CS686: Presentation

PairwiseNet: Pairwise Collision Distance Learning for High-dof Robot Systems

Kim et al.

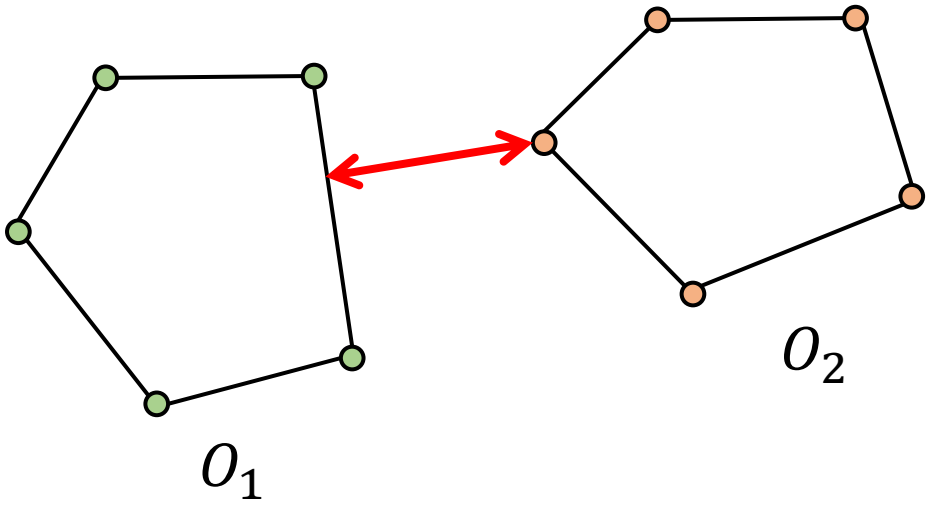
Minjae Song
(송민재)

KAIST

Background – Collision distance

Collision distance

: minimum distance between the robot and its nearest obstacle

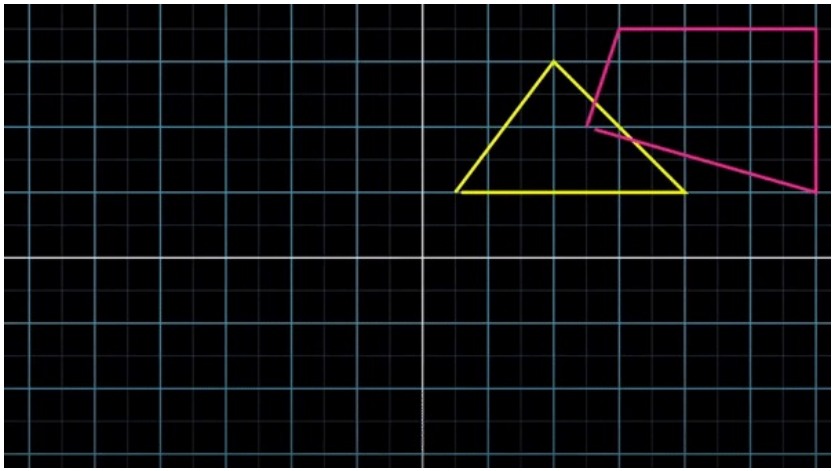


KAIST

Background

Geometrical method

GJK Algorithm (Gilbert et al., 88')

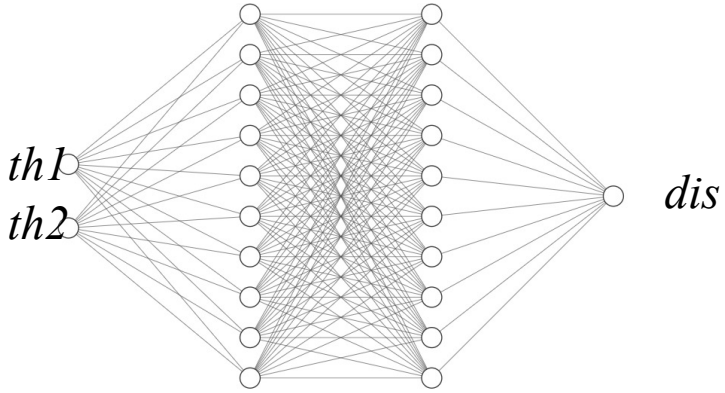


Pros
- Accurate
- Universal for any shape

Cons
- Slow for large number of queries

Data-driven method

ClearanceNet (Kew et al., 20')



