Latent Space I2A for Deep RL

Florian Klemt

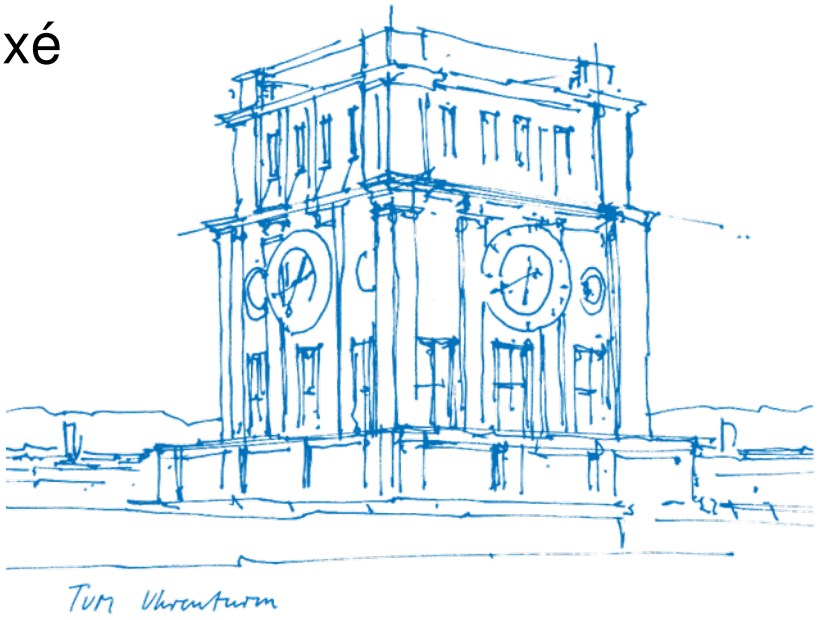
Technische Universität München

Dynamic Vision and Learning Group

Supervisor: Prof. Laura Leal-Taixé

Advisor: Tim Meinhardt

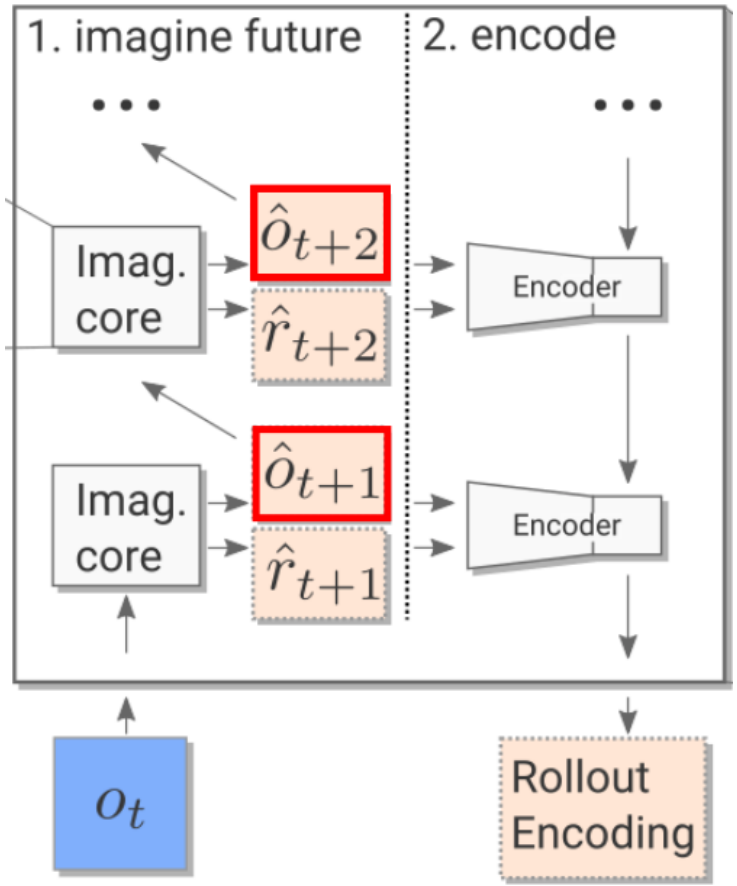
München, 13. August 2018



Problems of classical I2A

- Not scalable to large observations – rollouts are slow
- Not scalable to large action spaces – one rollout per action
- Environment model has problems predicting far into the future

Classical I2A rollout



$$o_t \rightarrow s_t \rightarrow \hat{o}_{t+1}$$

$$\hat{o}_{t+1} \rightarrow s_{t+1} \rightarrow \hat{o}_{t+2}$$

...

Solution to large observations

- Compress observation into a latent representation
- Compute rollout based on compressed state not on images

