FeUdal Networks for Hierarchical Reinforcement Learning
Supplementary material

Alexander Sasha Vezhnevets [1] **Simon Osindero** [1] **Tom Schaul** [1] **Nicolas Heess** [1] **Max Jaderberg** [1] **David Silver** [1] **Koray Kavukcuoglu** [1]

The supplementary materials contain this document and several videos of our FuN agent playing. The first section elaborates on how value functions are computed for policy gradients. The rest of the document presents additional experimental results that could not fit into the main text, but we feel are helpful in understanding FuN.

1. Learning value functions

We use value function estimators to stabilise policy gradients using advantage actor critic method (Mnih et al., 2016). There are 3 value functions to estimate: 1) for return on intrinsic reward for worker 2) for return on environment reward for worker 3) for return on environment reward for manager. Functions 2 and 3 can be the same, but we use different discount for the manager (we discount less). We compute 2 and 3 via a linear layer from Manager's dLSTM state. We compute 1 using a two-layer MLP which takes the pooled goal vector g_t and state representation z_t.

2. Qualitative analysis on Seaquest

To qualitatively inspect sub-policies learnt by the Worker we use the following procedure: first, we record goals emitted by Manager during the play; we then sample one of them and provide it as a constant input to the Worker for the duration of an episode and record its behaviour. This allows us to qualitatively inspect what kind of sub-policies emerge. Figure 4 plots sub-policies learnt on the seaquest game. Notice how different options correspond to rough spatial positions or manoeuvres for the agent's submarine – for instance sub-policy 3 corresponds to swimming up for air.

3. Intrinsic motivation weight α

This section look at the impact of the weight α, which regulates the relative weight of intrinsic reward (if $\alpha = 0$ then intrinsic reward is not used). We train agents with learning rate and entropy penalty fixed to $10^{-3.5}$ and only vary α between $[0, 1]$. Figure 6 shows scatter plots of agents final score vs α hyper-parameter. Notice a clear correlation between the score and high value of α on gravitar and amidar; however on other games the optimal value of α can be less than to 1.

4. Temporal resolution ablations

An important feature of FuN is the ability of the Manager to operate at a low temporal resolution. This is achieved through dilation in the LSTM and through the prediction horizon c. To investigate their influence we use two baselines: i) the Manager uses a vanilla LSTM with no dilation; ii) FuN with Manager's prediction horizon $c = 1$. Figure 5 presents the results. The non-dilated LSTM fails catastrophically, most likely overwhelmed by the recurrent gradient. Reducing the horizon c to 1 did hurt the performance, although interestingly less so than other ablations. It seems that even at high temporal resolution Manager captures certain properties of the underlying MDP and communicate them down to Worker in a helpful way. This confirms that by learning in two separate formulations FuN is able to capture richer structural properties of the environment and thus train faster.

5. Dilate LSTM agent baseline

One of innovations this paper presents is dLSTM design for a Recurrent network. In principle, it could alone be used in an agent on top of a CNN, without the rest of FuN structures. We evaluate such an agent as an additional baseline. We use the same hyper-parameters as for FuN – BPTT=400, discount $= 0.99$, learning rate sampled in the interval $LogUniform(10^{-4.5}, 10^{-3.5})$, entropy penalty $LogUniform(10^{-4}, 10^{-3})$. Figure 7 plots the learning curves for FuN, LSTM and dLSTM agents. dLSTM generally underperforms both LSTM and FuN. The power of dLSTM is in the ability to operate at lower temporal resolution, which is useful in the Manager, but not so much on it's own. Notice that plots here stop 100 epochs, unlike in the main text. The only reason for this is the finite nature of computational resources before the submission deadline.

6. DeepLab environments illustration

Figure 1 presents an illustration of the non-match and T-maze domains.

