Neural Cost-to-Go Function Representation for High Dimensional Motion Planning

Jinwook Huh, Daniel D. Lee, and Volkan Isler

***Abstract*— This paper presents c2g-HOF networks which learn to generate implicit cost-to-go functions for a wide range of robotic systems such as manipulators and non-holonomic systems. The c2g-HOF architecture consists of a cost-to-go function over the configuration space represented as a neural network (c2g-network) as well as a Higher Order Function (HOF) network which outputs the weights of the c2g-network for a given input workspace. Both networks are trained end-to-end in a supervised fashion using costs computed from traditional motion planners. Once trained, c2g-HOF can generate a smooth and continuous cost-to-go function directly from workspace sensor inputs (represented as a point cloud in 3D or an image in 2D). At inference time, the weights of the c2g-network are computed very efficiently and near-optimal trajectories are generated by simply following the gradient of the cost-to-go function. The experimental results indicate that planning with c2g-HOF is significantly faster than other motion planning algorithms, resulting in orders of magnitude improvement when including collision checking. We demonstrate cost-to-go based planning on a 7 DoF manipulator arm where motion planning in a complex workspace requires only 0.13 seconds for the entire trajectory.**

I. Introduction

Motion planning is an essential capability for industrial and home robot automation. It is a challenging problem for the case of complex robot systems (such as manipulators with many degrees of freedom or car-like systems subject to motion constraints) operating in complex environments [1]–[7].

Recently, several deep neural networks have been investigated for motion planning [8]–[13]. These approaches output a trajectory directly or generate critical points in C-space for efficient sampling-based planning. These neural network approaches commonly require hybrid approaches which combine neural networks with traditional planning algorithms to find a trajectory whereas c2g-HOF focuses on parameterizing a continuous cost-to-go function with a neural network and provides a compact and implicit representation of cost-to-go over the entire configuration space.

Implicit neural representations for 3D object reconstruction from 2D images are very successful [14]–[17]. We extend the implicit representation to the cost-to-go representation in the robot configuration space [18], [19]. The suggested neural representation of cost-to-go uses traditional motion planning methods as supervision. In this paper, *Higher Order Function (HOF)* network learns to generate cost-to-go functions directly from the robot's workspace provided as input. Specifically, consider the *cost-to-go* function

All authors are with the Samsung AI Center NY, 837 Washington St, New York, NY 10014

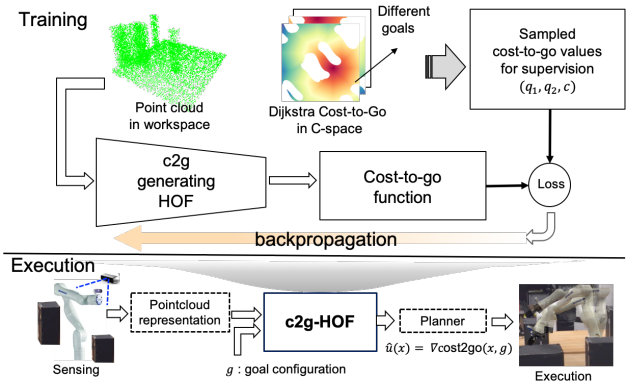
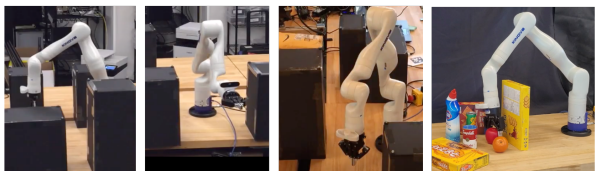


Fig. 1: System Overview. **Training architecture:** C2g generating HOF network encodes an input workspace and outputs the cost-to-go function represented as a radial basis function network. **Execution:** During run time, following the gradient of the cost-to-go function yields continuous collision-free trajectories.

$f_{\mathcal{W}}(s,t)$ which outputs the cost of going from a source configuration s to a terminal configuration t in a given workspace $\mathcal{W}$. Let $g(x) = f_{\mathcal{W}}(x,t)$ for a fixed t. The robot can arrive at t by following the gradient of g with respect to its inputs. We show that the HOF network can directly generate $f_{\mathcal{W}}(s,t)$ from the workspace $\mathcal{W}$ for a specific manipulator it is trained on.