$o_t \rightarrow s_t$

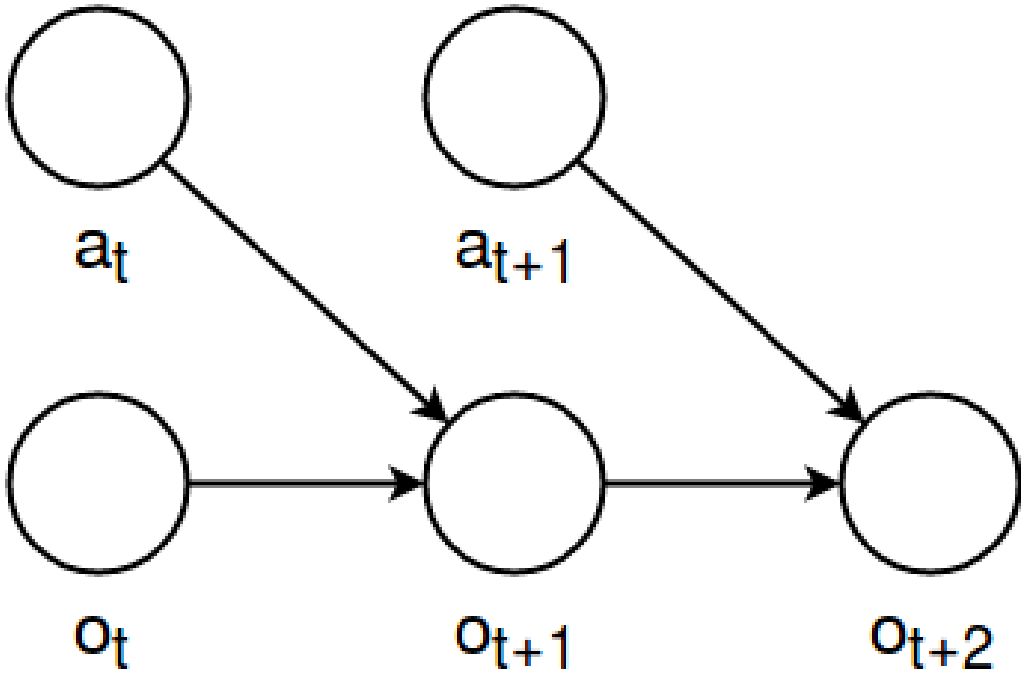
$s_t \rightarrow s_{t+1}$

Relevant Papers

- Imagination-Augmented Agents for Deep Reinforcement Learning, Weber et al.(2017)

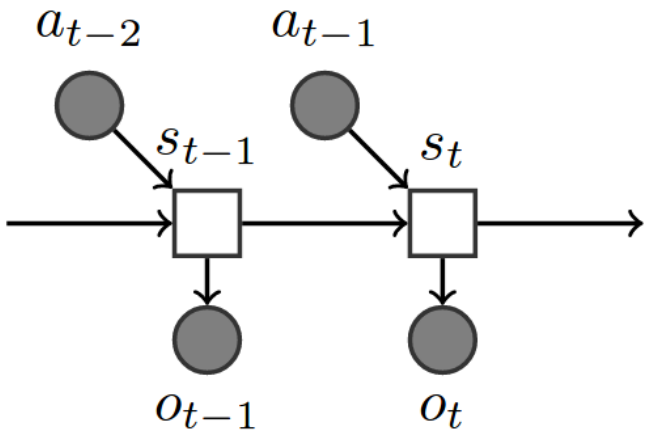
- Learning and Querying Fast Generative Models for Reinforcement Learning, Buesing et al.(2018)

Classical I2A rollout



dSSM-DET

deterministic State Space Model - Deterministic



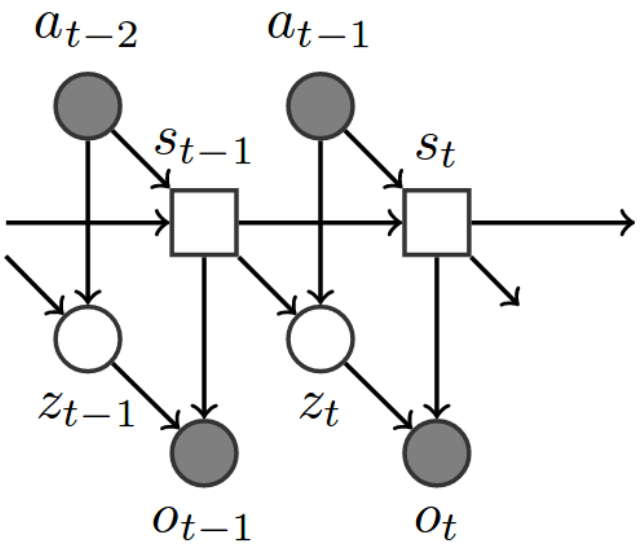
s_t: compact latent representation of observation at time t

a_t: action at time t

o_t: predicted output

dSSM-VAE

deterministic State Space Model - Variational Autoencoder



z_t: latent variable

sampled from gaussian

distribution given by learnable

mean and standard deviation

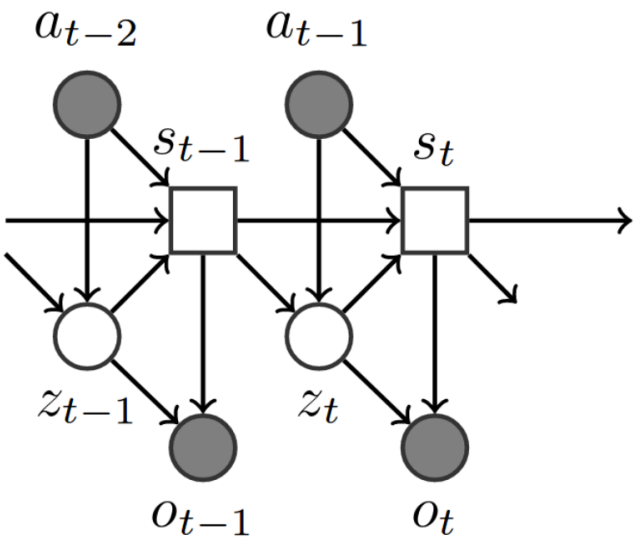
(based on s_{t-1} and a_{t-1})

