PROROK LAB, MPhil Project Proposals 2021-22

Projects:

Detailed project descriptions are listed below.
All projects will be jointly supervised by Prof. Prorok and a senior PhD student / Postdoc.

The Rules of Lying: Disentanglement in Multi-Agent Communication

Motivation:
Capturing information that enables complex inter-agent coordination requires new kinds of Neural Network architectures. Graph Neural Networks (GNNs) exploit the fact that inter-agent relationships can be represented as graphs, which provide a mathematical description of the network topology. In prior work, we developed a new model for learning to communicate to coordinate multi-agent systems in the presence of agents with potentially conflicting objectives [1]. This model also includes a post-hoc interpretability technique that enables the visualization of communicated messages. Interestingly, results show that it is possible to learn highly effective communication strategies capable of *manipulating other agents* to behave in such a way that it benefits the self-interested agents.

While this work showed the feasibility of generating interpretations for GNN-based agent interactions, the broader problem of ensuring that such interactions lead to desirable behaviors remains largely unsolved. Manipulative communication can be harmful. Hence, we are interested in further interpreting the emanating

communication patterns, such that appropriate *protective* measures can be taken. While a number of techniques within the broad domain of explainable AI exist (e.g., [5]), there is a dearth of work on the explainability of inter-agent communication, specifically.
c
Objectives:
The purpose of this project is to formalize the interpretation problem and propose a technique tailored for the interpretation of black-box multi-agent communication. Ideally, the technique should be able to identify which generative factors lead to falsified latent units, i.e., *lies*.

Approach:
In this project, we propose the use of 'disentanglement' techniques. These provide factorized representations of latents into individual (latent) units that are sensitive to changes in single generative factors [2, 3].

References:
[1] Jan Blumenkamp and Amanda Prorok. "The emergence of adversarial communication in multi-agent reinforcement learning." Conference on Robot Learning (CoRL, 2020).
[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.
[3] K. Ridgeway. A survey of inductive biases for factorial Representation-Learning. arXiv, 2016.
[4] Burgess et al., Understanding disentangling in β-VAE, 2018, https://arxiv.org/pdf/1804.03599.pdf%20
[5] Q.Huang, M.Yamada, Y.Tian, D.Singh, D.Yin, and Y.Chang. Graphlime: Local interpretable model explanations for graph neural networks. arXiv preprint arXiv:2001.06216, 2020.

Contact: asp45, jb2270

Interpreting Multi-Agent Interaction through Symbolic Regression

Motivation:
Although an abundance of work exists in the general domain of neural network interpretability (for a comprehensive review see [1] and the references therein), these approaches fall short in their ability to incorporate *relational* (e.g., inter-agent) information. A handful of very recent works address GNN interpretability in the case of static, structural prediction problems [2, 3]. The aforementioned methods, however, focus on the problem of identifying compact subgraphs that are likely to explain local node features. To interpret GNN-based control policies, however, we further require explanations of how node interactions influence local decision-making policies. There is a key deficiency in current work: *we are not yet able to interpret nor ensure learning-based multi-agent behavior*. Furthermore, by using a black-box model we cannot guarantee that our learned function will extrapolate to all regions of the state space.

Symbolic regression is the process of finding a symbolic expression that matches data from a black-box function. Although this problem is hard, recent work has shown that it is possible to use symbolic regression to determine the underlying physics equations that govern real-life phenomena [4].

Objectives:

The objective of this project is to apply symbolic regression to observed data originating from classical multi-agent dynamic processes, such as consensus, formation control, and flocking. The aim is to discover how closely symbolic regression is able to recover the underlying interaction principles.

Approach:
Various symbolic regression approaches exist, e.g., [4] combines neural network fitting with a suite of physics-inspired techniques, and [5] uses a symbolic metamodeling framework. In the first instance, the goal is to identify adequate techniques, and second, incorporate the necessary modifications to make the techniques amenable to the problem at hand.

References:
[1] D.C.Nguyen, P.Cheng, M.Ding, D.Lopez-Perez, P.N.Pathirana, J.Li, A.Seneviratne, Y.Li, and H.V.Poor. Enabling AI in future wireless networks: A data life cycle perspective. IEEE Communications Surveys & Tutorials, 2020.
[2] Q.Huang, M.Yamada, Y.Tian, D.Singh, D.Yin, and Y.Chang. GraphLIME: Local interpretable model explanations for graph neural networks. arXiv preprint arXiv:2001.06216, 2020.
[3] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. GNNExplainer: Generating Explanations for Graph Neural Networks. In NeurIPS, pages 9244–9255. 2019.
[4] Udrescu et al. AI Feynman: A physics-inspired method for symbolic regression; https://www.science.org/doi/10.1126/sciadv.aay2631
[5] Alaa and van der Schaal., Demystifying Black-box Models with Symbolic Metamodels, NIPS, 2019

Contact: asp45, rk627, jb2270

Graph Neural Network (GNN) Memory for Robotic Sequence Learning

Introduction:
Sequence learning is an important part of machine learning [1], used heavily in fields such as natural language processing [2] or reinforcement learning [3]. Traditional approaches to sequence learning assume a fixed temporal conditional structure, but in many cases (e.g. navigation, music) there are more succinct and efficient structures, perhaps hierarchical or containing loops respectively [4]. We can represent these structures as graphs. Graph neural networks (GNNs) excel at extracting meaning from large, structured data in an efficient manner [5]. Using GNNs, we can extract meaningful information from these structures.

