Multi-Robot Deep Reinforcement Learning via Hierarchically Integrated Models

Katie Kang*
University of California, Berkeley
katiekang@eecs.berkeley.edu

Gregory Kahn*
University of California, Berkeley
gkahn@berkeley.edu

Sergey Levine
University of California, Berkeley
svlevine@eecs.berkeley.edu

Abstract

Deep reinforcement learning algorithms require large and diverse datasets in order to learn successful perception-based control policies. However, gathering such datasets with a single robot can be prohibitively expensive. In contrast, collecting data with multiple platforms with possibly different dynamics is a more scalable approach to large-scale data collection. But how can deep reinforcement learning algorithms leverage these dynamically heterogeneous datasets? In this work, we propose a deep reinforcement learning algorithm with hierarchically integrated models (HInt). At training time, HInt learns separate perception and dynamics models, and at test time, HInt integrates the two models in a hierarchical manner and plans actions with the integrated model. This method of planning with hierarchically integrated models allows the algorithm to train on datasets gathered by a variety of different platforms, while respecting the physical capabilities of the deployment robot at test time. Our simulated and real world navigation experiments show that HInt outperforms conventional hierarchical policies and single-source approaches.

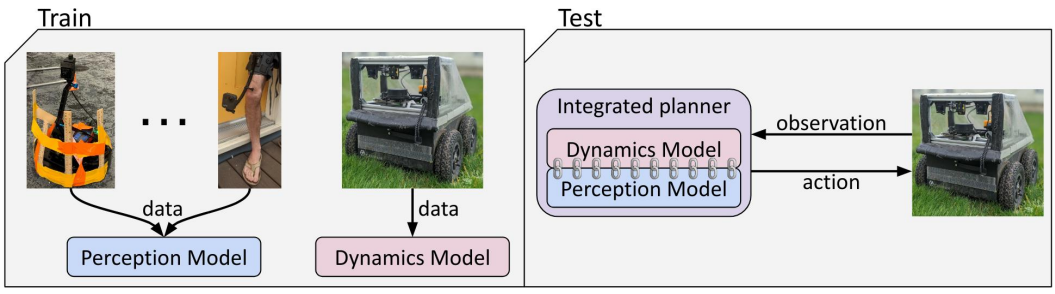


Figure 1: Overview of our hierarchically integrated models (HInt) algorithm. At training time, HInt separately trains a perception model and a dynamics model, then at test time, HInt combines the perception and dynamics model into a single model for integrated planning and execution. Our modular training procedure enables HInt to train the perception model using data gathered by multiple platforms, such as ground robots and even people recording video with a hand-held camera, while our integrated model at test time ensures the perception model only considers trajectories which are dynamically feasible.

NeurIPS 2020 3rd Robot Learning Workshop: Grounding Machine Learning Development in the Real World.

1 Introduction

Machine learning has emerged as a powerful tool for enabling robots to acquire control policies by learning directly from experience. One of the key guiding principles behind these advances in robot learning is to leverage large datasets. However, collecting these large and diverse datasets using a single robot can be prohibitively challenging; for example, in order for a robot to learn to navigate any sidewalk, the robot must likely gather data from many different cities, and it is typically impractical to physically transport a single robot to many different locations. What if we could instead train vision-based robot control policies on large and diverse datasets collected by a variety of different robots? An ideal method could use data from *any* platform that provides useful knowledge about the problem—such an approach would be far more scalable. Unfortunately, data from such heterogeneous platforms presents a challenge: different platforms have different physical capabilities, and therefore produce different data. In order to leverage such heterogeneous data, we must properly account for the underlying dynamics of the data collection platform.

The primary contribution of this work is HInt— hierarchically integrated models for acquiring image-based control policies from heterogeneous datasets. HInt makes it possible for a robot to operate in environments it has never been in before, by leveraging experiences gathered by other robots, with potentially differing dynamics. We demonstrate that HInt successfully learns image-based control policies from heterogeneous datasets in both simulated and real-world robotic navigation tasks, and outperforms methods that use single-source data or use conventional hierarchical policies.

2 Related work

Prior work has demonstrated end-to-end learning for vision-based control for a wide variety of applications [17, 2, 13, 9, 19, 11]. However, these approaches typically require a large amount of diverse data [7], which hinders the adoption of these algorithms for real-world robot learning. One approach for overcoming these data constraints is to use teams of robots to gather data; however, these approaches typically assume that the robots are the same [14], have similar underlying dynamics [3], or the data is from expert demonstrations [4, 22, 21]. Prior works have also investigated learning modular vision-based control policies [15, 10, 5, 18, 1, 16], many of which can leverage heterogeneous datasets [15]. However, these approaches can fail because the modules cannot communicate their capabilities and limitations to each other.

3 HInt: Hierarchically Integrated Models

Our goal is to learn image-based control policies that can leverage data from heterogeneous platforms. The key contribution of our approach, shown in Fig. 2, is to learn separate hierarchical models at training time, and combine these models into a single integrated model for planning at test time. The separate hierarchical model training allows our method to leverage datasets gathered by heterogeneous platforms, while the integrated planning enables our approach to directly map raw sensory inputs to robot actions that are dynamically feasible for the deployment robot.