Decoder depends on z_t

sSSM

stochastic State Space Model

dSSM variants can be seen as simplifications of sSSM

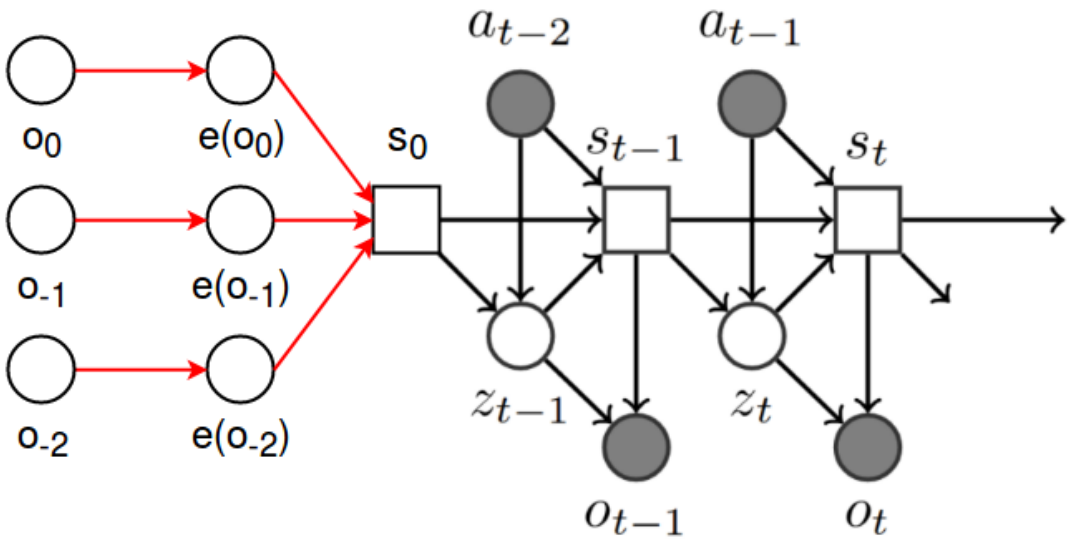


- Now the **transition and the decoder** depend on the sampled latent variable
- Explicitly models uncertainty of state s_{t+1}

State Space Models

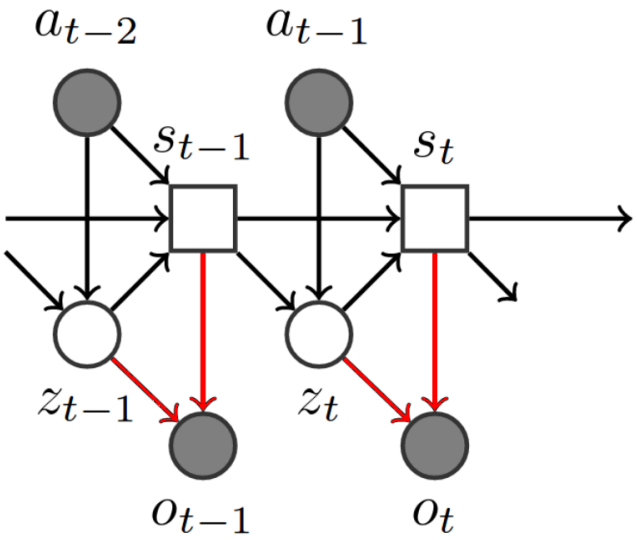
- dSSM-DET (deterministic State Space Model, deterministic decoder)
- dSSM-VAE (deterministic State Space Model, stochastic decoder)
- sSSM (stochastic State Space Model, stochastic decoder)

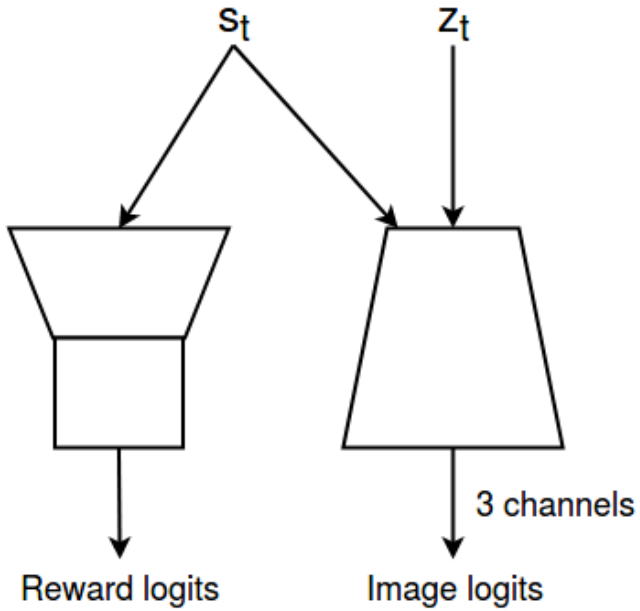
Environment Model Encoder



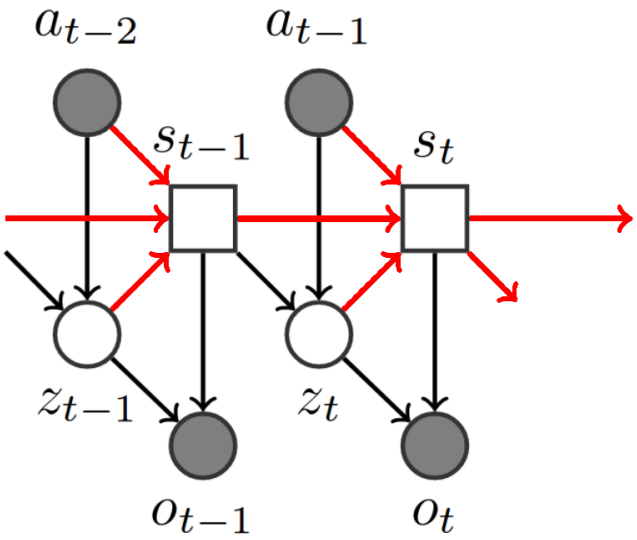
- encoding via space-to-depth operation and conv-stacks
- s_0 is based on the 3 previous encodings