Fig. 1 presents an overview of our method, which we call c2g-HOF: The cost-to-go generating HOF encodes an input workspace $\mathcal{W}$, and generates the weights of a radial basis function network (RBFN) which represents the cost-to-go function over the configuration space (C-space). During training, the values generated by the cost-to-go function are compared to values generated by a traditional planner such as Dijkstra's algorithm over grid samples or a probabilistic roadmap (PRM) over the configuration space. This way, the network learns to generate the cost-to-go function for any input workspace. During runtime, the weights are used to generate a compact neural network whose gradient yields the desired motion plan.

Our method c2g-HOF has a number of appealing properties. Its main advantage is that, once trained, it can instantly generate a continuous cost-to-go map over the entire C-map much faster than existing approaches. The cost-to-go function represented by a neural network (c2g-network) with weights from c2g generating HOF is compact and its gradient is readily computed to generate near-optimal trajectories. Finally, we note that c2g-network yields a continuous, smooth function although it is trained from discretely sampled data.



(a) (b)

Fig. 2: (a) Examples of target configurations located in narrow passageways with flipping end effector joints and rotating the base joint over obstacles. (b) Picking motion planning with collision avoidance based on c2g-HOF.

In summary, the core contributions of our work, which will be detailed in the full version, are: 1) c2g-HOF architecture for generating continuous cost-to-go functions directly from the workspace. 2) efficient trajectory planning based on c2g-network in C-space for any target configuration without slow iterative propagation of cost values and extensive collision checks. 3) demonstration of the method's applicability to real systems.

II. C2G-HOF WITH NEURAL REPRESENTATION

Let $f_{\mathcal{W}} : \mathcal{C} \times \mathcal{C} \rightarrow [0, \infty)$ be the cost-to-go function which, for any given configurations s, t in the configuration space $\mathcal{C}$ returns the cost to traverse a collision-free path from s to t. The objective is to train a neural network which outputs $f_{\mathcal{W}}$ from input workspace $\mathcal{W}$ and the workspace $\mathcal{W}$ is 3D obstacles represented as a point cloud.

During the training phase, we assume that a planner which can compute the cost-to-go between two configurations is available. The specific metric used for computing the cost (e.g. length or energy) is determined by the planner. c2g-HOF is trained in a supervised manner to closely mimic cost-to-go values given by the planner. Specifically, we use a uniform grid when possible or a Probabilistic Roadmap Method (PRM) to build a graph and use Diksjtra's method to compute shortest paths of high DoF manipulators. In addition, we use Reed-Shepp curves or optimal Rapidly exploring Random Trees (RRT*) for non-holonomic systems. The network is trained to output a cost-to-go function of two configurations in C-space.

The network architecture used in this paper, c2g-HOF, consists of two components: (1) the cost-to-go function represented as a neural network which can map any two configurations to a non-negative real number indicating the cost-to-go distance of these configurations (2) HOF network which outputs the weights of the cost-to-go function from the workspace. The HOF generating network requires an encoder for the workspace $\mathcal{W}$ provided as input. We use PointNet [20] for 3D point clouds. The c2g-network takes two configurations as input and outputs the optimal distance between two configurations. We choose radial basis function networks (RBFN) with fully connected layers since radial basis functions are a good architecture to represent continuous cost-to-go.

III. MOTION PLANNING BASED ON C2G-HOF

The robot trajectory to achieve the desired goal configuration is generated by following the gradient of the cost-to-go function starting from the initial configuration. In order to compute the gradient, we use the cost-to-go function as follows: we fix one configuration as the goal configuration, and perturb the other one in the direction of the gradient with respect to the current input configurations. The trajectory generated by following the cost-to-go gradient is then validated using a minimal number of collision checks.

