PPO Agent for LifeSteps: a simulation based on human daily life decisions

Daniel Bernardi
MSc Computer Engineering
Alma Mater Studiorum - Bologna
daniel.bernardi@studio.unibo.it
daniel_bernardi@outlook.it

Abstract

This paper presents an implementation of a Reinforcement Learning agent based on a Proximal Policy Optimization algorithm. The agent plays on LifeSteps, a simple custom Gymnasium Environment that simulates some basic human daily decisions (and their developments on life), also presented and developed for this paper.

1 Background

1.1 Gymnasium

Gymnasium is an API for Reinforcement Learning. It provides many environments that can be used to train and evaluate RL Algorithms, including the support support for incorporating new custom environments into the local installation. It can be useful because the custom environment, once registered in the local database of environments, will be ready to use as any of the others, garanteeing a standard usage and the possibility to use Wrappers and Utilities provided by the Gymnasium library.

1.2 Proximal Policy Optimization

The implementation of a Proximal Policy Optimization algorithm is one of the central parts of this project, this section will provide some background about it. Since the algorithm can be implemented in various ways, further explanations will be based on the project's implementation.

The ideas behind PPO come from Policy Gradient methods and Trust Region methods.

Policy Gradient methods are based on the computation of the policy gradient that can be used as input for a stochastic gradient ascent algorithm. To make good use of the already existing libraries that perform automatic differentiation, this gradient estimator can be obtained by differentiating an objective function with those softwares.

TRPO (Trust Region Policy Optimization, Schulman et al. [2015]) introduced a constraint to the size of the policy update. Alternatively to the constraint, it's possible to insert a penalty into the objective function, but it turns out difficult, sometimes, to define the correct hyperparameters about the penalty for problems that can change over time during the learning.

PPO implements the idea of a constrained policy update of TRPO in a simple way, based on the following formulas:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t]$$

Preprint. Under review.

With $r_t(\theta)$ being the ratio between the new and the old policy:

$$r_t(\theta) = \frac{\pi_\theta(a_t, s_t)}{\pi_{\theta_{old}}(a_t, s_t)}$$

And $\hat{A}_t$ being the advantage function.

The advantage function implemented in PPO is a truncated version of Generalized Advantage Estimation [Schulman et al., 2018]

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1},$$
$$where\ \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

In conclusion, the objective to maximize proposed by PPO is:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

with $L_t^{VF}(\theta)$ being the mean-squared error loss between the estimated and the target value function, $S[\pi_\theta](s_t)$ being the entropy of the policy on a given state, and c_1,c_2 two coefficients (hyperparameters).

A more detailed description of the algorithm can be found on the original PPO paper. What has been written here will be discussed in the implementation section.

2 LifeSteps

LifeSteps is a single player game where the person (or the agent) needs to make decision about what action to take on each timestep. Each action has some sort of development into the life of the player, which has to play in a smart way to avoid losing the game. The game is considered won when the player is still alive at the end of the simulation, which endures for a given number of timesteps.

Since the environment works according to the Gymnasium API, it's characteristics will be explained based on the implementation.

There are two gamemodes for LifeSteps, one simple (the *standard* mode), one more difficult (called *monopoly* mode).

2.1 The Standard gamemode

Since the *monopoly* gamemode is an extension of the *standard*, all of the details written here will be true also for the next section.

The *life* of the player is the main information contained in every environment's observation. It's implemented as an array of three integers in the interval $[0, 100)$. Each of these integers encode the quality of a characteristic of the player's life: it's *financial resources*, it's *health* and it's *social connections* richness.

There is also another value, called *friends*, which tells if the player has at least a true friend. This value changes nothing in this gamemode, but in the *monopoly* gamemode it will be important.

The actions that the player can perform are three: go to work, do some sport, enjoy a social occasion.