MLP

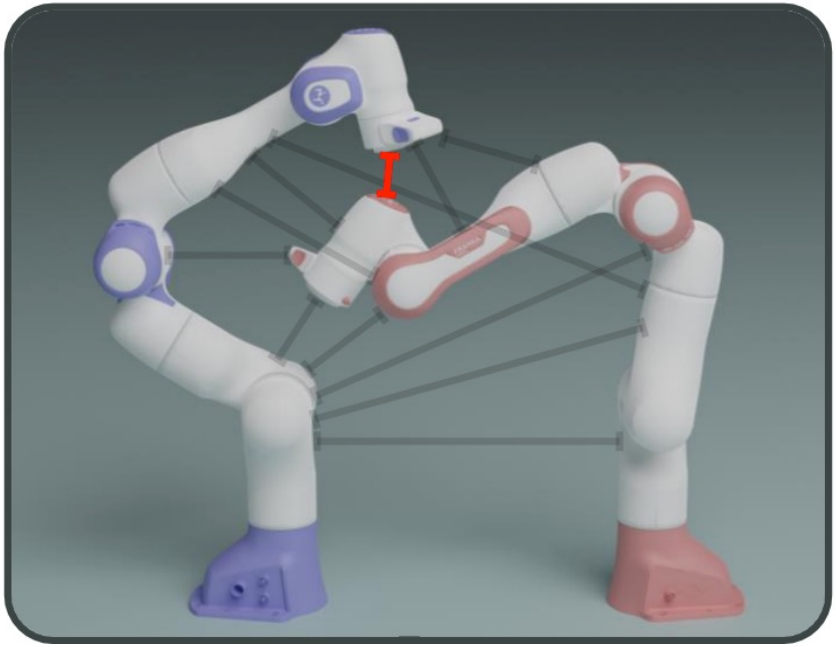
Pros
- Faster for large number of queries

Cons
- Unscalable
- Possibly not accurate

KAIST

Problem

- **Non convex objects**
- **<u>High dof robots</u>**

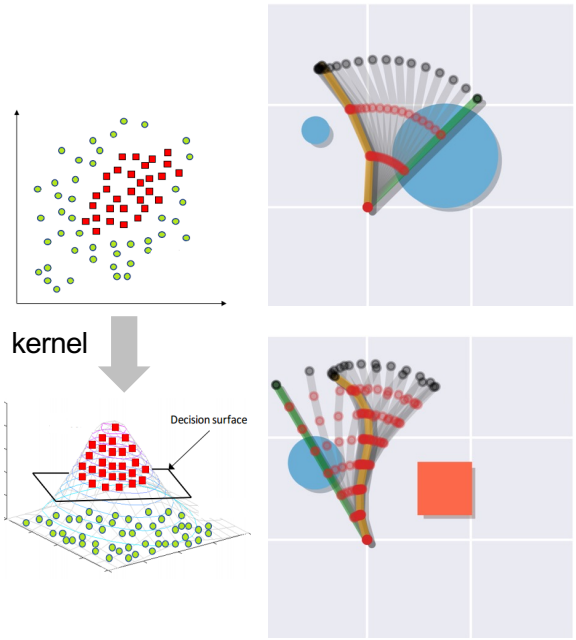


Computational burden becomes intensive.

