ODSC-East 2022

Reinforcement Learning for Recommender Systems

From Contextual Bandits to Slate-Q

Christy Bergman, DevAdvocate Ray RLlib

https://linkedin.com/in/christybergman
christy@anyscale.com
@cbergman on Twitter

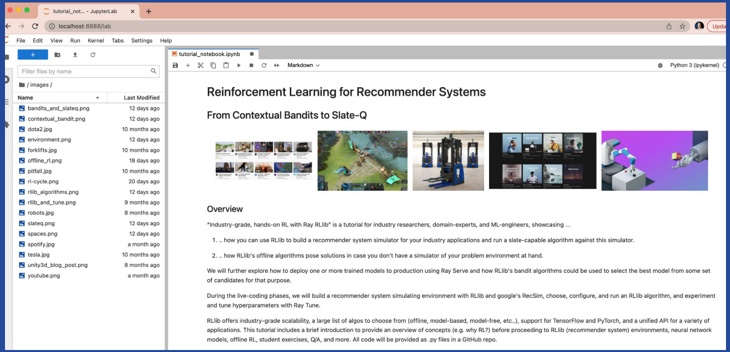
Avnish Narayan, Software Engineer Ray RLlib

https://linkedin.com/in/avnishn
avnish@anyscale.com
@narayan_avnish on Twitter

anyscale RLlib

Overview of the tutorial

Our iPython Notebook … and how to git it




```
$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab
```

$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab

anyscale

Overview of the tutorial

25 min: Intro to RL - What is RL and why should we use RL to solve Recommender Systems?
5 min: break
25 min: RLlib's Contextual Bandits and SlateQ on Google RecSim problem
5min: break
25min: Ray Tune, Offline RL, Ray Serve

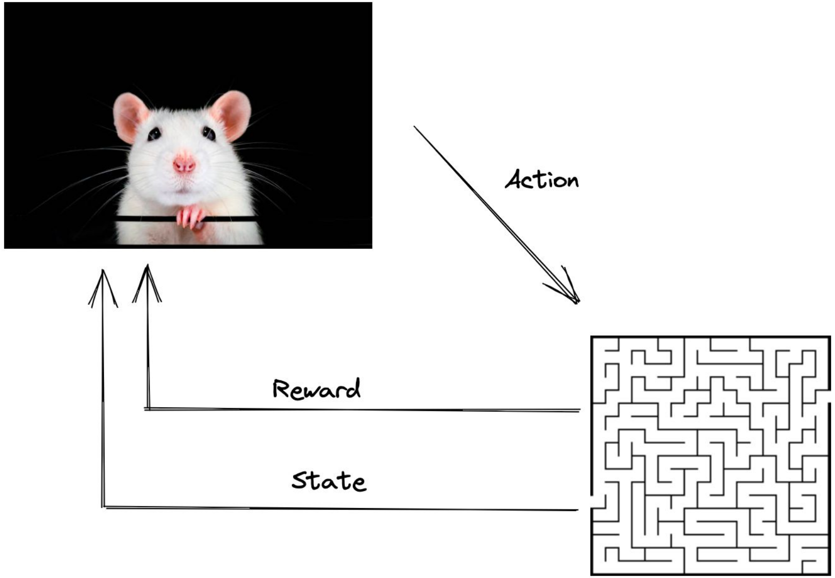
~1.5h

$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab

anyscale

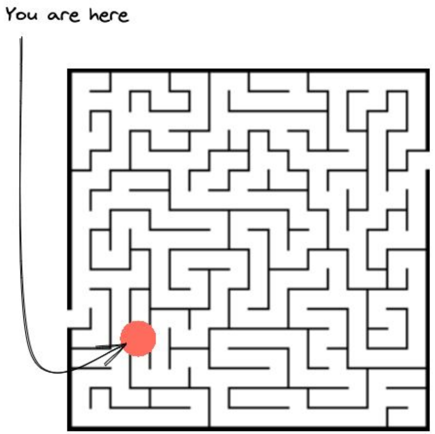
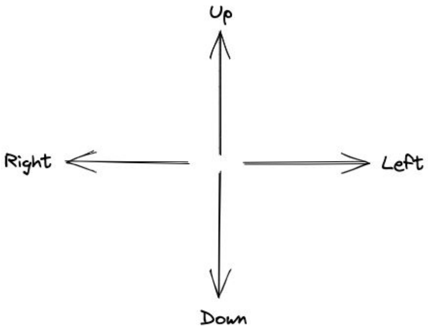
What is Reinforcement Learning (RL)?

- We want to teach a mouse to get to the center of a maze … how can we do that?
- Place bits of cheese in the maze along paths that we **do** want the mouse to take.
- Place bits of poison along paths that we **don't** want the mouse to take.
- Let the mouse repeatedly explore the maze for a fixed time from different random starting points.
- Eventually the mouse learns all paths from everywhere in the maze that maximize cheese and minimize poison.



What is Reinforcement Learning (RL)?

- The mouse can be at different positions in the maze. We call this its **state** or observation.
- The mouse can move up down left and right in the maze. We call this its **actions.**

State	Action
	

What is Reinforcement Learning (RL)?

The mouse gets cheese or poison depending on whether its taking the best action at its state. We call this its **reward.**

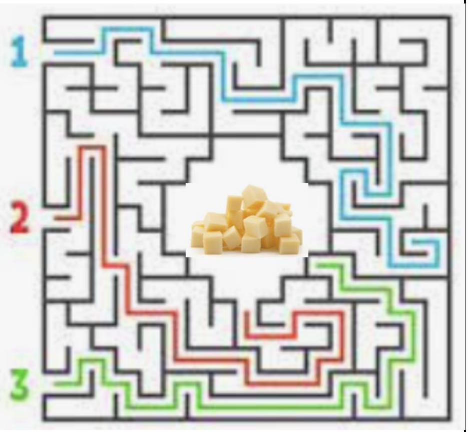
Reward

Cheese :)

Poison :(

What is Reinforcement Learning (RL)?