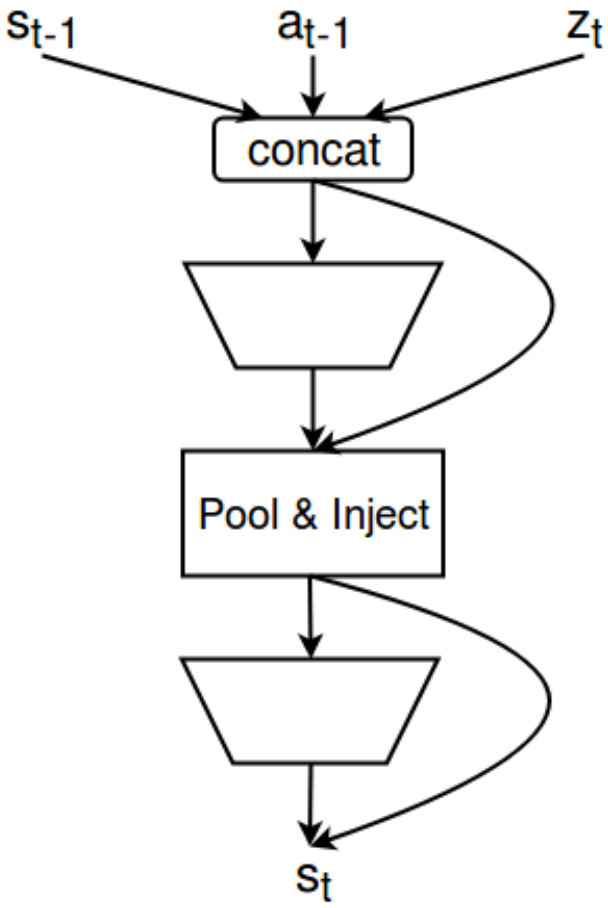
Environment Model Decoder





Environment Model State Transition





Training the dSSM-DET Model

Maximum Likelihood Estimation:

$$L(\theta) = \log p_\theta(o_{1:T} | a_{0:T-1}, \hat{o}_0) \tag{1}$$

$\hat{o}_0$: initial context

the encodings of the last 3 observations are used to generate s_0 in a learnable way

Training the sSSM Model

- We cannot directly evaluate $L(\theta)$ when using latent variables as it depends on the posterior $p(z_t|s_{t-1}, a_{t-1})$
- Approximate true posterior $p(z_t|s_{t-1}, a_{t-1})$ with approximated posterior $q(z_t|s_{t-1}, a_{t-1}, o_{t:T})$
- Use Kullback-Leibler divergence

$$\begin{aligned} & KL(q(z_t|s_{t-1}, a_{t-1}, o_{t:T})|p(z_t|s_{t-1}, a_{t-1})) \\ &= \int q(z_t|s_{t-1}, a_{t-1}) \log \frac{q(z_t|s_{t-1}, a_{t-1}, o_{t:T})}{p(z_t|s_{t-1}, a_{t-1})} dz \end{aligned} \tag{2}$$

- Reformulate as Evidence Lower Bound (ELBO)

Training the sSSM Model

Evidence Lower Bound:

$$ELBO_q(\theta) = \sum_{t=1}^{T} \mathbb{E}_q[\log p(o_t|s_t) + \log p(z_t|s_{t-1}, a_{t-1}) - \log q(z_t|st-1, a_{t-1}, o_t)] \quad (3)$$

ELBO = reconstruction loss + KL-Div between true and approximated posterior

https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Reconstruction loss

$p(o_t|s_t)$: Bernoulli Distribution

$\rightarrow$ Compute reconstruction loss with binary cross entropy

How can the image output be Bernoulli distributed?

- Interpret channels of decoder output scaled between 0 and 1 as probabilities of a Bernoulli distribution

Reward prediction

- Binary representation of the reward

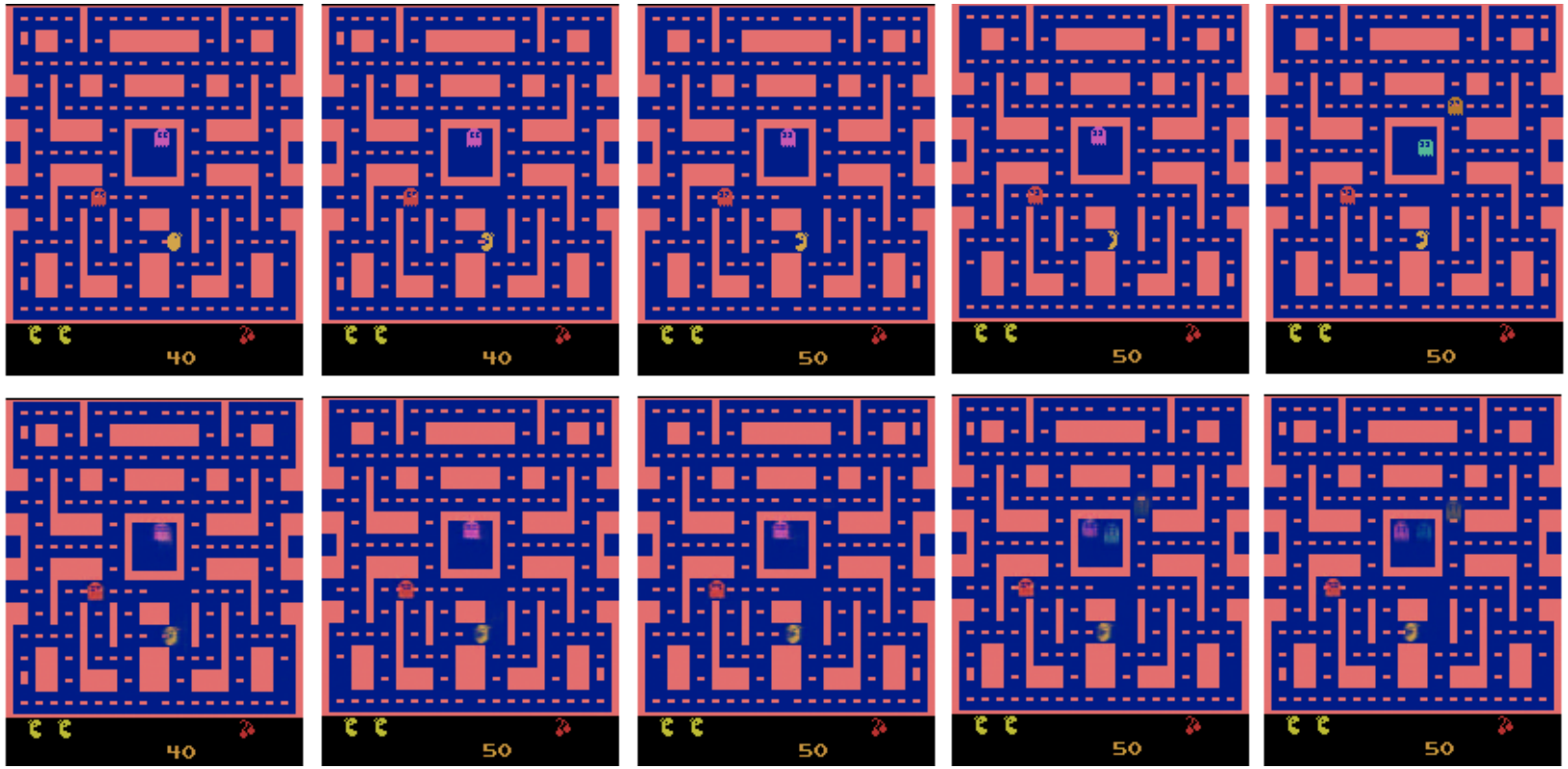
$$\sum_{n=0}^{N-1} b_{t,n} 2^n = \lfloor r_t \rfloor \tag{4}$$