KAIST

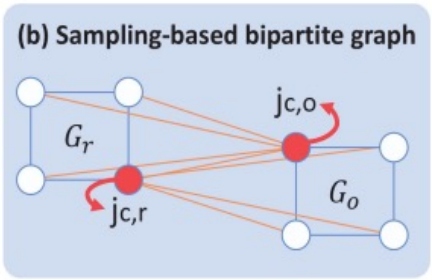
Related works

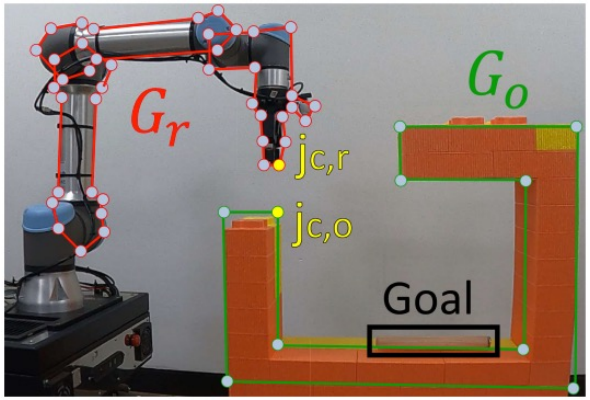
Configuration-based representation



DiffCo
Zhi et al., T-RO 2022

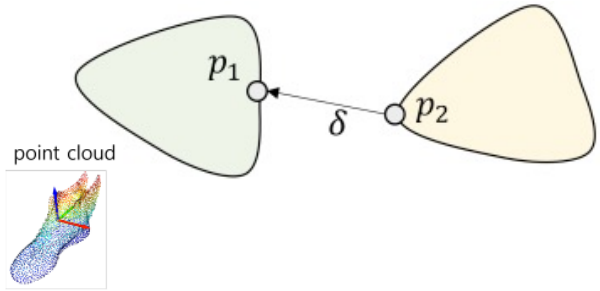
Graph-based representation

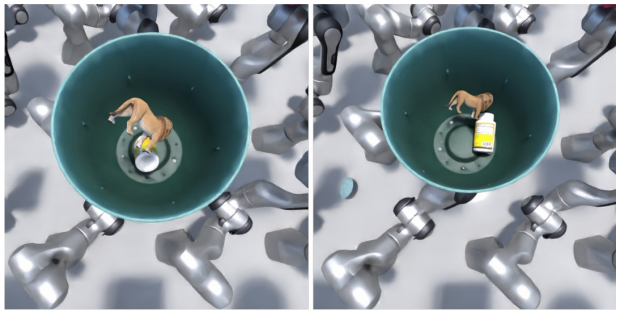




GraphDistNet
Kim et al., RA-L 2022