Objectives:
The student is to pick a robotics-based application of sequence learning of interest to them. For example, they could choose navigation, manipulation, pose estimation given noisy samples, multiagent coordination, etc. Sequences need not be temporally ordered.

Once a problem is identified, they need to find an underlying structure to their problem that they can represent using a graph. They must design methods to structure the graph in a way specific to their chosen application. Once the graph is constructed, they must develop GNN models to extract meaning from the graph and accomplish their task. We intend to run the models onboard real robots, so efficiency is important.

Background:

Knowledge of Pytorch, Pytorch Geometric, and the Robot Operating System are ideal. GNN or graph theory is useful, as well as understanding recurrent neural networks and transformers. A thorough understanding of multilayer perceptrons and basic machine learning concepts in a must. You should have some knowledge (not necessarily ML-based) of your area of interest.

References:

[1] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
[2] Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).
[3] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. PMLR, 2016.
[4] Morad, Steven D., Stephan Liwicki, and Amanda Prorok. "Graph Convolutional Memory for Deep Reinforcement Learning." arXiv preprint arXiv:2106.14117 (2021).
[5] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).

Contact: sm2558, asp45

Form Meets Function: Evolving Multi-Agent Environments

Motivation:

Conventional representations of a 2D/3D world environment for mobile robots stem from point-cloud and voxel-grid representations. These are fairly straight-forward to implement, and are convenient for implementing various transforms, rotations and scalar operations. However, these methods assume that the objects and entities in the world do not “evolve”, and, if they do, such approaches simply capture snapshots at different time points. Further, logical and semantic relationships between entities in the world are not captured implicitly in such representations.

Robot systems, in contrast, are designed to function efficiently as *guests* within these environments, assuming that the environment is a given. That is, the question often is: what is the best type of robot given a specific world (factory, industrial workspace, house etc)? We would like to consider a scenario where we have control over the host environment, and thus ask the question: what can we change about this world given a specific robot? The problem might appear hypothetical, and deviates from conventional robotic challenges; nevertheless, the solutions have strong impacts on smart and agent-centric design.

Objectives

We envision a situation where we, as designers, are allowed to morph [1] the *host* environment in order to better accommodate its *guests*. Towards this end, we will forbear conventional “static” representations and develop a framework that models the environment as a dynamic, morphable entity [2]. The project will have two broad approaches.

(A) Parameterised/structured: We will develop novel scalable representations that are freely morphing. For instance, in an overly simplistic case, one might imagine the world entities represented akin to scalable vectors and polynomials (ala SVG), and their relationships captured by a graph data structure. If these are parameterised appropriately, it will be possible to optimise a world and its contents based on a cost metric capturing their favourability to robotic tasks. More complex representations, such as k-d trees and Octrees [3], might also be useful in developing structured environments.

(B) Deep graphs: We will utilise the spatial representation offered by graph neural networks (GNNs) to learn 2D/3D structures and their ability to morph. This new representation must retain implicit semantic/logical connections between its objects ("a door is connected to a wall"), and yet enable dynamics between them ("door can be positioned anywhere on the wall"). Our objectives will be two-fold: (a) develop a graph environment representation that can be morphed at design-time to meet specified criteria, and, (b) develop a dynamic graph representation that can evolve continuously.

Approach:
We will build on graph representations (such as Octrees and k-d trees [3]), and then develop graph optimisation techniques that morph the structure to minimise a cost function. Our approach will follow deep graph structures (GNNs, deep GNNs [4]), and construct a machine learning pipeline that modifies the graph's properties (adjacencies and node distances, for instance). Additionally, we will also investigate whether such an algorithm can fabricate new nodes on the graph (or new graphs entirely) to meet the specified objectives [5]. The "quality" of a proposed environment will be measured by the performance of an agent deployed in it.