When performing a specific action, the corresponding value of *life* will receive a bonus. The difficulty of this simulation is on making the right choices. The agent must choose an action at each timestep, but that action will increase the score of just one of the three *life* components. The other two components will decrease in a deterministic way. For example, doing sports for a timestep will give a 10 points bonus to the health value, but at the same time the money and social values will decrease of 5 points.

The critical values (in this gamemode) are health and money. If they reach the 0 value at one observation, the game ends and the player loses.

The **reward is given to the player only at the end of the simulation** and is calculated in this way:

$$reward = max\ ts - current\ ts - difficulty$$

Table 1: Gamemodes initial settings

	Standard	Monopoly
difficulty	$[0, 100)$	$[0, 100)$
money start	$[25, 34)$	$[25, 34)$
health start	$[25, 34)$	$[25, 34)$
social start	$[25, 34)$	$[25, 34)$
friends start	0	0
trouble prob. start		0.0
trouble deficit		$[0, 4], 10$

if the player didn't reach the end of the game, otherwise the reward is:

$$reward = max(difficulty - min(life[i]), 1), i = [0, 1, 2]$$

with i being the index for the values about money, health and sociality, and *difficulty* being an initialization parameter of the environment. **If the player wins the game, the reward is 1**.

2.2 The Monopoly gamemode

Just like in the popular table game Monopoly, in this gamemode sometimes the player may have to pick up a *Chance card* that may affect positively or negatively the course of the game. In the environment context, the *Chance card* is an event that can happen with probability p, which increases at every timesteps, until the event comes up and p goes to zero again. In the occurrence of the event, a deficit on the health or money statistic of the player is applied, complicating the game.

This possibility of encountering troubles is balanced by another game mechanic, which in the standard mode had no effects: the *friends* state of the player. This information is either True or False, and it can change only in one way, from False to True, when the *social* development of the player reaches 40 points. This means that if the player reaches, for example, 43 points on sociality, it will acquire a friend, which will be permanent even if the sociality score goes below 40. This is both a simplification of the game and also a method to diversify the game conditions.

The parameters that define the initial settings of the game are explained in Table 1. The difficulty is a parameter defined by the player at the creation of the environment, while the starting values are randomly sampled inside their range. Table 2 summarizes the gameplay behaviour based on the increment or decrease of state's values at each timestep. The increment for a given value is 0 only if the corrispondent action is not chosen. The trouble deficit on one of the life statistics goes from 0 to 4 if the player has a friend, 10 otherwise.

2.3 An example of a LifeSteps simulation

To better describe how the game works, in Figure 2 there is an example of the first steps of an episode rendered in text mode.

M, H and S are the three life statistics values: money, health, sociality. The F value describes if the player has some friends. The last column tells if some trouble happened during the last step, indicading also how many points were lost. If the player loses pennies, the deficit goes to the money statistic, if the player get's hurt, the deficit goes to its health.

Let's take into consideration the second and third line. The selected action is sociality, so:

- the M value decreases of 5 points ($\Leftrightarrow 0 - 5$);
- the H value decreases of 3 points ($\Leftrightarrow 0 - 3$);
- the S value increases of 9 points ($\Leftrightarrow 10 - 1$);

(0,0,10) is the increase on the state for that timestep (since sociality was selected), (-5,-3,-1) is the constant decrease of the state at each timestep.