At the core of our hierarchically integrated models (HInt) reinforcement learning algorithm are two predictive models: a high-level, shared perception model and a low-level, robot-specific dynamics model. The perception model is an image-based predictive model that takes as input the current image and a sequence of future kinematic poses, and predicts future rewards, while the dynamics model takes as input the current robot state and a sequence of future low-level actions, and predicts future kinematic poses.

The perception model is trained using a large, heterogeneous dataset consisting of data gathered by a variety of robots, all with possibly different underlying dynamics. Meanwhile, the robot-specific dynamics model is trained only using data from the deployment robot. At test time, because the output predictions of the dynamics model are the input actions for the perception model, the dynamics and perception models can be combined into a single integrated model. This integrated model predicts future rewards, and is used to plan into the future to determine the actions that maximize future reward. This integrated model is essential because it enables the planner to holistically reason about the entire system.

4 Experiments

In our experiments, we aim to answer the following questions:

Q1: Does the ability of HInt to train from datasets gathered by heterogeneous platforms result in better performance compared to approaches trained from a single data source?

Q2: Does HInt's integrated model planning approach result in better performance compared to conventional hierarchy approaches?

In order to separately examine these questions, we investigate **Q1** by training the perception module with multiple different real-world data sources, including data from different environments and different platforms, and evaluating on a single real-world robot. To examine **Q2**, we deploy a shared perception module to systems with different low-level dynamics, in both simulation and the real world.

4.1 Comparison to Single-Source Models

We first examine **Q1**. For this experiment, perception data was collected in three different environments using three different platforms (Fig. 3), and the deployment robot is the Clearpath Jackal. We compared HInt with the single data source approach from [8], in which only data gathered by a single platform is used for training and the integrated model is trained end-to-end.

Fig. 4 shows the results comparing HInt to the single data source approach. In all environments[1], our approach is more successful in reaching the goal. Note that even when the single data source method is trained and deployed in the same environment, HInt still performs better because learning-based methods benefit from large and diverse datasets. Furthermore, the row labeled "Industrial" illustrates well how HInt can benefit from data collected with other platforms: although the Jackal robot had never seen the industrial setting during training, the training set did include data collected by a person with a video camera in this setting. The increase in performance from including this data ("Kobuki + Jackal + Human") shows that the Jackal robot was able to effectively integrate this data into its visual navigation strategy.

4.2 Comparison to Conventional Hierarchy

We next examine **Q2**. We compared our integrated approach with the conventional hierarchy approach, in which the perception model is used to output desired waypoints that are then passed to a low-level controller [5, 15, 18, 10, 1].

Tab. 1 shows the results comparing HInt versus conventional hierarchy in a simulated experiment, and Fig. 5 shows the comparison for a real world experiment. In both experiments, we showed that the conventional hierarchy approach can catastrophically fail if the higher level perception-based planner sets waypoints for the lower level dynamics-based planner without regard for the physical capabilities of the robot. In contrast, HInt's integrated planning approach enables the dynamics model to inform the perception model about which maneuvers are feasible.

[1] We could not run experiments in the office environment due to COVID-related closures.

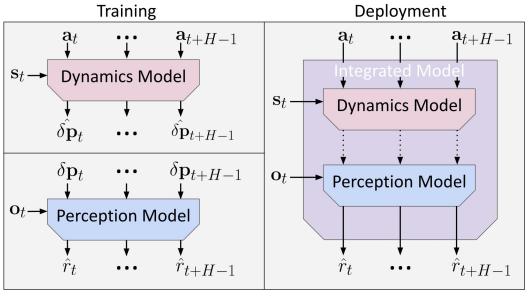


Figure 2: During training, we learn two separate neural network models. The dynamics model takes as input the current robot state and a sequence of future actions, and predicts future changes in poses. The perception model takes as input the current image observation and a sequence of future changes in poses, and predicts future rewards. When deploying, the dynamics and perception models are combined into a single integrated model that is used for planning and executing actions that maximize reward.

Figure 3: Training data was gathered by an indoor Yujin Kobuki robot in an office (3.7 hours), an outdoor Clearpath Jackal robot in an urban environment (3.5 hours), and a person with a video camera in an industrial area (1.2 hours).