- Bits are modeled as independent Bernoulli variables
- 2 additional bits: sign and zero indicator

Comparison of Environment Model architectures

- Stochastic architectures perform better at predicting the future
- I2A seems to have difficulties learning from the output of sSSM
- dSSM-DET performs best when combining it with I2A

dSSM-DET rollout prediction

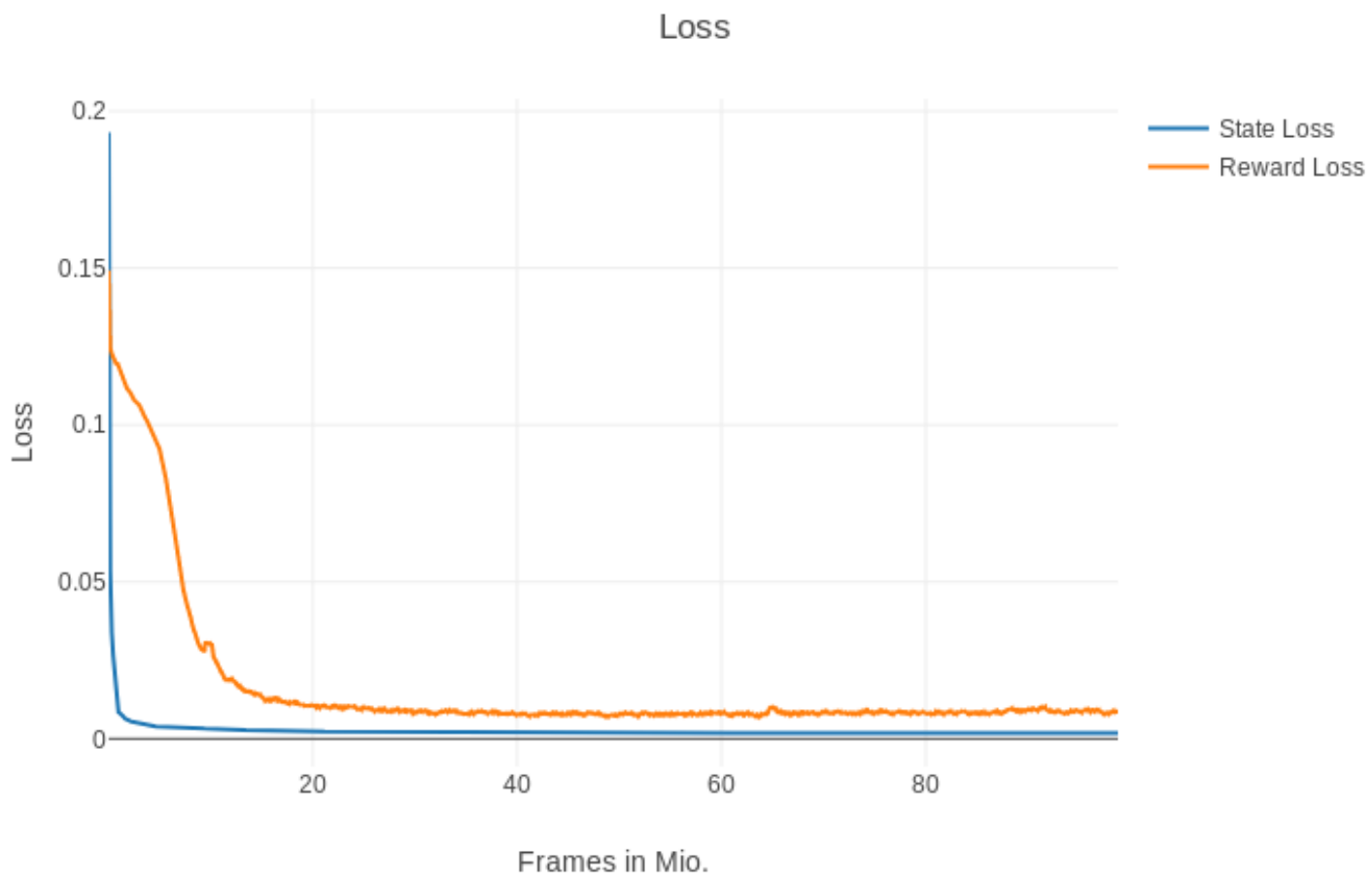


sSSM rollout prediction demo

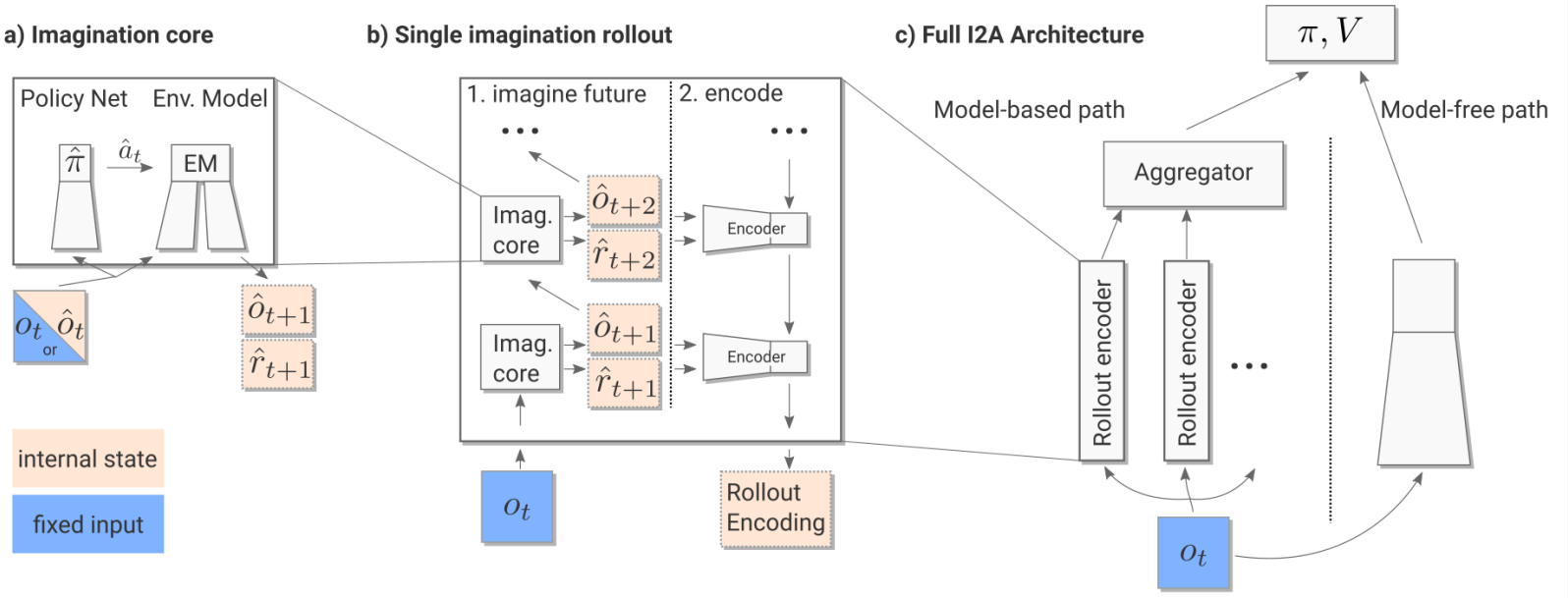
Why is MsPacman a hard prediction problem?

- Observation size: 160x200x3
- Sprites of pacman and ghosts are not static
- Blinking objects, essential objects are not seen in all frames
- Ghosts follow certain rules – the environment model has to learn them to predict ghost behaviour
- Some reward-types are very sparse