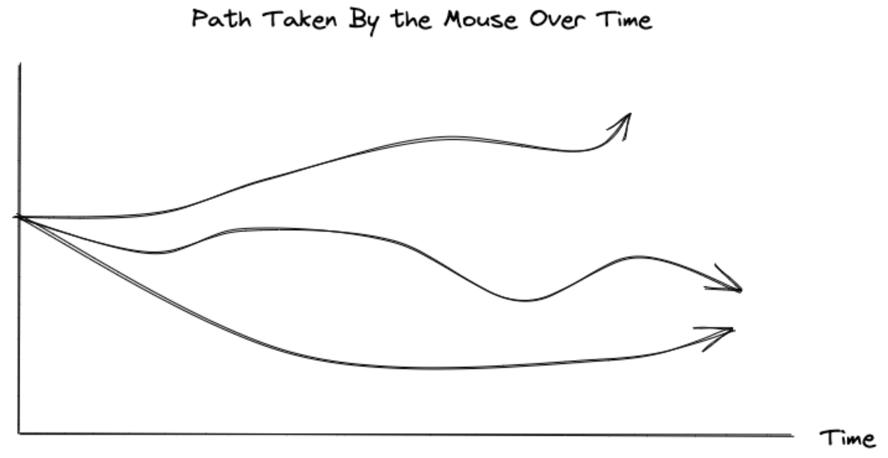
The mouse, or our **agent,** is incentivized to take the sequence of actions at the states which it visits that maximizes the total reward per episode in a particular maze **environment**.

Agent	Environment
	

What is Reinforcement Learning (RL)?

- The mouse collects experiences over a certain period of time. We call this collecting **episodes** of experience.
- An episode ends when the mouse reaches the center of the maze. We can also send a **done** signal to end the episode due to time out.

Episodes



Going from Mice and Mazes to the Real World

Autonomous Vehicle vision-based controller

State: camera images, lidar, sensor readings

Action: steering wheel angle, pressure on brake, pressure on gas pedal

Reward: +1 if close to destination & driving safely, 0 otherwise

Done: true if reached destination

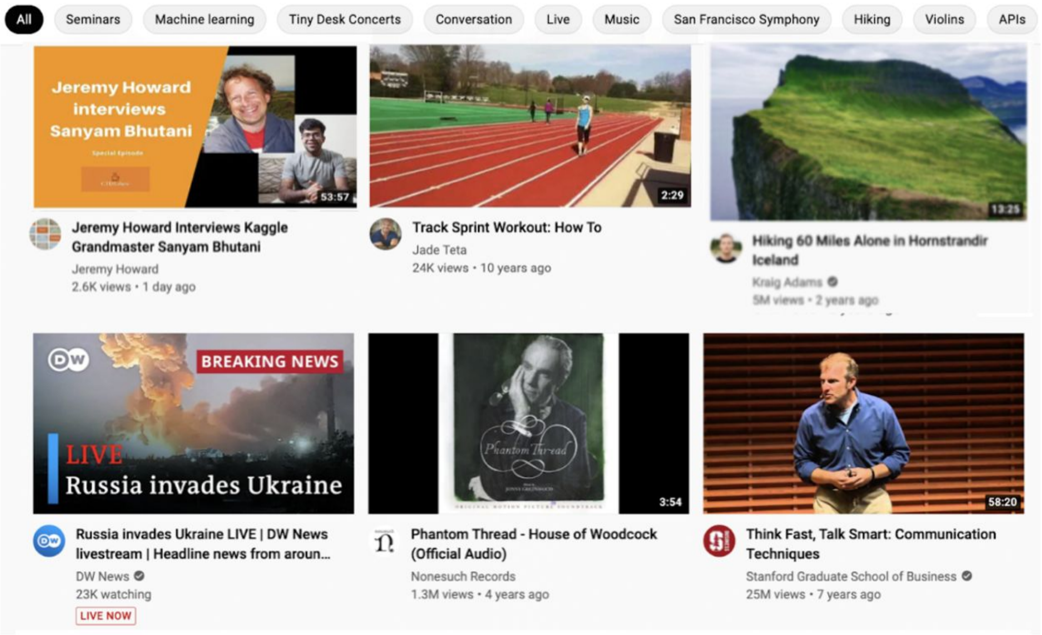
What are Recommender Systems and where do you find them?

What	Where
We want to serve up **personalized content**, based on **users interactions**, in order to improve each user's **experience**.	• Websites, web apps • Mobile apps • Purchasing apps (B2B, B2C) • Games • Online ads and offers • Emails • Chatbots • Call centers

Typical Supervised Learning Approach to Recommender System

High-level steps to build supervised DL Recommender

- Build User feature embeddings vectors
- Build Document feature embeddings vectors for content items
- Rank items per user based on "preferences of others or the crowd" (Collaborative and/or Content Filtering)
- Assign items to the user's feed onto "slate" positions based on each user's item rankings.



Why use RL over Supervised Learning for Recommendations?

Challenges not solved by typical Supervised Learning prediction

- High sparsity of data can lead to high bias.
- Cold-start problem for brand new items or users which have no data yet.
- Delay between in-product realtime user actions and updating the Models. Billiions of items, millions of users calculations are usually pre-processed in batch offline.
- Myopic algorithms that optimize only for short-term click-through rate can end up hurting long term engagement. Long-term satisfaction, typically measured as long-term user engagement, needs to be part of the model.
- Exploration. Besides offering users the same type of content they interacted with historically, offer education and new content .

Approach - Real-time Recommendations based on real-time sequential decision-making (RL)

- Historical data “Batch” pre-processed into traditional Supervised Learning models and the model input feature embeddings
- User and product feature embeddings used to define Environment for RL

