A Survey on Different Methods Applied in Humanoid Deep Reinforcement Learning

Yu-Wei Chang
Department of Electrical Engineering
National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan
qiyoudaoyi@gapp.nthu.edu.tw

Po-Hsiang Hsu
Department of Electrical Engineering
National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan
pohsianghsu@gapp.nthu.edu.tw

Abstract

This thesis surveys various Deep Reinforcement Learning (DRL) methods for humanoid environment in the Mujoco, focusing on Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG) Twin Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC). By comparing these algorithms, the study identifies their strengths, weaknesses, and performance impacts on feature engineering. Results show that while PPO and SAC benefit from feature engineering, TD3 and A2C perform worse, highlighting the need for careful feature selection. Future work includes integrating Physic-Informed Neural Networks (PINNs)[6], hyperparameter fine-tuning, and exploring advanced methods like Maximum-Entropy Reinforcement Learning using Energy-Based Normalizing Flows (MEOW) to enhance the robustness and effectiveness of DRL in humanoid control.

1 Introduction

The field of Deep Reinforcement Learning (DRL) has witnessed significant advancements over recent years, driven by the intersection of deep learning and reinforcement learning techniques. DRL algorithms have been successfully applied to a wide range of tasks, including game playing, robotic control, and autonomous driving, demonstrating their potential to solve complex, high-dimensional problems.

One particularly challenging application of DRL is in the domain of humanoid robot control. Humanoid robots, which mimic human form and movement, present unique challenges due to their high degrees of freedom, complex dynamics, and the need for precise, continuous[5] and stable control. The Mujoco (Multi-Joint Dynamics with Contact) simulation environment has emerged as a popular platform for testing and developing DRL algorithms for humanoid robots due to its accurate physical modeling and flexibility.

This thesis aims to provide a comprehensive survey of different DRL methods, mainly focusing on Proximal Policy Optimization (PPO)[7], Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG)[4], Twin Delayed Deep Deterministic Policy Gradient (TD3)[2], and Soft Actor-Critic (SAC)[3]. Apply those methods to humanoid robot control using the Mujoco humanoid

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

environment. The focus is on comparing the performance and characteristics of various DRL algorithms, identifying their strengths and weaknesses, and understanding the underlying factors that contribute to their effectiveness.

In the subsequent sections, we will review related work in the field, describe the methods and algorithms used, present the results of our experiments, discuss future work, and conclude with a summary of our findings. By systematically evaluating multiple DRL methods in a standardized environment, this thesis seeks to contribute to the ongoing research efforts in humanoid robot control and provide insights for future advancements in the field.

2 Related Work

In the field of Deep Reinforcement Learning (DRL), several algorithms have emerged as leading methods for solving continuous control tasks, such as those involved in humanoid robot control. In this section, we briefly explain five prominent DRL algorithms: Proximal Policy Optimization (PPO)[7], Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3)[2], and Soft Actor-Critic (SAC)[3].

2.1 On-Policy Algorithms

A key feature of this kind of algorithm is that all of these algorithms are on-policy: that is, they don't use old data, which makes them weaker in sample efficiency. But this is for a good reason: these algorithms directly optimize the objective you care about—policy performance—and it works out mathematically that you need on-policy data to calculate the updates. So, this family of algorithms trades off sample efficiency in favor of stability.

2.1.1 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO)[7] is an on-policy algorithm that strikes a balance between the stability and reliability of Trust Region Policy Optimization (TRPO) and the simplicity of vanilla policy gradient methods. PPO utilizes a clipped surrogate objective function that limits the size of policy updates, ensuring that the new policy does not deviate excessively from the old one. This clipping mechanism stabilizes training and makes PPO robust to hyperparameter variations. PPO has demonstrated strong performance across a wide range of reinforcement learning tasks, including robotic control, due to its ability to maintain a stable learning process.