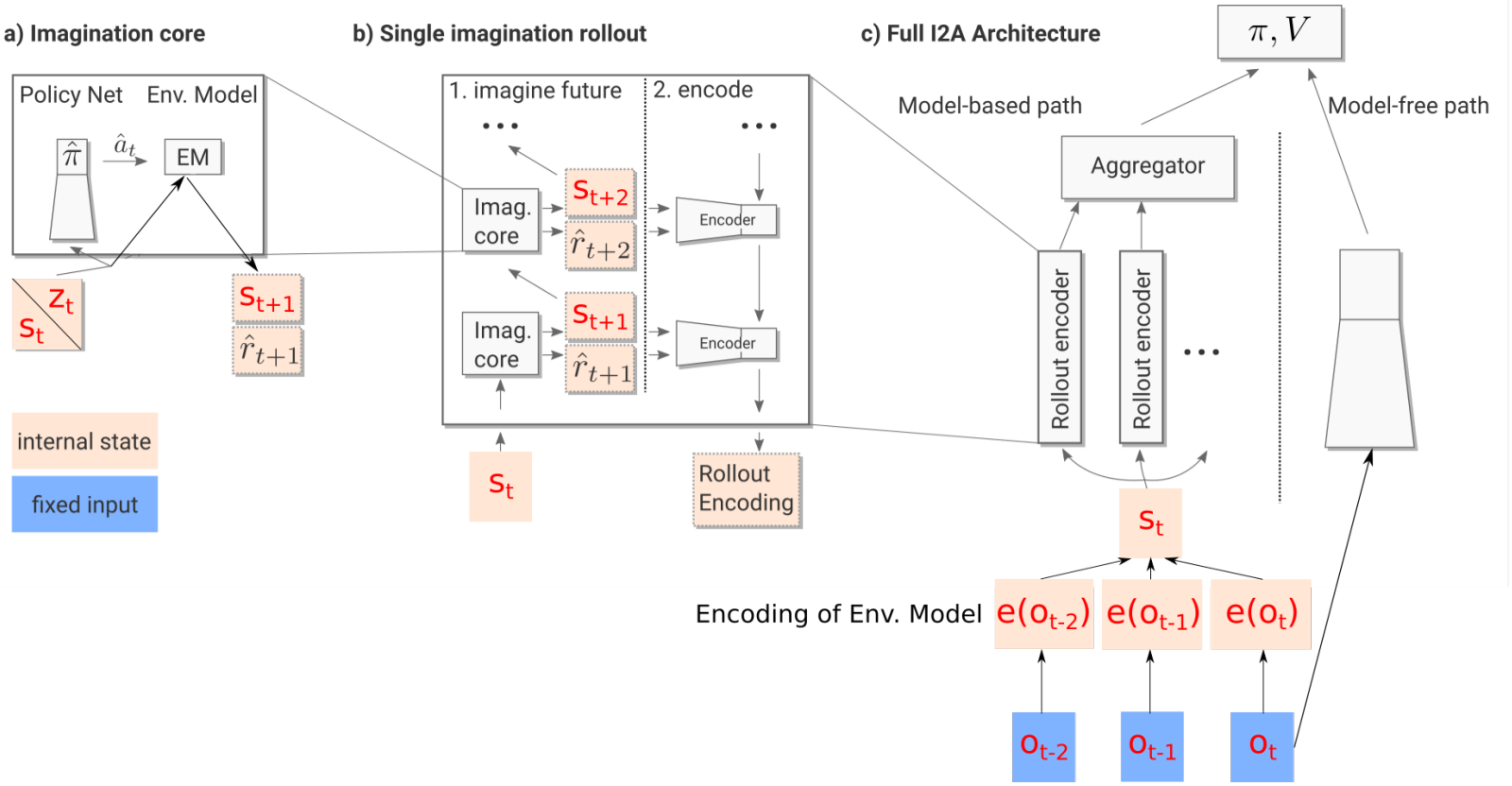
Environment Model training loss



Original I2A architecture



Modified I2A architecture



Why is MsPacman a hard reinforcement learning problem?

- The Agent needs to plan into the future $\rightarrow$ I2A
- The input is a raw image – the agent initially knows nothing about objects $\rightarrow$ Conv-Layers, Latent Space I2A
- Large observations $\rightarrow$ Latent Space I2A
