DRLViz: Understanding Deep Reinforcement Learning Memory for Navigation Tasks (Supplementary Material)

1 Introduction

This supplementary material provides the following contents to support our manuscript:(1) details on the training conditions and hyper-parameters used for the agent in section 6 of the manuscript. (2) Two additional interactions, one to display several episodes at the time, and the other to display disjoint time intervals. (3) We also used DRLViz with the *two colors* navigation problem, and (4) applied other memory reductions. Finally, we conclude by details on how to reproduce the training of the agent used in section 6.

2 Merge Interactions

Due to space limitations we did not introduce all implemented DRLViz interactions. In particular the ones related to merging multiple episodes in order to get an overview of the behavior of multiple agents. In this section we use the *health gathering supreme* scenario with a memory of 512 dimensions.

Merge episodes. DRLViz allows to merge the data of loaded episodes and display it as a single episode which can still be filtered and sorted. Such operation provide an overview of the agent behavior across episodes, grasp patterns in memory elements activations, compare them in episodes. Fig 2 shows DRLViz loaded with 10 merged episodes which corresponds to 5439 time-steps. A first observation is that such an increase in time-steps to display slows down DRLViz to the point that the t-SNE projection is no longer usable. However, the memory timeline which displays 5439 time-steps × 512 elements i.e slightly less than 2.8 millions values is still usable. In such view, one can observe that at each beginning of episodes the hidden states elements are overall inactive (gray) for around 2 steps, which allows to distinguish each episodes. In addition, some elements are constantly inactive during all the episodes, while some are constantly actives. This suggests that the agent has redundancy in its memory and thus that it may be reduced.

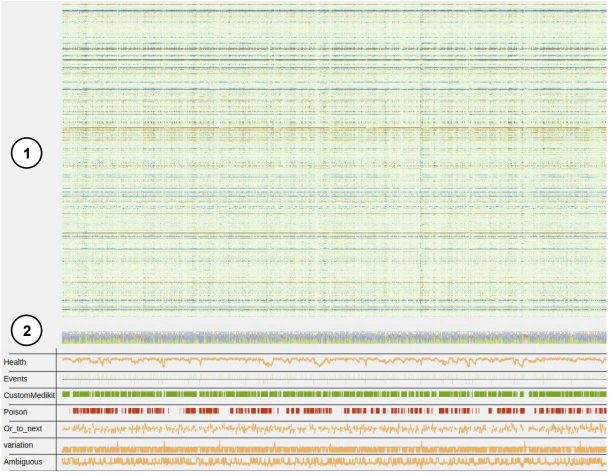

Figure 1: DRLViz loaded with 10 merged episodes (i. e. 5439 time-steps), despite being harder to read, the memory timeline ① can still be used to grasp patterns of elements activations across episodes. However, the derived metrics timeline ② is unsable.

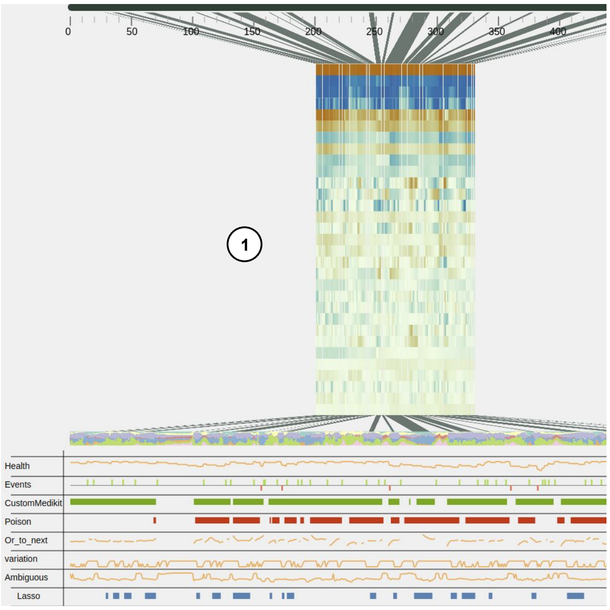

Figure 2: The merge of multiple time-steps intervals ① corresponding to a selection on the t-SNE projection.