2.1.2 Advantage Actor-Critic (A2C)

Advantage Actor-Critic (A2C) is a synchronous version of the Asynchronous Advantage Actor-Critic (A3C) algorithm. A2C combines the strengths of value-based and policy-based methods by simultaneously learning a policy (actor) and a value function (critic). The critic estimates the value of states, while the actor updates the policy parameters in the direction suggested by the advantage function, which measures the relative value of an action compared to the average. By leveraging both value and policy gradients, A2C achieves efficient learning and robust performance in various environments.

2.2 Off-Policy Algorithms

Off-policy algorithms are able to reuse old data very efficiently. They gain this benefit by exploiting Bellman's equations for optimality, which a Q-function can be trained to satisfy using any environment interaction data.

2.2.1 Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) is an off-policy algorithm that combines the advantages of DQN (Deep Q-Network) and policy gradient methods. It operates by learning deterministic policies in high-dimensional, continuous action spaces. DDPG maintains a critic network, which estimates the Q-value of state-action pairs, and an actor-network, which updates the policy deterministically by following the policy gradient of the expected return. The algorithm also employs a target network and experience replay to stabilize training and improve learning efficiency. DDPG is known for its ability to handle complex environments with continuous action spaces effectively.

2.2.2 Twin Delayed Deep Deterministic Policy Gradient (TD3)

Twin Delayed Deep Deterministic Policy Gradient (TD3)[2] is an improvement over DDPG designed to address issues of overestimation bias and variance. TD3 introduces three key modifications: the use of a pair of critic networks to provide more reliable Q-value estimates, the delayed updating of the target networks to reduce error accumulation, and the addition of noise to the target action to smooth out Q-value estimation. These enhancements make TD3 more stable and efficient, especially in high-dimensional continuous action spaces, making it well-suited for complex tasks like humanoid control.

2.2.3 Soft Actor-Critic (SAC)

Soft Actor-Critic (SAC)[3] is an off-policy algorithm that aims to improve sample efficiency and stability by incorporating an entropy term into the reward function. This entropy term encourages exploration by maximizing the policy's entropy, leading to more diverse actions and avoiding premature convergence to suboptimal policies. SAC uses a stochastic actor, a value network, and two Q-networks to provide more accurate value estimates and improve training stability. Its entropy-augmented objective and off-policy nature make SAC highly effective in continuous action environments, showing superior performance and robustness compared to many other DRL algorithms.

In summary, these algorithms—DDPG, TD3, PPO, A2C, and SAC—represent a spectrum of approaches to solving continuous control problems in DRL. Each has its unique strengths and weaknesses, making them suitable for different types of environments and tasks. The following sections will explore the methods and results of applying these algorithms to the Mujoco humanoid environment, highlighting their performance and practical considerations.

3 Methods

In this thesis, feature engineering plays a crucial role in preparing the state and action spaces for effective learning by the DRL algorithms. The Mujoco humanoid environment provides a rich set of features representing the robot's state, including joint angles, velocities, and external forces. The primary goal of feature engineering is to ensure that these features are normalized and scaled appropriately to enhance the learning process.

3.1 Feature Engineering

1. **Positional Values:** As we only care about if the humanoid falls or not, we discard both x and y coordinates.
2. **Orientation and Angular Values:** Then, we takes the orientation of the torso into account. Also, angles from each part of the body are considered.
3. **Torso Velocity:** Torso velocity is important for us to get the information of how the humanoid is moving in space.
4. **Angular Velocity:** In addition to plain angular values, we incorporate angular velocities to help the agent understand the motion happening to the humanoid.
5. **Mass and Inertia:** For each body part, we extract its information about the mass, center of mass, and inertia relative to the center of mass of the whole body. This information helps the agent to adapt to each body part. We normalize the vector of the center of mass and the 6 components of inertia and append them at the end.
6. **Center of Mass Based Velocity:** For each body part, we also take its center of mass-based velocity into account. We normalize the velocity vector and append this information at the end.
7. **Constraint Force Generated as the Actuator Force:** For each joint, there is an actuator force generated by the torque we applied to the humanoid. This is also useful to the agent. We normalize the force vector and append this information at the end.