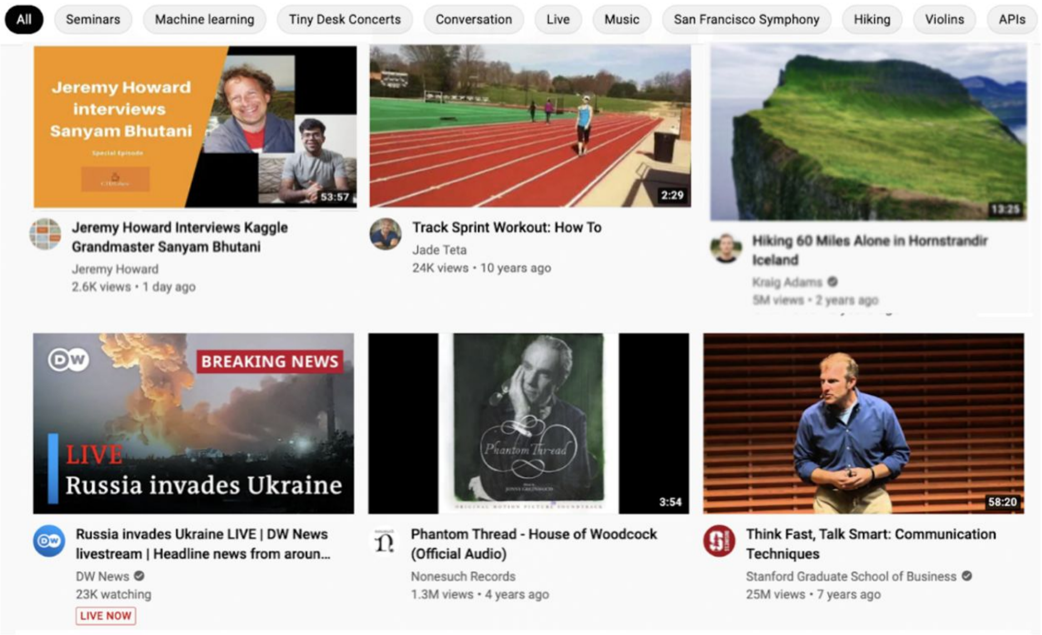
Going from Mice and Mazes to RecSys

State: user actions, user feature embeddings, doc feature embeddings

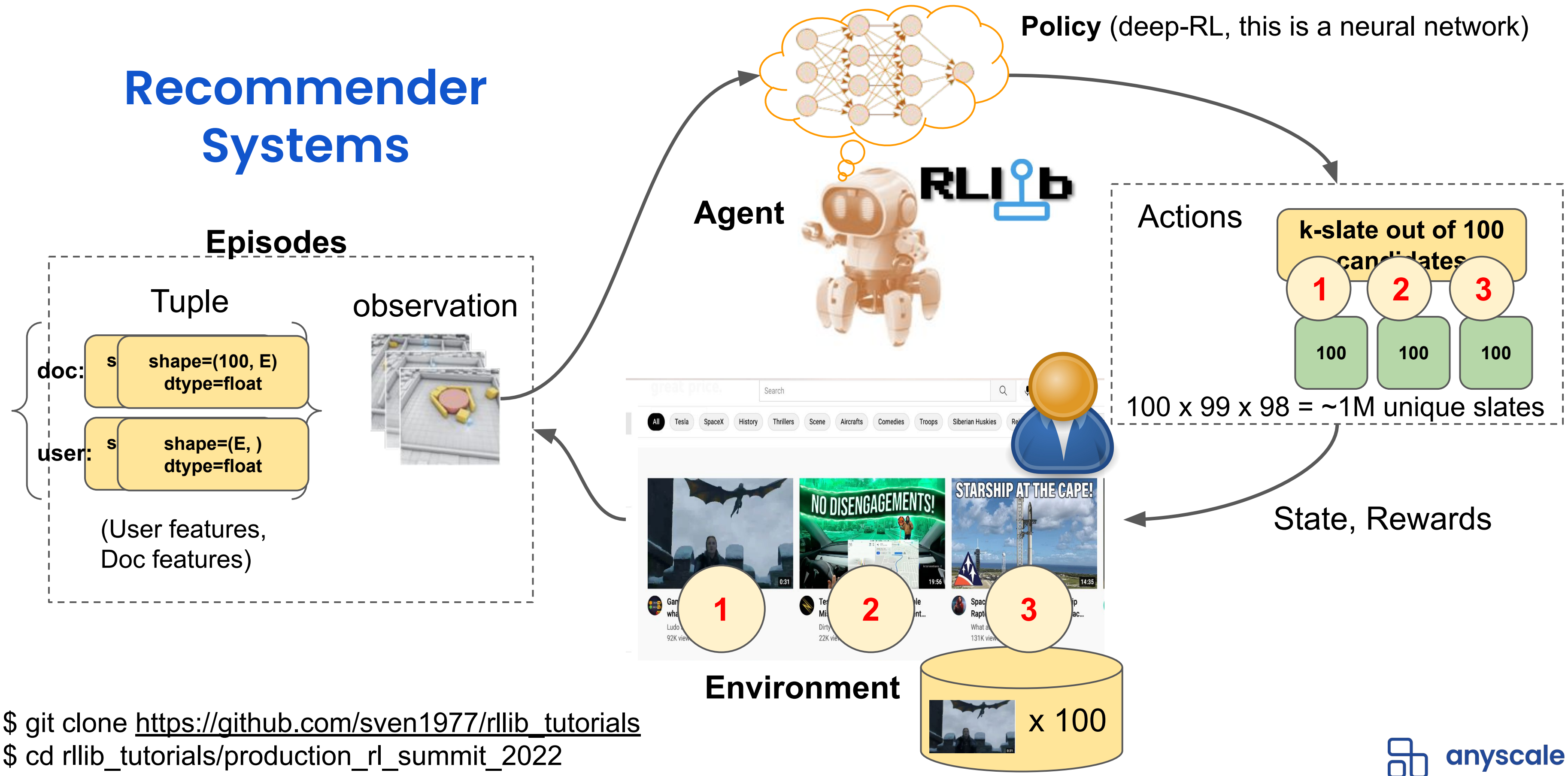
Action: click_slate_0_0, skip_slate_0_0, click_slate_0_1, skip_slate_0_1, …

Reward: +1 if user clicks a piece of content AND long-term satisfaction has increased, 0 otherwise

Done: true if X days since user's last interactive session



RL Ingredients and RLlib code abstractions



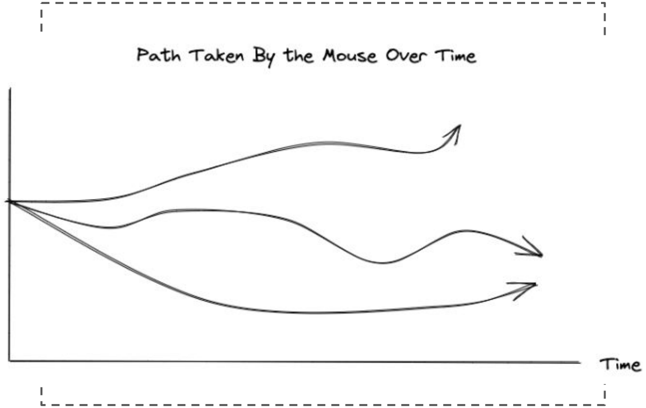

```
$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab
```