Merge intervals. Some selections such as brushing trajectories, or lasso on t-SNE projections, may yield time intervals related to different time-steps depending on the selections. DRLViz displays those intervals in the memory timeline aligned with the time-step t in which they occurred. Therefore, each time intervals may be scattered across the memory timeline and any patterns in the hidden states activations may be hard to grasp. To tackle such issue, we implemented an interaction which allows to compact those intervals and put them together. In order to preserve the relation between the time steps of those intervals and still compact them. A link is created to on both the top of the memory to tight them to time steps, and the bottom of the memory to link them with the derived metrics timeline.

3 Arnold and Death-match

As mentioned in the manuscript, we used DRLViz with the the model and scenrio from [2]. The data displayed in Fig. 3, is extrated from a pretrained model available on this github repository[1]. We adapted this code to extract the data required by DRLViz without altering the model itself and thus without needing to retrain it. To launch experiments, we used the following command line: "./run.sh shotgun –n_bots 7".

The limit of DRLViz while using this scenario is that it cannot display all actions available to model (29) in this case. In to use action distribution views, we decidaed to sum them based on their direction (e. g., fire + forward will be summed with move camera + forward).

4 Two Colors Navigation Problem

As mentioned in the manuscript, DRLViz is applicable to many ViZDoom navigation tasks. The following presents one of them and insights discovered. The agents used on this tasks are similar to the ones used in section 6 of our manuscript and can be found in GitHub [1].

The **two colors navigation problem** is similar to the one introduced in the scenarios (Sec. 6) of our manuscript, as the agent must gather items in order to survive in an environment which reduces its health h periodically. The environment contains walls and 2 types of item: armors and a pillar. The agent must gather armors depending on their colors. The color it must gather is determined by the color of the pillar. Therefore, the agent must first discover the pillar, remember its color and then gather armors with matching colors. The pillar's color, positions as well as the armors positions are different at each episodes.

Figure 4 shows the result of visualizing *two colors* from which one can have the following observations:

[1] https://github.com/glample/Arnold

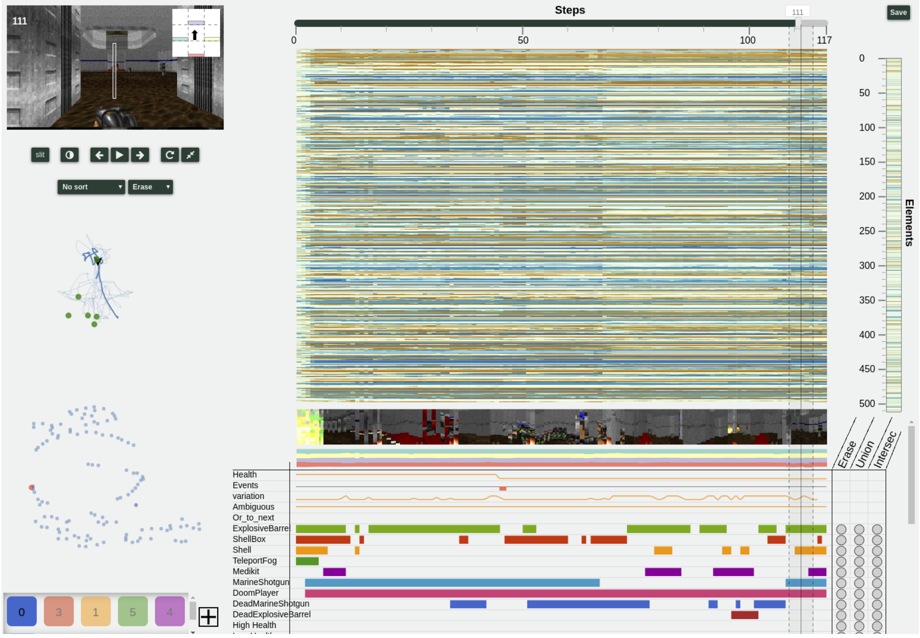

Figure 3: Visualizing the arnold model on the doom shotgun death-match with DRLViz. The memory timeline, and the t-SNE projection ① reflects the agent looking for the pillar before gathering armors.

