Autonomous Navigation using Inverse Reinforcement Learning

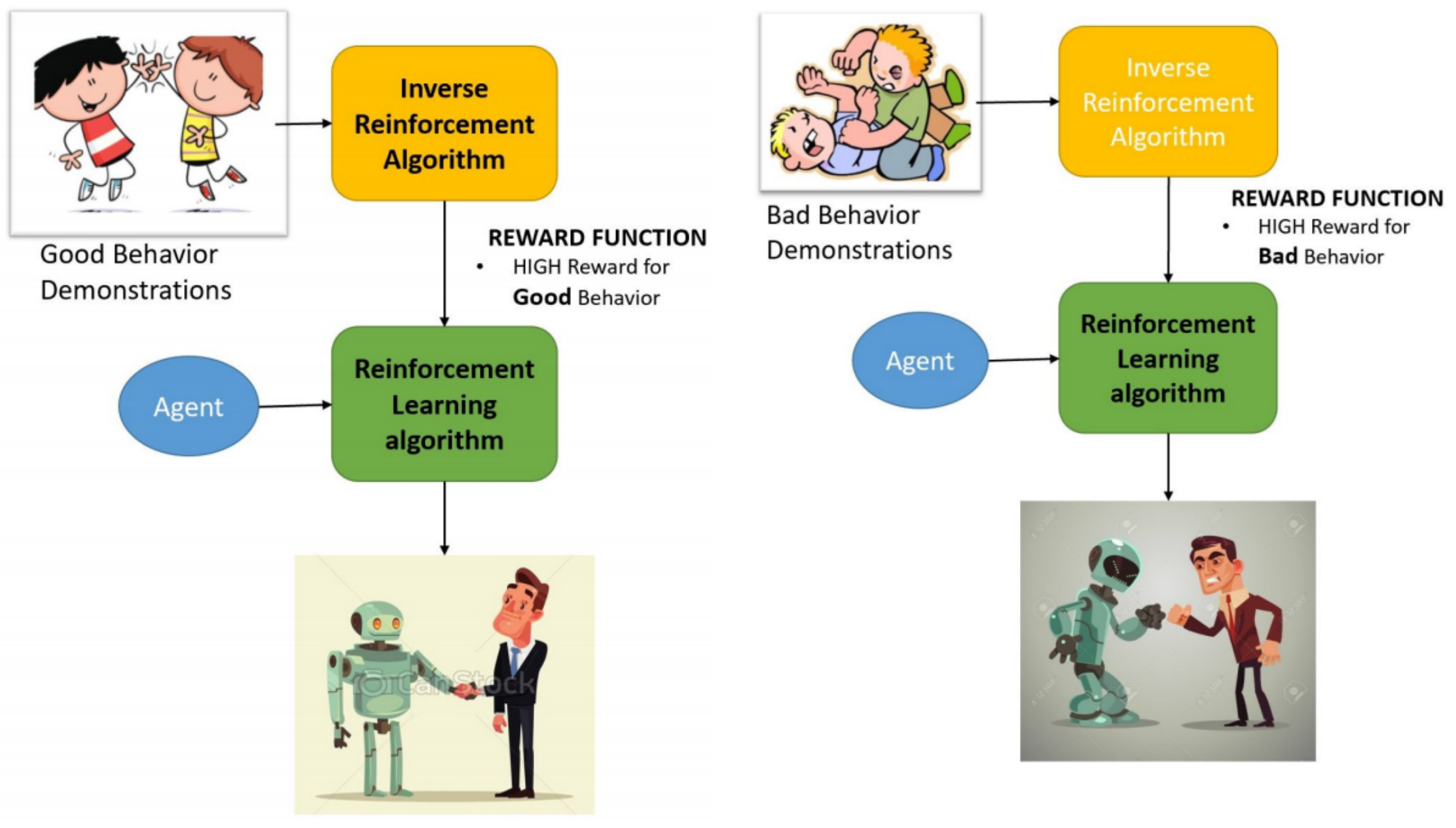
Anirudh.G.J, Bharath Y P, Lakshwin Shreesha and Siddharth Kumar
Under the guidance of Professor P.Meena
Department of Electrical and Electronics, BMSCE, Bangalore

1. Abstract

Programming an autonomous vehicle for every unique scenario it encounters is a time consuming and inefficient task. Widely used Machine Learning algorithms such as Supervised Learning and Unsupervised Learning offer possible solutions, but even they fail to generalize to situations beyond what they have been trained for. A promising area of research to enable such generalizations is the domain of Inverse Reinforcement Learning, wherein an agent observes an expert and infers *good* behavior. Inspired these algorithms, we have developed a computationally simple variant which applies the concept of IRL to encoded trajectories. To do so, we propose spatially and temporally encoding input video frames and applying IRL on the resulting encoded vectors.