		Perception Data Sources				
		Single-Source [8]			HInt (ours)	
		Kobuki (Office)	Jackal (Urban)	Human (Industrial)	Kobuki (Office) + Jackal (Urban)	Kobuki (Office) + Jackal (Urban) + Human (Industrial)
Test Env	Urban	0%	13%	N/A	**100%**	**100%**
	Industrial	0%	N/A	0%	33%	**87%**

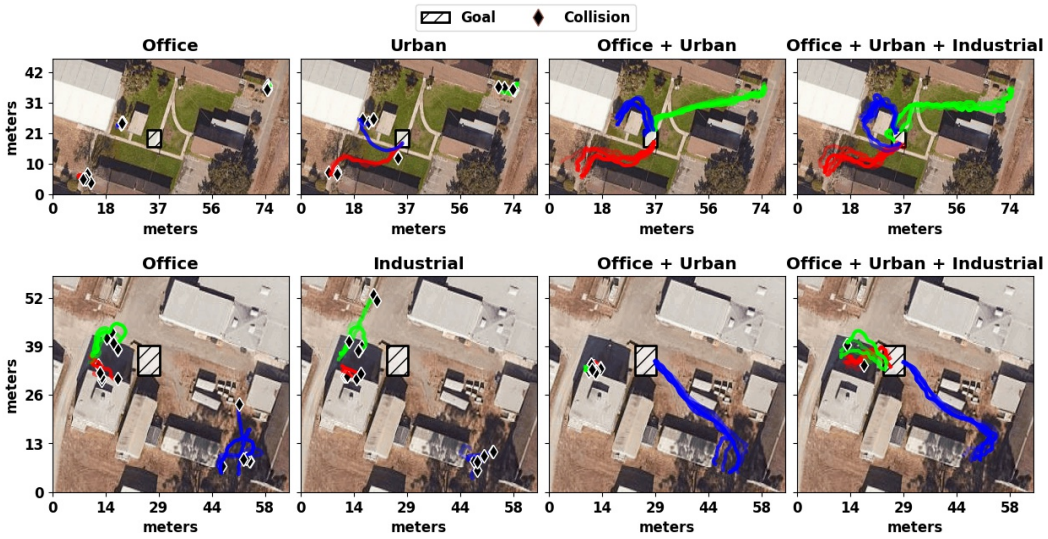
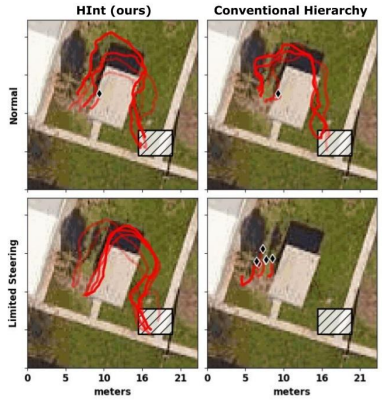


Figure 4: Comparison of single data source models [8] versus our multiple data source approach in an urban and industrial environment. Each approach was evaluated from the 3 same start locations in each environment (corresponding to the red, green, and blue lines), and was ran 5 times from each start location. The quantitative results show what percentage of the 15 trials successfully reached the goal.

	Normal	Limited Steering	Right turn only	0.25 second lag
Conventional Hierarchy	96%	68%	0%	0%
HInt (ours)	**96%**	**84%**	**56%**	**40%**

Table 1: Comparison of conventional hierarchy (e.g., [5, 10, 1]) versus HInt (ours) approaches at deployment time in a simulated environment. Four different robot dynamics models were evaluated—normal, limited steering, right turn only, and 0.25 second lag. Both approaches were evaluated from the same 5 starting positions, with 5 trials for each starting position.



	Normal	Limited Steering
Conventional Hierarchy	80%	0%
HInt (ours)	**80%**	**100%**

Figure 5: Comparison of conventional hierarchy vs HInt (ours) approaches in a real world experiment, on two robots with different dynamics: a Jackal robot with full dynamical range (normal), and a Jackal robot with its steering limit to 40% of its full dynamical range (limited steering). Both approaches were evaluated with 5 trials each from the same starting position. The table displays percentage of trajectories that successfully reached the goal.

References

[1] Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. In *CoRL*, 2019.

[2] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[3] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-scale multi-robot learning. In *CoRL*, 2019.

[4] Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *ICML*, 2019.

[5] Wei Gao, David Hsu, Wee Sun Lee, Shengmei Shen, and Karthikk Subramanian. Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. In *CoRL*, 2017.

[6] M. Goslin and M. Mine. The panda3d graphics engine. In *IEEE Computer*, 2004.

[7] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*, 2018.

[8] Gregory Kahn, Pieter Abbeel, and Sergey Levine. BADGR: An autonomous self-supervised learning-based navigation system. *arXiv:2002.05700*, 2020.

[9] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, 2018.

[10] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. In *ICRA*, 2019.

[11] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *ICRA*, 2019.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[13] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 2016.

[14] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *IJRR*, 2018.

[15] Antonio Loquercio, Ana I Maqueda, Carlos R Del-Blanco, and Davide Scaramuzza. Dronet: Learning to fly by driving. In *RA-L*, 2018.

[16] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Scaling Local Control to Large-Scale Topological Navigation. *arXiv preprint arXiv:1909.12329*, 2019.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[18] Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. In *CoRL*, 2018.

[19] Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert. Learning monocular reactive uav control in cluttered natural environments. In *ICRA*, 2013.

[20] Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1999.

[21] Wen Sun, Anirudh Vemula, Byron Boots, and J Andrew Bagnell. Provably efficient imitation learning from observation alone. In *ICML*, 2019.

[22] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. In *ICML Workshop on Imitation and Intent*, 2018.

[23] Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv:1509.01149*, 2015.