En.601.787 ML for Trustworthy AI
Superhuman Imitation Learning in low-resource environment

NIKHIL SHARMA (NSHARM27), KRITIKA SHARMA (KSHARM19), and AMMAR AHMED PALLIKONDA LATHEEF (APALLIK1)

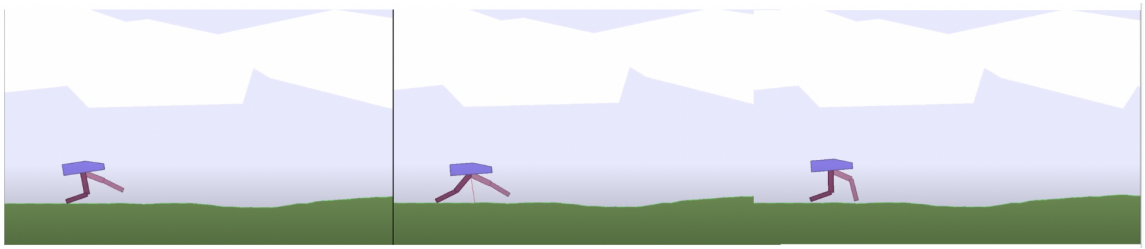

Fig. 1. A model trained using a great reward function

In this project proposal, our objective is to expand superhuman imitation learning (IL) to bipedal walking, explicitly focusing on incorporating superhuman autonomy objectives within the imitation learning framework. While imitation learning has been successfully applied in various fields, such as self-driving cars, surgical procedures, and video games, it primarily aims to mimic human behavior. However, this approach may not yield optimal outcomes for entities with non-human capabilities, particularly in low-resource settings. We propose an innovative methodology that utilizes the Minimum Suboptimal Inverse Optimal Control (MinSub IOC) function to address this limitation. This approach aims to enhance performance beyond the highest level demonstrated by humans for each component of the cost function. We intend to implement this methodology within the context of the bipedal walking task in the OpenAI Gym environment, which serves as our testing platform. Given the constraints of a limited dataset in a low-resource setting, we employ a form of gradual data augmentation to enhance the performance of our system. By applying the MinSub IOC function and employing gradual data augmentation, we aim to achieve superhuman performance in the bipedal walking task despite resource limitations. This project proposal seeks to push the boundaries of imitation learning and showcase its potential for enhancing performance in non-human domains with constrained resources.

Additional Key Words and Phrases: Imitation Learning, Reinforcement Learning, Bipedal Walking, Data Augmentation

ACM Reference Format:
Nikhil Sharma (nsharm27), Kritika Sharma (ksharm19), and Ammar Ahmed Pallikonda Latheef (apallik1). 2018. En.601.787 ML for Trustworthy AI Superhuman Imitation Learning in low-resource environment . In *Final report, course project*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Superhuman autonomy refers to the ability of an AI system to perform tasks with a level of autonomy and decision-making that exceeds human capabilities. This means that the AI system can operate and make decisions independently,

Manuscript submitted to ACM

without human intervention or oversight, and achieve superior performance in various domains. This could potentially revolutionize fields like robotics, transportation, and healthcare.

During our research, we were greatly influenced by the research papers by Dr. Brian D. Ziebart, namely his work in [7], [14] and [12] which provide essential insights into the fields of AI and machine learning. These works contribute to developing autonomous systems with superior capabilities, fair decision-making, and accurate modeling of human behavior.

These papers collectively contribute to advancing AI systems by enhancing their autonomy, modeling human behavior, and ensuring fairness. Building upon these, our work on "Superhuman Imitation Learning For Low Resource Setting" aims to explore the application of superhuman imitation learning in scenarios with limited resources using Bipedal Walker in OpenAI Gym.

We also incorporated Imitation Learning in our methodology. It is a technique by which an agent learns a policy that tries to mimic the behavior seen in demonstrations [1]. Imitation learning has been successfully used in various tasks like self driving [2][3], surgery [4], and video games [5][6]. But these methods generally aim to mimic the behavior of the demonstrations. To improve performance, they try to either curate the data by using only expert demonstrations, or augment the capabilities of the agent by giving it enhanced sensors compared to humans, like in the driving task [3].

