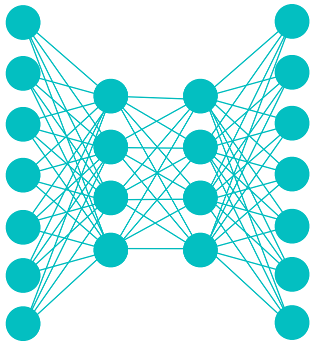
Lecture Notes for

Neural Networks and Machine Learning

World Models & Course Retrospective



Logistics and Agenda

- Logistics
  - Final Paper Due at end of Finals
- Agenda
  - Student Presentations
    - SAC
  - World Models
  - Class Retrospective

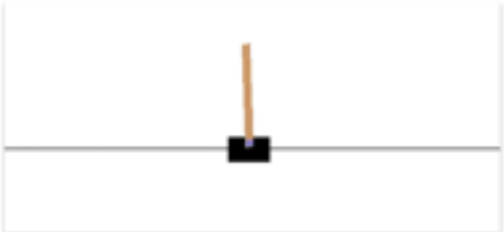
Last Time

```python
import gym


if __name__ == "__main__":
    env = gym.make("CartPole-v0")

    total_reward = 0.0
    total_steps = 0
    obs = env.reset()

    while True:
        action = env.action_space.sample()
        obs, reward, done, _ = env.step(action)
        total_reward += reward
        total_steps += 1
        if done:
            break
```



Action Space: One input, [0, 1] pull left or pull right

Obs Space: Dynamic state variables (continuous and four dimensional)

End: When more than 15 degrees off or too far from center

Reward: +1 for each time step

- Want to approximate $Q(s,a)$ when the state space is potentially large. Given s_t, we want the network to give us a row of actions that we can choose from:
  $[\, Q(s_t,a_1),\ Q(s_t,a_2),\ Q(s_t,a_3),\ \ldots\ Q(s_t,a_N) \,]$
- This allows us to make a loss function which incentives the actual Q-function behavior we desire from a sampled tuple (s, a, r, s')

$$\mathscr{L} = \Big[\underbrace{Q(s,a)}_{\text{from current network params}} - \underbrace{[r_{s,a} + \gamma \max_{a' \in A} Q^*(s',a')]}_{\text{from older network params (better stability)}}\Big]^2$$

Periodically Update
Params of Q^* from Q

$$\underbrace{\mathscr{L} = \left[Q(s,a) - [r_{s,a}]\right]^2}_{\text{if no next state (env is done)}}$$

$$V_0 = \max_{a \in A} \mathbf{E}[r_{s,a} + \gamma V_s] = \max_{a \in A} \sum_{s \in S} p_{a,0 \to s} \cdot (r_{s,a} + \gamma V_s)$$

- Define intermediate function Q

$$Q(s,a) = \sum_{s' \in S} p_{a,s \to s'} \cdot (r_{s,a} + \gamma V_{s'})$$

- With some nice properties/relations:

$$V_s = \max_{a \in A} Q(s,a)$$

$$Q(s,a) = r_{s,a} + \gamma \max_{a' \in A} Q(s',a')$$

Deep Q-Learning Reinforcement Learning

M. Lapan Implementation for Frozen Lake

And with Atari!

Paper Presentation

Soft Actor-Critic:
Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

Tuomas Haarnoja[1] Aurick Zhou[1] Pieter Abbeel[1] Sergey Levine[1]

Abstract

Model-free deep reinforcement learning (RL) algorithms have been demonstrated on a range of challenging decision making and control tasks. However, these methods typically suffer from two major challenges: very high sample complexity

of these methods in real-world domains has been hampered by two major challenges. First, model-free deep RL methods are notoriously expensive in terms of their sample complexity. Even relatively simple tasks can require millions of steps of data collection, and complex behaviors with high-dimensional observations might need substantially more.

World Models

The Problem

World Models

Can agents learn inside of their own dreams?