References:
[1] Co-Reyes JD, Sanjeev S, Berseth G, Gupta A, Levine S. Ecological Reinforcement Learning. arXiv preprint arXiv:2006.12478. 2020 Jun 22.
[2] Kamalaruban P, Devidze R, Cevher V, Singla A. Environment Shaping in Reinforcement Learning using State Abstraction. arXiv preprint arXiv:2006.13160. 2020 Jun 23.
[3] Octrees. [online] http://www.open3d.org/docs/latest/tutorial/geometry/octree.html
[4] Gallicchio C, Micheli A. Fast and deep graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence 2020 Apr 3 (Vol. 34, No. 04, pp. 3898-3905).
[5] Sudhakaran S, Grbic D, Li S, Katona A, Najarro E, Glanois C, Risi S. Growing 3D Artefacts and Functional Machines with Neural Cellular Automata. arXiv preprint arXiv:2103.08737. 2021 Mar 15.

Contact: asp45, ashankar@cse.unl.edu

Can We Change the World? Environment-Shaping for Multi-Agent Learning

Motivation:
To control a robot system, we must first obtain a mathematical model that represents the given system. To write an algorithm for a particular task, say multi-robot rendezvous [1], we must first establish the dynamics of the task, define our assumptions, and set up suitable parameters (How many agents? What are their kinematic properties? What can they sense? How do they communicate?) . However, there already exist a variety of classical algorithms and controllers for driving mobile robots, and for solving common tasks. Why, then, is robot control still an open problem? One reason is that our system models are imperfect abstractions, and that our algorithms make several implicit assumptions in order to function. For instance, an optimal controller for driving 4-wheeled cars may not account for unmodeled dynamics in the world. Thus, most of our algorithms must contend with approximate system models, and work on idealised environments, using best estimates of system state. Additionally, reinforcement learning methods make implicit assumptions about the environment, rewards and actions that may be faulty [2].

Objective:

If the environment and the target system were perfectly modeled, our algorithms should work perfectly. But, instead of attempting to improve the system model of a given robotic platform, in this project, we are taking a reverse approach: we wish to develop hypothetical ("imaginary") systems that are best suited for a given control problem/algorithm.
We are interested in asking: what are the properties that an environment and the robot(s) must exhibit that make them conducive towards a particular problem? Prior work has looked into such considerations [3,4]. For classical systems, the answers might be fairly obvious. For instance, a maze solving robot will thrive in mazes that are too simple. But we wish to refine our query further: can we develop mazes that are best suited for the dynamics of a particular type of robot?

Approach:
The project will utilise reinforcement learning (RL) to shape and evolve different aspects of a Markov Decision/Reward Process (MDP). In the first phase, we will establish baselines by considering standard "well-known" problems in robotics and computer science. If our approach arrives at the expected obvious solutions, then we will have quantifiable comparisons against the standards as well as the state-of-the-art. For instance, in the case of a maze-solver that utilises the A* search algorithm, a favourable environment will represent the heuristic that is used. Simpler base examples include automatically generating and populating a data structure that works as the best-case scenario for a given sorting/search algorithm. Afterwards, the approach will be expanded to include more complex scenarios and applications that involve potentially nonlinear systems.

References:
[1] Roy N, Dudek G. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. Autonomous Robots. 2001 Sep;11(2):117-36.
[2] Reda D, Tao T, van de Panne M. Learning to locomote: Understanding how environment design matters for deep reinforcement learning. InMotion, Interaction and Games 2020 Oct 16 (pp. 1-10).
[3] Ha D. Reinforcement learning for improving agent design. Artificial life. 2019 Nov 1;25(4):352-65.
[4] Boudet JF, Lintuvuori J, Lacouture C, Barois T, Deblais A, Xie K, Cassagnere S, Tregon B, Brückner DB, Baret JC, Kellay H. From collections of independent, mindless robots to flexible, mobile, and directional superstructures. Science Robotics. 2021 Jul 21;6(56):eabd0272.

Contact: asp45, ashankar@cse.unl.edu

Graph Morphogenesis with a GNN-based Autoencoder

Motivation:
In this project, the goal is to develop a method which can encode and then reconstruct a graph. This problem has several applications in disparate fields:

- An autoencoder for graphs would be useful for creating embeddings for graph-structured data. In robotics, this can be used as a decentralised method of compressing a representation of the connectivity of a swarm, and allow robots to reconstruct the state of individual other robots (even those not in the immediate neighbourhood). In computer vision, this can be used to create an embedding for 3D meshes.
- The act of morphogenesis can be used in self-assembling structures [1]. Imagine a building made out of nanobots is hit with a cannonball. This algorithm could tell the building how to rebuild itself.

- In the field of biology, this method could be used to predict the future behaviour of cells---how they will move and divide over time. With a sufficiently advanced model, this could be used to optimise the conditions for healing wounds [2,3].