These methods suffer from the disadvantage of purely learning the policy based on human demonstrations. We argue that the policy learnt by human experts is only optimal for humans, who are beings with limited physical capabilities, large reaction times, and finite stamina. On the other hand, a non-human entity with superior capabilities might benefit from trying to learn a policy that can outperform humans instead.

The iterative nature allows continual improvement, exceeding the initial gold standard and achieving superhuman capabilities. Deploying the algorithm iteratively enables the model to learn from mistakes, discover new strategies, and enhance performance. Superhuman learning provides an exciting opportunity to surpass gold standard trajectories and approach true superhuman capabilities. Applying this concept to bipedal walking presents a unique challenge, combining the complexity of human locomotion with stability issues, making it an exciting domain for superhuman autonomy exploration.

The method in [7] was originally applied to a relatively simple cursor-pointing task. We decided to use it on Bipedal Walker, an OpenAI gym game. Bipedal Walker is a classic control problem in robotics and AI, which involves designing an agent that can control the movement of a bipedal walker, a humanoid robot with two legs. In this environment, an agent controls a humanoid robot with two legs. The goal is to teach the agent to walk as far as possible while maintaining balance. The agent receives a reward based on how far the robot walks, with higher rewards for longer distances. To solve the Bipedal Walker problem, reinforcement learning algorithms can train the agent to make decisions that maximize the reward signal. This involves designing a policy that maps the current state of the environment to an action that the agent should take.

Additionally, to overcome the challenges associated with a small dataset, we have incorporated an approach to gradually

augment our dataset. This iterative process involves adding our model to the dataset after each iteration and removing the worst-performing one. This self-improvement cycle not only conserves data but also fosters continuous learning and improvement of the model, ensuring that its performance escalates with each iteration.

2 RELATED WORK

2.1 Imitation learning

The key idea behind imitation learning is that an agent can learn an effective policy by observing and replicating the actions of expert demonstrators, removing the need to unnecessarily explore the vast search space. There have been a wide variety of approaches proposed for Imitation Learning over the last several decades, including Behavior Cloning (BC) [9], Dataset Aggregation (DAgger) [10], and Generative Adversarial Imitation Learning (GAIL) [11].

Behavior Cloning (BC) is a simple yet effective approach that involves learning a policy by directly mapping observed states to actions using supervised learning [9]. BC is prone to compounding errors, especially with small datasets. The inaccuracies may keep compounding and lead to encountering states not present in the expert demonstrations.

Generative Adversarial Imitation Learning (GAIL) takes a different approach, employing a generative adversarial network (GAN) framework to learn a policy by matching the distribution of state-action pairs between the expert and the agent [11]. GAIL has been shown to produce policies that closely resemble expert behavior while requiring fewer demonstrations compared to other imitation learning methods.

In [7] the authors propose a novel approach to autonomous decision-making. The paper introduces the concept of subdominance minimization, which optimizes an agent's behavior to minimize instances of being dominated by other agents. The agent tries to learn a policy that performs unambiguously better than every human demonstration according to every feature of the cost function (or at least, it attempts to learn such a policy). They devise a Minimum Suboptimal Inverse Optimal Control (MinSub IOC) function that attempts to identify the demonstration in each feature that it least dominates over and then tries to optimize this function. The research demonstrates the effectiveness of this approach in achieving superhuman performance in various domains, including robotics and game playing.

The research in [14] presents a framework for modeling human behavior using maximum entropy inverse optimal control (MaxEnt IOC). The authors demonstrate how MaxEnt IOC can estimate the underlying decision-making process of humans by observing their behavior. This approach is based on the idea that human behavior can be modeled as a rational process of decision-making that seeks to maximize a specific utility function. MaxEnt IOC is a probabilistic framework that estimates a human's underlying utility function by observing their behavior in a particular task. The estimated utility function can then predict their behavior in new situations. The paper demonstrates the MaxEnt IOC approach's effectiveness in various domains, including driving behavior and pedestrian navigation.