In addition to that, the S value is now above the 40 threshold, which means that the player now has friends.

```
| Life ->   M: 34,   H: 25,   S: 30  -   F: alone.., I just started playing!
| Life ->   M: 29,   H: 22,   S: 39  -   F: alone..     |  Last Action: social      | all ok
| Life ->   M: 24,   H: 19,   S: 48  -   F: many!!!    |  Last Action: social      | all ok
| Life ->   M: 29,   H: 16,   S: 47  -   F: many!!!    |  Last Action: work        | all ok
| Life ->   M: 24,   H: 23,   S: 46  -   F: many!!!    |  Last Action: sport       | all ok
| Life ->   M: 29,   H: 20,   S: 45  -   F: many!!!    |  Last Action: work        | all ok
| Life ->   M: 34,   H: 17,   S: 44  -   F: many!!!    |  Last Action: work        | all ok
| Life ->   M: 29,   H: 24,   S: 43  -   F: many!!!    |  Last Action: sport       | all ok
| Life ->   M: 31,   H: 21,   S: 42  -   F: many!!!    |  Last Action: work        | ...you lost some pennies....but Leandro helped, -3 points loss
| Life ->   M: 26,   H: 28,   S: 41  -   F: many!!!    |  Last Action: sport       | all ok
| Life ->   M: 21,   H: 25,   S: 50  -   F: many!!!    |  Last Action: social      | all ok
| Life ->   M: 26,   H: 22,   S: 49  -   F: many!!!    |  Last Action: work        | all ok
| Life ->   M: 27,   H: 19,   S: 48  -   F: many!!!    |  Last Action: work        | ...you lost some pennies....but Leandro helped, -4 points loss
```

Figure 1: The start of an episode in monopoly mode

Table 2: Gamemodes increments/decrements per timestep

	Standard	Monopoly
money inc	$0, 10$	$0, 10$
health inc	$0, 10$	$0, 10$
social inc	$0, 10$	$0, 10$
money dec	-5	-5
health dec	-3	-3
social dec	-1	-1
trouble prob. inc		0.03

3 Implementation

This section will discuss everything related to the implementation of the algorithm, the architectures of the neural networks used in the project, and the structure of the project files and Jupyter Notebook.

3.1 Project files

The project is programmed entirely in python and the code is subdivided in more files to semplify its understanding. The main python files are:

- *agent.py*: implements the PPO Agent, incapsulates the deep learning architecture, and all the methods used for training and evaluating the player inside of any gymnasium environment.
- *memory.py*: creates the memory structures (numpy arrays) that store all the information needed for the calculation of the gradients, the advantages and the returns of each batch.
- *utils.py*: implements some utility functions such as the calculation of advantages and returns.

The folder *logs* contains (Tensorboard) logged information collected during the training of the models inside the notebook. The folder *gym_projects* contains all the file needed to register the LifeSteps environment into the local Gymnasium installation (the specific environment implementation can be found in *gym_projects/life_sim/envs/lifesteps.py*). The folder *models* contains the main checkpoints of the deep learning models.

The Jupyter Notebook with all the code about the training process is *final.ipynb*.

3.2 Notebook: final.ipynb

Everything is implemented in Python language, using extensively the Numpy library for the management of the data, and Tensorflow to implement the deep learning models and gradients calculation.

The reproducibility of the notebook is guaranteed at the Numpy and Tensorflow levels, and all the evaluation are performed on the same environments, initialed using the same seeds. CUDA behaviour is not guaranteed to be deterministic.

The notebook is subdivided into 4 main Steps:

- Step 1: training and evaluation in *standard* gamemode and difficulty 40;

- Step 2: fine-tuning and evaluation of previous' step models in *monopoly* gamemode and difficulty 40;
- Step 3: training and evaluation of new models in Step 2 settings;
- Step 4: training and evaluation of models with more units in Step 2 settings;

Further details can be found inside the notebook. The notebook was executed three times with different seeds. Each run can be seen in it's specific file called *final-seed{n}.ipynb* with n being the seed used for the run. It's advisable to read one of those notebooks instead of the main one.

The code was executed on a laptop with an Intel® Core™ i7-8565U CPU and an Nvidia MX250 2GB GPU.

3.3 PPO Actor-Critic

The Actor-Critic architecture was implemented using two distinct neural networks (=> no weights sharing) with the same topology, except for the output layer. There are two hidden dense layer with 32 units each and tanh activations, and output heads with linear activations. For the Step 4 the units were increased to 128 on both models.