2. Introduction

An Markov Decision Process (MDP) is a set of states (S), actions (A) and state-transition probabilities (θ). Additionally, each state-action pair corresponds to a reward (R). These rewards are accumulated over several iterations to calculate the value of a state and the value of an action. A policy (π) describes a set of actions to be taken over the state space. The optimal policy ($\pi*$), is one such special mapping which maximizes the expected discounted sum of rewards between two given states (start and goal) in an episodic task (which repetitively solves the same problem). The goal of IRL is to learn a reward function for each state based on parts of a given policy (a demonstration). In a broader sense, the goal is to be able to generate a policy over a state space (S), which correlates to what an expert demonstrates.


Good Behavior Demonstrations
Inverse Reinforcement Algorithm
REWARD FUNCTION
• HIGH Reward for Good Behavior
Agent
Reinforcement Learning algorithm
Bad Behavior Demonstrations
Inverse Reinforcement Algorithm
REWARD FUNCTION
• HIGH Reward for Bad Behavior
Agent
Reinforcement Learning algorithm

3. Trajectory Embedding

To proceed with the proposed approach, a suitable representation of the entire trajectory in a lower dimensional space is essential. The essence of trajectory embedding is to successfully represent an entire trajectory, a set of continuous time series images in a lower dimensional state space such that the spatial and temporal information of each image in the trajectory is captured, we monitor this by the loss. Given any frame s_i in a sequence $S = s_1, s_2, ..., s_N$, the embedding is computed as $u_i = \phi(s_i; \theta)$, where ϕ is the neural network CNN and LSTM encoder parameterized by θ

7. References

[1] Abbeel, P., and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning, 2004.

[2] Debidatta Dwibedi Et al., Temporal Cycle-Consistency Learning, 2019.

[3] Ziebart Et al., Maximum Entropy Inverse Reinforcement Learning, 2008.

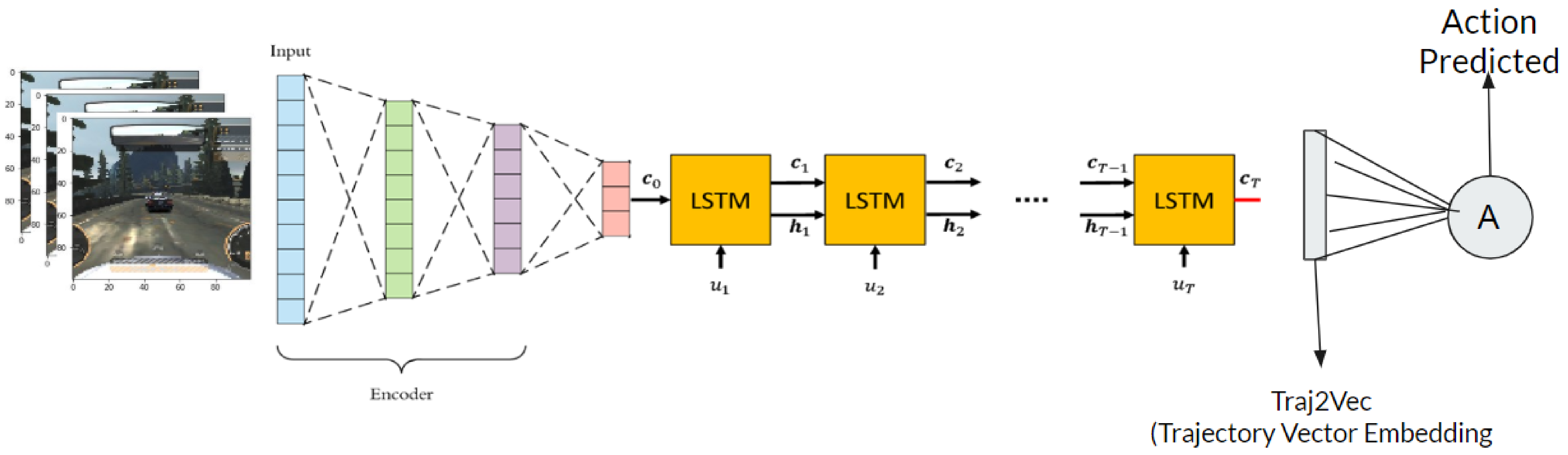
[4] Chelsea Finn Et al., Guided Cost Learning, 2016.