2.2 Superhuman Imitation Learning

Superhuman autonomy is a recent concept that aims to develop policies for autonomous agents that not only emulate human behavior, but also unambiguously outperform it in various aspects of a given task [7].

They propose the MinSub IOC function that minimizes the subdominance of the minimum cost trajectory $\mathcal{E}^*(w)$ with respect to its weights, w and the set of demonstration trajectories $\tilde{\mathcal{E}}_i$ (see section 3 for definition of notation).

2.3 Application of MinSub IOC

This technique has been applied on two tasks, a cursor pointing task [7] and a fairness task [12]. On the cursor pointing task, it achieves superhuman performance on 78% of the samples. In [12], the authors frame fair machine learning as an imitation learning task. By leveraging subdominance minimization (Ziebart et al., 2022) and policy gradient optimization methods (Sutton and Barto, 2018), they are able to achieve superhuman performance in 100% of the tasks.

But the superhuman imitation learning method has not been applied on a real world task that is filled with complex interconnections of features. The authors mentioned that they plan to explore more complex tasks in the future, but they have not done that yet. We test the viability of this approach on a complex real world task: autonomous driving.

2.4 Gradual Data Augmentation for Low-resource Setting

Gradual fine-tuning [15], originally introduced in the realm of Natural Language Processing (NLP), advocates for a multi-stage adaptation of models, yielding substantial improvements without altering the model or learning objective. Drawing parallels to this approach, our project implements a concept of gradual data augmentation in the context of bipedal walking. We iteratively integrate the newly refined model into our dataset. If it is good, it will get chosen as one of the support vectors in the subsequent iterations, otherwise, it will be discarded. This process, similar to [15], allows for progressive learning and improvement in a low-resource setting, enabling us to maximize the performance of our bipedal walking model.

3 PROPOSED METHOD

Problem formulation: We want to test [7] in a more complex and realistic task like driving where the applications will have more realistic real-life impact and the method will be validated for more complicated tasks.

Problem Setup:

- **Defining the state action pairs** At a particular timestep t, the Markov Decision Process is defined as: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, where the State, S includes set of observations about the environment O_t from the Carla sensors: GNSS, Depth Camera, IMU, Lane invasion detector, LIDAR, radar sensor and Object detector. The actions for a particular timestep t, are defined as a_t which include: steering angle, throttle, and brake values. A demonstrator produces trajectories, $\tilde{\xi} = (\tilde{s}_1, \&tildea_1, \ldots, \tilde{a}_{T-1}, \tilde{s}_T)$, of states, $\tilde{s}_t \in \mathcal{S}$, and actions, $\tilde{a}_t \in \mathcal{A}$, according to the demonstrator's policy, π. There are K state-action features, $\mathbf{f} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^K_{>0}\Big)$
- **Expert demonstrations**: Our expert demonstrations include state action pairs by humans and [8] pre-trained models by CARLA team. $D = \{d1, d2, ...dn\}$ and for each $d1$ we have sequence of state-action pairs (s_t, a_t).
- **Feature Extractor**: We will be using a pre-trained encoder-only transformer-based model that will take in the observations and produce complex features/embeddings that captures the temporal relationships.
- **Learning the reward function**: We learn the reward function from the minimal subdominance inverse optimal control training function [7]. We define the reward function as a neural network that takes in a feature and action a_t and outputs the scalar reward.

- **Reinforcement Learning**: We train the model using reinforcement learning to maximise the sum of rewards by updating the policy that maps the state to action.
- **Evaluation**: We evaluate the model in different settings such as rainy, dirt and in unseen paths and compare the performance to humans.
- **Iterative data augmentation**: We do iterative data augmentation with MS IOC to mitigate the requirement of large amounts of gold standard trajectories. We will start will novice demonstrations and train a set of RL models based on steps 3-6. We then make these the expert demonstrations and continue but with a slight change of including the output model from iteration 1 to be included in the demonstrations. The final set of models should achieve similar performance or surpass the model that used state-action pairs from expert demonstrations.