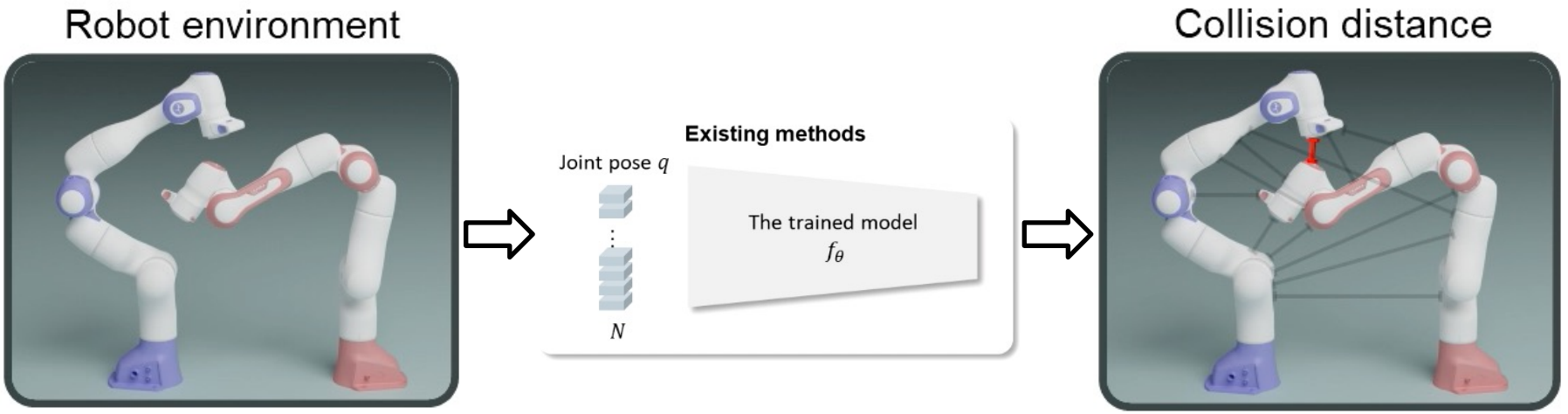
Pointcloud-based representation





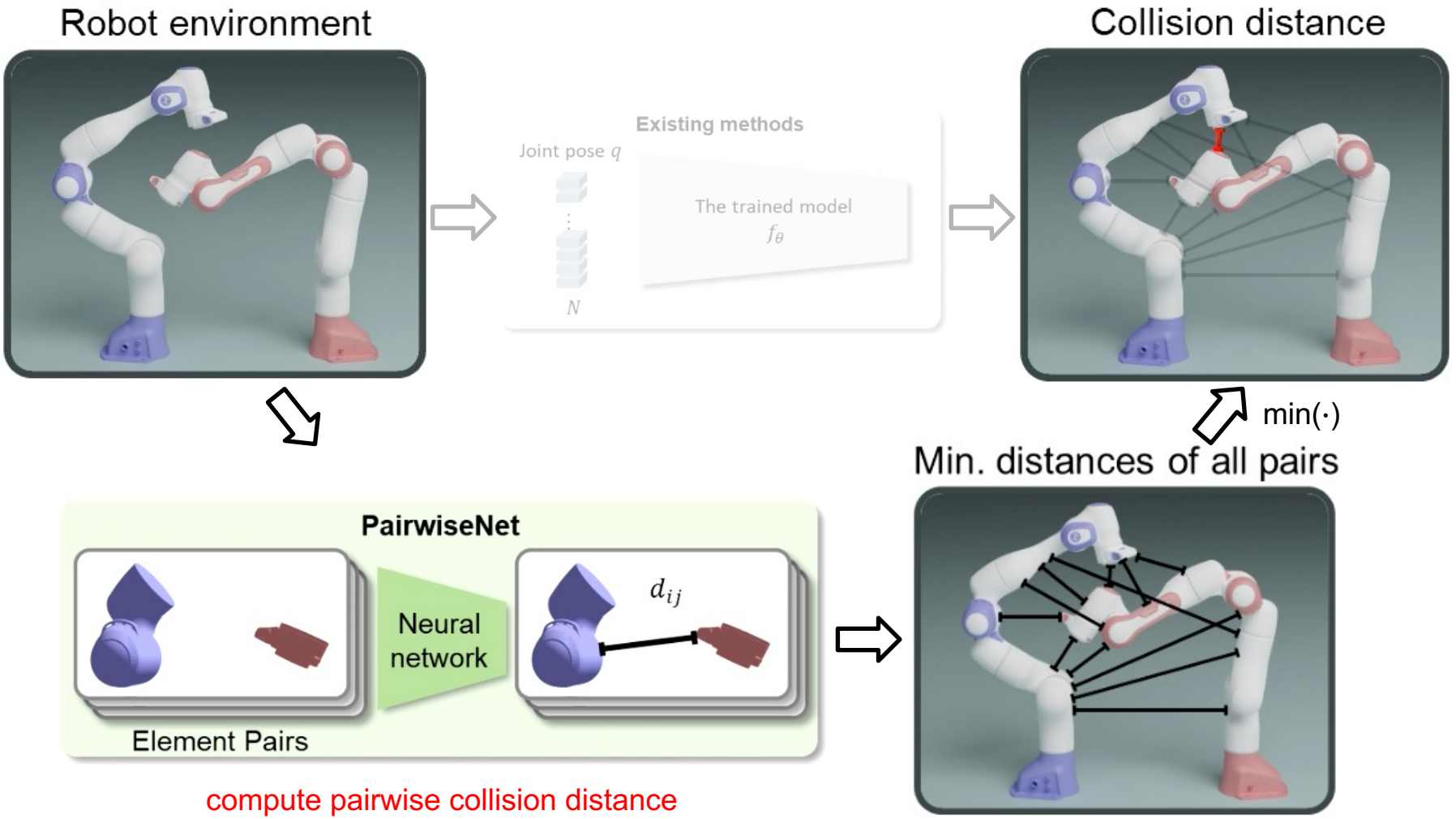
LOCC
Son et al., RSS 2023

Estimate global collision distance

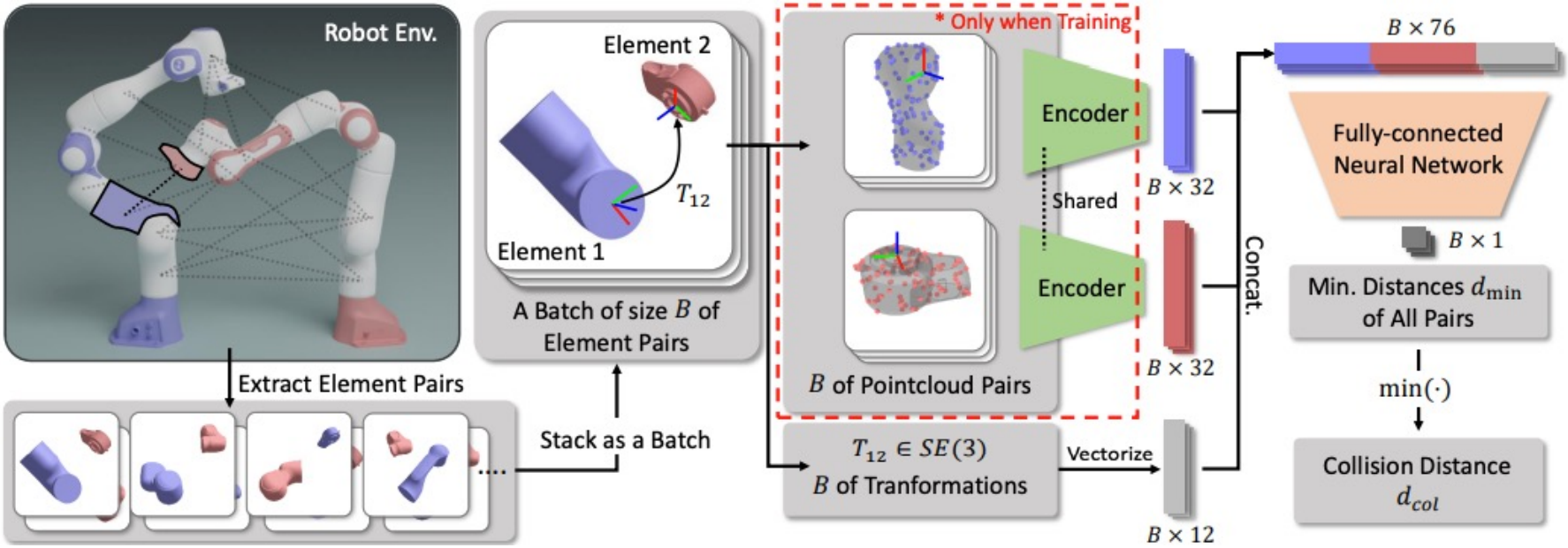


- bad performance at high DOF robots
- sensitive to environment change
- poor generalizability