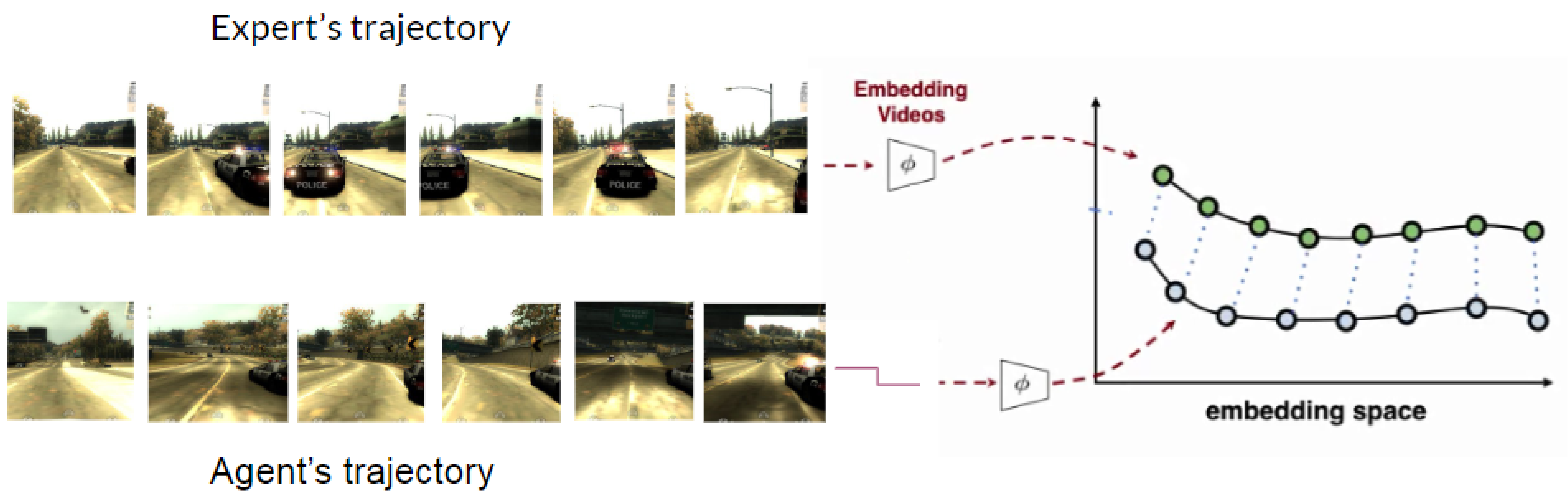
[5] Markus Wulfmeier Et al., Maximum Entropy Deep Inverse Reinforcement Learning, 2016.