- Dense rewards (eat), that do not help much in learning the way to get large (sparse) rewards fast (power pills, kill ghosts)

MsPacman

- Environment MsPacmanNoFrameskip-v0 from OpenAI Gym
- Original input size: (210, 160, 3) cropped to (200, 160, 3)
- Reward:

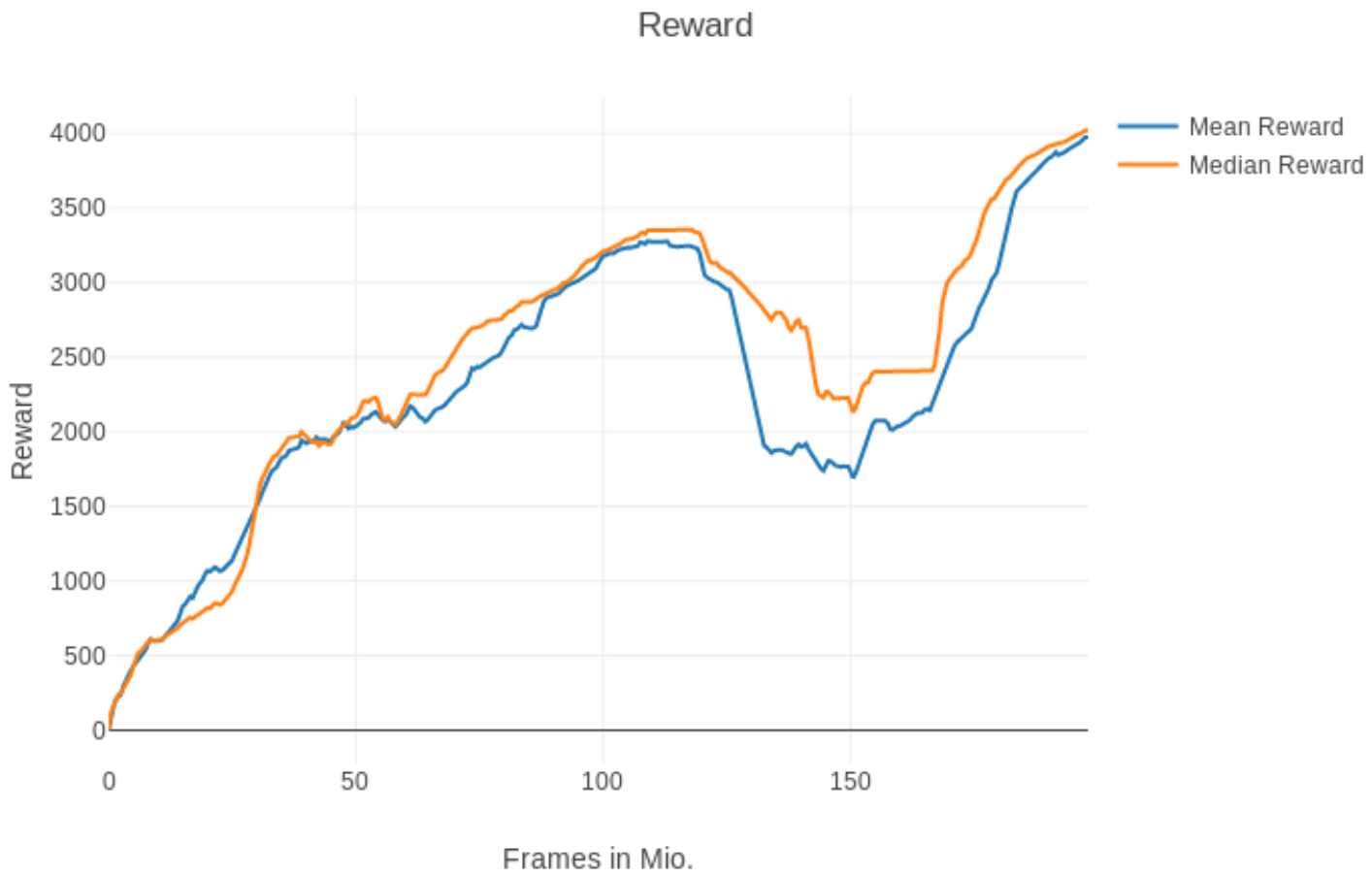
Eating food	10
Eating power pill	50
Eating ghost	200,400,800,1600
Killed by ghost	-100
Eating Fruits	100-5000