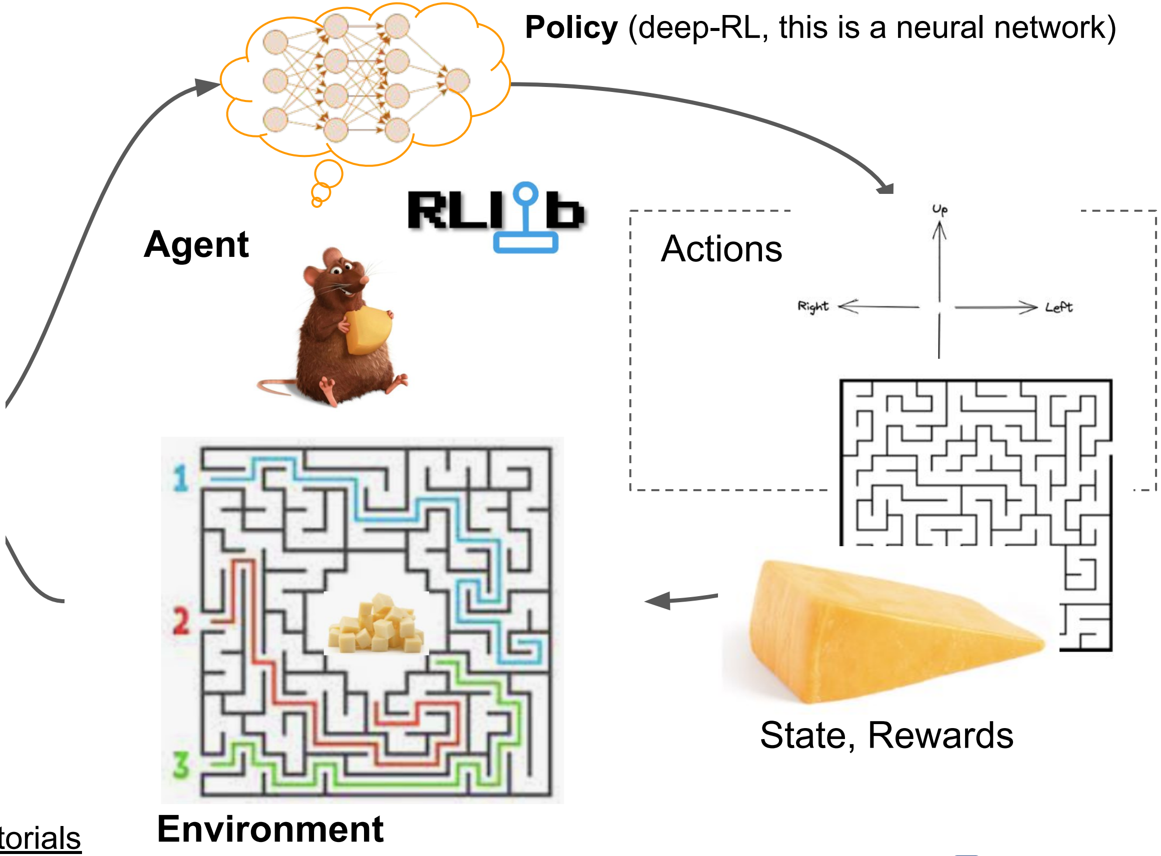
anyscale

RL Ingredients and RLlib code abstractions

Recommender Systems

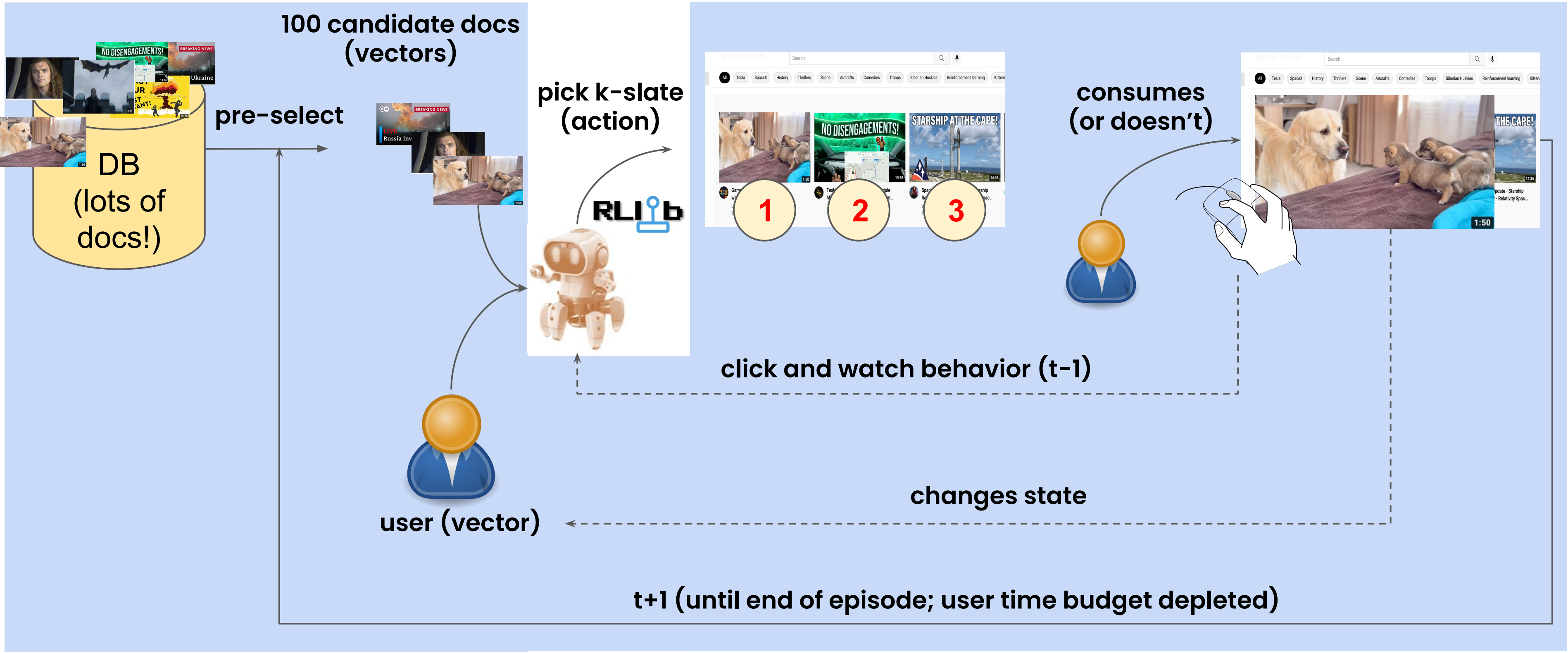
Episodes





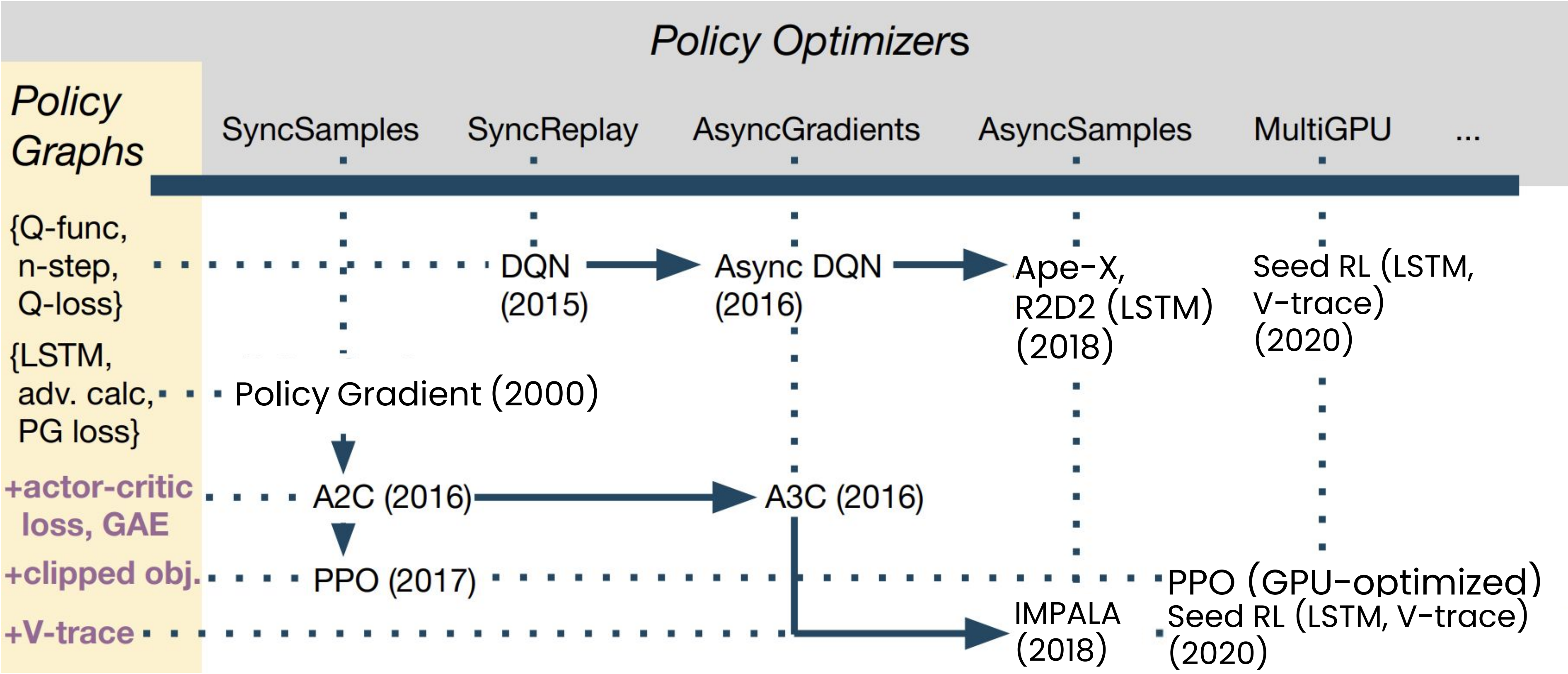
$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab

A Recommender System in Action

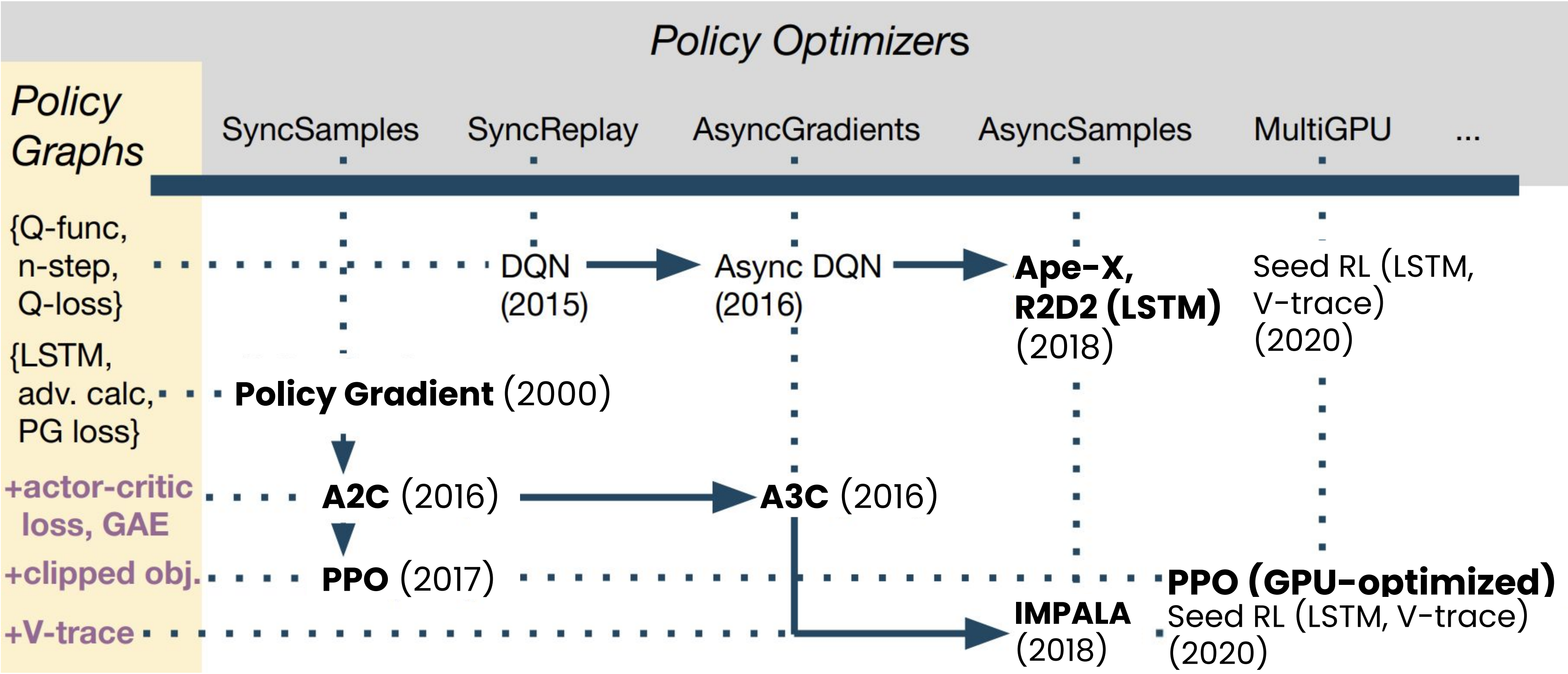


$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab

RLlib Parallelize and Distribute

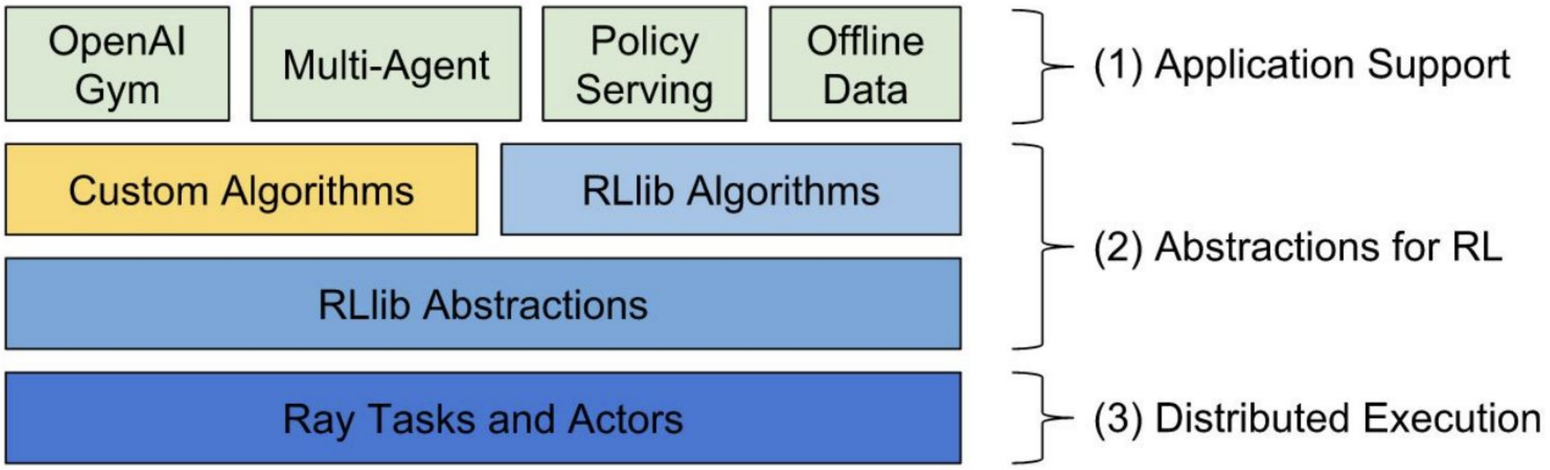


RLlib Parallelize and Distribute