Existing work does not quite exhibit the functionality needed to accomplish these tasks. Variational Graph Auto-Encoders exist [4], but they simply create an embedding in each node which can be used for link prediction. They do not compress the nodes themselves into a fixed-length embedding, and they are only formulated from a centralised point of view, so they are not very useful for the aforementioned applications. Conversely, methods that implement Neural Cellular Automata [1] encode the graph state but not the graph connectivity. Neural Cellular Automata are often implemented with CNNs, which implies a specific graph connectivity. In [1], the graph state is encoded in the network parameters of a 3D CNN, which is used to perform node prediction on a 3D grid.

Objectives:
In this project, the objective is to encode both the node states and the graph structure itself. Ideally, this should be done in a decentralised context (that is, only using local operations). Furthermore, the method should have the ability to construct arbitrary graphs given different inputs, meaning that the encoding for a specific graph should be stored in a latent state, as opposed to the network parameters themselves. The finished method should be demonstrated on an application (e.g., as listed in the Motivation section).

Method:
The method should include a mechanism for compressing an existing graph into a fixed-length encoding, as well as a mechanism for reconstructing a graph given an encoding. In order to do this, we recommend using an autoencoder, allowing both the encoder and decoder to be trained end-to-end, without the need for labelled training data. For the architecture of the encoder, we recommend using a Graph Neural Network (GNN). The decoder will likely be much more challenging---it will take some creativity to design an “inverse GNN”.

References:
[1] Sudhakaran, Shyam, et al. "Growing 3D Artefacts and Functional Machines with Neural Cellular Automata." arXiv preprint arXiv:2103.08737 (2021).
[2] Yamamoto, Takaki, et al. Graph-Based Machine Learning Reveals Rules of Spatiotemporal Cell Interactions in Tissues. 23 June 2021, p. 2021.06.23.449559. bioRxiv, https://www.biorxiv.org/content/10.1101/2021.06.23.449559v1.
[3] Shim, Gawoon, et al. “Overriding Native Cell Coordination Enhances External Programming of Collective Cell Migration.” Proceedings of the National Academy of Sciences, vol. 118, no. 29, July 2021. www.pnas.org, https://doi.org/10.1073/pnas.2101352118.
[4] Kipf, Thomas N., and Max Welling. "Variational graph auto-encoders." arXiv preprint arXiv:1611.07308 (2016).

Contact: asp45, rk627, bmd39

A GNN-based Differentiable Simulator for Environment Optimization

Motivation
Reinforcement Learning (RL) is increasingly popular in both academia and industry applications. The conventional RL focuses on how to encourage the agent to predict suitable behaviors through maximizing the reward in a given environment. Recent work [1] from OpenAI discovered the emergent behavior of agents to build multi-object shelters using moveable boxes to hide themselves in hide-and-seek games. Yet,

how the modifications of the environment could impact the agent with a rule-based behavior mechanism remains a question to be answered.

Approach
To explore further on this question, we choose graph representations to describe the complex environment, as they are a powerful data structure to represent data in our daily life. The complexity of graph data has imposed significant challenges on existing machine learning algorithms, which yields the need for Graph Neural Networks (GNNs) [2]. Recently, the researchers have developed GNNs-based learnable models to implement an inductive bias for object- and relation-centric representations of complex systems [3]. In this proposal, we intend to develop a general framework that can optimize graph-represented environments using the GNNs-based differentiable simulator [4]. Then, we are interested in demonstrating his performance across several multi-agent case studies, e.g., Sokoban (push boxes), hide-and-seek and interior layout optimization of buildings.

Project Objectives
- Develop an optimization framework based on GNN-based differentiable simulator to process a graph-represented environment and objects inside it;
- Show that the differentiable simulator works by demonstrating it into different scenarios.
- In layout optimization, we will design a scoring metric based on the quantitative parameters as the reward mechanism of RL to guide the optimization process of the architectural layout;
- Extension: In gaming scenarios, design self-supervising R L[1] to encourage emergent agent behavior and their manipulation in environment;
- Extension: Optimize office layouts for improved multi-agent movement (e.g., social distancing). See Appendix 1.

Contact: asp45, ql295

References:
[1] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch. Emergent tool use from multi-agent autocurricula.arXiv preprint arXiv:1909.07528, 2019.
[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graphneural networks.IEEE transactions on neural networks and learning systems, 32(1):4–24, 2020.
[3] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia. Graph networks as learnable physics engines for inference and control. InInterna-tional Conference on Machine Learning, pages 4470–4479. PMLR, 2018.
[4] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin. Scalable differentiable physics for learning and control. arXiv preprint arXiv:2007.02168, 2020
[5] Sanchez-Gonzalez et al.]Learning to Simulate Complex Physics with Graph Networks http://proceedings.mlr.press/v119/sanchez-gonzalez20a.html

---

Heterogeneous Risk Attitudes in Multi-Agent Teams

Motivation:
Decision making under uncertainty has received extensive attention in game theory and robotics. The most common approaches to these problems are risk-neutral (optimizes the expectation). Many also focus on risk-averse methods when safety is concerned [1]. In bandit problems, risk-seeking behavior can naturally

arise in exploration strategies such as upper confidence bound and Thompson Sampling [2]. Recently, risk-adaptive approaches have gained interest, but these centralized approaches address single decision makers [3].

