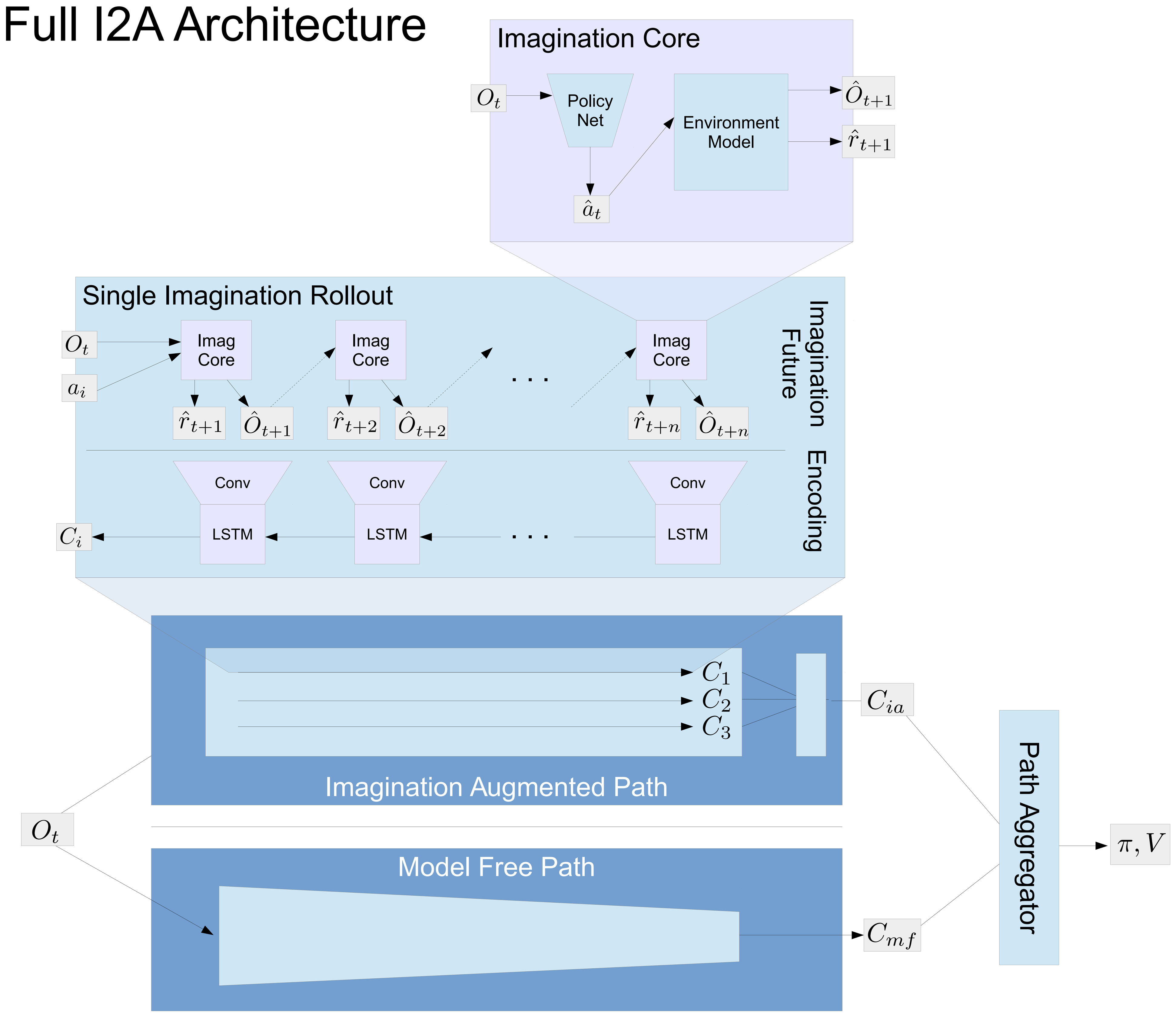
Imagination Augmented Agents for Deep Reinforcement Learning

