GNN with Attention Mechanisms for Path Planning

David Alvear

King Abdullah University of Science and Technology

Reducing Collision Checking for Sampling-Based Motion Planning Using Graph Neural Networks

Chenning Yu
Computer Science and Engineering
UC San Diego
chy010@ucsd.edu

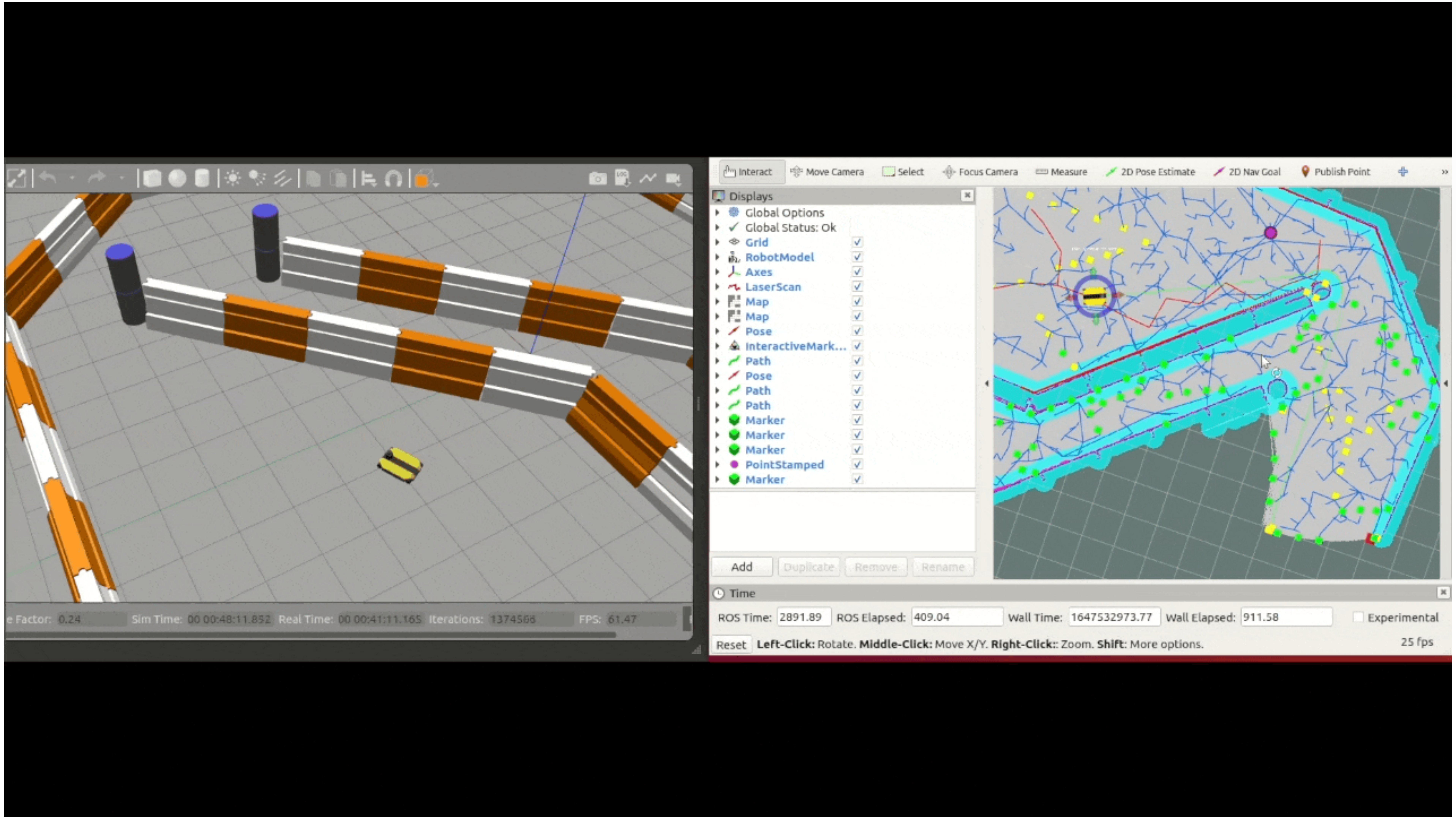
Sicun Gao
Computer Science and Engineering
UC San Diego
sicung@ucsd.edu

Abstract

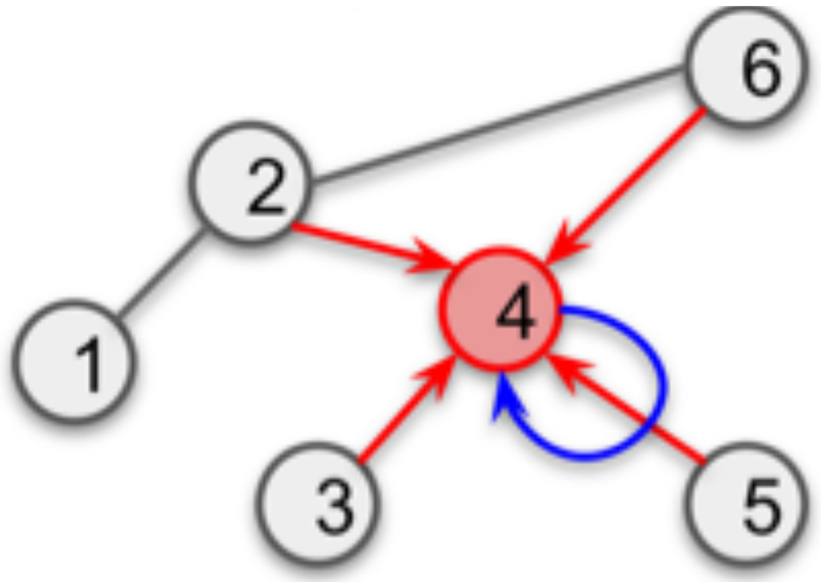
Sampling-based motion planning is a popular approach in robotics for finding paths in continuous configuration spaces. Checking collision with obstacles is the major computational bottleneck in this process. We propose new learning-based methods for reducing collision checking to accelerate motion planning by training graph neural networks (GNNs) that perform path exploration and path smoothing. Given random geometric graphs (RGGs) generated from batch sampling, the path exploration component iteratively predicts collision-free edges to prioritize their exploration. The path smoothing component then optimizes paths obtained from the exploration stage. The methods benefit from the ability of GNNs of capturing geometric patterns from RGGs through batch sampling and generalize better to unseen environments. Experimental results show that the learned components can significantly reduce collision checking and improve overall planning efficiency in challenging high-dimensional motion planning tasks.