The original paper of PPO specifies how the algorithm should be programmed, but many variants can actually be implemented: changing the initialization of the neural networks, or the way in which the returns and advantages are calculated. The notebook follows the original PPO characteristic with some further additions.

3.3.1 Training Loop

Gymnasium provides the possibility to create a vector of multiple environments. This characteristic was exploited in the notebook, creating an AsyncVec (asynchronous vector) of 32 LifeStep's environments to better exploit the multiple cores during training. Even if not strictly necessary for LifeSteps, this decision speeded up the training in a considerable way, particularly in the monopoly gamemode where many epochs of training were needed to approximate a good enough policy.

PPO algorithm works by optimizing the objective for each **minibatch**, which is a portion of the total **batch** of $timesteps * environments$ observations. The training loop iterates for a specified number of **updates**, which depends on the number of parallel environments and the maximum duration of the training (measured in timesteps). For each update, 128 actions are selected for each environment (serialized process w.r.t. the timesteps, parallelized process w.r.t. the environments), forming a batch of $128 * 32$ sets of informations stored in Numpy NDArrays (observations, rewards, terminations, truncations, state-values). Then, the rewards are normalized and the advantages and returns are calculated for the whole batch.

The batch gets shuffled and subdivided in minibatches, for each of those we perform the optimization step for a specified number of epochs.

Some characteristic of the implementation:

- The learning rate and the ϵ are linearly annealed from the starting value to 0 at the end of the training;
- Orthogonal initialization of the weights in the neural network ([Engstrom et al., 2020] and [Andrychowicz et al., 2021] for a more specific setup);
- Reward scaling to the $[-1, 1]$ range;
- Advantages normalization

The **advantages** are calculated for all the steps in the batch. The **returns** are the discounted sum of the actual rewards, with the exception of the last value, which is calculated using the state-value given by the neural network at the first observation outside the scope of the current batch.

The optimizer used is Adam, same as the original PPO paper. There was not a exhaustive search for the optimal training hyperparameters, which remain similar to the ones specified in the original paper.

Table 3: Training results

Model	GM	Units	LR	c_2	Seed	Updates	Evaluation[3]
Step 1	std	32	2.5e-4	0	14	350	**1.0**
					77	350	0.92
					39	350	1.0
Step 2	mon	32	1e-4	5e-3	14	900	-1.14
					77	2343	-2.04
					39	2343	**0.66**
Step 3	mon	32	2.5e-4	5e-3	14	2300	**0.72**
					77	1150	0.66
					39	3500	0.40
Step 4	mon	128	2.5e-4	5e-4	14	1750	**0.86**
					77	2300	0.86
					39	1250	-0.96
random-bs	std				14		-122.3
random-bs	mon				14		-126.1

3.3.2 Loss function

PPO is based on gradient ascent, but the popular machine learning frameworks that provides automatic differentiation are based on gradient descent methods. That's why the original PPO objective is changed and minimized for the **actor network**:

$$Loss_{actor} = -L_t^{CLIP}(\theta) - c_2 S[\pi_\theta](s_t)$$

The loss minimized on the **value network** is the mean-squared error between the returns and the approximated state-values.

4 Results

Tensorboard was used to log all the informations needed to analyze the 4 steps of training. Table 3 summarized the results, specifying some hyperparameters. The parameters that didn't change for all the training Steps are listed below[1]:

- The max number of timesteps for each episode was 100;
- Difficulty is always 40;
- Early stopping of the training was after 6 consecutive evaluations with reward > 0;
- $\gamma = 0.99$
- $\lambda = 0.95$
- $Epochs = 4$

In the Evaluation column of Table 3 there is the value of an evaluation step performed after each of the four trainings. The evaluation is always performed on 50 episodes generated by the same 50 seeds. The number of Updates refers to the lenght of the training at the time of the early stopping[2].

5 Discussion

The Deep Learning architecture solves easily the game in standard gamemode and difficulty 40 (and possibly we could even go to difficulty at 50).

