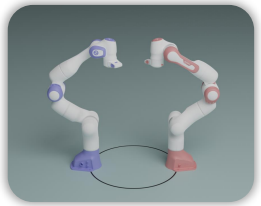
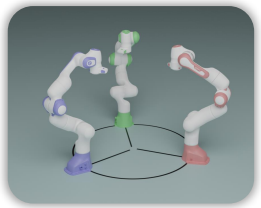
  : contains 50% collision data points $d_{\mathrm{col}} < 1\mathrm{cm}$

  35% near-collision data points $1\mathrm{cm} < d_{\mathrm{col}} < 5\mathrm{cm}$

  15% non-collision data points $5\mathrm{cm} < d_{\mathrm{col}}$

Env.	Model	Training	Accuracy (ϵ=0.0)	AUROC	near-Accuracy (ϵ=0.0)	near-AUROC
Fig. 3.5 (Two arms)	JointNN	*none*	0.9765	0.9942	0.8799	0.9491
		balanced	0.9767	0.9942	0.8854	0.9552
		active	**0.9809**	**0.9960**	**0.9018**	**0.9643**
Fig. 3.5 (Three arms)	JointNN	*none*	0.9691	0.9868	0.8189	0.8875
		balanced	0.8887	0.9118	0.6422	0.7229
		active	**0.9765**	**0.9918**	**0.8609**	**0.9280**

Comparison with Uniform Sampling Update

Training Method		*uniform*				*active*			
Env.	Model	Accuracy (ϵ=0.0)	AUROC	near-Accuracy (ϵ=0.0)	near-AUROC	Accuracy (ϵ=0.0)	AUROC	near-Accuracy (ϵ=0.0)	near-AUROC
	JointNN	0.9788	0.9953	0.8913	0.9585	0.9809	0.9960	0.9018	0.9643
	PosNN	0.9821	0.9969	0.9079	0.9720	0.9828	0.9971	0.9117	0.9735
	ClearanceNet	0.9659	0.9879	0.8278	0.9005	0.9707	0.9900	0.8516	0.9186
	SE3NN	0.9878	0.9986	0.9375	0.9869	0.9886	0.9987	0.9416	0.9877
	JointNN	0.9715	0.9888	0.8321	0.9028	0.9765	0.9918	0.8609	0.9280
	PosNN	0.9809	0.9949	0.8869	0.9535	0.9821	0.9955	0.8937	0.9587
	ClearanceNet	0.9478	0.9590	0.7105	0.7463	0.9532	0.9619	0.7377	0.7746
	SE3NN	0.9847	0.9967	0.9092	0.9701	0.9858	0.9971	0.9157	0.9738

Direct Point Cloud Distance Computation

- Three advantages of PairwiseNet compared to the direct point cloud distance computation
  - **GPU memory consumption** : Allows larger batch size of joint configurations.
  - **Inference complexity** : Independent of point cloud data size.
  - **Inference time on CPU** : Enables rapid inference even without GPU.

Method	Inference time for 1000 joint poses (s)					
	Two arms (64 pairs)		Three arms (192 pairs)		Four arms (384 pairs)	
	CPU	GPU	CPU	GPU	CPU	GPU
Direct distance	50.14	0.2325	151.4	0.3903	308.9	0.7966
Direct distance (Batch)	55.68	out of memory	174.7	out of memory		
PairwiseNet	0.1054	0.0988	0.1590	0.0998	0.2025	0.1045
PairwiseNet (Batch)	0.0362	0.0207	0.1070	0.0224	0.2121	0.0253