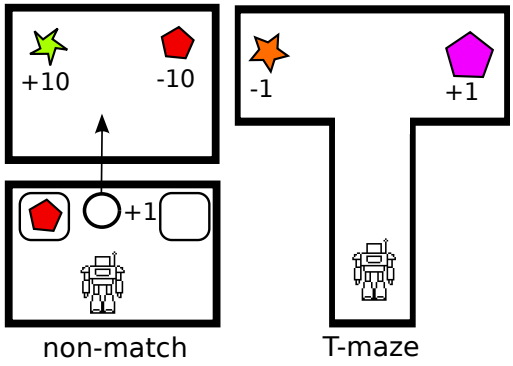

Figure 1. Schematic illustration of t-maze and non-match domains

7. Comparison to Option-Critic

Here we present plots for FuN on 2 ATARI games that Option-Critic (Bacon et al., 2017) was evaluated on, but which are not included in the experiments in the main text: Zaxxon and Asterix. Figure 2 presents our results. We took the approximate performance of Option-Critic from the original paper – 8000 for Asterix and 6000 for Zaxxon. Plots in the original paper also suggest that score stagnates around these levels, notice that our score keeps going up.

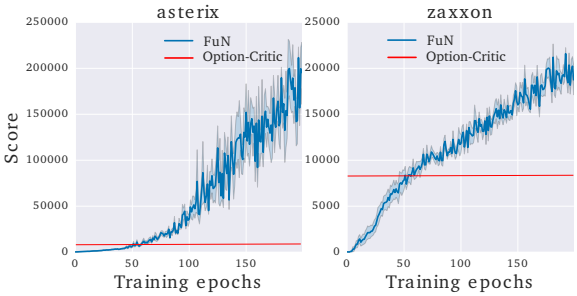

Figure 2. Comparison to Option-Critic on Zaxxon and Asterix. Score for Option-Critic is taken from the original paper

8. Videos

This supplementary material contains videos of FuN agent playing several ATARI games and DeeMind Lab games. All ATARI videos are in original resolution. Videos from Lab environment are from agent's perspective in the resolution at which the agent perceives. There is an additional video for water maze game, which is made from a top-down camera to allow a better view of agent's policy. The agent was not trained with this as input, but had an egocentric view as in other videos. We have also in-painted a visualisation of embedded goal vectors w into the videos in the top left corner (figure 3).

Figure 3. Embedded goal vectors w is in-painted into the videos (not part of agents observation). Each pixel corresponds to an element of the vector with intensity being proportional to the value. Dark pixels correspond to high negative values, white to high positive and grey is zero. Notice how the embedded goals are diverse, yet smoothly varying.

References

Bacon, Pierre-Luc, Precup, Doina, and Harb, Jean. The option-critic architecture. In *AAAI*, 2017.

Mnih, Volodymyr, Badia, Adria Puigdomenech, Mirza, Mehdi, Graves, Alex, Lillicrap, Timothy P, Harley, Tim, Silver, David, and Kavukcuoglu, Koray. Asynchronous methods for deep reinforcement learning. *ICML*, 2016.

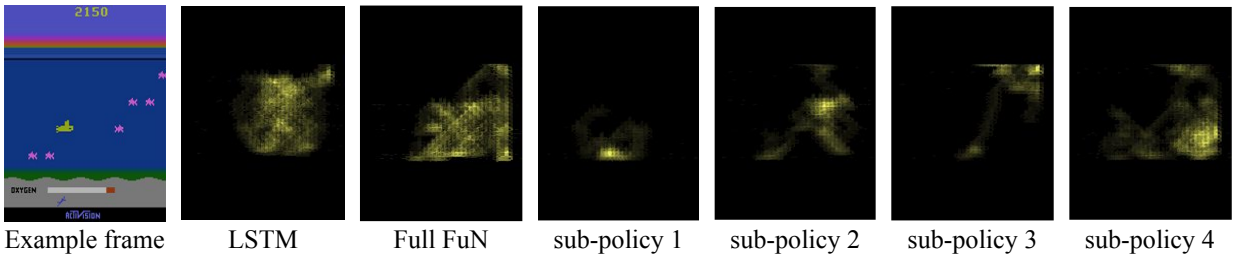

Figure 4. Visualisation of sub-policies learnt on sea quest game. We sample a random goal and feed it as a constant conditioning for the Worker and record its behaviour. We filter out only the image of the ship and average the frames, acquiring the heat-map of agents spatial location. From left to right: i) an example frame of the game ii) policy learnt by LSTM baseline iii) full policy learnt by FuN followed by set of different sub-policies. Notice how sub-policies are concentrated around different areas of the playable space. Sub-policy 3 is used to swim up for oxygen.