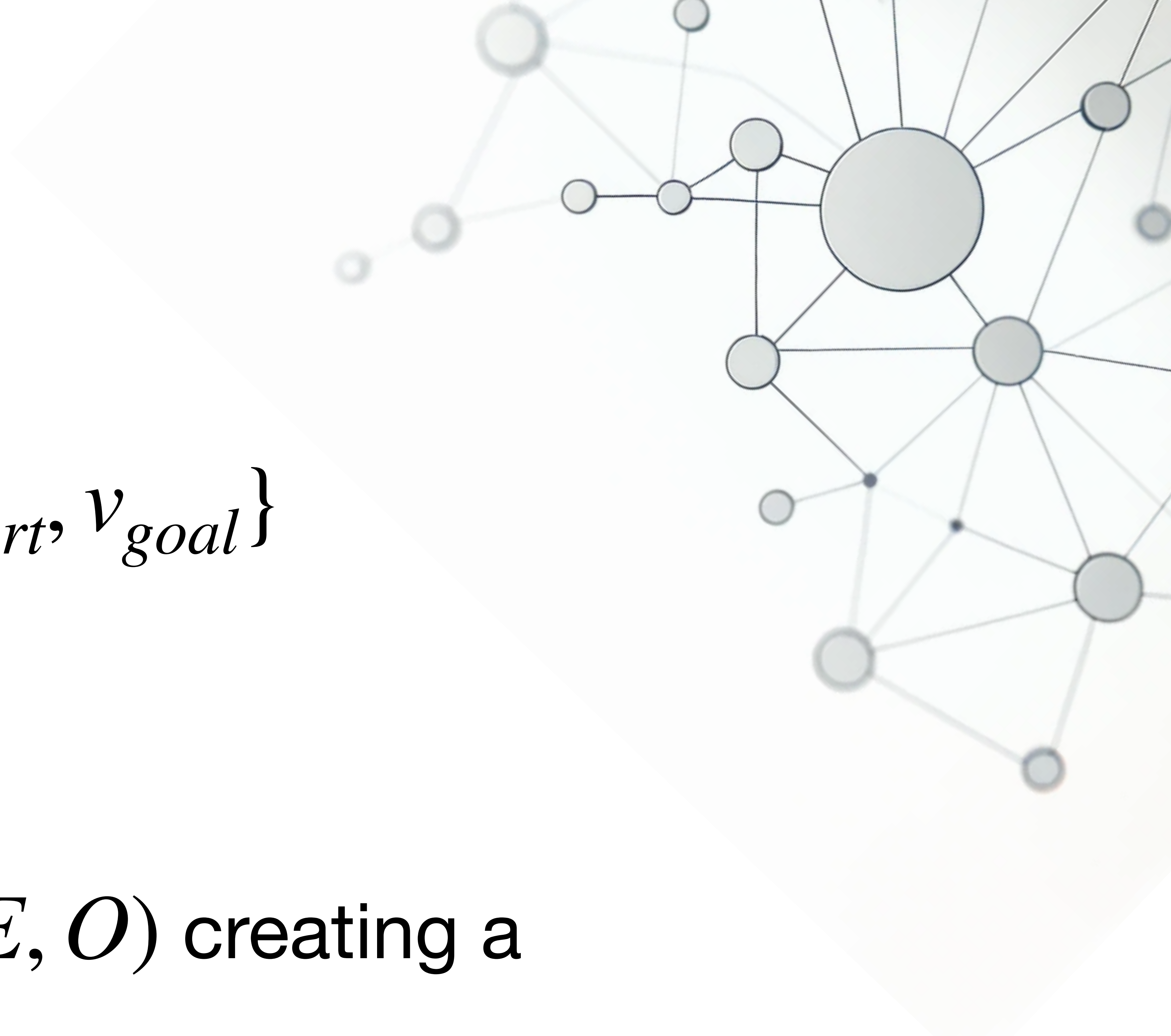
- C. Yu and S. Gao, "Reducing collision checking for sampling-based motion planning using graph neural networks," in Advances in Neural Information Processing Systems, vol. 34, pp. 4274–4289, 2021.