- The t-SNE projection had distinct clusters regrouping steps during which the agent was looking for the pillar (Fig.4 ①). This suggest that the agent may learned the role of the pillar. This is also observed with using playback and derived metrics timeline in which the agent avoid most items before the pillar enters its field of view.
- While overviewing the memory timeline, a first observation is that around a third of the hidden state elements had constant activations during the episode. This phenomenon is observed across episodes, therefore one can the hypothesis that those elements may not be used by the agent.
- In the action probabilities temporal stacked graph, one can observe that the agent only used *forward+right* and *forward+left* actions (Fig.4 ②). This results in the VARIATION derived metric being constant during the whole episode. One can also observe that the agent's decisions AMBIGUITY metric is constantly low during the episode, which can be interpreted as the agent being confident.
- In the memory timeline, some hidden state elements were inactive (gray) until the pillar entered the agent's field of view, and then remained active for the rest of the episode (Fig.4 ③). One can interpret those elements as flags that determine if the agent has seen the pillar. In addition, those flags

had different activations when the pillar was red or green. This suggest that those elements encodes which color the agent should focus on.

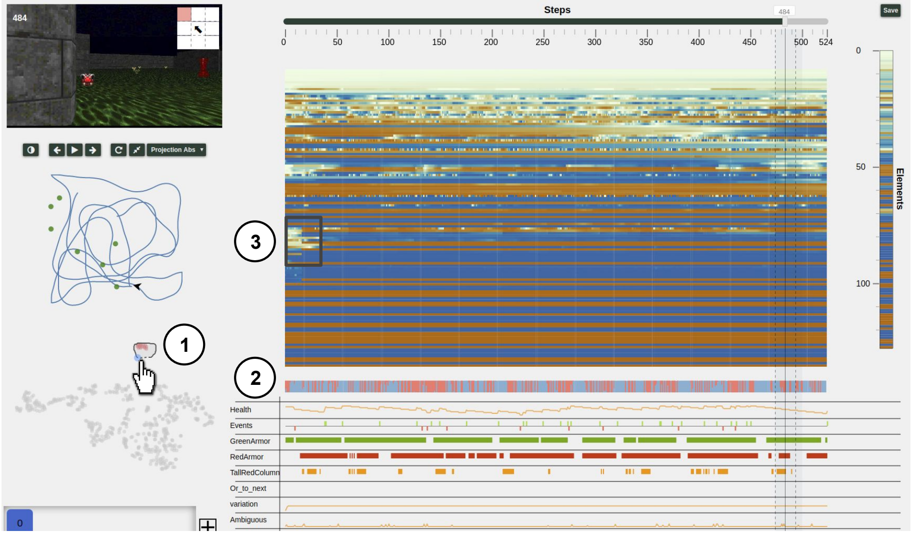

Figure 4: Visualizing the two colors navigation problem with DRLViz.The memory timeline, and the t-SNE projection ① reflects the agent looking for the pillar before gathering armors (also represented in ③. The actions distribution graph ② shows that the agent only choose between two actions.

5 Reproducing our Results

To reproduce the result we demonstrated in Section 8 of the manuscript, one can download the trained model on our GitHub repository[2], and launch the script `reduce.py`. Such script will generate one episode of the agent exploring the *health gathering scenario*. The number of episodes to generate, as well as the memory mask can be modified as parameters. We also provide an online prototype[3] which can be used on the navigation problems we presented.

References

[1] Edward Beeching. edbeeching/3d_control_deep_rl, July 2019. original-date: 2019-03-19T07:25:59Z.

[2] https://github.com/sical/drlviz

[3] https://sical.github.io/drlviz/

[2] Guillaume Lample and Devendra Singh Chaplot. Playing FPS games with deep reinforcement learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.