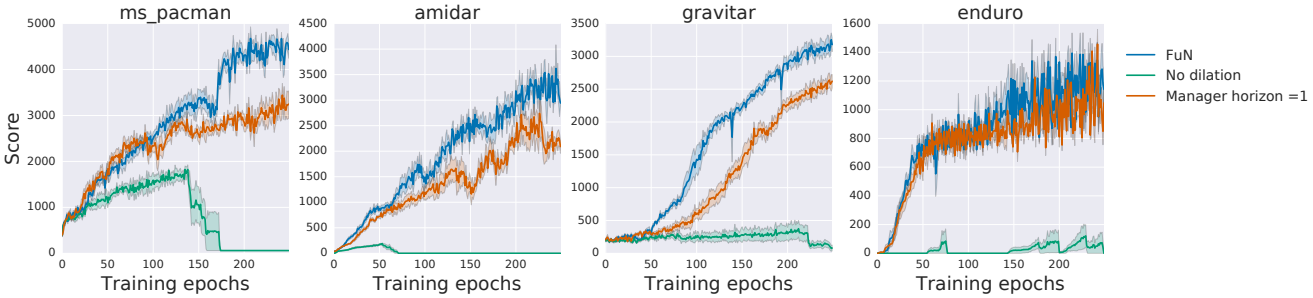

Figure 5. Learning curves for ablations of FuN that investigate influence of dLSTM in the Manager and Managers prediction horizon c. No dilation – FuN trained with a regular LSTM in the Manager; Manager horizon =1 – FuN trained with $c = 1$.

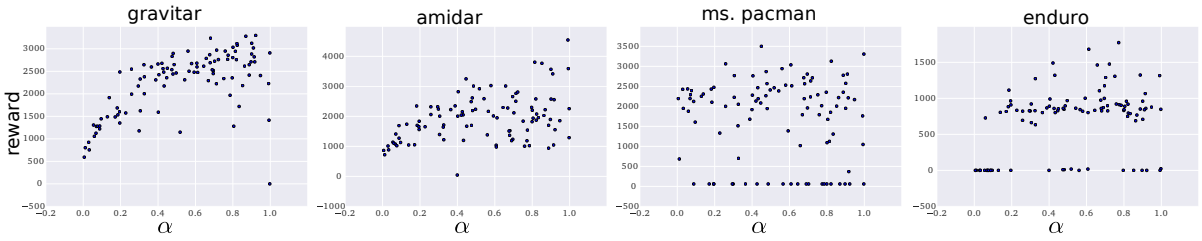

Figure 6. Scatter plot of agents reward after 200 epochs vs intrinsic reward weight α.

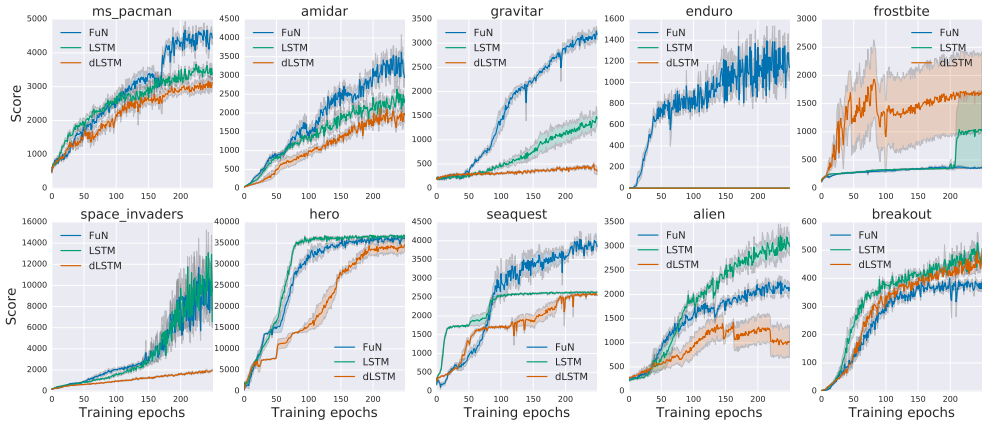

Figure 7. Learning curves for dLSTM based agent with LSTM and FuN for comparison.