Main Idea

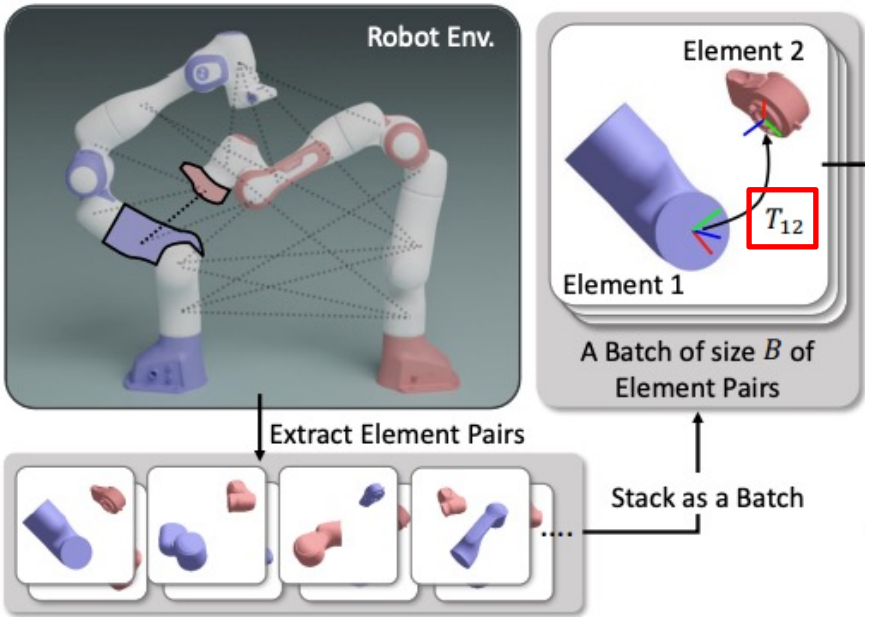

Robot environment
Existing methods
Joint pose q
The trained model
f_θ
N
Collision distance
min(·)
Min. distances of all pairs
PairwiseNet
Neural network
d_{ij}
Element Pairs
compute pairwise collision distance

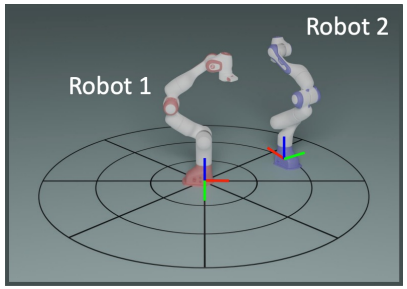
Overall framework



KAIST

Overall framework

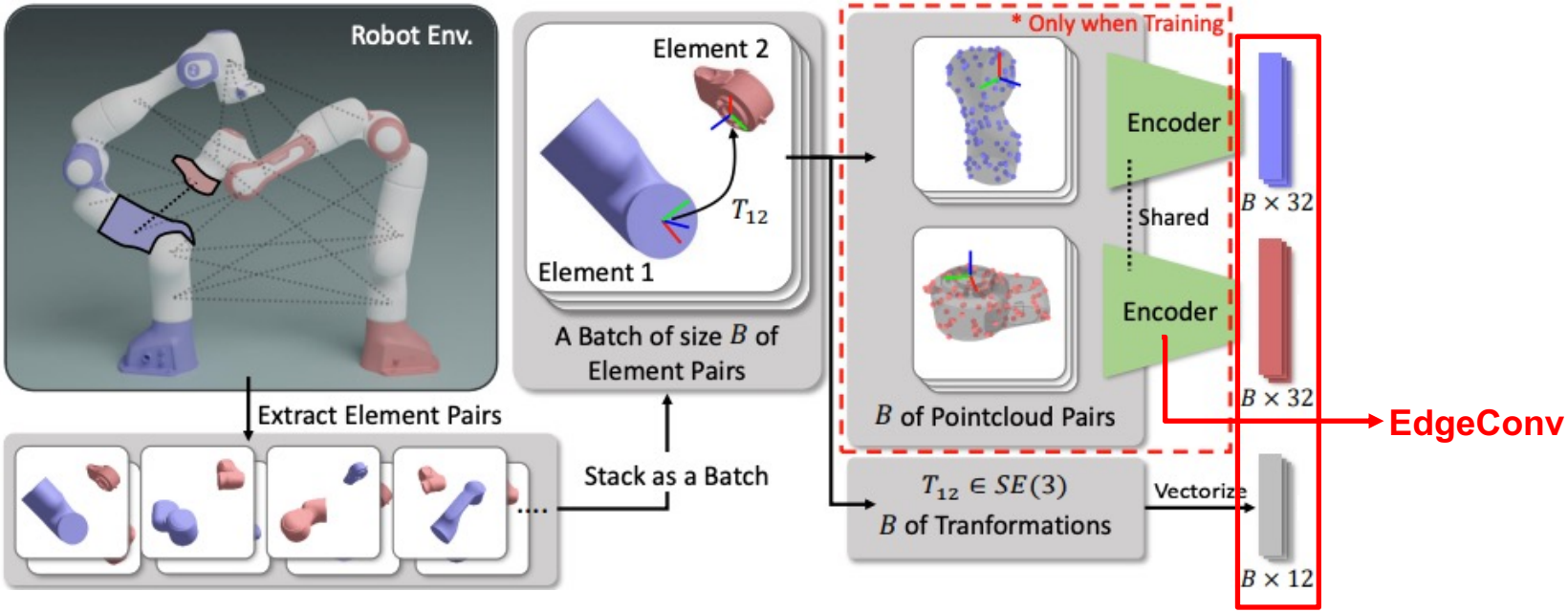




- 320 different combination
- 3million data
- Pybullet collision library (GJK algorithm)