Table 1: Off-Policy Algorithms Hyperparameters

Algorithm	Learning Rate	Discount γ	Batch Size	Horizon T	GAE λ
PPO	$3 \cdot 10^{-4}$	0.99	64	2048	0.95
A2C	0.007	0.99	-	5	1

Table 2: On-Policy Algorithms Hyperparameters

Algorithm	Learning Rate	Discount γ	Batch Size	Replay Size	Smoothing τ
DDPG	0.001	0.99	256	10^6	0.001
TD3	0.001	0.99	256	10^6	0.005
SAC	$3 \cdot 10^{-4}$	0.99	256	10^6	0.0005

8. **Center of Mass Based External Force:** For each body part, there is an external force, e.g. gravity, normal force. This is also useful to the agent. We normalize the force vector and append this information at the end.

3.2 Reinforcement Algorithms

Here, we go over the settings we have for implementation of each algorithm in Table 1 and Table 2. Most of them are inspired by the original paper of which the algorithm was proposed.

4 Experiments

This section outlines the experiments conducted to evaluate the performance of various deep reinforcement learning algorithms on the Mujoco humanoid simulation. The goal of these experiments is to assess each algorithm's ability to learn efficient and stable control policies for humanoid robots.

4.1 Experimental Setup

The experiments were carried out using the Mujoco simulation environment, which is specifically designed for testing the dynamics of multi-joint humanoid robots with complex interactions with their environment. The humanoid model used in the experiments is equipped with 17 degrees of freedom and various sensors that provide comprehensive state information.

Each algorithm was evaluated under identical conditions to ensure comparability. The simulations ran for a total of 5 million timesteps, and the performance was evaluated every 10 epochs to monitor the learning progress. The primary metrics for evaluation were the total cumulative reward and the stability of the learning curve, indicating the effectiveness and consistency of the policy learned by each algorithm.

4.2 Implementation Details

The experiments involved five reinforcement learning algorithms: Soft Actor-Critic (SAC), Advantage Actor-Critic (A2C), Proximal Policy Optimization (PPO), Twin Delayed DDPG (TD3), and Deep Deterministic Policy Gradient (DDPG). Each algorithm was implemented with and without feature engineering (No F.E./F.E.). So we will have ten results finally, and we will compare them in Result Part.

4.2.1 Software Configuration

The following software was used across all setups:

- Python version 3.11.9
- OpenAI Gymnasium-Humanoid V4
- PyTorch 2.3.1 (cu121)

Table 3: Reward results

Algorithm	Reward w/o F.E.	Reward w/ F.E.	Improvement
PPO	473	487	2.9%
SAC	6557	6888	5%
DDPG	57.159	61.485	7.57%
TD3	367.80	74.119	-
A2C	131.18	78.781	-

4.2.2 Hardware Configuration

Since SAC and TD3 require lots of training resources and time, so I deploy the training environment into different hardware configurations to handle the computational demands of various training scenarios:

- Intel Core i7-13700 with 32GB DDR5 RAM and NVIDIA RTX 4080 was used for the initial experiments involving SAC and its variants without features engineering.
- AMD Ryzen 9 7950X with 128GB DDR5 RAM and NVIDIA RTX 4090 was used to train SAC for extended steps (20M, 10M, and 50M).
- AMD Ryzen 9 5900HS with 24GB DDR4 RAM and NVIDIA RTX 3070 Laptop GPU supported TD3, A2C, PPO, and their no features engineering configurations.
- Another setup involving Ryzen 9 5900HS but with 32GB DDR4 RAM and NVIDIA RTX 2070 Super was used for TD3 no features engineering and DDPG.
- MacBook Air M2 with 24GB of RAM was utilized for running DDPG without a features engineering, highlighting the versatility and adaptability of the implementation across various computing platforms.