Objectives:
We are interested in distributed decision making. Specifically, we propose to explore heterogeneous risk attitudes among a team of robots. While there is a wealth of research on heterogeneity, none investigates heterogeneity in the agents' attitudes towards risk. Potential research questions include:
What robotics applications or missions may benefit from heterogeneous risk attitudes? How should risk attitudes be models in multi-agent systems? Given a team with heterogeneous risk attitudes, how should we allocate tasks or form coalitions? Given a defined space of tasks, how should we compose a team?

Approach:
Possible approaches may include stochastic or non-stationary multi-armed-bandit problems [4], multi-agent reinforcement learning, and online coalition formation and task allocation.

References:
[1] Majumdar, Anirudha, and Marco Pavone. "How should a robot assess risk? towards an axiomatic theory of risk in robotics." Robotics Research. Springer, Cham, 2020. 75-84.
[2] Liu, Xingchi, et al. "Risk-Aware Multi-Armed Bandits With Refined Upper Confidence Bounds." IEEE Signal Processing Letters 28 (2020): 269-273.
[3] Rudolph, Max, Sonia Chernova, and Harish Ravichandarr. "Desperate Times Call for Desperate Measures: Towards Risk-Adaptive Task Allocation." arXiv preprint arXiv:2108.00346 (2021).
[4] Besbes, Omar, Yonatan Gur, and Assaf Zeevi. "Stochastic multi-armed-bandit problem with non-stationary rewards." Advances in neural information processing systems 27 (2014): 199-207.

Contact: asp45, malencia@seas.upenn.edu

Path Optimisation in Continuous Graphs for Exploration in Reinforcement Learning

Motivation:
While the most commonly used algorithms in reinforcement learning are model-free, model-based algorithms can significantly boost performance, particularly in environments with a sparse reward [1]. This is because they introduce a planning component to explore a tree of possible actions originating from the current state, in a manner reminiscent of model predictive control. The value function is updated via bootstrapping with simulated data from a finite time horizon, allowing the model to learn from states which have never been visited.

However, there are several issues with the use of planning in model-based reinforcement learning. First, the act of planning typically requires a discrete action space (alternatively, continuous actions may be sampled from a distribution, but in that case the tree of possible actions cannot be explored exhaustively [2]). Another, bigger problem is that planning can only take place on a short time horizon, as the number of possible trajectories grows exponentially in the number of steps.

In this project, we take inspiration from model-based reinforcement learning to supplement real experience with planning. However, we opt for an optimisation-based approach instead of the standard breadth-first search approach in order to plan a path all the way to the goal (instead of on a fixed time horizon).

Furthermore, instead of mixing this simulated data (which is likely inaccurate) with real experiences to directly train the value function, it is simply used to inform off-policy actions. Then, the value function is trained indirectly with the resulting real experience.

Objectives:
In reinforcement learning methods such as Hindsight Experience Replay [3] and Curriculum Learning [4], the act of learning basic behaviours first is used as a tool to develop more complex behaviour. Similarly, in this project, the objective is to introduce a 'waypoints' system in lieu of breadth first search, which allows agents to consider intermediate states which are not limited to one timestep in the future.

To build an intuition about the benefit of allowing an agent to choose its own waypoints instead of simply performing a breadth-first search, let us consider an example. Assume we are given a pathfinding agent which has been trained to navigate to any goal point within 10 units of its current position. The objective is to extend this functionality so that the agent may navigate to an arbitrary goal location. Now, if we give an agent positioned at [0,0] a goal at position [20,0], then the behaviour will be undefined, because it will consider this a completely new goal. The point [20,0] is too far away from any known goal in the state space for the behaviour to be inferred through interpolation or extrapolation. Furthermore, it is unlikely that a naive model-based reinforcement learning algorithm could solve this with breadth-first search, because it is intractable to search over the number of possible trajectories over 20 timesteps. However, in theory, the agent *should* have a way to reason about how to get to the goal using its existing knowledge---the agent knows how to get to [10,0], and once it is at [10,0], it would know how to [20,0]. So, if the agent chose [10,0] as a waypoint, it could reach the goal without any further training. With multiple waypoints, it could extend its range even further.

In this project, the objective is to use optimisation to select waypoints, and then use those waypoints as intermediate goals in order to select actions. Note that while there are some parallels to model-based reinforcement learning, this method is fundamentally different because the simulated trajectories are not directly used to update the value function. Instead, this is simply a way of selecting actions as a means of exploration, particularly in environments with sparse rewards. Note that the number of waypoints can also be altered, either by a scheduled reduction over the course of the training process, or programmatically as a function of the confidence in getting to the goal. When the number of waypoints drops to 0, the method has no effect.