KAIST

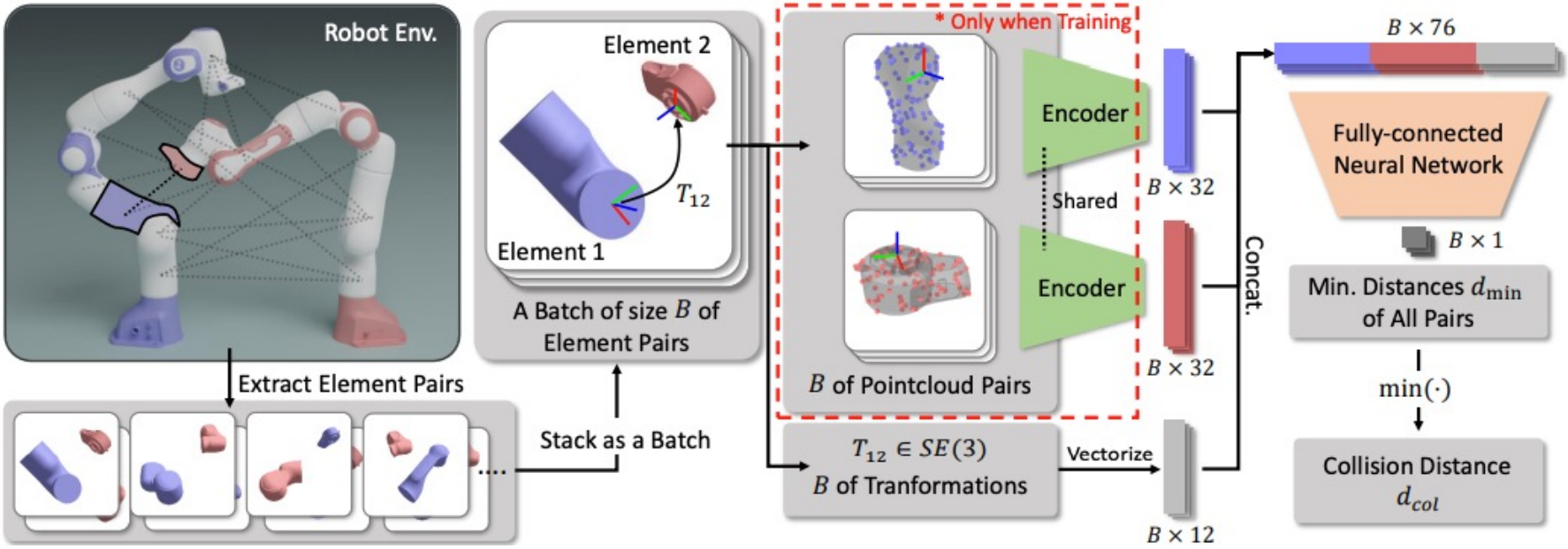
Overall framework



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

KAIST

Overall framework



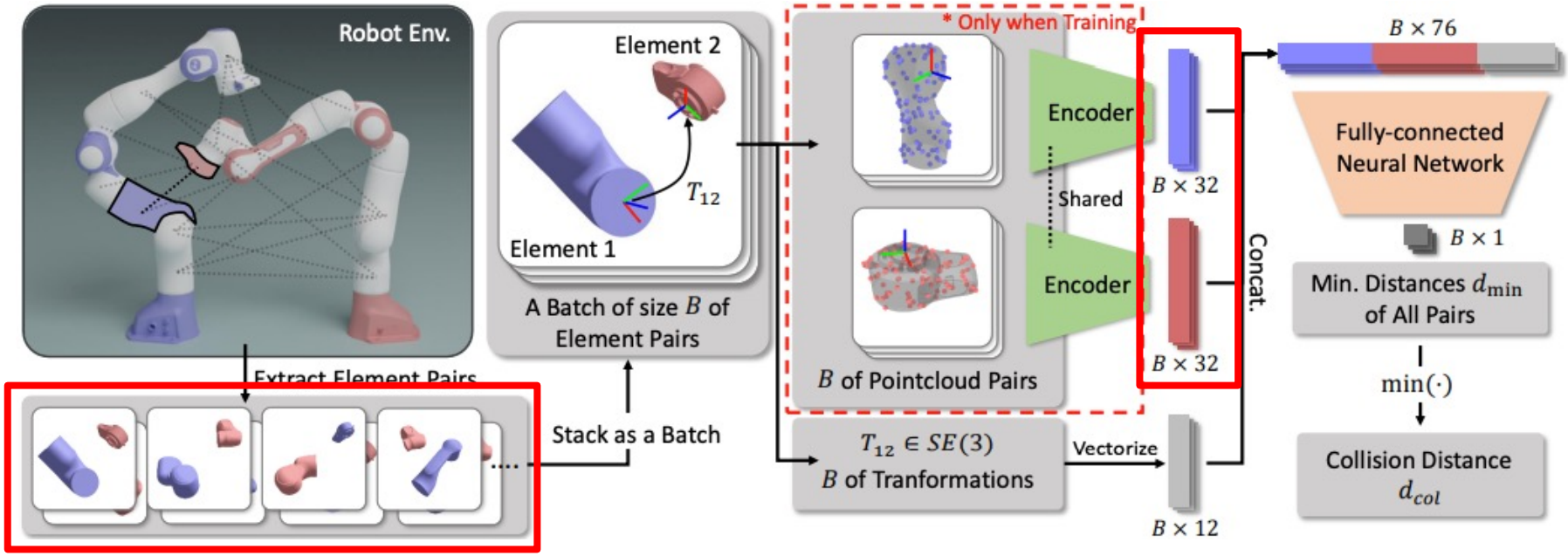
$$\hat{d}_{\text{col}}(q) = \min_{(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q)) \in \mathcal{S}(q)} f_\psi(\mathcal{P}_i, \mathcal{P}_j, T_{ij}(q))$$