Path Planning Problem

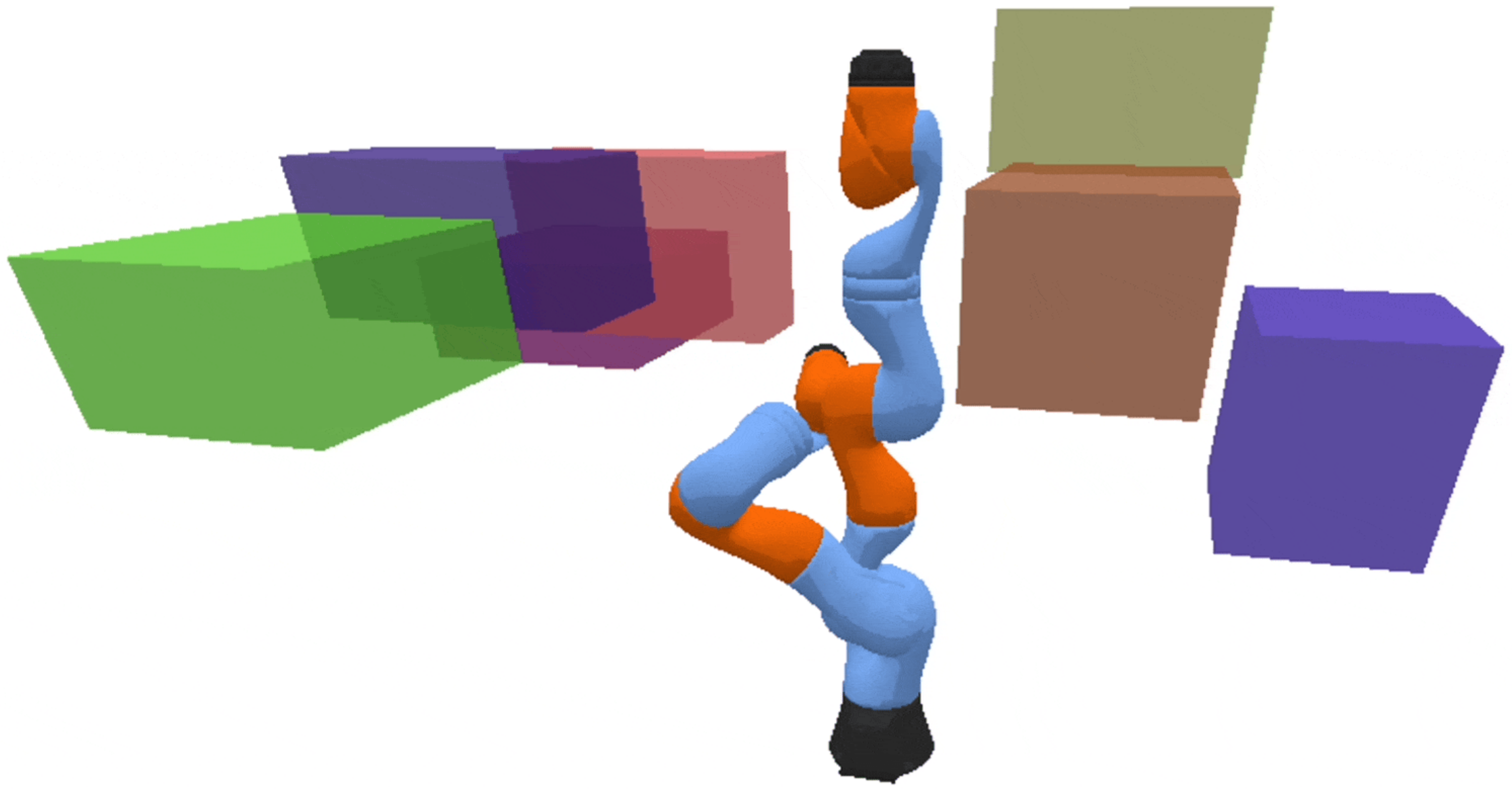


Path Exploration

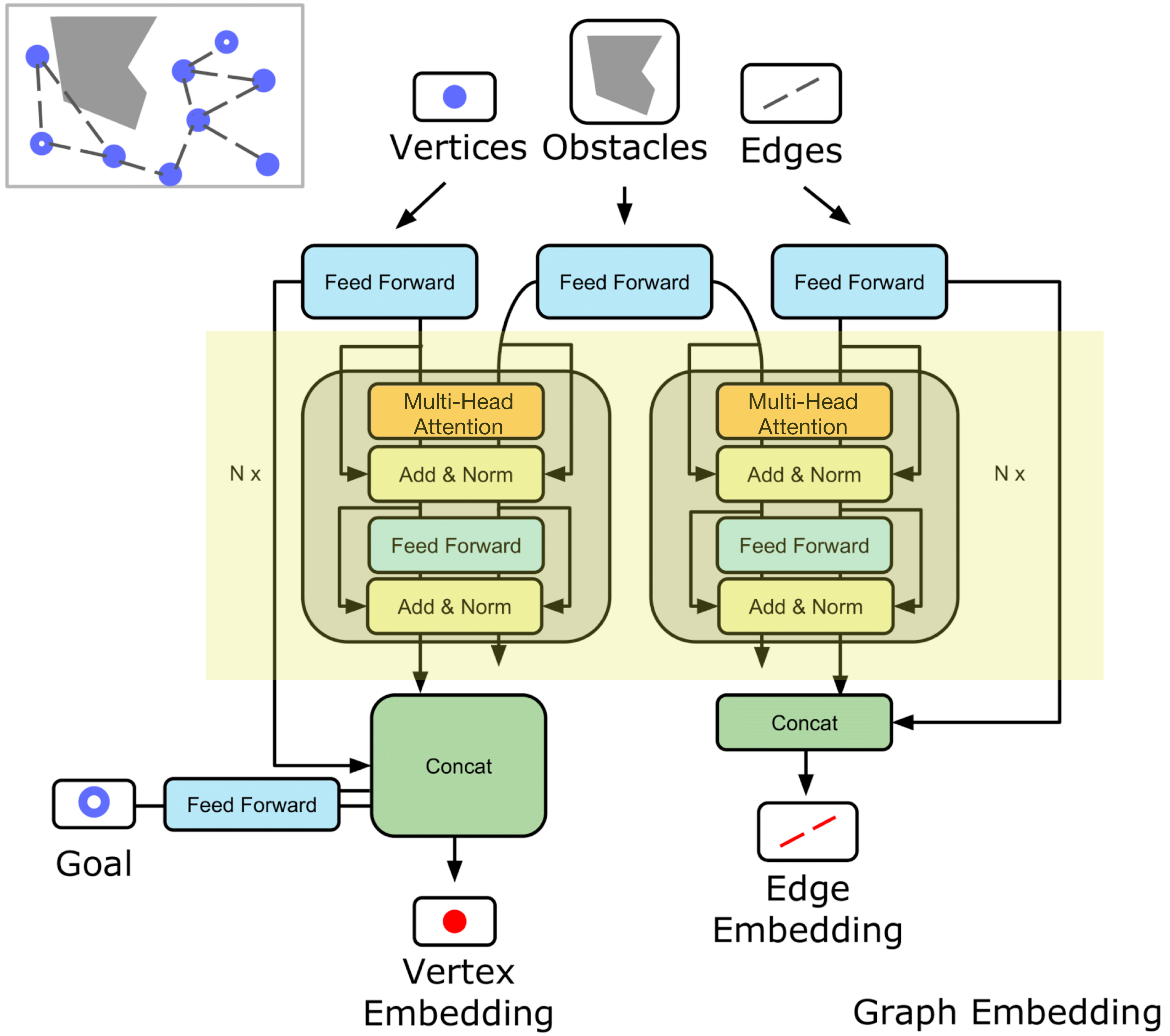


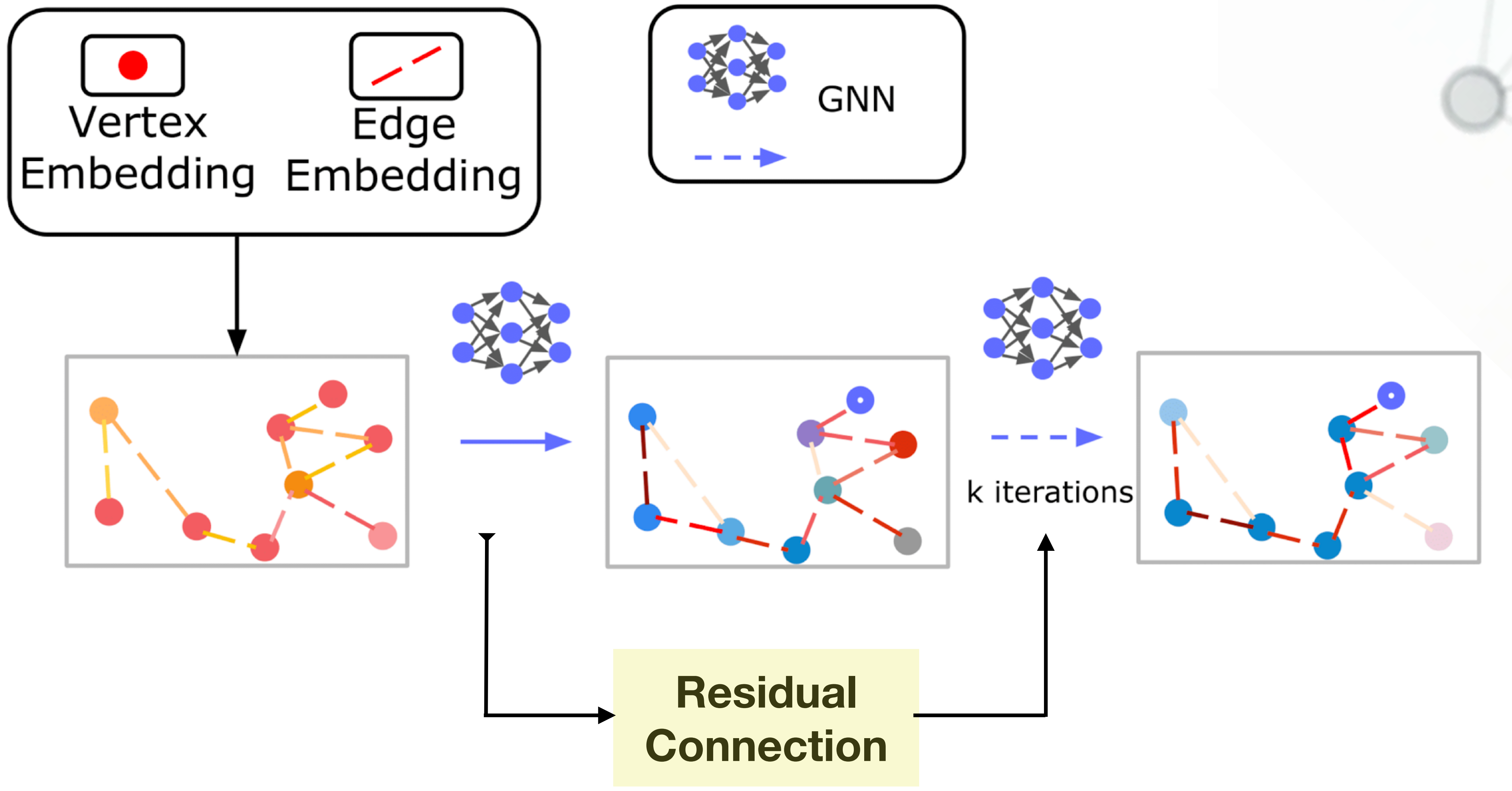


- Problem definition: sample n nodes $\{V_{free}, V_{obs}\}$ and $\{v_{start}, v_{goal}\}$
- Sample using K-NN and create the graph
  $G = \{v : \{v_s, v_g\} \cup V_{free} \cup V_{obs}, E : KNN(V_{free}) \cup KNN(V_{obs})\}$
- Explore the graph predicting the edge priority $\eta = N_E(V, E, O)$ creating a tree τ of explored edges until v_{goal} is in τ
- Nodes features: $\{v_i, (v_i - v_g)^2, v_i - v_g\}$
- GNN Update: $h_i^{l+1} = \sigma(h_i^l W_0^l + \sum_{j \in N} \frac{1}{c_{ij}} h_j^l W_1^l)$
- Train model using reference algorithm **Dijkstra**



