Imagination Augmented Agents for Deep Reinforcement Learning

Angela Denninger, Felix Schober, Florian Klemt, Max-Philipp Schrader

Imagination Augmented Agent Architecture

- **Adopted implementation** of the paper *Imagination Augmented Agents for Deep Reinforcement Learning* by DeepMind [1] (I2A)
- We were not able to replicate the specifc results of DeepMind using their proposed design choices, as they used a custom implementation of Atari games and we used **OpenAI Gym as an Atari environment**. [1,4]
- Combines **model based and model free** Reinforcement Learning Architectures
- Different **Imagination Rollouts explore an imagined future** of available actions

Full I2A Architecture

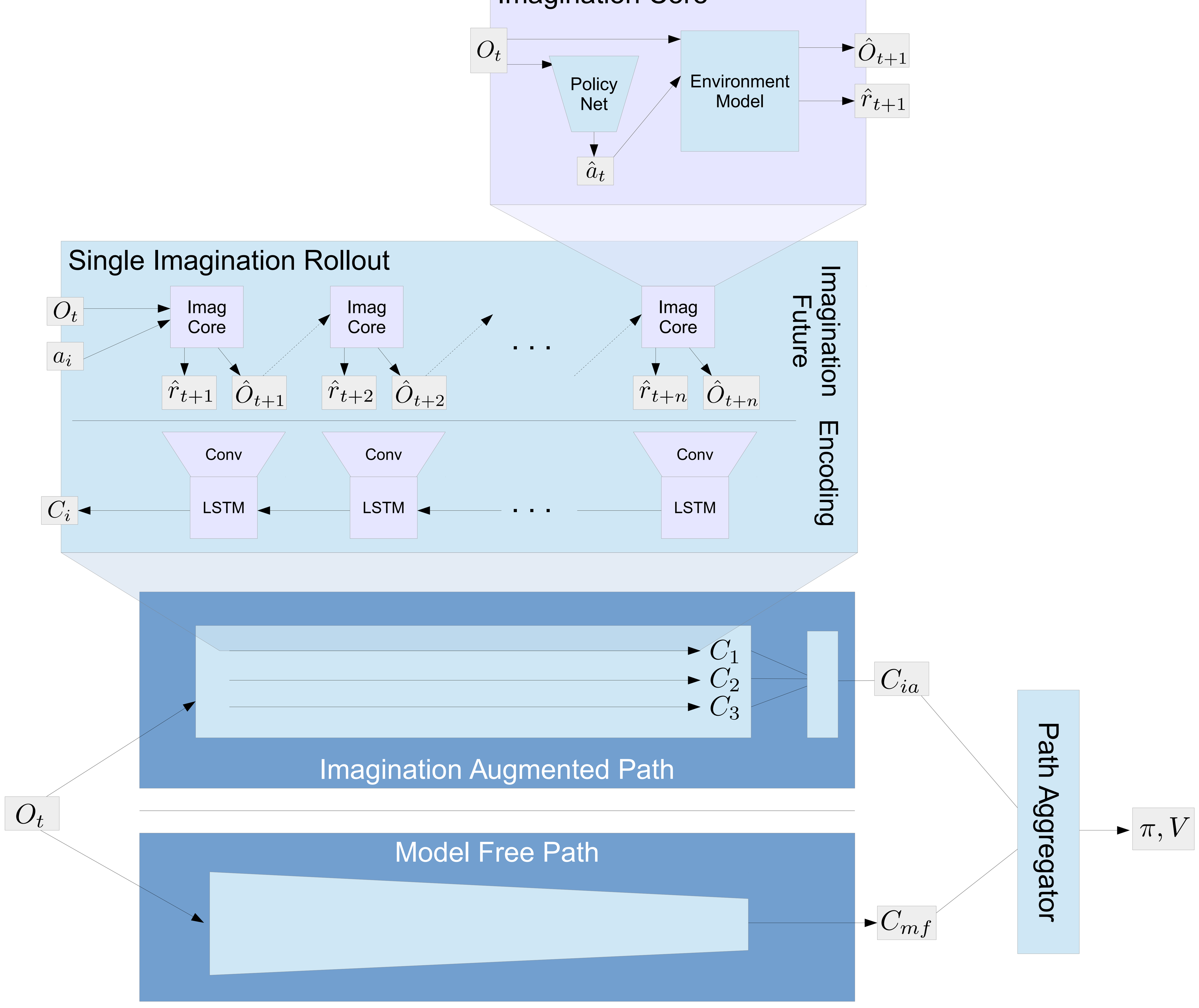

Fig. 1: Full Imagination Augmented Agent – Architecture

Environment Model

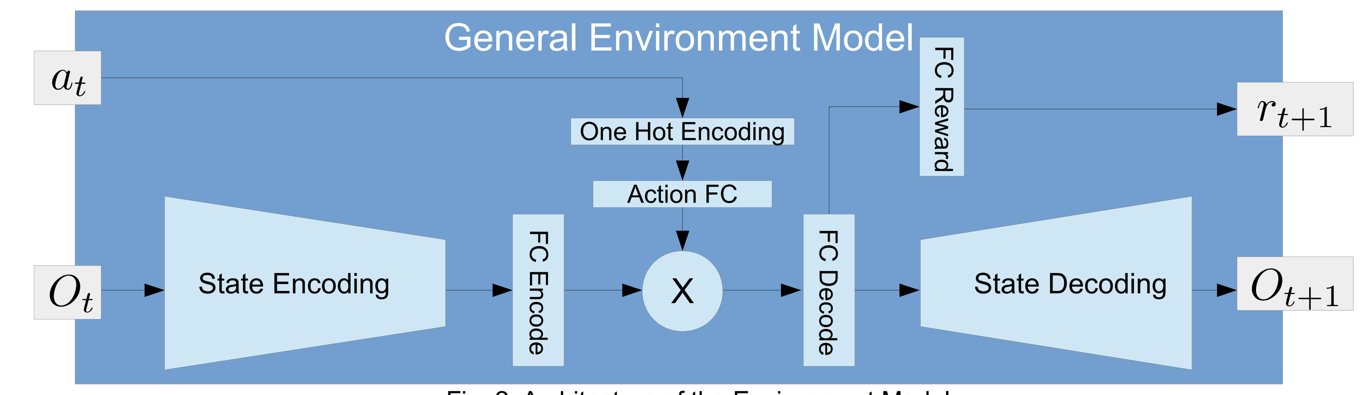

Fig. 2: Architecture of the Environment Model

- **… predicts the next state and reward**
- The Environment Model differs from the ones proposed in the paper due to different environment state sizes