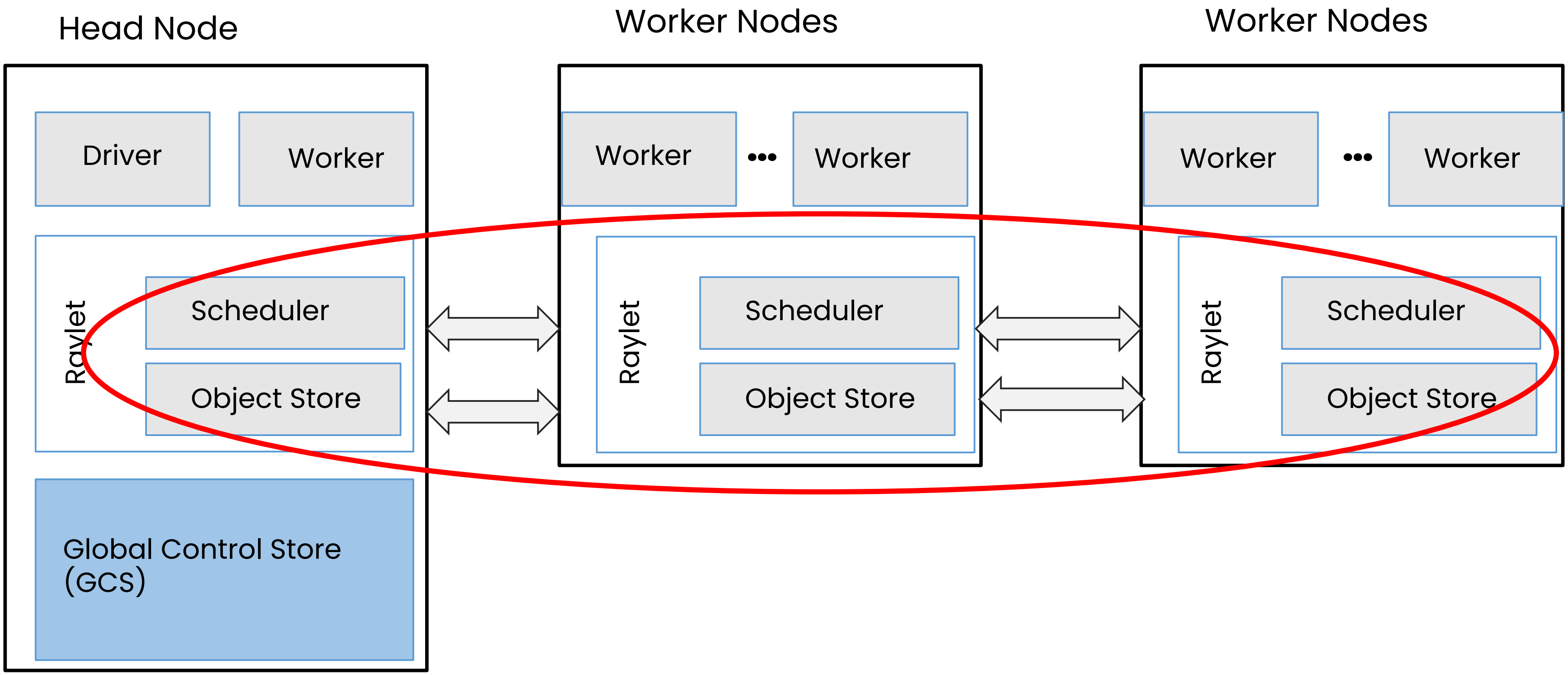


RLlib Parallelize and Distribute

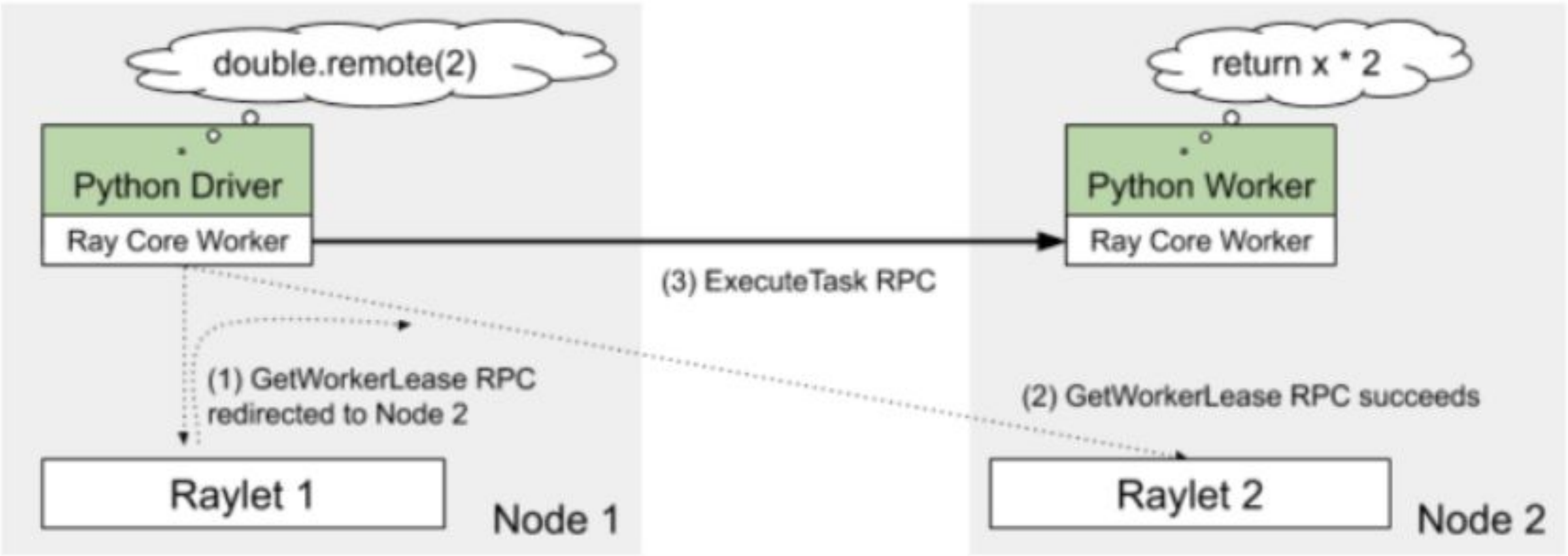
- Beyond a "collection of algorithms",
- RLlib's abstractions let you easily implement and scale new algorithms (multi-agent, novel losses, architectures, etc)