$$L = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\mathcal{P}_i, \mathcal{P}_j, T_{ij}, d_{ij}) \in \mathcal{D}_{\text{train}}} ||f_\psi(\mathcal{P}_i, \mathcal{P}_j, T_{ij}) - d_{ij}||^2$$

KAIST

Contribution (1)

Efficient inference strategy

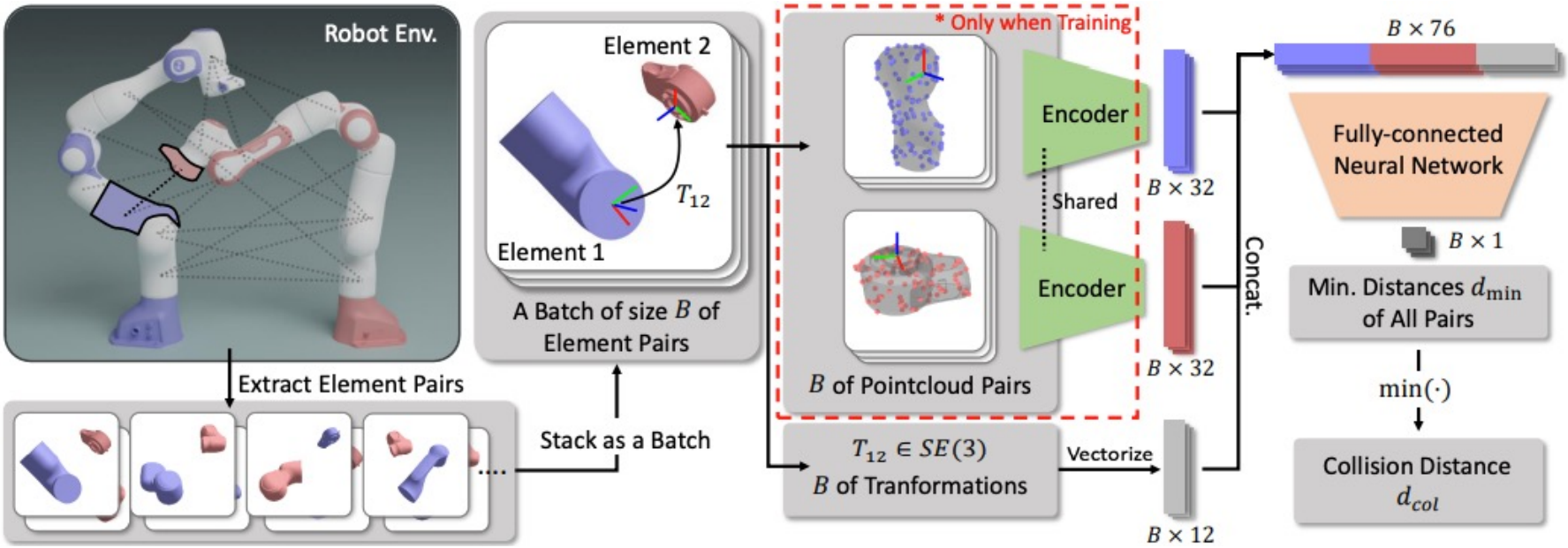


- Fully utilize batch computation
- No need of running encoder multiple times

KAIST

Contribution (2)