- We used the architecture proposed in [3]. The model takes **one hot encoded actions and the current state** as input to **predict the next state and the reward**. We tried the current state in different settings. The last three recorded frames, combined to three channels, worked best.
- In the latent space the Action FC and the State Encoding are combined by element wise multiplication
- For training we found the **Negative Log Likelihood Loss** in combination with **Adam** and a **learning rate of 10^{-4}** generate the best results for Pong, MsPacMan, and Breakout

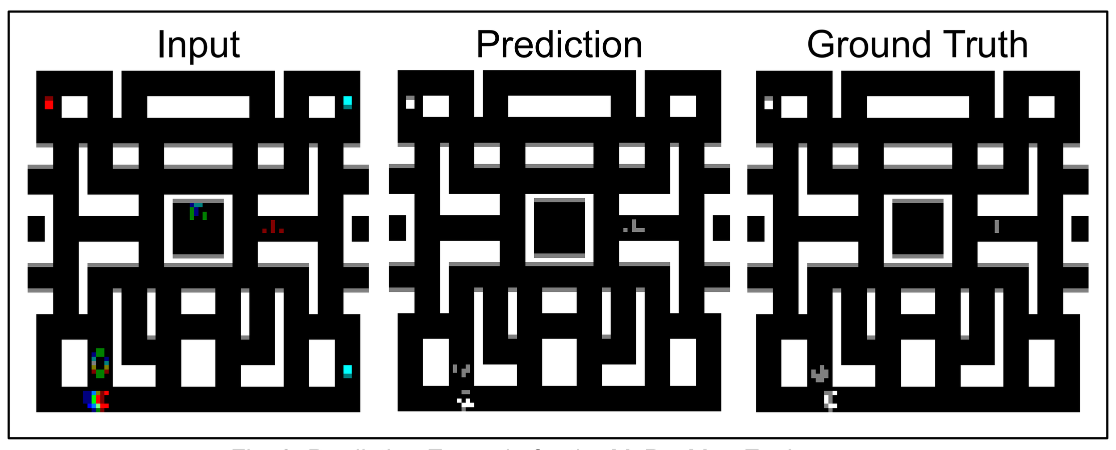

Fig. 3: Prediction Example for the MsPacMan Environment

Imagination Augmented Path (IAP)

- … uses rollouts to imagine the best future action
- The IAP consists of **one Imagination Rollout** for all available **actions** a_i
- All Imagination Rollout outputs C_i will be aggregated by concatenating them to C_{im}

Single Rollout

- … evaluates **how a selected action performs in the future**
- Imagines the future by chaining multiple imagination cores. At the beginning it takes the current state as well as a start action. Finally the predicted state $\hat{O}_{t+1}$ gets passed into the next Imagination Core.
- After performing n rollout steps a **convolutional LSTM encodes the result** of the Imagination Rollout