What does a Ray Cluster look like …



Scaling to Multiple Nodes

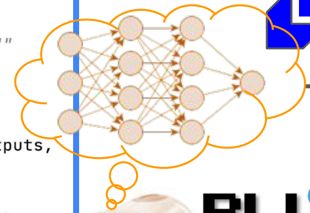


Tasks are sent to remote workers if there are no local resources available, transparently scaling Ray applications out to multiple nodes.

anyscale

And why should

TensorFlow &
PyTorch Policies

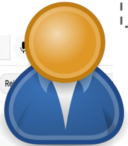


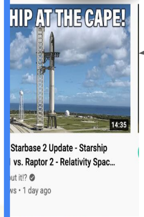
```
class CustomModel(TFModelV2):
    """Example of a keras custom model that just delegates to an fc-net."""

    def __init__(self, obs_space, action_space, num_outputs, model_config,
                 name):
        super(CustomModel, self).__init__(obs_space, action_space, num_outputs,
                                          model_config, name)
        self.model = FullyConnectedNetwork(obs_space, action_space,
                                           num_outputs, model_config, name)

    def forward(self, input_dict, state, seq_lens):
        return self.model.forward(input_dict, state, seq_lens)

    def value_function(self):
```

```
ModelCatalog.register_custom_model(
    "my_model", TorchCustomModel
    if args.framework == "torch" else CustomModel)

config = {
    "model": {
        "custom_model": "my_model",
    },
}
```

```
    def forward(self, input_dict, state, seq_lens):
        input_dict["obs"] = input_dict["obs"].float()
        fc_out, _ = self.torch_sub_model(input_dict, state, seq_lens)
        return fc_out, []

    def value_function(self):
        return torch.reshape(self.torch_sub_model.value_function(), [-1])
```

actions





anyscale

And why should

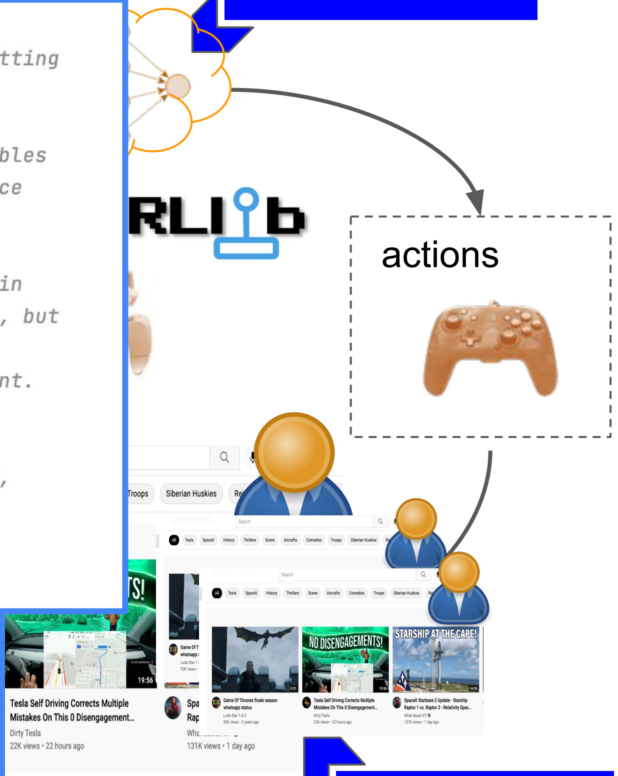
TensorFlow &
PyTorch Policies

```
# === Settings for Rollout Worker processes ===
# Number of rollout worker actors to create for parallel sampling. Setting
# this to 0 will force rollouts to be done in the trainer actor.
"num_workers": 2,
# Number of environments to evaluate vector-wise per worker. This enables
# model inference batching, which can improve performance for inference
# bottlenecked workloads.
"num_envs_per_worker": 1,
# If using num_envs_per_worker > 1, whether to create those new envs in
# remote processes instead of in the same worker. This adds overheads, but
# can make sense if your envs can take much time to step / reset
# (e.g., for StarCraft). Use this cautiously; overheads are significant.
"remote_worker_envs": False,
# Timeout that remote workers are waiting when polling environments.
# 0 (continue when at least one env is ready) is a reasonable default,
# but optimal value could be obtained by measuring your environment
# step / reset and model inference perf.
"remote_env_batch_wait_ms": 0,
```

```
Mod
con
    },
}
```

```
ace, num_outputs,

    def forward(self, input_dict, state, seq_lens):
        input_dict["obs"] = input_dict["obs"].float()
        fc_out, _ = self.torch_sub_model(input_dict, state, seq_lens)
        return fc_out, []

    def value_function(self):
        return torch.reshape(self.torch_sub_model.value_function(), [-1])
```

```
$ git cl
$ cd rl
$ jupy
```



parallelize and
distribute

anyscale

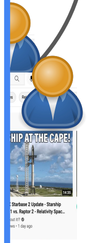
```
# === Settings for Rollout Worker
# Number of rollout worker actors
# this to 0 will force rollouts to
"num_workers": 2,
# Number of environments to evalua
# model inference batching, which
# bottlenecked workloads.
"num_envs_per_worker": 1,
# If using num_envs_per_worker > 1
# remote processes instead of in t
# can make sense if your envs can
# (e.g., for StarCraft). Use this
"remote_worker_envs": False,
# Timeout that remote workers are
# 0 (continue when at least one en
# but optimal value could be obtai
# step / reset and model inference
"remote_env_batch_wait_ms": 0,
```