We validate c2g-HOF by generating the trajectory and see that the robot performs successfully even in the presence of narrow passages such as inserting the link between obstacles while avoiding collisions (Fig. 2a). For comparison, we test the performance of RRT, RRT-smooth, PRM and PRM-Smooth and Table. I shows comparison results with a physical 7 DoF manipulator. We average planning times and trajectory lengths of 30 trajectories (10 trajectories $\times$ three workspaces). We also verify that c2g-HOF network can generate near-optimal trajectories for non-holonomic mobile robot systems while avoiding obstacles and compare with traditional planning approaches (Fig. 3). Overall, these results establish c2g-HOF as an efficient and practical method which can learn from existing motion planners and generalize to generate continuous near-optimal cost-to-go functions in novel environments.

	Preproc	**Traj generation**	**Postproc**	**Total time**
c2g-HOF	0.0023	0.129	-	**0.131**
RRT	-	13.577	1.201	14.778
PRM	3.647	0.025	0.354	4.026

TABLE I: Computation time breakdown. Preprocessing time (seconds) for c2g-HOF is the time to generate the weights of the cost-to-go function by running the HOF network. c2g-HOF is up to two orders of magnitude faster on the average.

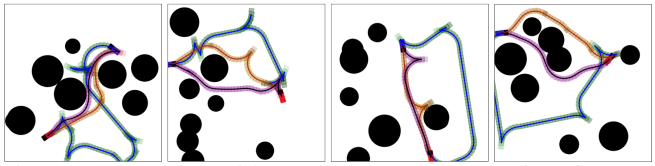


Fig. 3: Trajectories with non-holonomic constraints for various start and goal configurations in various cluttered environments by RRT (blue), RRT* (orange), and c2g-HOF (magenta).

IV. CONCLUSION

This paper presented c2g-HOF for generating implicit cost-to-go functions for high degree of freedom manipulators and non-holonomic mobile robots. Learning the cost-to-go function in high-dimensional configuration spaces requires a large number of samples which, in turn, makes learning difficult. We overcome this challenge by introducing a novel Higher Order Function (HOF) architecture which learns to generate a continuous cost-to-go function. Our experiments demonstrated that c2g-HOF exhibits significant performance improvement over conventional approaches for several kinds of manipulators, including a physical 7 DoF manipulator. For real applications, we build a complete motion planning system based on c2g-HOF (Fig 2b).

REFERENCES

[1] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[2] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[3] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[4] J.-C. Latombe, "Robot motion planning," 1991.

[5] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[6] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1478–1483.

[7] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[8] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 7087–7094.

[9] A. H. Qureshi, M. J. Bency, and M. C. Yip, "Motion planning networks," *arXiv preprint arXiv:1806.05767*, 2018.

[10] D. Molina, K. Kumar, and S. Srivastava, "Learn and link: Learning critical regions for efficient planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 605–10 611.

[11] C. Zhang, J. Huh, and D. D. Lee, "Learning implicit sampling distributions for motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2018, pp. 3654–3661.

[12] R. Kumar, A. Mandalika, S. Choudhury, and S. S. Srinivasa, "Lego: Leveraging experience in roadmap generation for sampling-based planning," *arXiv preprint arXiv:1907.09574*, 2019.

[13] B. Ichter, E. Schmerling, T.-W. E. Lee, and A. Faust, "Learned critical probabilistic roadmaps for robotic motion planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9535–9541.

[14] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[15] E. Mitchell, S. Engin, V. Isler, and D. D. Lee, "Higher-order function networks for learning composable 3d object representations," in *Int. Conf. Learn. Rep.*, 2019.

[16] A. P. S. Kohli, V. Sitzmann, and G. Wetzstein, "Semantic implicit neural scene representations with semi-supervised training," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 423–433.

[17] V. Sitzmann, S. Rezchikov, B. Freeman, J. Tenenbaum, and F. Durand, "Light field networks: Neural scene representations with single-evaluation rendering," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[18] J. Huh, V. Isler, and D. D. Lee, "Cost-to-go function generating networks for high dimensional motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8480–8486.

[19] J. Huh, D. D. Lee, and V. Isler, "Learning continuous cost-to-go functions for non-holonomic systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5772–5779.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Conf. Comp. Vision Pattern Recognition*, 2017, pp. 652–660.