Approach:
Consider an MDP (S, A, T, R) where S is the state space, A is the action space, T(s,s') is the transition probability function, and R(s) is the reward function. A trajectory of states is written as $\tau = (s_0, s_1, s_2, \ldots)$.

Also, we define the value function as $V([s, g]) = E [\Sigma_t \gamma^t R([s_t, g]) \mid s_0 = s]$. This is equivalent to the standard definition of a value function, except that we decompose the state into the current state s and the goal g.

Now, let us define a graph where the vertices are states and the edge weights are defined by the probability that transition is possible between two states by following the current policy:

$e(s_i, s_j) = P(s_j \mid s_i \in \tau)$

The transition probability $e(s_i, s_j) = E [1(s_j \in \tau) \mid (s_i \in \tau) \wedge (i < j)]$ can be learned by using positive samples s_j from a sampled trajectory and negative samples s_j from the intended goal state that was not reached.

Then, the value of getting from starting state s_0 to goal g via waypoint w is given as:

$H(s, w, g) = e(s, w) * V([s, w]) + e(w, g) * V([w, g])$

Or, more generally:

$H(X) = \Sigma_i [e(x_i, x_{i+1}) * V([x_i, x_{i+1}])]$, where X is a list of waypoints, including s and g.

Note that neither the transition probability function e nor the value function V is a conservative vector field, so the value of H depends on the path taken from s to g. This means that by perturbing w, the value of H will change (H can be interpreted as the path integral). By using optimisation ($w' = \text{argmax}_w H(s,w,g)$), it is possible to find a local optimum for w. We can do this by simply following the partial derivative of H with respect to w:

$w' = \partial H(s,w,g) / \partial w$

By sampling multiple possible waypoints w and performing optimisation on each one, it is possible to find a global optimum for w. Then, we simply take the action that corresponds to using the first waypoint as a virtual goal.

References:
[1] Moerland, Thomas M., Joost Broekens, and Catholijn M. Jonker. "Model-based reinforcement learning: A survey." *arXiv preprint arXiv:2006.16712* (2020).
[2] Deisenroth, Marc Peter, Carl Edward Rasmussen, and Jan Peters. "Model-based reinforcement learning with continuous states and actions." *16th European Symposium on Artificial Neural Networks (ESANN 2008)*. d-side, 2008.
[3] Andrychowicz, Marcin, et al. "Hindsight experience replay." *arXiv preprint arXiv:1707.01495* (2017).
[4] Soviany, Petru, et al. "Curriculum learning: A survey." *arXiv preprint arXiv:2101.10382* (2021).

Contact: asp45, rk627

GNN-based resilient multi-sensor fusion with abnormal data reconstruction

Motivation:
Multi-modality sensor information, including visual image, LiDAR point cloud, IMU, GPS, Radar, depth sensor, sonar, gyroscope and so on, has been widely used in mobile robot localization and navigation, autonomous vehicle driving and mobile manipulator systems. Multi-sensor fusion acts as a critical role and its performance will greatly affect the perception, localization and control ability of a robotic system. Recent research has demonstrated that, with some very simple external systems or easy-implemented methods, the current state-of-the-art multi-sensor fusion systems can be easily attacked and even totally fail [1].

From the aspect of algorithmic frameworks, existing multi-sensor fusion approaches can be divided into filter-based, optimization-based and deep-learning approaches. Nowadays, most of the self-driving systems are based on the prior-fusion based learning approaches, aggregating in raw data level maintains most of the complementary information and does not need any hand-engineering preprocessing operations, showing advanced performance. However, most of the existing approaches assume that the input sensor data is perfect or only contains some noises. ***Resilient multi-sensor fusion*** is still an open challenge.

Objectives:

The main objective of this project is to build a resilient system which can keep a reliable multi-sensor fusion performance in the presence of external interference and even attacks. The first main challenge is how to estimate the reliability of each sensor data in real-time and how to address the un-reliable data. As different sensors' data may contain correlational and complementary information (and the historical information of each sensor itself may also contain correlational information), spatio-temporal consistency information can be used to estimate the uncertainty and then reconstruct the failed data. Another main challenge is how to aggregate the multi-modality sensor data with different frequency and sensor-specific noises, i.e., how to aggregate heterogeneous information and learn joint representation with large modality gaps.