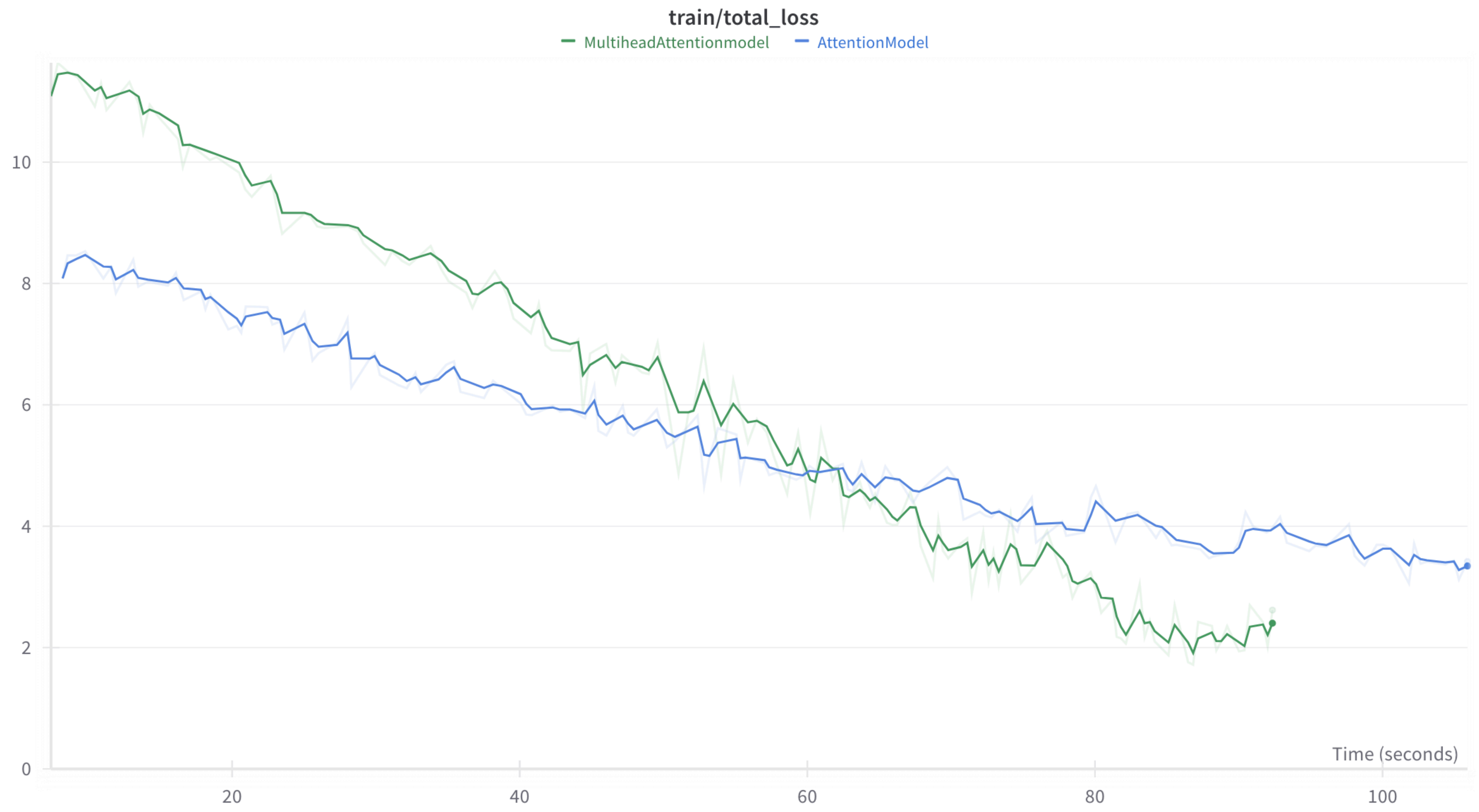
Multi-Head Attention Model



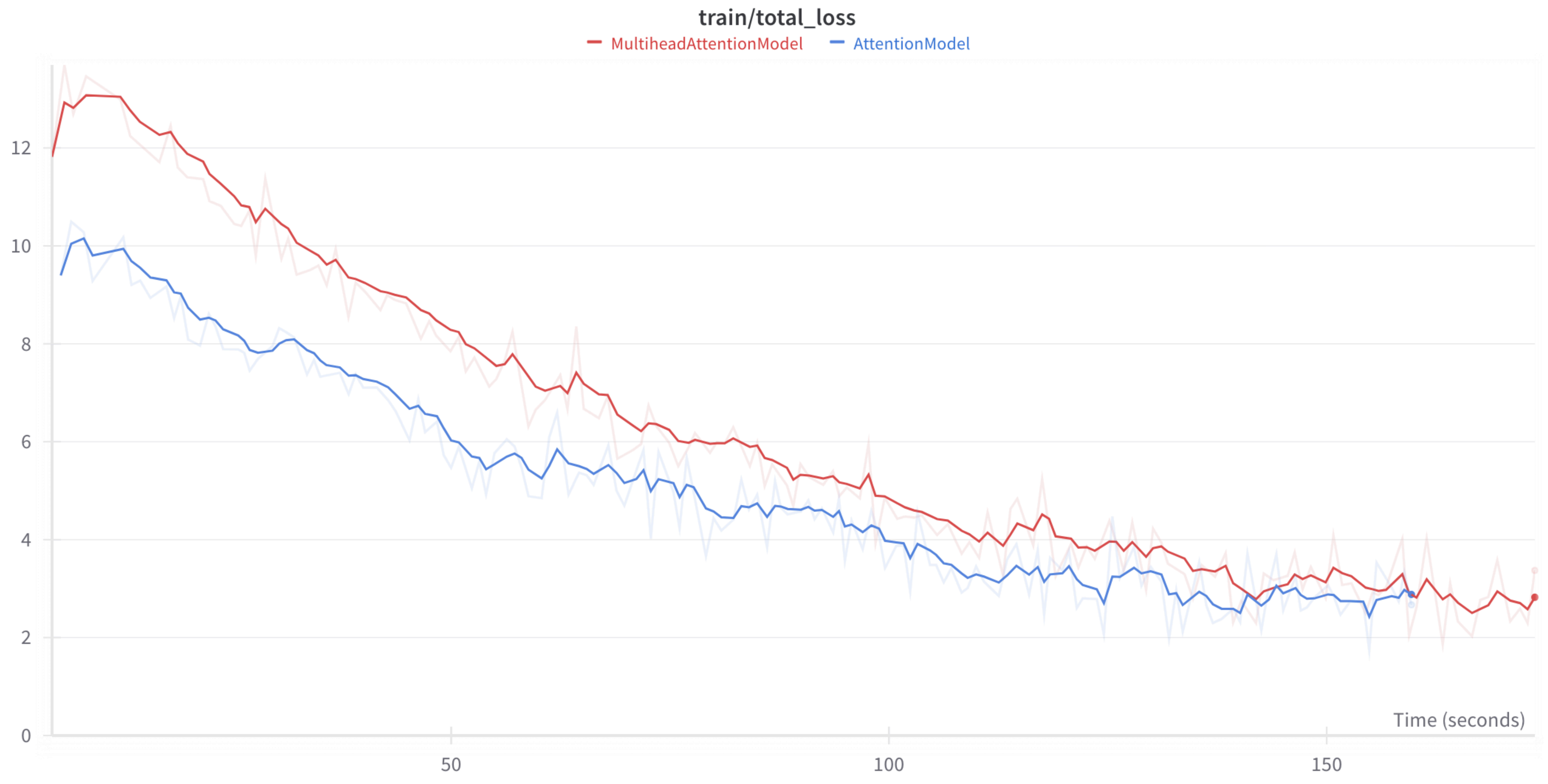


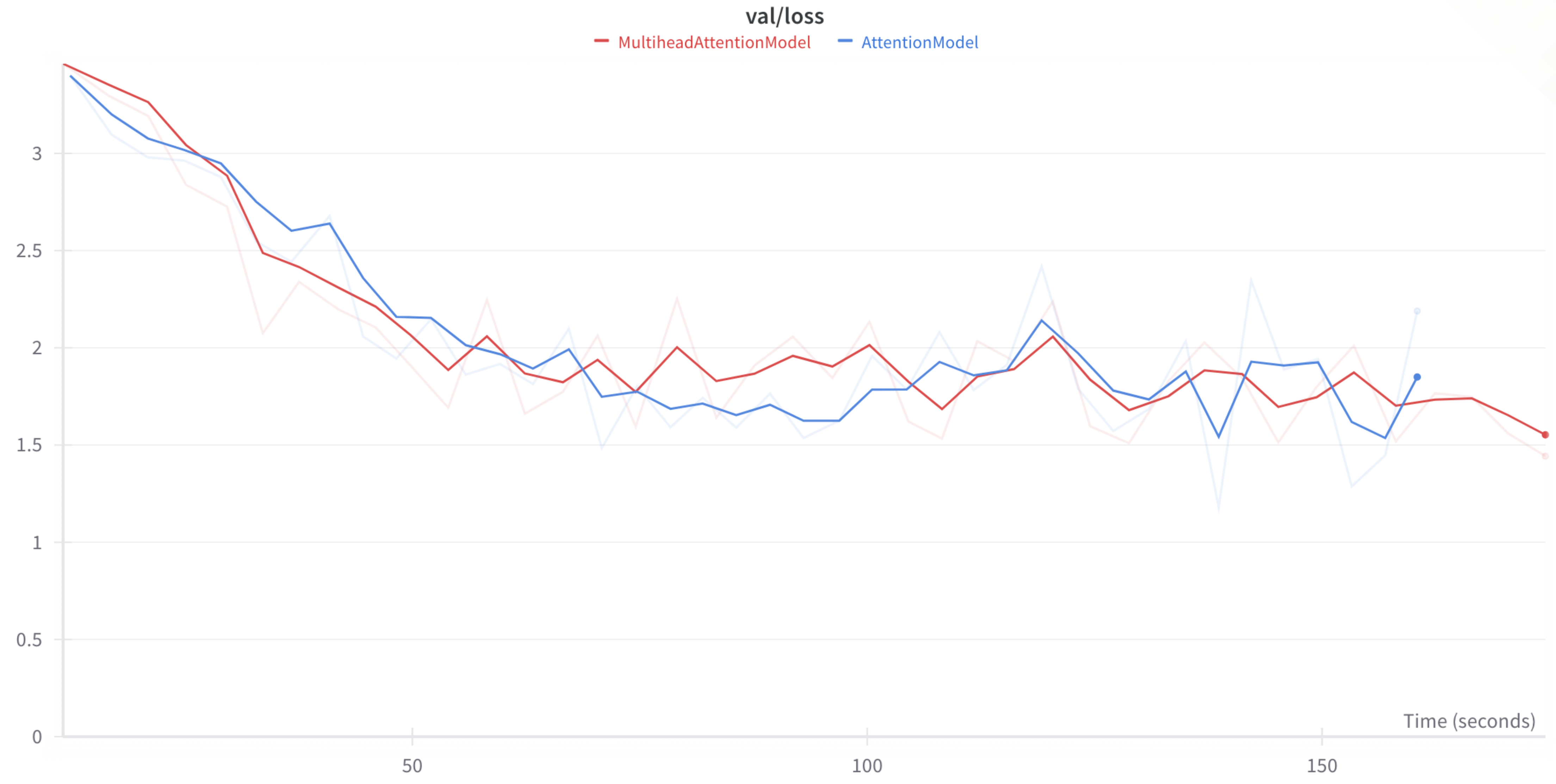
$$L_{\eta_E} = -\log\gamma_{N_0} \quad where \quad \gamma_k = \frac{e^{\eta_k}}{\sum_{j\in E_i\cup E_f} e^{\eta_j}}$$