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

Imagination Augment Agent Architecture

- **Adopted implementation** of the paper *Imagination Augmented Agents for Deep Reinforcement Learning* by DeepMind [1]
- We were not able to replicate the results of DeepMind using their proposed design choices, as they used a custom implementation of Atari games and we used **OpenAI Gym as 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



Path Aggregator

- The aggregator first concatenates the output of both paths
- Followed by a fully connected net which outputs the policy and the value

Model Free Path

- Uses the **convolutional layers of A3C** model free architecture[2] and does not include the fully connected layer

Imagination Augmented Path

- In the Imagination Augmented Path there is **one Imagination Rollout** for all available **action** a_i
- All Imagination Rollout outputs C_i will be aggregated by concatenating them to C_{im}

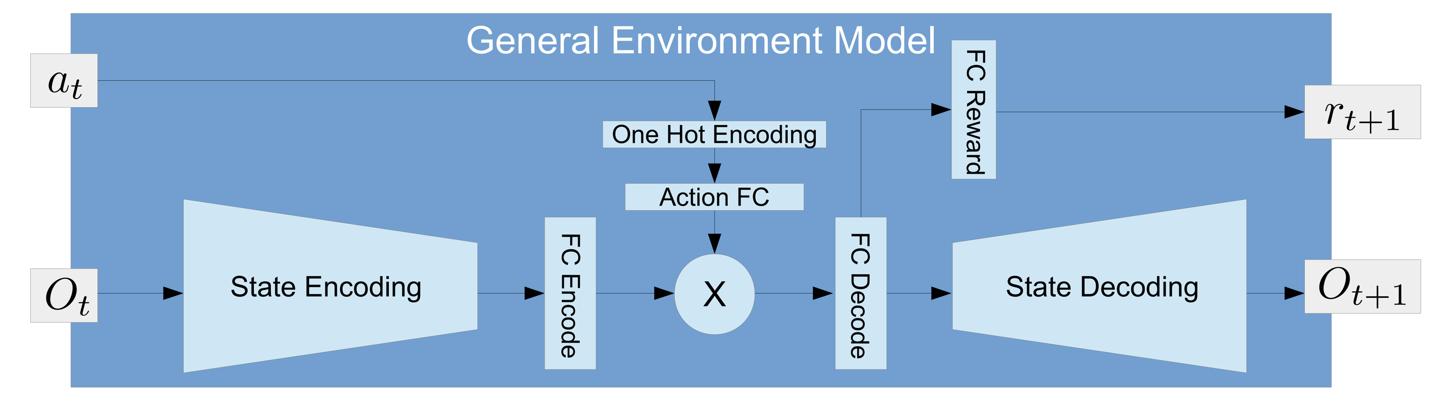
Single Imagination Rollout

- Each 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)

- **…** consists of a Policy Net and an Environment Model
- Output: predicted reward $\hat{r}_{t+1}$ and the next state $\hat{O}_{t+1}$
- The policy net predicts the next action to perform, the policy net is a simple policy net. We like in [1] used A3C [2].

Environment Model



- The Environment Model differs from the ones proposed in the paper due to different state sizes
- We used the architecture proposed in [3]. The model takes one hot encoded **actions and the current frame as input** and **predicts next state and reward**
- In the latent space the Action FC and the State Encoding are combine by element wise multiplication
- For training we found **bla bla** to generate the best results

BILDER VON INPUT OUTPUT UND GROUND TRUTH

Evaluation

- 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. Which has technical been not feasible for us.
- **Hier Graphen??? Welche Graphen Willen wir hier mit welcher Erklärung einfügen?**

Summary

- **Scott Reed**, DeepMind, 01/30/2017: “Oh… **That’s a very ambitions project**” and “What you want to use real PacMan?”
- Due to computational resources, 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 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 mschr@mit.edu