- Fruit rewards are extremely sparse and higher value fruits only appear in later levels.

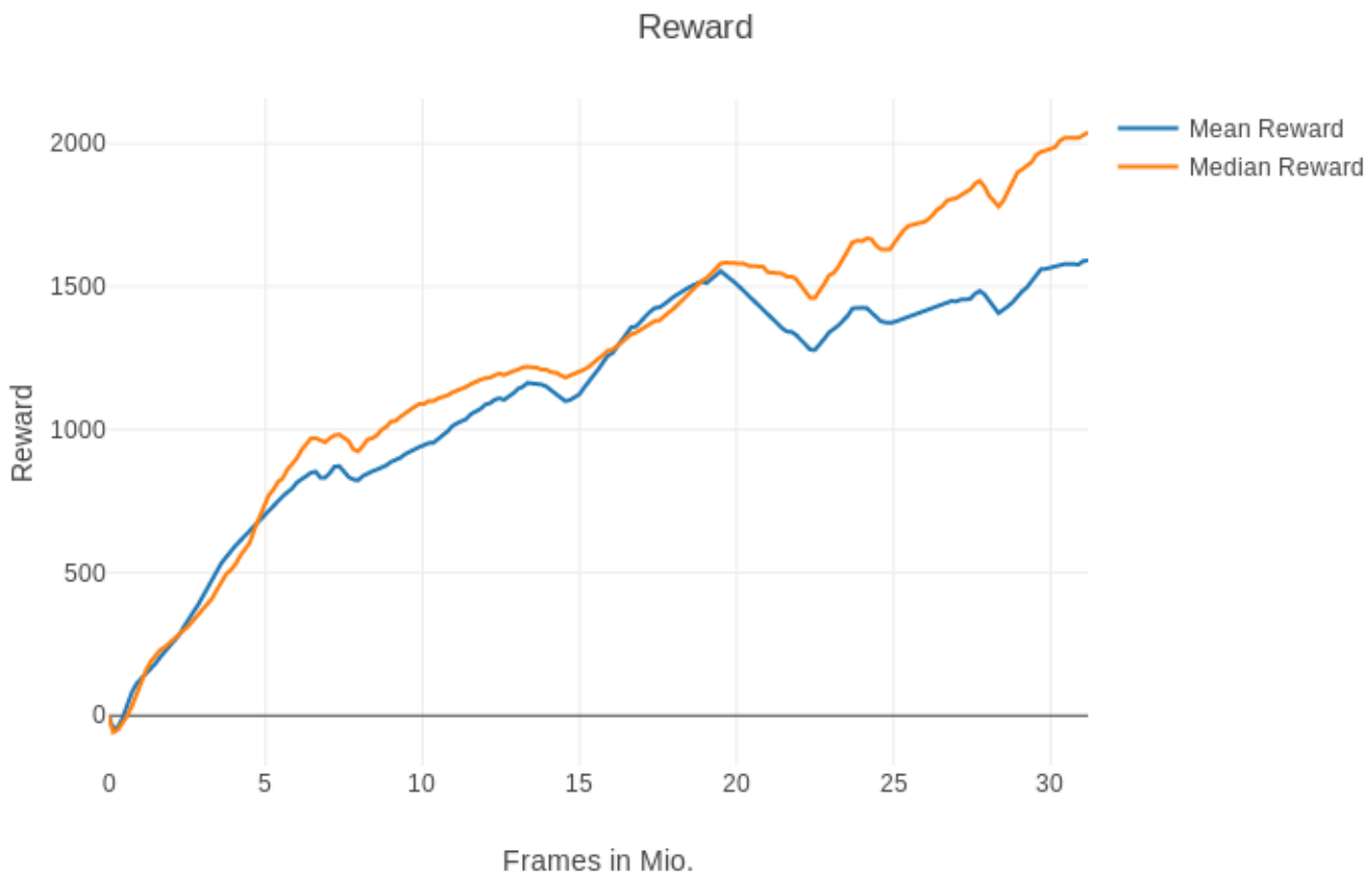
Specifics of I2A run of MsPacman

- Environment Model: dSSM_DET
- Optimizer: RMSProp, learning rate=7e-4
- Framestack: 4
- Frameskip: 4
- Distill coefficient: 10
- Entropy coefficient: 0.02
- Rollout length: 2, 5

dSSM-DET MsPacman rollout length 2



dSSM-DET MsPacman rollout length 5



I2A with dSSM-DET environment model

MsPacman demo

Conclusion

- Pytorch framework for I2A
- Implementation of latent environment model generation
- Implementation of I2A using the latent space approach

Our code will be published on github.

If you are interested send us an email to get the link.

Potential future work

- Compact latent representation for both paths
- Use GANs for the environment model
- Smaller latent space, the latent space in the paper is still pretty large
- Train environment model together with the policy network.

  For complex problems, other state-of-the-art methods may not be good enough to get a representative training set for the environment model training.
- Select promising actions to focus rollouts on

Thank you for your attention! Questions?