Training Maze 2D



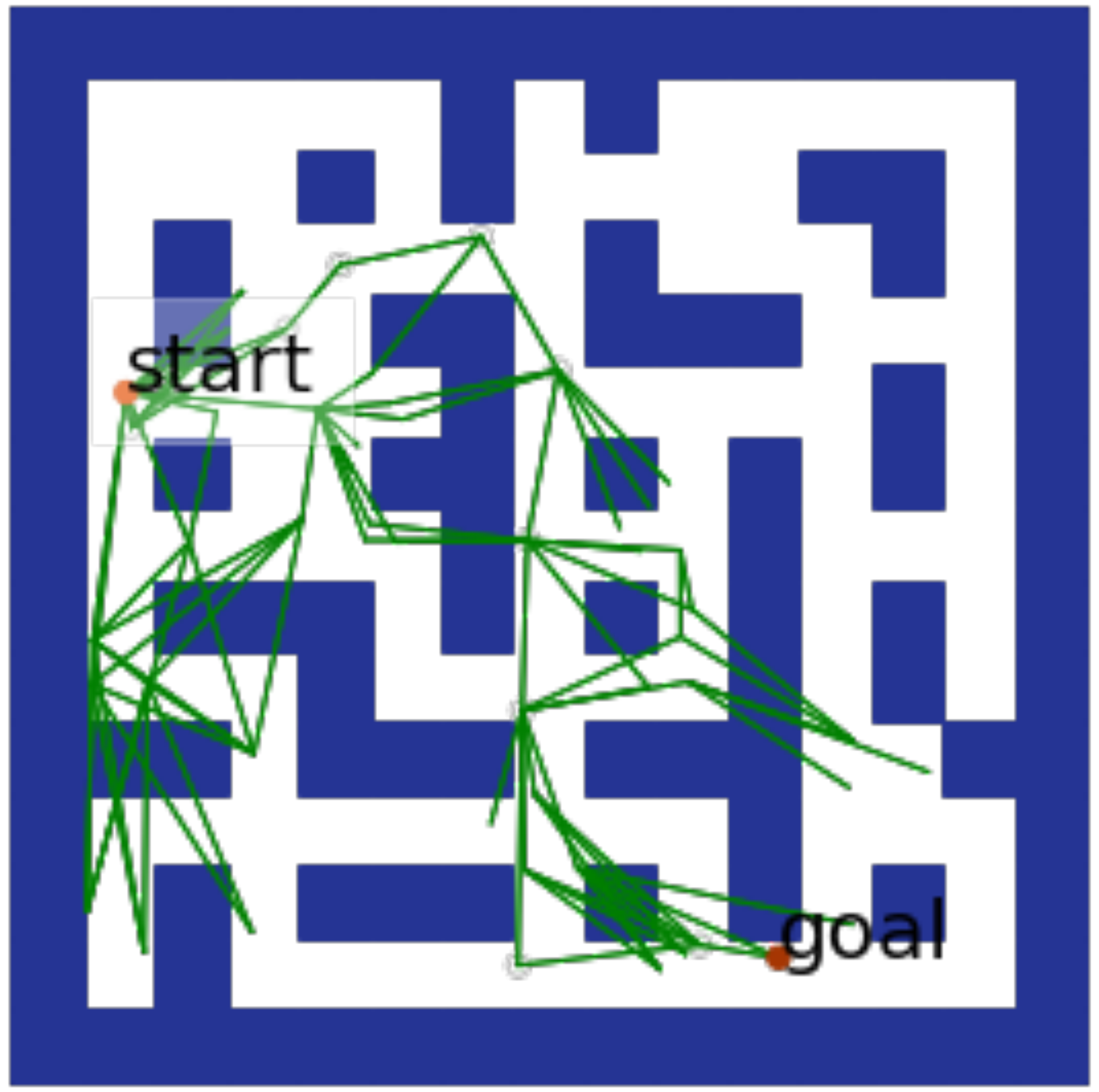
Training KUKA 14DoF

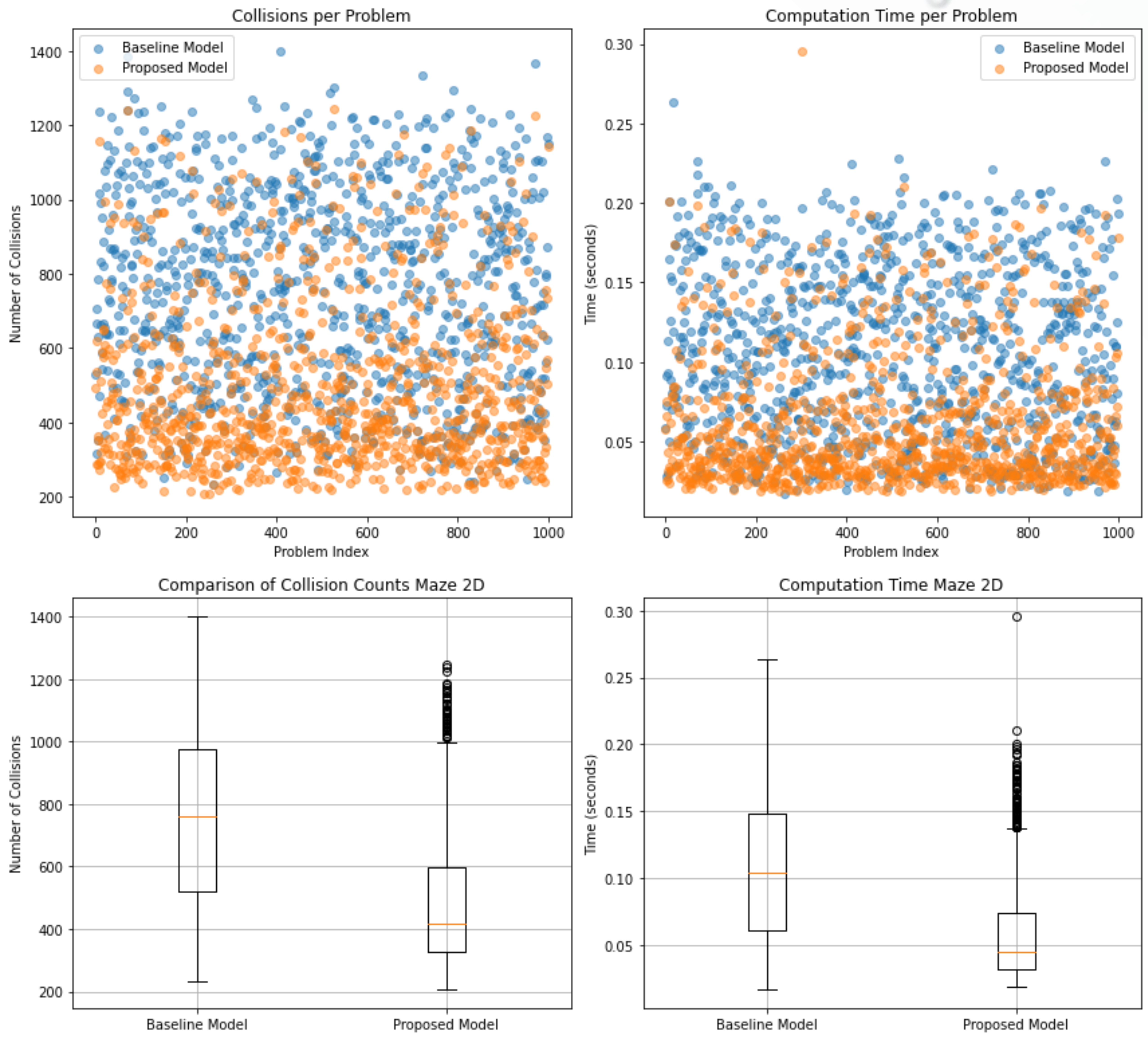




Evaluation

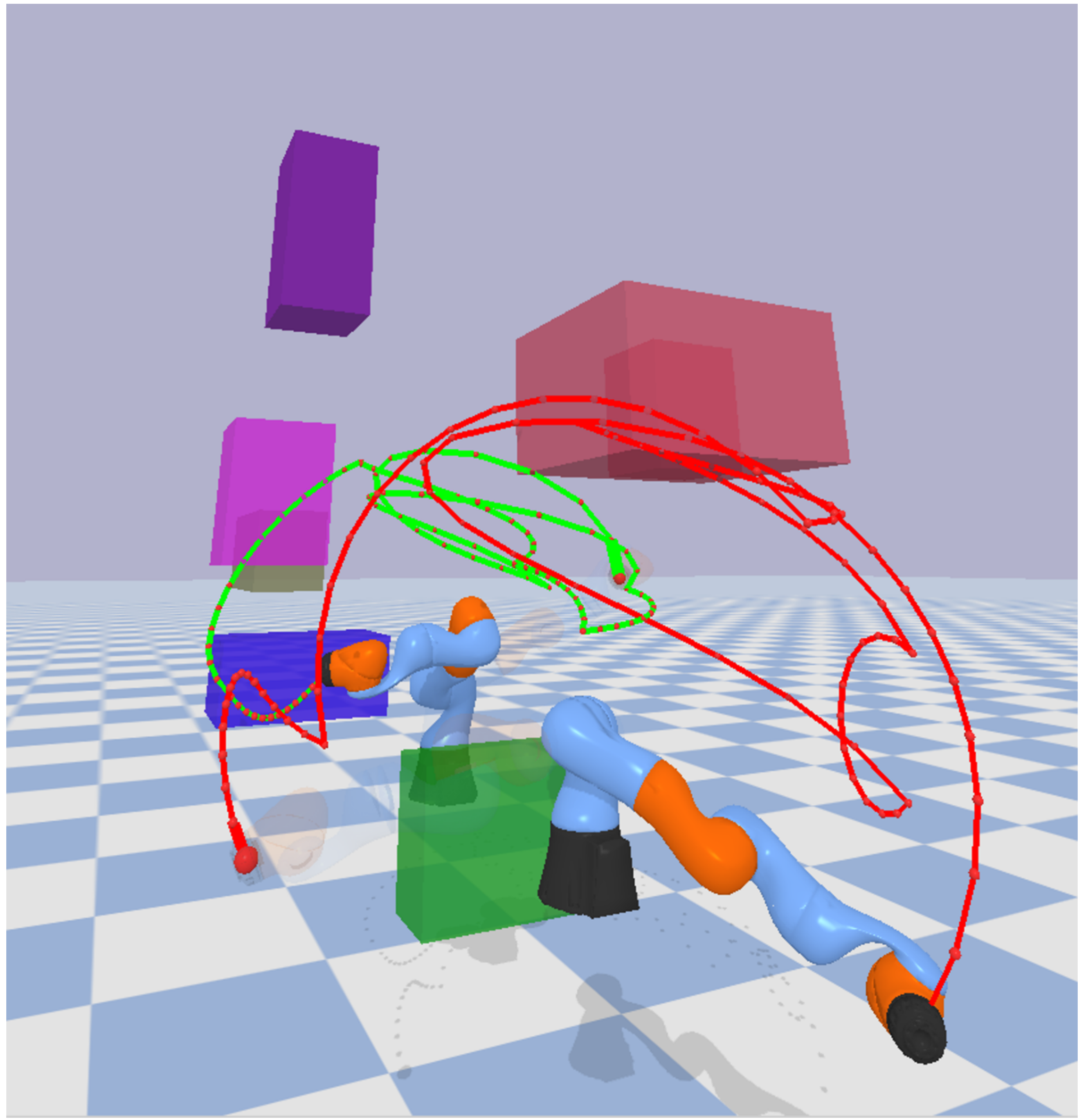
Maze 2D

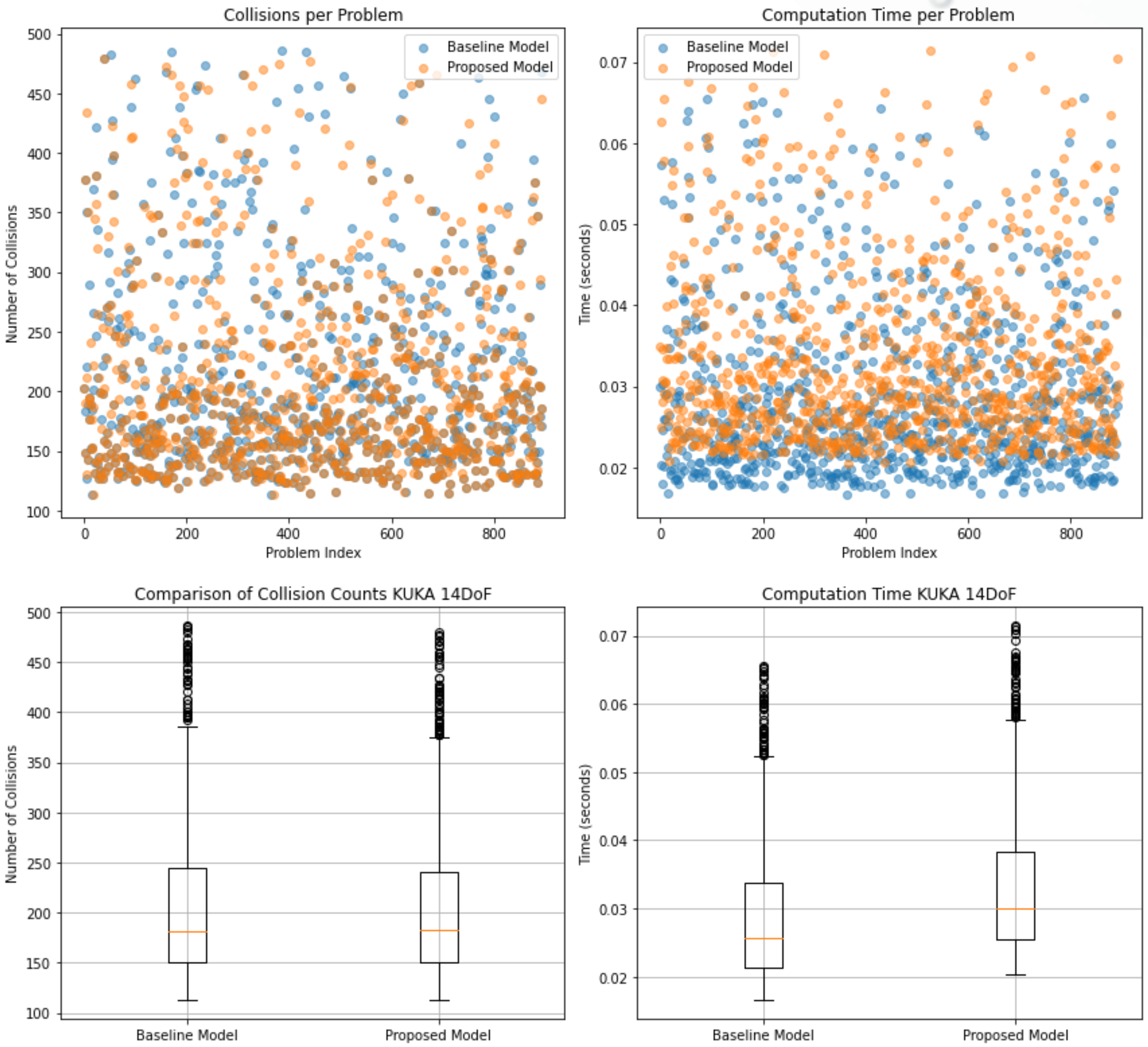




Evaluation

KUKA 14DoF





Thanks

References

- C. Yu and S. Gao, "Reducing collision checking for sampling-based motion planning using graph neural networks," in Advances in Neural Information Processing Systems, vol. 34, pp. 4274–4289, 2021.