[1] An exaustive search of the optimal hyperparameters was not performed and this is not an exhaustive list.

[2] Step 2 is the only one that starts from pre-trained neural networks, so there is need to considerate also the number of updates of Step 1.

[3] The evaluation is the mean reward on the same 50 episodes for all the runs

```
| Life ->   M: 34,   H: 25,   S: 30  -   F: alone.., I just started playing!
| Life ->   M: 29,   H: 22,   S: 39  -   F: alone..     |  Last Action: social      | all ok
| Life ->   M: 24,   H: 19,   S: 48  -   F: many!!!     |  Last Action: social      | all ok
| Life ->   M: 29,   H: 16,   S: 47  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 24,   H: 23,   S: 46  -   F: many!!!     |  Last Action: sport       | all ok
| Life ->   M: 29,   H: 20,   S: 45  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 34,   H: 17,   S: 44  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 29,   H: 24,   S: 43  -   F: many!!!     |  Last Action: sport       | all ok
| Life ->   M: 31,   H: 21,   S: 42  -   F: many!!!     |  Last Action: work        | ...you lost some pennies....but Leandro helped, -3 points loss
| Life ->   M: 26,   H: 28,   S: 41  -   F: many!!!     |  Last Action: sport       | all ok
| Life ->   M: 21,   H: 25,   S: 50  -   F: many!!!     |  Last Action: social      | all ok
| Life ->   M: 26,   H: 22,   S: 49  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 27,   H: 19,   S: 48  -   F: many!!!     |  Last Action: work        | ...you lost some pennies....but Leandro helped, -4 points loss
```

Figure 2: The start of an episode in monopoly mode

```
| Life ->   M: 45,   H: 32,   S: 70  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 50,   H: 29,   S: 69  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 45,   H: 36,   S: 68  -   F: many!!!     |  Last Action: sport       | all ok
| Life ->   M: 50,   H: 31,   S: 67  -   F: many!!!     |  Last Action: work        | ...got hurt while cooking...but Dudu helped, -2 points loss
| Life ->   M: 45,   H: 38,   S: 66  -   F: many!!!     |  Last Action: sport       | all ok
| Life ->   M: 40,   H: 35,   S: 75  -   F: many!!!     |  Last Action: social      | all ok
| Life ->   M: 45,   H: 32,   S: 74  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 40,   H: 39,   S: 73  -   F: many!!!     |  Last Action: sport       | all ok
| Life ->   M: 45,   H: 36,   S: 72  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 50,   H: 33,   S: 71  -   F: many!!!     |  Last Action: work        | all ok
| Life ->   M: 45,   H: 40,   S: 70  -   F: many!!!     |  Last Action: sport       | all ok

--! Game finished with reward 1 !--
```

Figure 3: The end of an episode in monopoly mode

Step 2 needs some discussion. Its goal was to understand if the pre-trained networks could reduce the time of training needed to play the game in a different gamemode, similarly to what can be done in supervised learning. The results suggest that Transfer Learning in this particular case is not ideal, since the Evaluation score was good, but not excellent, and the number of update steps to obtain it was almost the same needed in Step 3. This result actually isn't that much of a surprise. Since the state is encoded into a 4 values array of integers, intuitively the features that the networks need to learn are actually very simple. On the other end, the policy must be much different because without a starting boost on the social score (to obtain a friend) the game becomes almost impossible, thanks to the Chance cards which give big deficits to the player's score.

There is not a lot to say about Step 3 and Step 4. Both the networks seem to solve the problem almost perfectly.

5.1 A deeper analysis

During the evaluation steps, after each training loop, for each model the data about percentages of choosen actions (on the evaluation episodes) was collected. In a very simplicistic way it describes, along with the episodes examples in the appendix, the behaviour of the agent. It can be seen that, in the monopoly gamemode, an higher selection of social actions is needed to win the game. This data is described in Table 4.