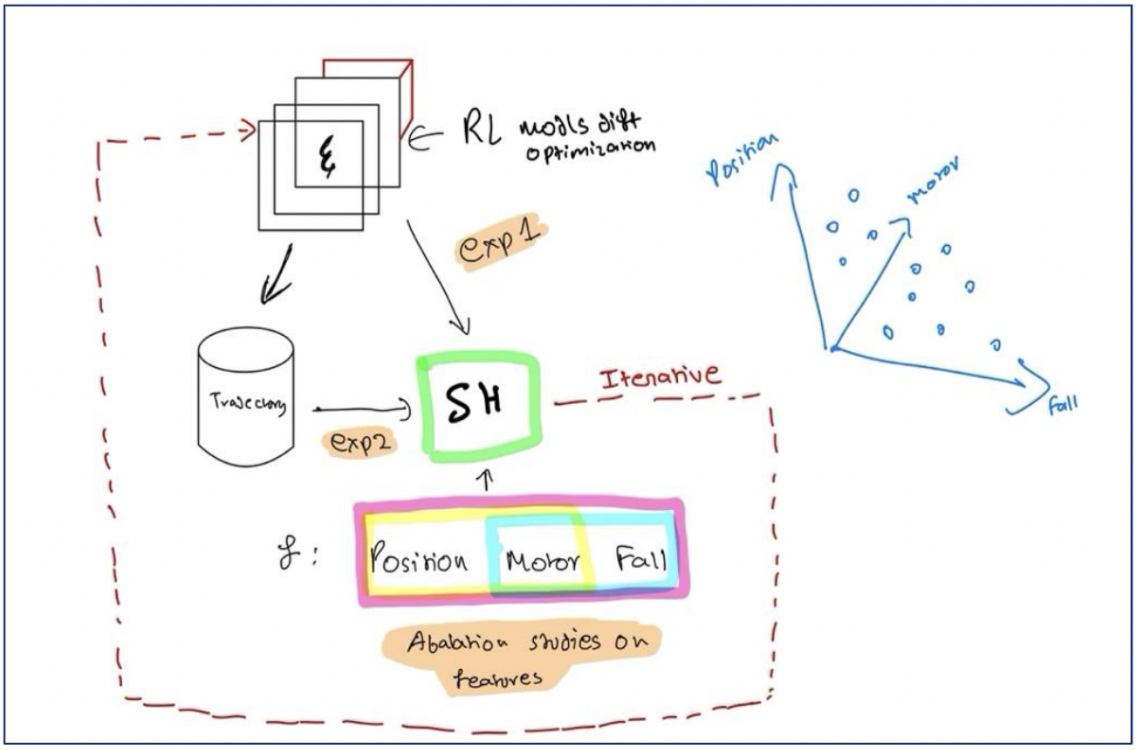

Fig. 2. Our Problem setup. Our iterative method to gradually augment the data contains the SH module, or the superhuman algorithm at the center (green box). We iteratively provide it the input data (either individual trajectories or models) and add the superhuman model to the dataset. We also do ablation studies where we choose to only provide access of a subset of the features to the superhuman algorithm.

Algorithm 1 Inverse Optimal Control for 3 dimensional task

1: **Step 1:** Use the trained model from [13] as the initial optimal behavior for our problem. The weights of this model were released to be used for autonomous driving research. These weights will be used as our cost weights w.
2: **Step 2:** Find support vectors given the value of alpha and demonstrations: $\tilde{\Xi}_{\mathrm{SV}_k}(\mathbf{w}, \alpha_k)$ given ξ^*
3: **Step 3:** The Hinge slope α and cost weights w will be updated exactly as described in [7].
We optimize each α_k using stochastic exponentiated gradient descent where the α_k is regularized:

$$\alpha_k \leftarrow \alpha_k e^{\eta_t \left(f_k(\tilde{\xi}) - f_k(\xi^*) - \lambda \alpha_k \right)}$$

using an appropriately decaying learning rate η_t.
We employ a similar exponentiated subgradient update for cost weights w :

$$\mathbf{w} \leftarrow \mathbf{w} \odot \exp\left(-\eta_t \partial_{\mathbf{w}} \operatorname{subdom}^k_{\alpha_k,1}\left(\xi^*(\mathbf{w}), \tilde{\xi} \right) \right)$$