DAVID HA Google Brain Tokyo, Japan	JÜRGEN SCHMIDHUBER NNAISENSE Swiss AI Lab, IDSIA (USI & SUPSI)	March 27 2018	NIPS 2018 Paper	YouTube Talk	Download PDF

https://worldmodels.github.io

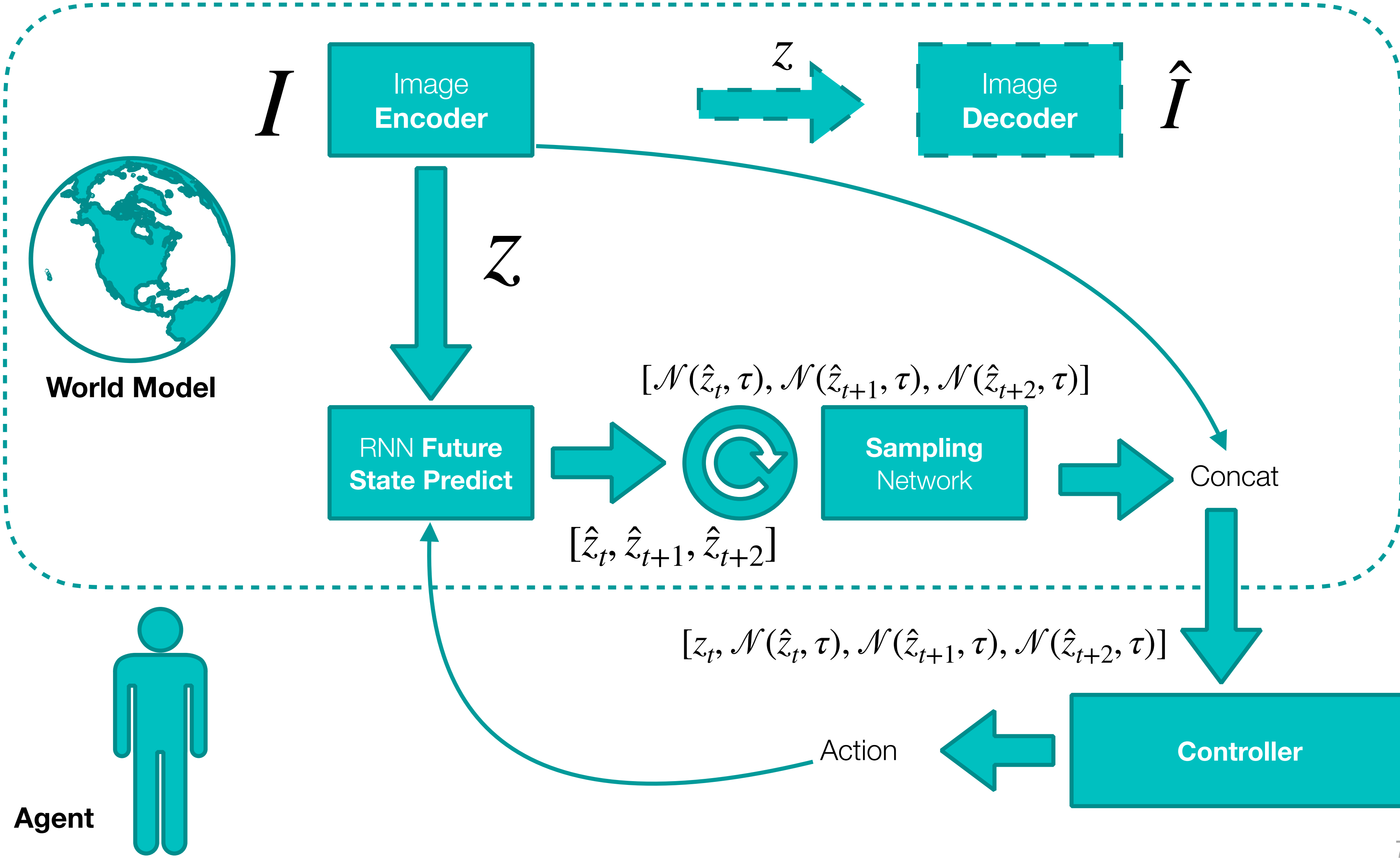
A Motivation

Agents can dream! What a time to be alive!