Approach:
Recent work has shown the potential of using Graph Neural Networks (GNNs) to aggregate heterogeneous information for prediction and recommendation tasks [2]. In addition, the authors in [3] utilize GNN to achieve the spatio-temporal sensor information kriging and virtual sensor information generation in traffic prediction tasks. This research points out a promising direction using GNN to reconstruct un-reliable sensor information and build a resilient multi-sensor fusion system in robotics and self-driving applications. This is the main purpose of this MPhil project. In addition, with the reconstruct sensor data and the presented resilient sensor fusion system, a reliable vehicle navigation system is encouraged to be further developed with an end-to-end imitation-learning framework.

References:
[1] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing, 29th USENIX Security Symposium, 2020.
[2] C. Zhang, D. Song, and C. Huang, Heterogeneous Graph Neural Network, KDD, 2019.
[3] Y. Wu, D. Zhuang, A. Labbe, L. Sun, Inductive Graph Neural Networks for Spatiotemporal Kriging, AAAI, 2021.

Contact: asp45, zl457, sn611

GNN-based resilient perception and navigation for autonomous driving applications

Motivation:
Mobile robot navigation and autonomous vehicle driving are the hottest topics in recent years in the robotics community. A large number of approaches are provided, especially the learning-based approaches, which show great potential in pushing the next generation of intelligent transportation and logistics. However, recent research has demonstrated that, with some very simple and easy-to-implement methods, the current state-of-the-art vehicle navigation systems can be easily attacked and even totally fail [1,2,3]. In addition, self-driving accidents occur frequently, also resulting from the un-robust perception and navigation systems.

In the perception and navigation systems, the detection of surrounding dynamic objects, such as other vehicles (human-driving or autonomous), bicycles and pedestrians, acts as a critical role. The object detection performance will directly affect the subsequent trajectory planning and vehicle control modules.

Mis-detection (misidentification or missed detection) is also one of the most widely occurring issues in existing self-driving accidents [4]. Building a robust and reliable detection system is of great importance.

Objectives:
The main objective of this project is to build a resilient system which can keep a reliable surrounding object detection performance in the presence of external disturbances and even attacks. Traditional self-driving systems track and predict the trajectory of each dynamic object independently, which can easily be disturbed by external interference, unreliable sensor data or even attacks. In this project, we will learn the relation among surrounding objects as well as the whole scenario information. The underlying logic is that the motion of each object is also affected by its surrounding objects (and even traffic signals or signs) and there should exist a certain amount of consistency among the objects. Exploiting this spatio-temporal relation provides additional information thus helping to achieve a better perception performance.

Approach:
In order to achieve this, Graph Neural Networks [5] (GNNs) will be utilized for relation learning and feature aggregation. Potential extension: Using the information learnt above helps to estimate the confidence level of each object's state and re-construct the un-reliable object detection information. Based on this, a resilient navigation system can be further built (end-to-end or modular) with imitation learning or reinforcement learning strategies.

References:
[1] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. Chen, M. Liu, and B. Li, Invisible for both Camera and LiDAR: Security of Multi-sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks, IEEE Symposium on Security and Privacy, 2021.
[2] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, C. Cadena, Out-of-Distribution Detection for Automotive Perception, arXiv:2011.01413, 2020.
[3] B. Nassi, Y. Mirsky, D. Nassi, R. B. Netanel, O. Drokin, and Y. Elovici, Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks, ACM SIGSAC Conference on Computer and Communications Security, 2020.

Contact: asp45, zl457, sn611

Sim-to-real via Real-to-sim: Improving real-world performance through real-world learning

Motivation:
Recent work in multi-robot research shows promising results in coordination and control problems using data-driven approaches. Eventually, we want to be able to run such approaches on real robots in the real world. Such data driven approaches require a large number of training samples, ranging from thousands to millions, and therefore rely on simulations for data collection instead of real-world experience and interaction. Replicating all real-world detail in simulation is impossible due to computational constraints, and training in simulations that make simplified assumptions of the real-world and then deploying to the real-world will result in a performance gap, which is also referred to as reality-gap. The research field that deals with training policies in simulations and then run them in another domain (e.g. the real-world) is called sim-to-real transfer.

Sim-to-real transfer has previously been addressed through approaches such as domain randomization and domain adaptation. A fairly new approach is referred to as sim-to-real via real-to-sim, where a policy is trained in a simplified simulation, and the real-world observation is then transformed to an equivalent simulation observation, which the policy can act on.

Objectives:
The objective of this project is to create a novel multi-robot simulation environment that entirely relies on automatically gathered real-world samples of robot-environment and robot-robot interactions, thus avoiding the expensive and time-consuming process of implementing complex motion models and fine-tuning physics engines of simulations. By following this approach, we hope to be able to collect real-world samples of any robot and use this as a motion and dynamics model that considers all real-world effects that are present in the corresponding environment for the corresponding robot model.
Eventually, we expect to see a significantly smaller dynamics reality gap when deploying policies to real robots compared to using available numerical simulation environments.