Algorithm 2 IOC for low resource setting

1: **Step 1:** Train different sub-optimal functions with different optimizations. No model should be good in all the action space. These weights will be used as our cost weights w. Let us call this pool of models δ.
2: **Step 2:** Find support vectors given the value of alpha and demonstrations: $\tilde{\Xi}_{\mathrm{SV}_k}(\mathbf{w}, \alpha_k)$ given ξ^*
3: **Step 3:** The Hinge slope α and cost weights w will be updated exactly as described in [7].
We optimize each α_k using stochastic exponentiated gradient descent where the α_k is regularized:

$$\alpha_k \leftarrow \alpha_k e^{\eta_t \left(f_k(\tilde{\xi}) - f_k(\xi^*) - \lambda \alpha_k \right)}$$

using an appropriately decaying learning rate η_t.
We employ a similar exponentiated subgradient update for cost weights w :

$$\mathbf{w} \leftarrow \mathbf{w} \odot \exp\left(-\eta_t \partial_{\mathbf{w}} \operatorname{subdom}^k_{\alpha_k,1}\left(\xi^*(\mathbf{w}), \tilde{\xi} \right) \right)$$

4: **Step 4:** The final output model from superhuman will be added to the pool of models δ. Then repeat the superhuman learning loop till the model converges to optimal reward.

4 RESULTS

4.1 Effect of Changing Reward Function

We show the effect of changing the rward function weights. In each case, we train the model to convergence using the PPO method.

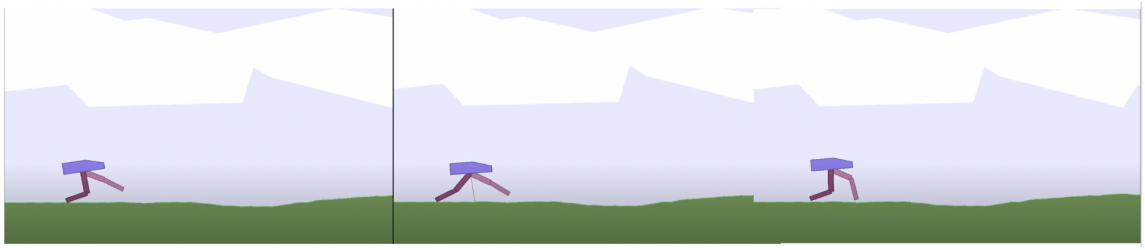

Fig. 3. We provide the model with the default reward weights given by the box2D-py library. The model never falls, reaches the end of the map, and moves while conserving its energy.

4.1.1 *Great Reward Function.*

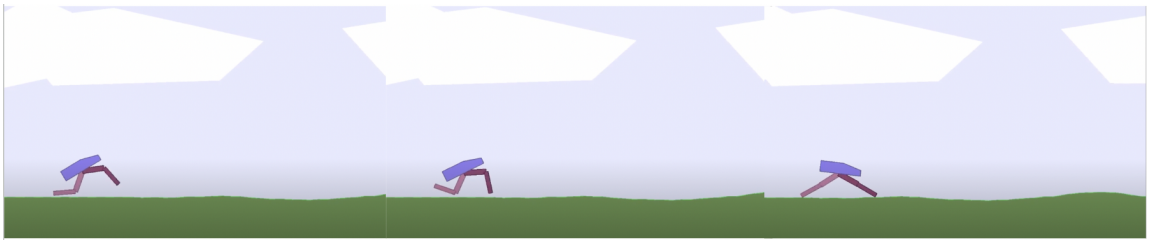

Fig. 4. We provide the model with the default reward weights given by the box2D-py library, but invert the motor reward weight. The model never falls, reaches the end of the map, but wastes its energy while moving, flourishing its legs with each step.

4.1.2 *Inverted Motor Reward Weight.*

Fig. 5. We provide the model with the default reward weights given by the box2D-py library, but remove the fall reward weight. The model always falls, but still moves forward, and moves while conserving energy.