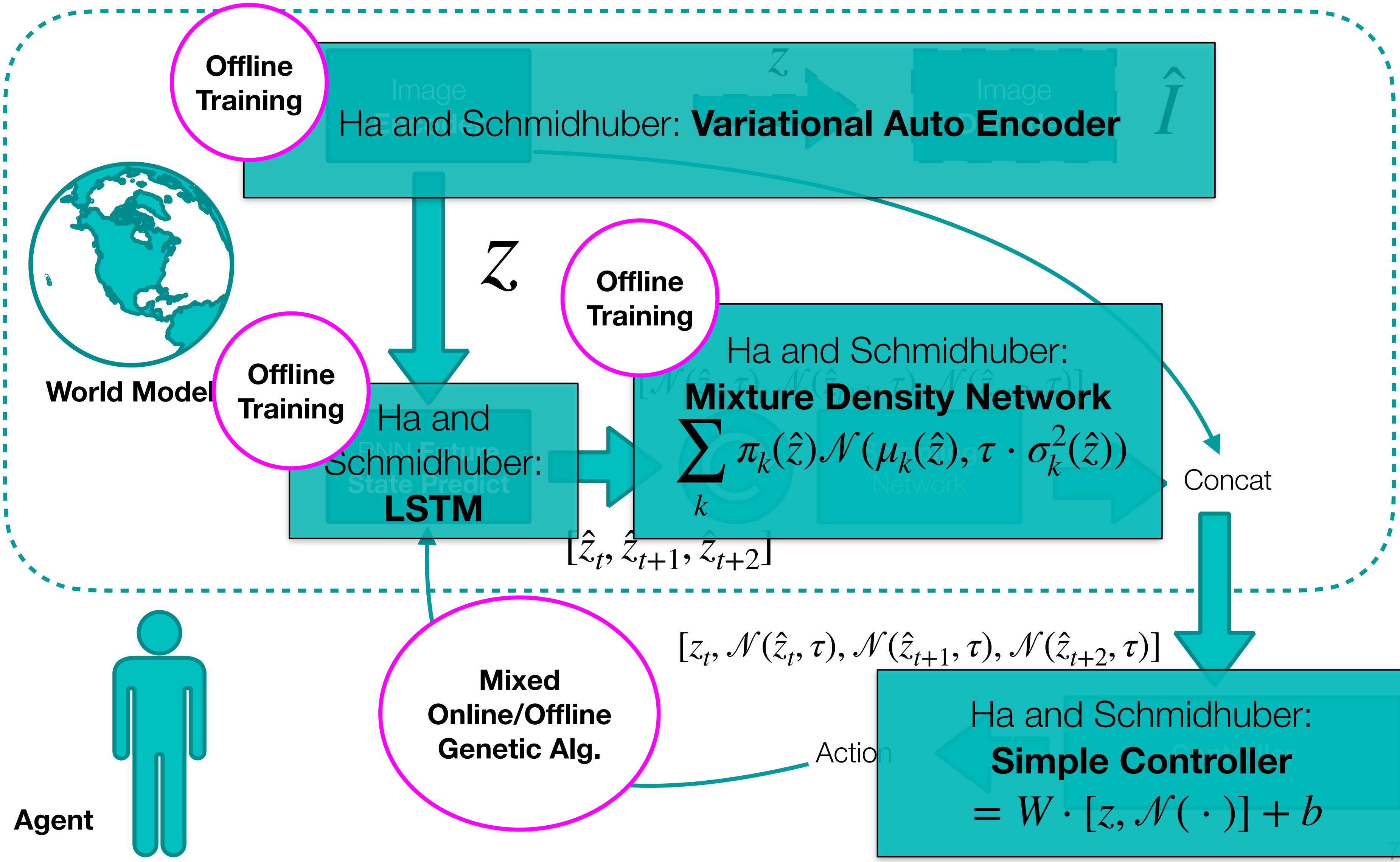
And academia can dream about driving the hype train!

Maybe we should be more careful about the way we describe what an agent does… because they don't dream. That's fluff.