Applicability to system

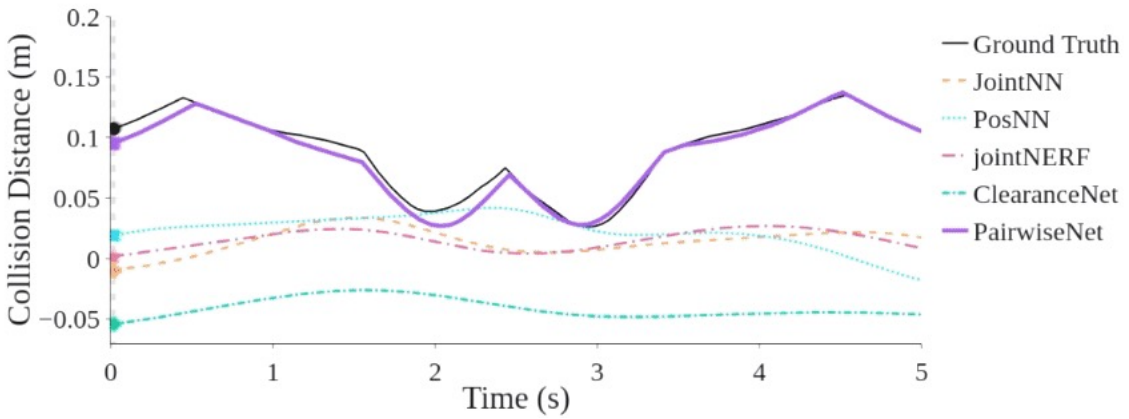


- Changing base / adding additional robot
- No need of re-training

KAIST

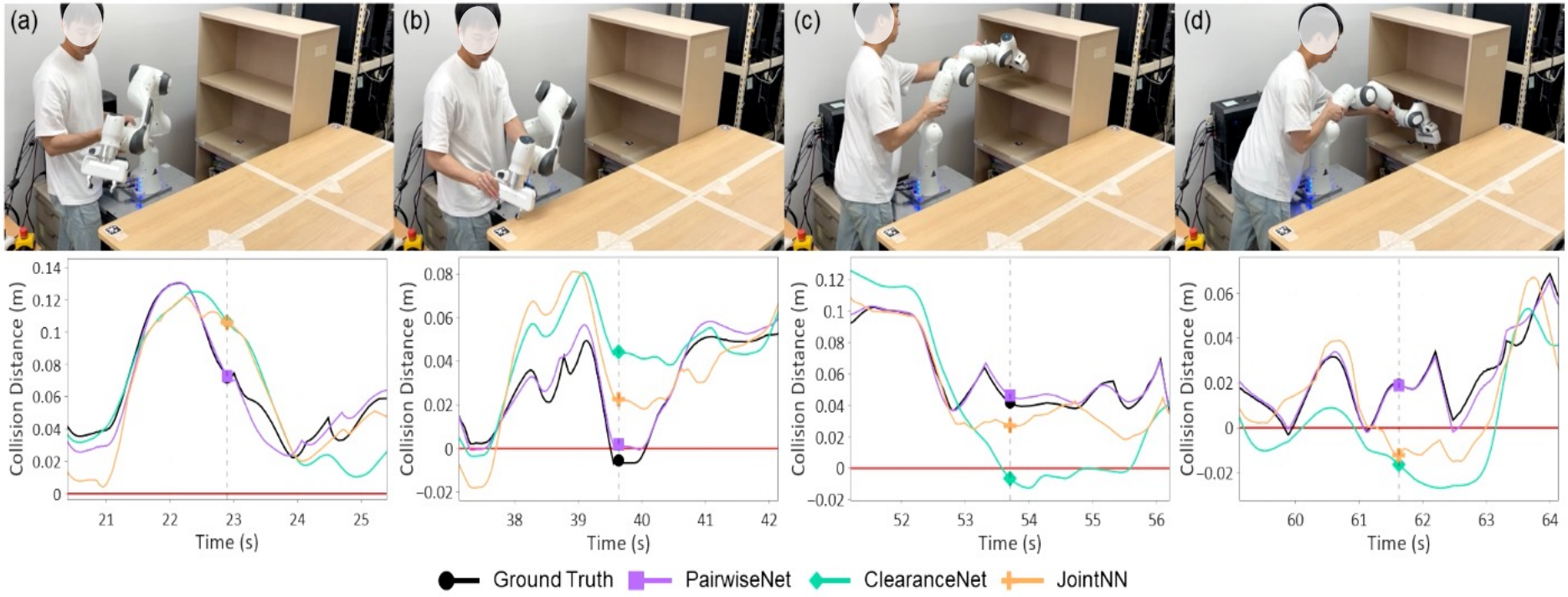
Results

- **MSE**
- **AUROC**
- **Accuracy**
- **Safe-FPR**



Results

- **Real-world experiment**



Discussion

- **Validation of the use of point cloud representation?**

- **Strong assumption, known geometry. Any suggestion for improvement?**

Q&A

- **Any question?**

Quizz

Q1. Which is NOT a typically used representation formula when estimating collision distance?

a. point cloud
b. graph
c. RGB image
d. configuration
e. voxel

Quizz

Q2. Describe the definition of collision distance.