4.1.3 *No Fall Penalty.*

4.2 Superhuman Imitation Learning without Gradual data augmentation

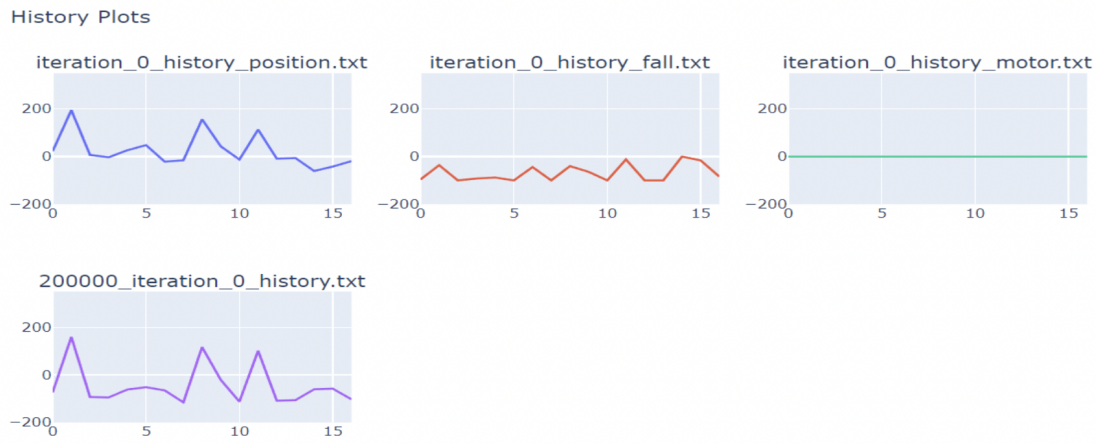

Fig. 6. In a single run with the same trajectories used for the data, we run multiple iterations and plot the varying performance for (from left to right) position, motor, fall, and the sum of the three feature rewards, respectively.

4.3 Superhuman Imitation Learning with Gradual Data Augmentation

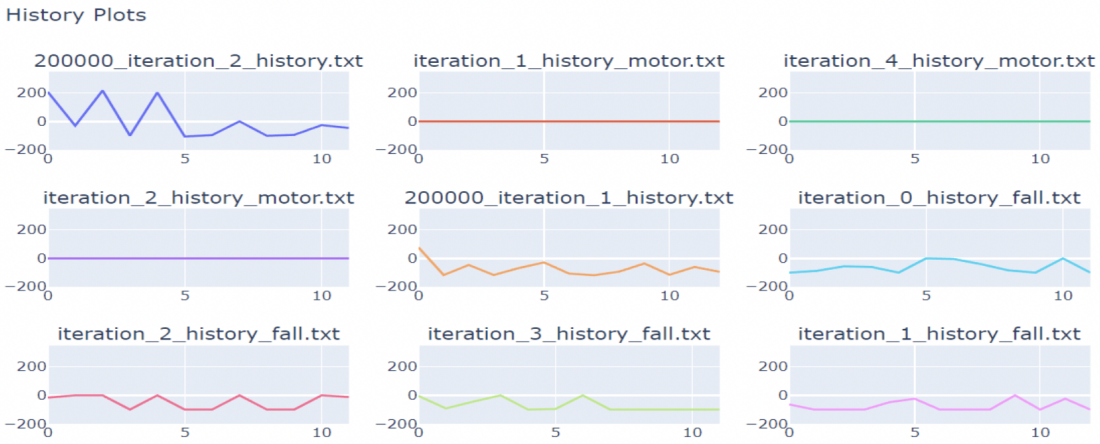

Fig. 7. We apply gradual data augmentation to help with our low resource environment. We run multiple iterations and plot the varying performance for (from left to right) position, motor, fall, and the sum of the three feature rewards, respectively.

4.4 Ablation Studies

We conduct ablation studies to understand the effect of different features on superhuman behavior. We selectively remove features and see the fluctuation on rewards obtained.