Approach:
The goal is to train a mapping from current state (e.g. position, velocity, acceleration) and desired action (e.g. desired velocity in world-coordinates) to the next state (e.g. an updated position) using a neural network (or, in the multi-robot case, potentially a GNN (Graph Neural Network)). The parameters for this model have to be determined using gradient descent and imitation learning, given a dataset that was gathered in the real-world environment in which the policy is to be deployed. This dataset can be obtained using a motion capture system that is already installed in our lab. In the single-robot case, it contains only robot-environment interactions, and in the multi-robot case also robot-robot interactions, which is particularly relevant for drones. Effects such as downwash (i.e. the force applied to another drone due to the air mass moved by another drone) are particularly hard to simulate, so following a data-driven approach potentially saves a lot of time..
After collecting the samples and training the model, it can be used to train an action policy. We then compare the performance of the policy using this new data-driven dynamics model to a policy trained in a simplistic simulation environment and a numerical simulation environment (such as PyBullet).

References:
[1] J. Zhang, L. Tai, P. Yun, Y. Xiong, M. Liu, J. Boedecker and W. Burgard, VR-goggles for robots: Real-to-sim domain adaptation for visual control, IEEE Robotics and Automation Letters, 2020.
[2] J. Blumenkamp, S. Morad, J. Gielis, Q. Li and A. Prorok: A Framework for Real-World Multi-Robot Systems Running Decentralized GNN-Based Policies (under review).
[3] J. Blumenkamp, Q. Li and A. Prorok: Evaluating the Sim-to-Real Gap of Graph Neural Network Policies for Multi-Robot Coordination, Robot Swarms in the Real World Workshop at International Conference on Robotics and Automation (ICRA), 2021. Online.
[4] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan and V. Vanhoucke, Using simulation and domain adaptation to improve efficiency of deep robotic grasping, IEEE international conference on robotics and automation (ICRA) (pp. 4243-4250), 2018.
[5] Batra et al. Decentralized Control of Quadrotor Swarms with End-to-end Deep Reinforcement Learning https://arxiv.org/pdf/2109.07735.pdf
[6] Sanchez-Gonzalez et al.]Learning to Simulate Complex Physics with Graph Networks http://proceedings.mlr.press/v119/sanchez-gonzalez20a.html

Contact: asp45, jb2270

[Appendix 1] Motivation for a User Case Study

The pandemic has reshaped the way of using interior spaces like workplaces [1], especially with the indoor social distancing rules. As the plan for returning to offices is resumed with the delivery of mass vaccination and the release of lockdown, the rearrangement of office layout is expected to be a cost-effective solution to ensure safe access to the workplace and minimize possible transmission. The current design solutions implemented under the pandemic condition are knee-jerk, such as switching to a simple one-way system and putting barriers or 2-meter-signs around the place [2]. However, the effectiveness of those solutions remains to be questioned. Compared to the one-size-fits-all strategy, a case-specified solution based on careful evaluation and design is expected to be more effective. The solutions with scientific thinking and evidence are required, while machine learning methods can play a key role in seeking the best possible spatial layout solution. The rationality performed by artificial intelligence can assist the re-layout decisions, with a careful assessment of different possible arrangements of walls, corridors, furniture, facilities and points of interest (e.g. meeting room, restroom and tea points). In this case, the efficiency of design is expected to be improved, and machine learning is applied to look at the design of the office environment for the future.

Project Overview and Objectives
As an extension of the previous MPhil project, this part of the project is an interdisciplinary study of the application of machine learning in computer-aided architectural design. We consider a specific scenario of in-pandemic offices as the user case study.
The interior layout of buildings, as a description of the spatial structure of an architectural environment, is selected as an observation input to the GNN-based differential simulator. The layout optimization process is driven by RL based on a pre-designed scoring metric as the reward mechanism.

We intend to address the following research objectives:

- Explore and identify the quantitative parameters for the in-pandemic office design;
- Investigate the application of the framework in office optimization for in-pandemic offices;
- Find out optimal layout options for the case study with the tool;
- Conduct a post-optimization evaluation.

The major expected outcome from this project is the demonstration of the optimization process. As a further step, we expect a validation of the framework performance, by conducting a post-optimization study. The study may involve a cross-comparison among the other machine learning-based optimization algorithms and the human expert-based evaluation of optimization results.

Contact: Amanda Prorok (asp45); Ronita Bardhan (rb867); Qingbiao Li (ql295); Jiayu Pan (jp844)
References:
[1] Most Workers Want To Work From Home After Covid-19, 2021. URL https://yougov.co.uk/topics/economy/articles-reports/2020/09/22/most-workers-want-work-home-after-covid-19. [Online; accessed 141/09/21].

[2] Department for Business Energy and Industrial Strategy and Department for Digital CultureMedia and Sport. Working safely during coronavirus (COVID-19), 2021.