The Main Idea



Implementation

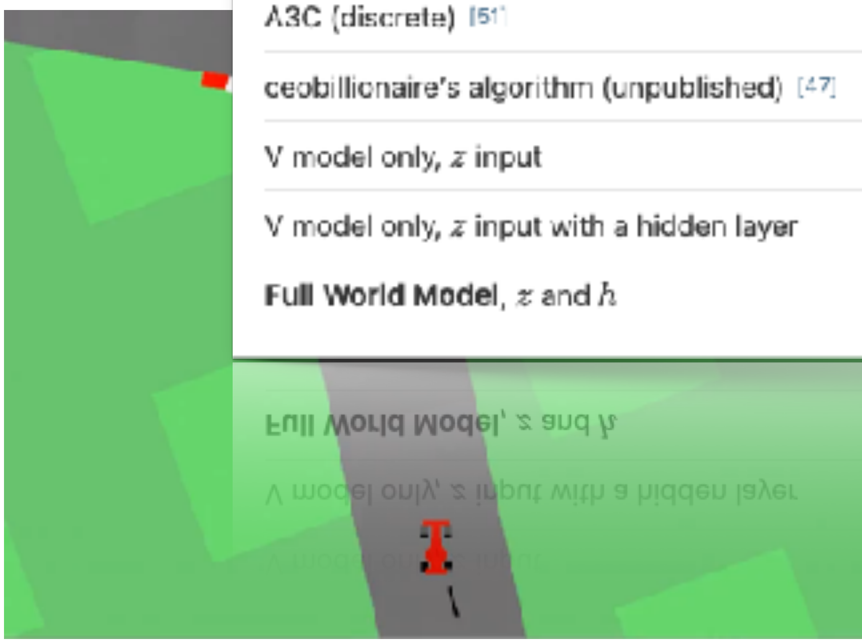


An Example, Racing

- Schmidhuber and Ha Methods:
  - Collect 10,000 rollouts from a random policy.
  - Train VAE (V) to encode each frame
  - Train
  - Evol
    cum

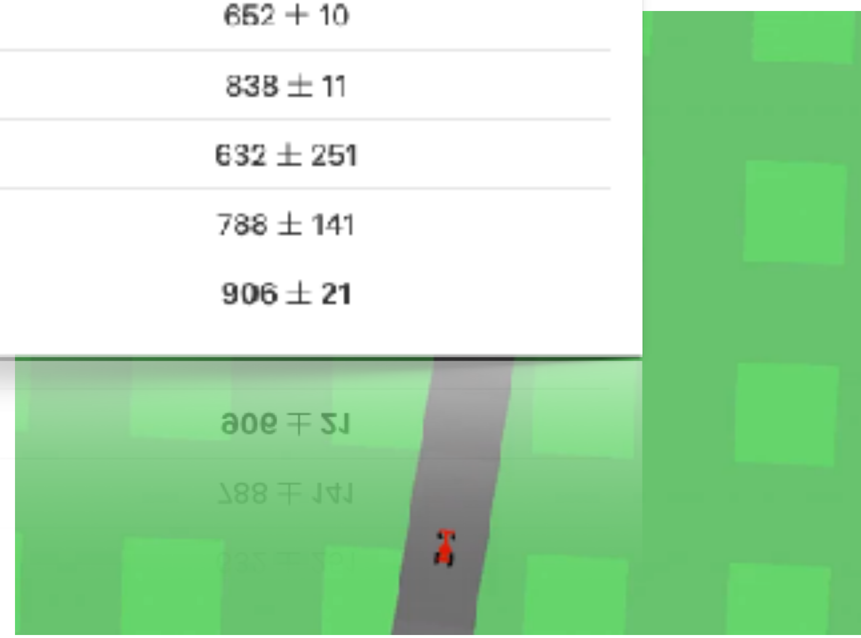
Model	Parameter Count
VAE	4,348,547
	422,368
	867