Another experiment we run is that during step 2 of our algorithm where we choose support vectors, we for each of our experiments choose 1) model weights as the demonstrations to choose support vectors from and 2) the trajectories of

the models as our demonstrations. For this, we see a smoother convergence for trajectories but overall the improvement was not huge and the model still fluctuated a lot. The smoother convergence makes sense as when we choose trajectories of the models our database increases exponentially giving the model more data points to choose from.

To perform this study we run the following experiments:

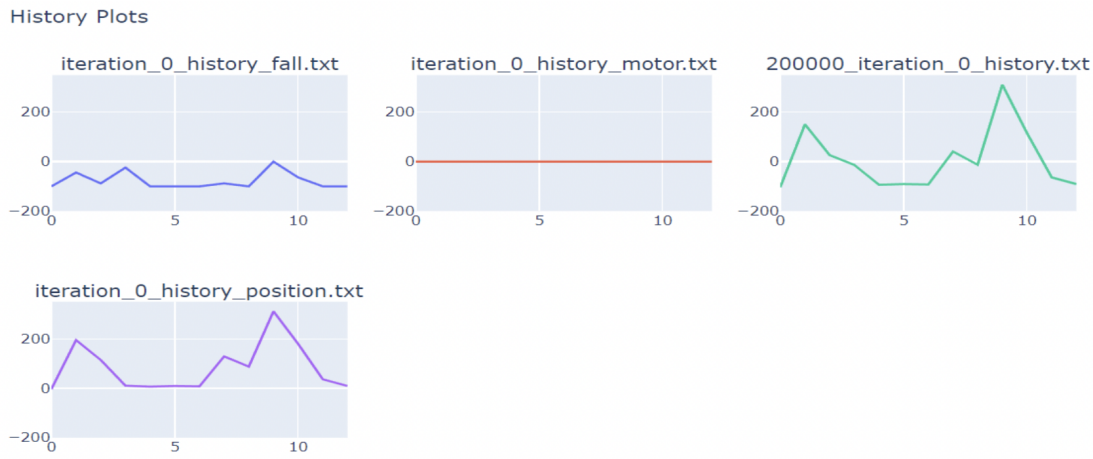

Fig. 8. We remove the fall parameter. In a single run with the same trajectories used for the data, we run multiple iterations and plot the varying performance for position, motor, fall, and the sum of the three feature rewards, respectively.

4.4.1 *Removing the fall parameter.*

Fig. 9. We remove the position parameter. In a single run with the same trajectories used for the data, we run multiple iterations and plot the varying performance for position, motor, fall, and the sum of the three feature rewards, respectively.

4.4.2 *Removing the position parameter.*

features	iterative	result
fall + motor	no	Slow convergence and takes a long time
fall + motor	yes	same as no iterative but has smoother convergence although both fluctuate and fall starves motor of good support vectors
fall + motor + position	no	Better convergence and better allocation Added some tricks to only allow 10% of support vectors to be chosen for any feature but Fall is still troublesome
fall + motor + position	yes	same as above and achieves higher rewards after 5 iterations
motor + position	no	The problem of fall starving is gone and it achieves the most reward of 297

4.5 Using deterministic TD3 instead of PPO

With limited time we could not run the TD3 till convergence since just 3 iterations took 5 hours. While we did not run it fully we saw a consistent increase in rewards across different runs and a more linear increase in individual feature rewards. This shows that our implementation of superhuman algorithm is correct. Reward: Iteration 1: fall -100.0 motor -0.1030901416015625 position 27.93383742160156 different_reward

Iteration 2: fall 0.0 motor -0.07880137240409851 position -84.6798233875959 different_reward

Iteration 3: fall 0.12 motor -0.0928503 position 12.6798233875959 different_reward

5 CONTRIBUTIONS

- Superhuman paper: The superhuman paper has only been used for a very simple task i.e. the cursor-pointing task. This task has no obstacles or a nonlinear space. We are applying the superhuman method to a more complex task i.e. bipedal walker. This shows a flaw in the superhuman method while scaling to a more complex environment and in a low-resource setting.
- We have done an extensive ablation study on the effects of features and choosing models vs trajectories in the step 2 of the superhuman algorithm. We see a better convergence for trajectories as database.
- Learning superhuman behavior even with novice demonstrations: Using the iterative learning setup in our proposed method we want to reach superhuman behavior that can challenge the best demonstrations only using the novice demonstrations. This will make for a realistic setting and make it more applicable to practical usage where expert demonstrations are not always present.