Imagination Core (IC)

- … predicts the next state based on an internal selected action $\hat{a}_t$
- Consists of a Policy Net and a pretrained Environment Model
- The policy net predicts the next action to perform. As proposed by [1] we used architecture from the A3C paper [2] as our policy net
- Output: predicted reward $\hat{r}_{t+1}$ and the next state $\hat{O}_{t+1}$

Model Free Path

- … provides the network with an option to **deal with insufficient future predictions**
- Uses the **convolutional layers of A3C** model free architecture [2] but does not include the fully connected layer

Path Aggregator

- … calculates based on both paths a policy π and value V
- First, the output of the paths C_{im} and C_{mf} gets concatenated
- This then is followed by a fully connected net which outputs the policy and the value

Evaluation

- **Scott Reed**, DeepMind, 01/30/2018: "Oh… **That's a very ambitious project**" and "What you want to use real PacMan?"
- For **training** the I2A network we used the **asynchronous method** proposed in DeepMinds A3C paper [2]
- Due to computational resources, we were not able to train a very strong model. **DeepMind trained** their I2A model **for 10^9 Atari environment steps**.
- Nevertheless we were able to **train a working version for Pong**, but we can **not proof the advantages** of the I2A-Architecture **with such a simple environment**
- We saw a changing duration length, which is based on learning and it's ability to win faster in the end.

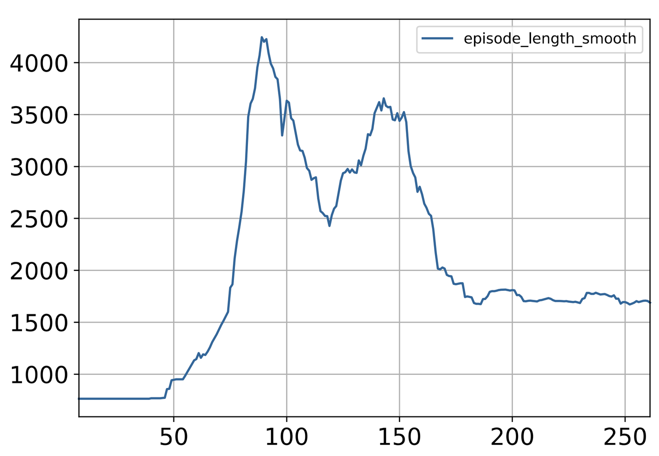

Fig. 4: Pong episodes played

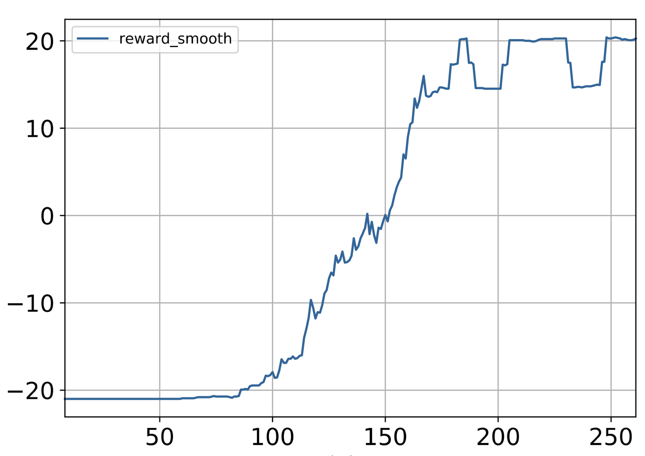

Fig. 5: Reward archived per episode of Pong

Summary

- As described above, we were not able to train a sufficiently strong model, but we were able to **implement a working I2A model,** which is able to **learn and play Atari Games**
- Our code will be published as **Open-Source on Github** [5] after the class

Literature

[1] Racanière, Sébastien, et al. "Imagination-Augmented Agents for Deep Reinforcement Learning." Advances in Neural Information Processing Systems. 2017.
[2] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International Conference on Machine Learning. 2016.
[3] Leibfried, Felix, Nate Kushman, and Katja Hofmann. "A deep learning approach for joint video frame and reward prediction in atari games." arXiv preprint arXiv:1611.07078 (2016).
[4] Brockman, Greg, et al. "Openai gym." arXiv preprint arXiv:1606.01540 (2016).
[5] https://github.com/mpSchrader/I2A-for-Deep-RL

Get in Touch

Angela Denninger	angela.denninger@tum.de
Felix Schober	felix.schober@tum.de
Florian Klemt	florian.klemt@tum.de
Max-Philipp Schrader	m.schrader@tum.de