A standard behaviour for the agent in monopoly mode is to improve the social score as first move, as seen in Figure 2.

On the other hand, the policy does something strange at the end of a monopoly episode. As can be seen in Figure 3, the player has a very high social score. The actual social score that it needed to win the game (in that difficulty) was 30. It doesn't change much, since the presence of a friend that would reduce the score deficit of maximum 4. But anyway, the agent it's expected to behave in a more safe and robust way on higher difficulty, otherwise the final reward would go down. This lack of robustness in some case, with a policy that doesn't behave well at the end of the episodes, can be seen in Figure 4, which is an episode played by the agent trained in Step 4.

A more robust behaviour can be observed at the end of a standard episode, played by the Agent trained during Step 1. Recalling the standard gamemode rules, at difficulty 40, with 100 timesteps of episode length: the game is lost if, at the end of a fully played episode, either money or health scores are below 40; the game is lost if the episode is played for less than 100 timesteps; There are no Chance cards.

```
| Life ->   M: 41,   H: 23,   S: 58  -   F: many!!!    |  Last Action: work       | all ok
| Life ->   M: 46,   H: 20,   S: 57  -   F: many!!!    |  Last Action: work       | all ok
| Life ->   M: 41,   H: 27,   S: 56  -   F: many!!!    |  Last Action: sport      | all ok
| Life ->   M: 36,   H: 33,   S: 55  -   F: many!!!    |  Last Action: sport      | ...got hurt while cooking...but Dudu helped, -1 points loss
| Life ->   M: 41,   H: 30,   S: 54  -   F: many!!!    |  Last Action: work       | all ok
| Life ->   M: 46,   H: 27,   S: 53  -   F: many!!!    |  Last Action: work       | all ok
| Life ->   M: 41,   H: 34,   S: 52  -   F: many!!!    |  Last Action: sport      | all ok
| Life ->   M: 36,   H: 31,   S: 61  -   F: many!!!    |  Last Action: social     | all ok
| Life ->   M: 41,   H: 28,   S: 60  -   F: many!!!    |  Last Action: work       | all ok
| Life ->   M: 36,   H: 31,   S: 59  -   F: many!!!    |  Last Action: sport      | ...got hurt while cooking...but Dudu helped, -4 points loss
| Life ->   M: 41,   H: 28,   S: 58  -   F: many!!!    |  Last Action: work       | all ok

--! Game finished with reward -2 !--
```

Figure 4: Policy is not perfect

```
| Life ->   M: 65,   H: 51,   S: 49  -   F: many!!!    |  Last Action: work
| Life ->   M: 60,   H: 58,   S: 48  -   F: many!!!    |  Last Action: sport
| Life ->   M: 65,   H: 55,   S: 47  -   F: many!!!    |  Last Action: work
| Life ->   M: 60,   H: 62,   S: 46  -   F: many!!!    |  Last Action: sport
| Life ->   M: 65,   H: 59,   S: 45  -   F: many!!!    |  Last Action: work
| Life ->   M: 60,   H: 56,   S: 54  -   F: many!!!    |  Last Action: social
| Life ->   M: 65,   H: 53,   S: 53  -   F: many!!!    |  Last Action: work
| Life ->   M: 60,   H: 60,   S: 52  -   F: many!!!    |  Last Action: sport
| Life ->   M: 65,   H: 57,   S: 51  -   F: many!!!    |  Last Action: work
| Life ->   M: 70,   H: 54,   S: 50  -   F: many!!!    |  Last Action: work
| Life ->   M: 65,   H: 61,   S: 49  -   F: many!!!    |  Last Action: sport
| Life ->   M: 60,   H: 68,   S: 48  -   F: many!!!    |  Last Action: sport
| Life ->   M: 65,   H: 65,   S: 47  -   F: many!!!    |  Last Action: work
```

Figure 5: The end of an episode in standard mode

In Figure 5 the text shows that the Agents behaviour in standard mode is more robust, since the scores are more balanced (based on the decrease-per-timestep). In fact, money would go below 40 after 5 timesteps without working, health after 5 timesteps without doing sports, and sociality after 7 timesteps without doing social actions.