6 CHALLENGES & FUTURE WORK

- The OpenAI gym library was not customizable enough on its own to implement this algorithm. So we had to modify the library.
- The algorithm has a high time complexity and is not very scalable. It is difficult to perform parameter tuning. It takes 2 hours for just 1 iteration with low data and just 200000 episodes which is not enough for convergence.
- We used the PPO method, which gave models with highly varied model performance with each run even with the same initial reward function.

For future work we plan to apply it on a use-case with real-world utility, like driving. Granted more powerful GPUs we would like to run the model till convergence as we belive that we will see the full power of superhuman behavior when we run it for 100 iterations and the iterative data augmentation for 10 iterations. This would require 4 40GB A100 to run.

REFERENCES

[1] Dixon, M.F., Halperin, I., Bilokon, P. (2020). Inverse Reinforcement Learning and Imitation Learning. In: Machine Learning in Finance. Springer, Cham. https://doi.org/10.1007/978-3-030-41068-1_11

[2] H. M. Eraqi, M. N. Moustafa and J. Honer, "Dynamic Conditional Imitation Learning for Autonomous Driving," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 22988-23001, Dec. 2022, doi: 10.1109/TITS.2022.3214079.

[3] Ahn, J., Kim, M. Park, J. Autonomous driving using imitation learning with look ahead point for semi structured environments. Sci Rep 12, 21285 (2022). https://doi.org/10.1038/s41598-022-23546-6

[4] B. Li et al., "3D Perception based Imitation Learning under Limited Demonstration for Laparoscope Control in Robotic Surgery," 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 7664-7670, doi: 10.1109/ICRA46639.2022.9812010.

[5] Artemij Amiranashvili, Nicolai Dorka, Wolfram Burgard, Vladlen Koltun and Thomas Brox, Scaling Imitation Learning in Minecraft, CoRR, abs/2007.02701, 2020, 17 Jul 2020 https://dblp.org/rec/journals/corr/abs-2007-02701.bib

[6] Panse, A., Madheshia, T., Sriraman, A., & Karande, S.S. (2018). Imitation Learning on Atari using Non-Expert Human Annotations. AAAI Conference on Human Computation Crowdsourcing.

[7] Ziebart, B., Choudhury, S., Yan, X. & Vernaza, P.. (2022). Towards Uniformly Superhuman Autonomy via Subdominance Minimization.Proceedings of the 39th International Conference on Machine Learning, in Proceedings of Machine Learning Research 162:27654-27670 Available from https://proceedings.mlr.press/v162/ziebart22a.html.

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun, CARLA: An Open Urban Driving Simulator, Proceedings of the 1st Annual Conference on Robot Learning, PMLR 78:1-16, 2017.

[9] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. volume 3, pp. 88–97. MIT Press, 1991.

[10] Stephane Ross, Geoffrey Gordon, Drew Bagnell, A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR 15:627-635, 2011.

[11] Ho, J., & Ermon, S. (2016). Generative Adversarial Imitation Learning. NIPS.

[12] Memarrast, O., Vu, L., & Ziebart, B. (2023). Superhuman Fairness. arXiv preprint arXiv:2301.13420v1 [cs.LG].

[13] Chen, Dian & Krähenbühl, Philipp. (2022). Learning from All Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR).

[14] BD Ziebart, AL Maas, JA Bagnell, AK Dey (2009). Human Behavior Modeling with Maximum Entropy Inverse Optimal Control. AAAI Spring Symposium: Human Behavior Modeling.

[15] Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. 2021. Gradual Fine-Tuning for Low-Resource Domain Adaptation. In Proceedings of the Second Workshop on Domain Adaptation for NLP, pages 214–221, Kyiv, Ukraine. Association for Computational Linguistics.