Method	Average Score over 100 Random Tracks
DQN [53]	343 ± 18
A3C (continuous) [52]	591 ± 45
A3C (discrete) [51]	652 ± 10
ceobillionaire's algorithm (unpublished) [47]	838 ± 11
V model only, z input	632 ± 251
V model only, z input with a hidden layer	788 ± 141
Full World Model, z and h	906 ± 21



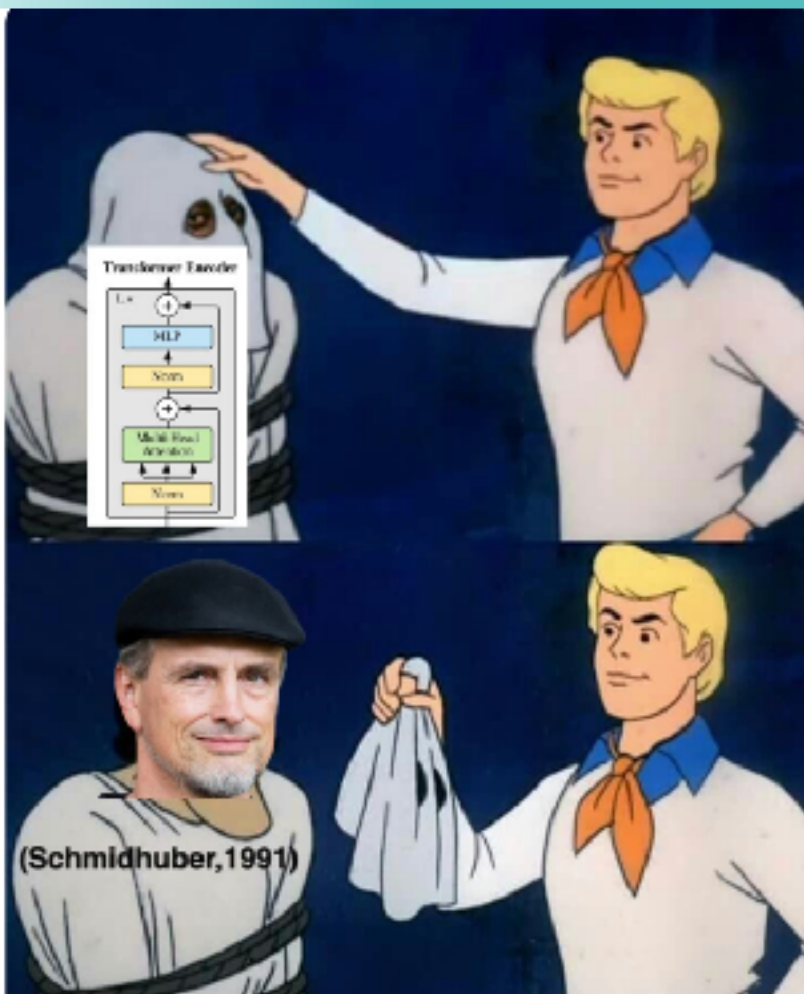
Only use VAE Encoding

https://worldmodels.github.io



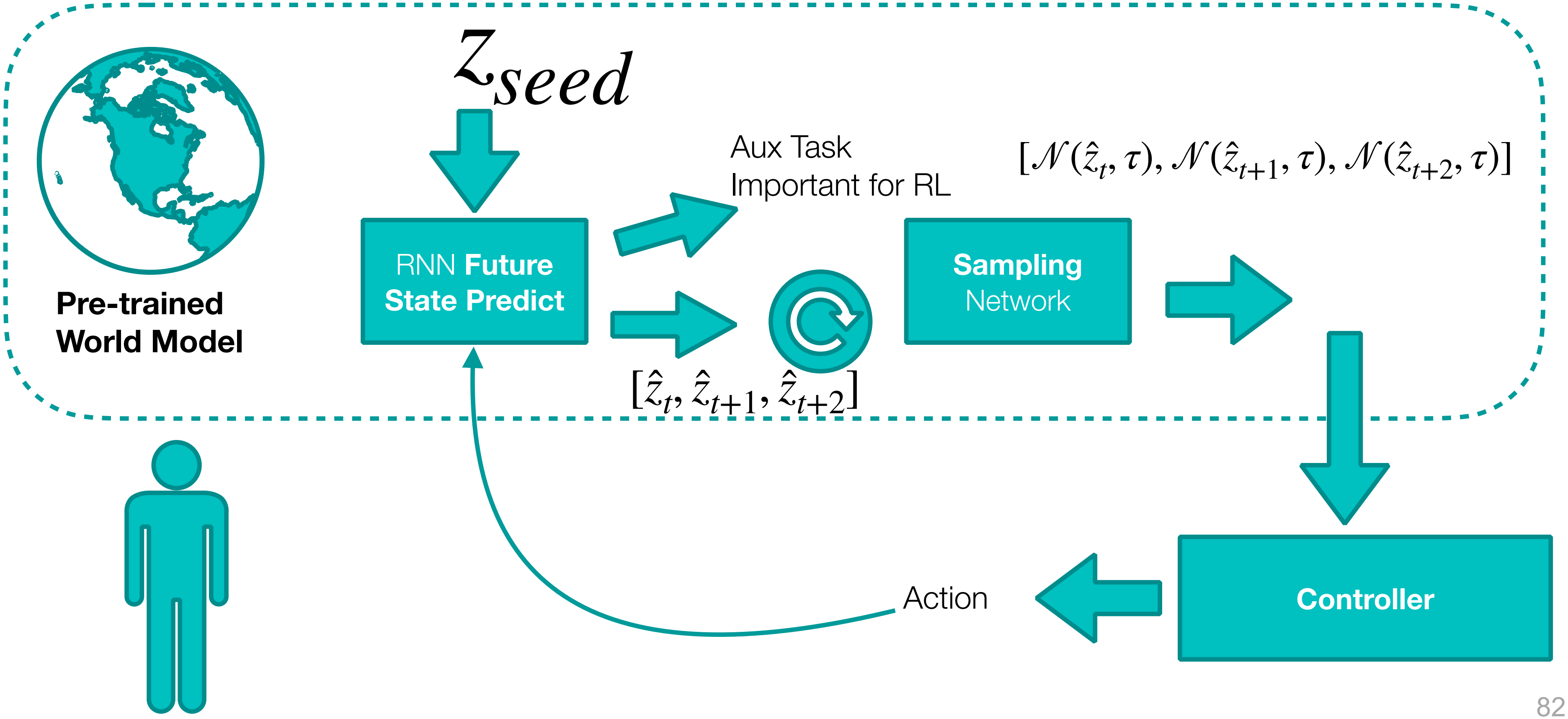
Full World Model

World Models II

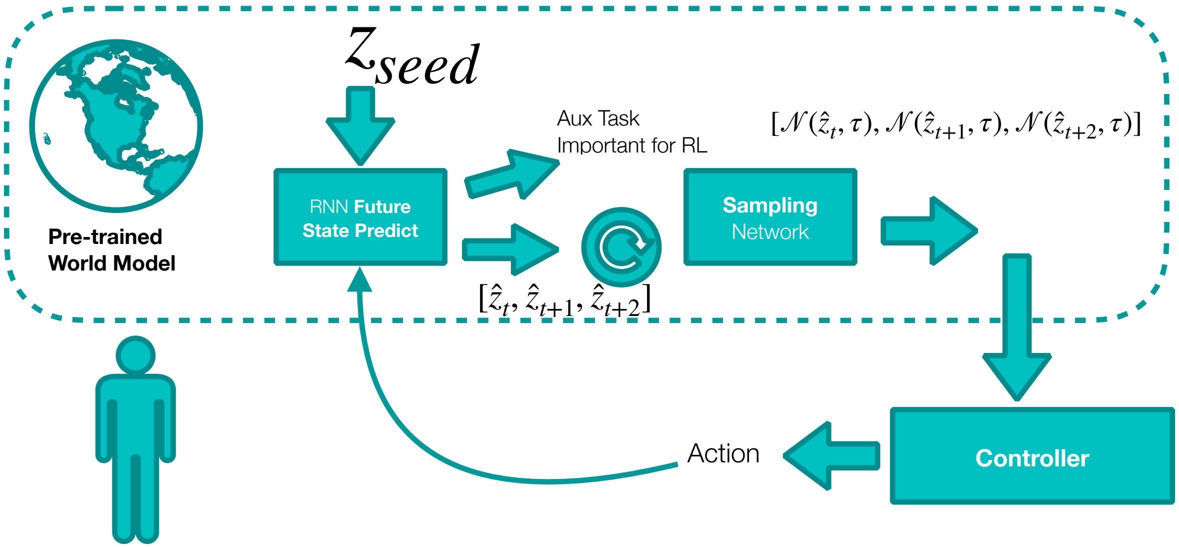


Can we learn without the environment?

- What if we sample from the world model to train our controller?



VizDoom Training Example



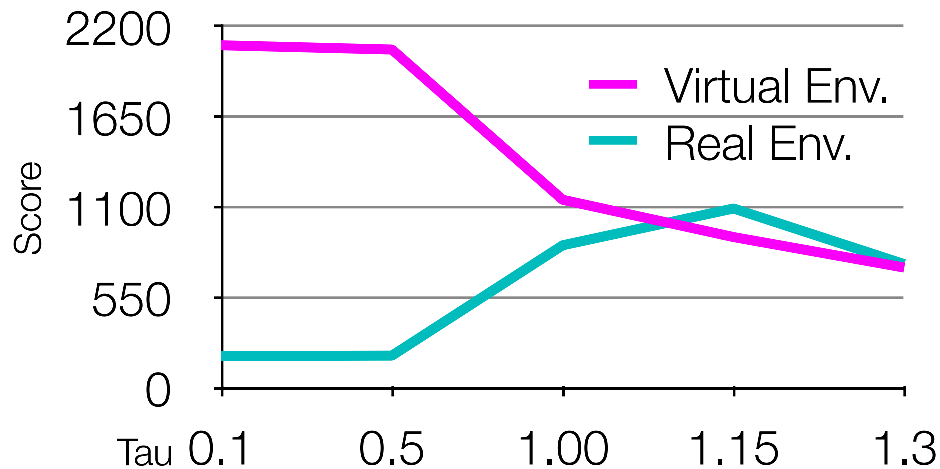