4. Approach

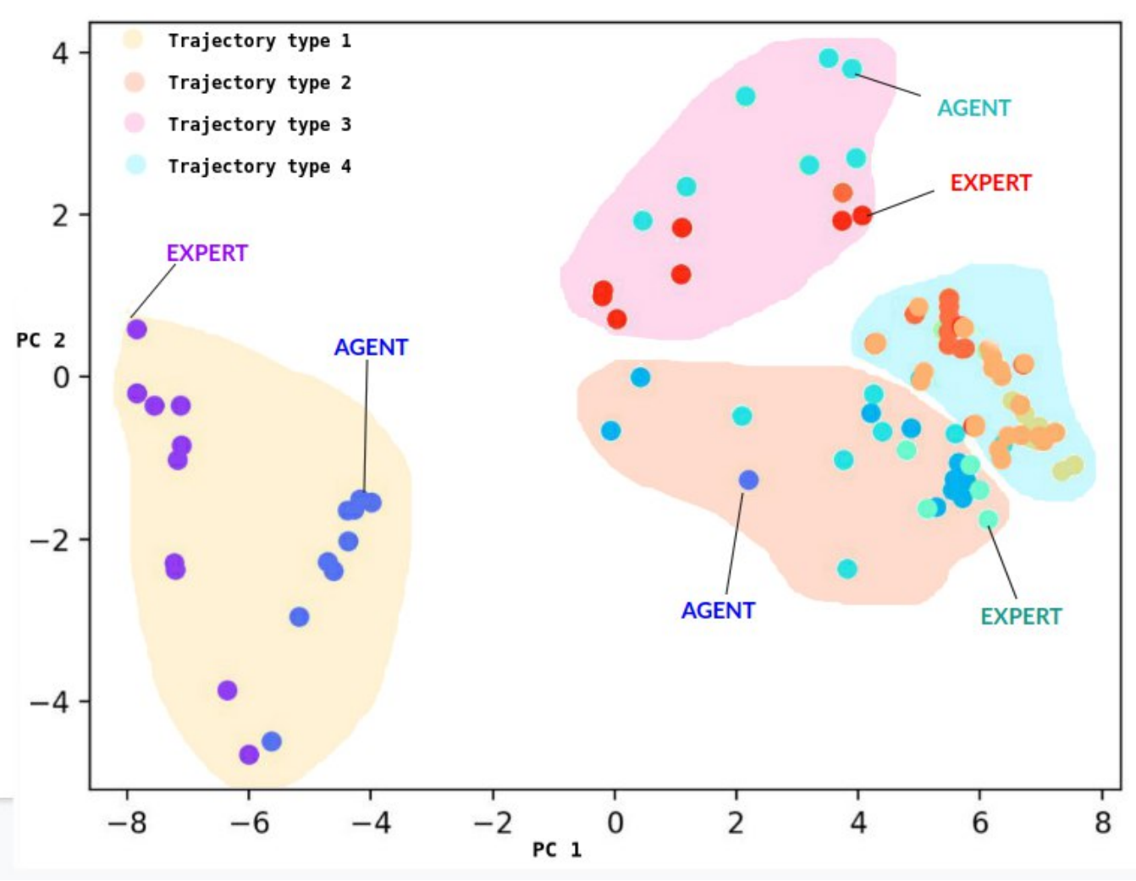
For the purpose of this work, the demonstrations received by the algorithm are assumed to be performed by an expert, meaning that they are assumed to be optimal. A demonstration (D) consists of numerous examples, each of which is a trace of the optimal policy through state space. These are represented in the form of sequences of state-action pairs (s, a). One method to generate a policy is to generate state values for each state based on state features. It is assumed that a weighted combination of state features can provide a quantitative evaluation of a state. The first problem, then, is to learn a mapping from state features to state values that produces a policy for which state-action pairs are consistent with the given examples. The second problem is to learn a non-linear mapping from these values to state reward, which produces a policy consistent with the given examples (as described for the first problem)

In this approach we develop a measure to quantify how good a particular action in a trajectory is by comparing the embeddings. These low dimensional embedding vectors contain essential features about the environment it is in. By using this metric we try to model an agent's behavior to that of an expert. We adopt Inverse reinforcement learning to treat this metric as a reward and penalize actions of the agent to achieve ideal behavior which the expert implicitly conveys.





5.Results and Analysis



In order to visualize the similarity between two embeddings we project the n-dimensional embedding into a 2-D space using PCA. The graph shows such a comparison. Each cluster depicts a separate environment state, in which one category of blobs represents the expert's embedding and the subsequent blob represents the agent's learned representation. The figure showcases our results wherein the random agent infers the motive of the expert by using the Kendall tau metric to guide its learning.

The results above were developed as a result of extensive training on the Udacity self-driving simulator.

6. Conclusions

The main aim of this project was to extract an approximation of the human's reward function for a particular task—driving a car in this case. We have demonstrated through this project that the model has learnt the intentions of the user in completing the task. The main significance that we intend to prove through this demonstration is that the reward function is a more robust and transferable definition of any task in a real world environment.