5 Results

Table 3 shows the reward outcomes for various algorithms with and without feature engineering (F.E.). The results indicate that both the Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) algorithms benefit from feature engineering, showing improvements of 2.9% and 5%, respectively. The Deep Deterministic Policy Gradient (DDPG) algorithm sees the most significant improvement at 7.57%. However, for the Twin Delayed DDPG (TD3) and Advantage Actor-Critic (A2C), the results with feature engineering are significantly worse, indicating a decrease in performance. We think the reason is that two algorithms are unsuitable for the task.

5.1 Analysis of Features Engineering Impact on SAC and PPO Performance

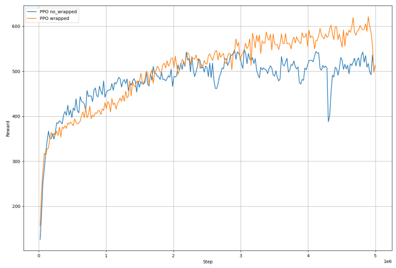
The experimental results demonstrate the impact of using custom observation space features engineering on the performance of SAC and PPO algorithms. The data, visualized through reward progression over time, provides insights into the initial performance, learning stability, and long-term effectiveness of the reinforcement learning strategies under different observation settings.

5.1.1 Initial Performance

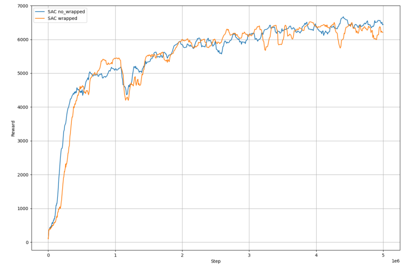
Both SAC and PPO algorithms exhibited higher initial rewards when utilizing the observation space features engineering, suggesting that the additional information provided might be advantageous for initializing more effective policies. This observation is consistent with the hypothesis that enriched observations can enhance the agent's understanding of the environment, potentially leading to better initial decision-making processes.

5.1.2 Learning Stability

The plots reveal distinct differences in learning stability between the two algorithms. The SAC algorithm, when equipped with the features engineering, showed increased variability in reward



(a) PPO with no F.E. and F.E.



(b) SAC with no F.E. and features F.E.

Figure 1: Comparisons of PPO and SAC with and without features engineering

progression, indicating potential challenges in learning stability. This might be due to the complexity or noise introduced by the enriched observation space, which could affect the algorithm's ability to consistently learn effective policies. In contrast, the PPO algorithm maintained a stable upward trend in rewards, similar to its performance without the features engineering, but with slightly higher peaks, suggesting that PPO might be better suited to leverage the enriched data without compromising learning stability.

5.1.3 Long-Term Performance

Over the course of the experiments, both algorithms achieved comparable levels of performance by the end, with or without the features engineering. However, the PPO algorithm with the features engineering showed a slight advantage in terms of reaching higher reward levels faster than its no-feature engineering counterpart. This could indicate that the features engineering not only aids in quicker learning but also helps in maintaining robust performance over time.

5.2 Analysis of Features Engineering Impact on DDPG, A2C and TD3 Performance

The experimental results illustrate the complex influence of feature engineering on the performance of the DDPG, A2C, and TD3 algorithms. While DDPG exhibits substantial improvement, TD3 and A2C performance deteriorates significantly with the integration of feature engineering. This disparity in results suggests that the benefits of feature engineering depend heavily on the specific attributes and operational mechanisms of each algorithm.

5.2.1 DDPG Performance

Although DDPG shows an enhancement of 7.57% in performance with feature engineering, its performance is poor compared with SAC and PPO. This is because humanoid has a high dimensional continuous action space, and DDPG often struggles in high-dimensional action spaces compared to other algorithms like SAC and PPO. So this is the reason that the reward of DDPG is smaller than SAC and PPO.