Model	Parameter Count
VAE	4,446,915
MDN-RNN	1,678,785
Controller	1,088

- Collect 10,000 rollouts from a random policy
- Train VAE (V) to encode each frame
- Train MDN-RNN to predict z and “if survived” in next frame
- Evolve Controller (C) to maximize the expected survival time inside the virtual environment.
- Use learned policy from on actual Gym environment
- Call it training inside a “dream” because marketing

Learned Policy

- Important to optimize the temperature control of the MDN

$$\sum_k \pi_k(\hat{z})\mathcal{N}(\mu_k(\hat{z}), \tau \cdot \sigma_k^2(\hat{z}))$$



Temperature	Score in Virtual Environment	Score in Actual Environment
0.10	2086 ± 140	193 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	918 ± 546	1092 ± 556
1.30	732 ± 269	753 ± 139
Random Policy Baseline	N/A	210 ± 108
Gym Leaderboard [34]	N/A	820 ± 58

More Complex Models

- Random Policy makes it hard to exploit "hard to get to" regions of the state space
- Solution: Iterative algorithm
  - Initialize *M*, *C* with random model parameters
  - Rollout to actual environment *N* times. Agent may learn during rollouts. Save all actions and observations during rollouts
  - Train *M* to model $P(x_{t+1}, r_{t+1}, a_{t+1}, d_{t+1} \,|\, x_t, a_t, \hat{z}_t)$ and train *C* to optimize expected rewards in *M*
  - Repeat rollout of new policy if not converged
- Leave that investigation to future work…