6 Conclusions

7 Useful Links and some references to useful arcticles

Project code (github)

PPO implementation details

CleanRL PPO implementation

8 Submission of papers to NeurIPS 2022

Please read the instructions below carefully and follow them faithfully.

8.1 Style

Papers to be submitted to NeurIPS 2022 must be prepared according to the instructions presented here. Papers may only be up to **nine** pages long, including figures. Additional pages *containing only*

Table 4: Agent's behaviour during the evaluation steps.

	work (%)	sport (%)	sociality (%)
Step 1	54.0	33.9	12.1
Step 2	52.8	32.6	14.6
Step 3	53.0	33.0	14.0
Step 4	53.4	32.4	14.2

acknowledgments and references are allowed. Papers that exceed the page limit will not be reviewed, or in any other way considered for presentation at the conference.

The margins in 2022 are the same as those in 2007, which allow for $\sim$15% more words in the paper compared to earlier years.

Authors are required to use the NeurIPS LaTeX style files obtainable at the NeurIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

8.2 Retrieval of style files

The style files for NeurIPS and other conference information are available on the World Wide Web at

`http://www.neurips.cc/`

The file `neurips_2022.pdf` contains these instructions and illustrates the various formatting requirements your NeurIPS paper must satisfy.

The only supported style file for NeurIPS 2022 is `neurips_2022.sty`, rewritten for LaTeX 2_ε. **Previous style files for LaTeX 2.09, Microsoft Word, and RTF are no longer supported!**

The LaTeX style file contains three optional arguments: `final`, which creates a camera-ready copy, `preprint`, which creates a preprint for submission to, e.g., arXiv, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

Preprint option If you wish to post a preprint of your work online, e.g., on arXiv, using the NeurIPS style, please use the `preprint` option. This will create a nonanonymized version of your work with the text "Preprint. Work in progress." in the footer. This version may be distributed as you see fit. Please **do not** use the `final` option, which should **only** be used for papers accepted to NeurIPS.

At submission time, please omit the `final` and `preprint` options. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `neurips_2022.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections 9, 10, and 11 below.

9 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by ½ line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow ¼ inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section 11 regarding figures, tables, acknowledgments, and references.

10 Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

10.1 Headings: second level

Second-level headings should be in 10-point type.

10.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

11 Citations, figures, tables, references

These instructions apply to everyone.

11.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated…

If you wish to load the `natbib` package with options, you may add the following before loading the `neurips_2022` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{neurips_2022}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

11.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[4] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

[4] Sample of the first footnote.

Table 5: Sample table title

	Part	
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Note that footnotes are properly typeset *after* punctuation marks.[5]

11.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

11.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 5.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 5.

12 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

13 Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NeurIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`

[5] As in this example.

- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

  ```
  \newcommand{\RR}{I\!\!R} %real numbers
  \newcommand{\Nat}{I\!\!N} %natural numbers
  \newcommand{\CC}{I\!\!\!\!C} %complex numbers
  ```

  Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

13.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: `https://neurips.cc/Conferences/2022/PaperInformation/FundingDisclosure`.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to autmoatically hide this section in the anonymized submission.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section 9.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [N/A]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

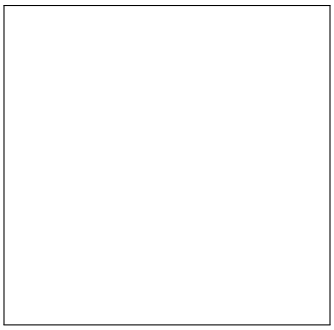

Figure 6: Example A.

References

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=nIAxjsniDzg`.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=r1etN1rtPB`.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL `http://arxiv.org/abs/1502.05477`.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

A Appendix

Example A

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.