5.2.2 TD3 and A2C Performance

Conversely, TD3 and A2C register a decline in performance when augmented with feature engineering, attributed to several key factors:

1. **Overfitting and Complexity:** The complexity introduced by additional features might lead to overfitting, where TD3 and A2C fail to generalize from overly detailed or irrelevant signals, resulting in decreased performance in varied or long-term scenarios.
2. **Exploration Inefficiency:** Enhanced features may adversely affect the exploration strategies critical for TD3 and A2C, hindering their ability to explore beneficial action spaces effectively due to increased dimensionality or misleading gradients.

3. **Bias-Variance Tradeoff:** The integration of more observational data can potentially reduce bias but increase variance if the added features do not consistently align with successful outcomes, complicating policy updates and stability.
4. **Algorithmic Sensitivity:** The intrinsic mechanisms of TD3 and A2C might be less compatible with the types of modifications introduced by feature engineering. For example, TD3's efforts to mitigate overestimation bias might be compromised by noisy or variable features, whereas A2C might struggle with abrupt changes in policy space due to volatile features.

In conclusion, while feature engineering holds potential in enhancing reinforcement learning algorithms, its implementation in DDPG, A2C, and TD3 underscores the need for careful selection and customization of features to match the specific requirements and sensitivities of each algorithm. This analysis highlights the importance of further research and tailored experimentation to optimize the integration of feature engineering in complex environments, ensuring improvements in observation space translate into more effective decision-making and learning.

6 Future Work

6.1 PINN Integration

Physic-Informed Neural Network, or PINN, is what inspired this project. The PINN is basically a neural network that is aware of the laws of physics. It computes various partial derivatives such as gradients of a velocity flow or divergence of a fluid flow. Those derivatives can then be used to compute an additional loss term to further fine tune the model more efficiently. Our initial motive was to integrate such networks to reinforcement learning. However, due to time constraints, we had no choice but to cut it out in the end. In the future, we aim to integrate this concept into humanoid to make it more robust.

6.2 Hyperparameters Fine-Tuning

Fine-tuning hyperparameters is also something on which we had no time to test. This can be used to potentially alleviate the problem that our algorithm gets stuck in local minimums just like what happened in our experiments.

6.3 MEOW

Lastly, we are aware that our dear Professor Lee has published a paper "Maximum Entropy Reinforcement Learning via Energy-Based Normalizing Flow"[1]. In the paper, they utilize a method called Maximum-Entropy Reinforcement Learning using Energy-Based Normalizing Flows or MEOW as they call it. We believe integrating such algorithms with PINN can have substantial improvement in physic-related environments. And in the future, we will go in this direction.

7 Conclusion

This thesis explored various Deep Reinforcement Learning (DRL) algorithms—PPO, A2C, DDPG, TD3, and SAC—for controlling humanoid robots in the Mujoco simulation environment. The study compared these algorithms, highlighting the significant impact of feature engineering. Results showed that while PPO and SAC improved with feature engineering, TD3 and A2C performed worse, indicating the need for careful feature selection. Future work includes integrating Physic-Informed Neural Networks (PINNs), hyperparameter fine-tuning, and exploring Maximum-Entropy Reinforcement Learning using Energy-Based Normalizing Flows (MEOW). These efforts aim to enhance the robustness and effectiveness of DRL in humanoid robot control. This thesis provides valuable insights for advancing DRL research in complex control tasks.

References

[1] Chen-Hao Chao, Chien Feng, Wei-Fang Sun, Cheng-Kuang Lee, Simon See, and Chun-Yi Lee. Maximum entropy reinforcement learning via energy-based normalizing flow. *arXiv preprint arXiv:2405.13629*, 2024.

[2] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[3] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[4] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[5] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, and David Silver Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[6] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have the main claims made in the abstract and introduction accurately reflected upon the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitation we have both on time and algorithm suitability.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have showed how to implement our methodology and disclose all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All resources are avaliable on GitHub.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: They are thoroughly discussed in our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have compared several baselines and reported appropriate information about the statistical significance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have discussed the machines used in our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the code of ethics in every aspect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use our own assests.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide instructions on how to load our pre-trained model on GitHub.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.