Course Retrospective

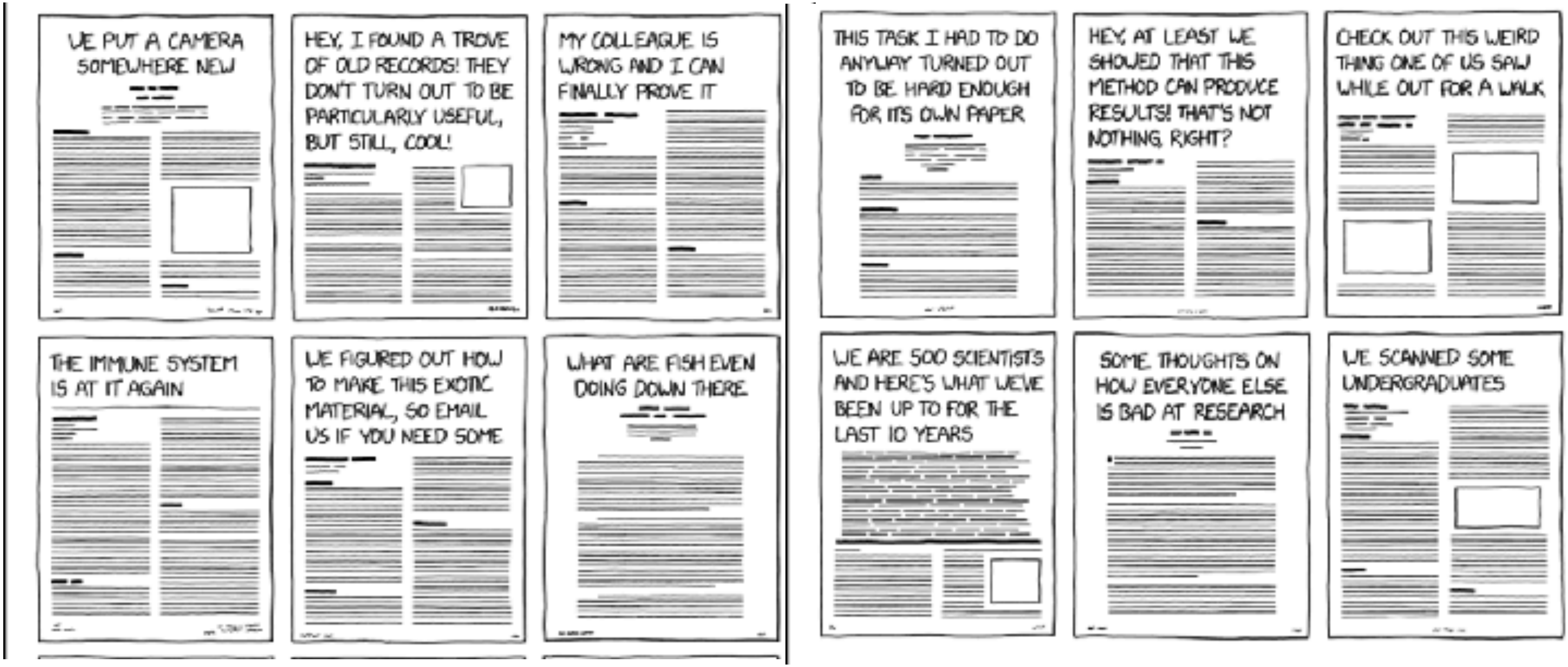
Day 1 of python: How can I learn python ?

Day 3 of python: machine learning engineer positions near me

Course Retrospective

- **Ethics**: The Guidelines, ConceptNet NumberBatch
- **CNN Visualization**: Filters, Heatmaps, Grad-CAM, Circuits
- **CNN Fully Convolutional**: R-CNN, YOLO, Mask-RCNN, YOLACT and others
- **Style Transfer**: Gatys, FastStyle, WCT
- **Multi-task and Multi-modal**: ATLAS, Self-consistency
- **GANs**: Goodfellow to Wasserstein to BigGAN
- **RL**: Value, Q-Learning, Deep Q-Learning and World Models
- What was good, **bad**, **ugly**? What could be **changed**?

Types of Scientific Papers



Thanks for a great semester!!!

Please fill out the course evaluations!!

Lecture Notes for

Neural Networks and Machine Learning

World Models and Course Retrospective

Next Time:

None!

Reading: Nope