RLlib Algorithms

- High-throughput architectures
  - Distributed Prioritized Experience Replay (Ape-X)
  - Importance Weighted Actor-Learner Architecture (IMPALA)
  - Asynchronous Proximal Policy Optimization (APPO)
  - Decentralized Distributed Proximal Policy Optimization (DD-PPO)
- Gradient-based
  - Advantage Actor-Critic (A2C, A3C)
  - Deep Deterministic Policy Gradients (DDPG, TD3)
  - Deep Q Networks (DQN, Rainbow, Parametric DQN)
  - Policy Gradients
  - Proximal Policy Optimization (PPO)
  - Soft Actor Critic (SAC)
  - Slate Q-Learning (SlateQ)
- Derivative-free
  - Augmented Random Search (ARS)
  - Evolution Strategies
- Model-based / Meta-learning / Offline
  - Single-Player AlphaZero (contrib/AlphaZero)
  - Model-Agnostic Meta-Learning (MAML)
  - Model-Based Meta-Policy-Optimization (MBMPO)
  - Dreamer (DREAMER)
  - Conservative Q-Learning (CQL)
- Multi-agent
  - QMIX Monotonic Value Factorisation (QMIX, VDN, IQN)
  - Multi-Agent Deep Deterministic Policy Gradient (contrib/MADDPG)
- Offline
  - Advantage Re-Weighted Imitation Learning (MARWIL)
- Contextual bandits
  - Linear Upper Confidence Bound (contrib/LinUCB)
  - Linear Thompson Sampling (contrib/LinTS)
- Exploration-based plug-ins (can be combined with any algo)
  - Curiosity (ICM: Intrinsic Curiosity Module)

25+ available algorithms (model-free/based; offline RL; meta RL; evolutionary strategies)

tions



llelize and
ribute

$ git clone https://github.com/sven1
$ cd rllib_tutorials/production_rl_sun
$ jupyter-lab

anyscale

RLlib Algorithms

- High-throughput architecture
  - Distributed Priorit

25+ available algorithms (model-free/based; offline RL; meta-RL; evolutionary strategies)

Tuned examples for most of our algorithms on popular environments.

master ray / rllib / tuned_examples / ppo /

Go to file | Add file | ...

sven1977 [RLlib] Slate-Q tf implementation and tests/benchmarks. (#22389) 6522935 17 days ago History

..

atari-ddppo.yaml	[RLlib] Deprecate `vf_share_layers` in top-level PPO/MAML/MB-MPO conf...	14 months ago
atari-ppo.yaml	[RLlib] Deprecate `vf_share_layers` in top-	nths ago
cartpole-appo-vtrace-fake-gpus.yaml	[RLlib] Optionally don't drop last ts in v-trac	nths ago
cartpole-appo-vtrace-separate-losses.yaml	[RLlib] Add all simple learning tests as `fram`	nths ago
cartpole-appo-vtrace.yaml	[RLlib] Minor fixes/cleanups; chop_into_seq	nths ago
cartpole-appo.yaml	[RLlib] Refactor: All tf static graph code sho	nths ago
cartpole-ddppo.yaml	[RLlib] Auto-framework, retire `use_pytorch`	ears ago
cartpole-grid-search-example.yaml	[RLlib] Auto-framework, retire `use_pytorch`	ears ago
cartpole-ppo-fake-gpus.yaml	[RLlib] Move existing fake multi-GPU learnin	nths ago
cartpole-ppo-hyperband.yaml	[RLlib] Auto-framework, retire `use_pytorch`	ears ago
cartpole-ppo.yaml	[RLlib] Deprecate `vf_share_layers` in top-	nths ago
frozenlake-appo-vtrace.yaml	[RLlib] Upgrade gym version to 0.21 and dep	nths ago
halfcheetah-appo.yaml	[RLlib] Refactor: All tf static graph code sho	nths ago
halfcheetah-ppo.yaml	[RLlib] Auto-framework, retire `use_pytorch`	ears ago
hopper-ppo.yaml	[RLlib] Auto-framework, retire `use_pytorch`	ears ago
humanoid-ppo-gae.yaml	[RLlib] Auto-framework, retire `use_pytorch` in favor of `framework=....	2 years ago

20 lines (20 sloc) | 506 Bytes

```yaml
cartpole-appo:
    env: CartPole-v0
    run: APPO
    stop:
        episode_reward_mean: 150
        timesteps_total: 200000
    config:
        # Works for both torch and tf.
        framework: tf
        num_envs_per_worker: 5
        num_workers: 1
        num_gpus: 0
        observation_filter: MeanStdFilter
        num_sgd_iter: 6
        vf_loss_coeff: 0.01
        vtrace: false
        model:
            fcnet_hiddens: [32]
            fcnet_activation: linear
            vf_share_layers: true
```

$ git clone https://gi
$ cd rllib_tutorials/p
$ jupyter-lab

ale

And why should I use RLlib?

Because these companies here do!

Thx for presen… Summits!

Tuned examples for most of our algorithms on popular environments.

master ▾ ray / rllib / tuned_examples / ppo /

Go to file | Add file ▾ | ···

sven1977 [RLlib] Slate-Q tf implementation and tests/benchmarks. (#22389) ✕ 6522935 17 days ago History

..

atari-ddppo.yaml	[RLlib] Deprecate `vf_share_layers` in top-level PPO/MAML/MB-MPO conf...	14 months ago
atari-ppo.yaml	[RLlib] Deprecate `vf_share_layers` in top-	…nths ago
cartpole-appo-vtrace-fake-gpus.yaml	[RLlib] Optionally don't drop last ts in v-trac	…nths ago
cartpole-appo-vtrace-separate-losses.yaml	[RLlib] Add all simple learning tests as `fram`	…nths ago
cartpole-appo-vtrace.yaml	[RLlib] Minor fixes/cleanups; chop_into_seq	…nths ago
cartpole-appo.yaml	[RLlib] Refactor: All tf static graph code sho	…nths ago
cartpole-ddppo.yaml	[RLlib] Auto-framework, retire `use_pytorch`	…ears ago
cartpole-grid-search-example.yaml	[RLlib] Auto-framework, retire `use_pytorch`	…ears ago
cartpole-ppo-fake-gpus.yaml	[RLlib] Move existing fake multi-GPU learnin	…nths ago
cartpole-ppo-hyperband.yaml	[RLlib] Auto-framework, retire `use_pytorch`	…ears ago
cartpole-ppo.yaml	[RLlib] Deprecate `vf_share_layers` in top-	…nths ago
frozenlake-appo-vtrace.yaml	[RLlib] Upgrade gym version to 0.21 and dep	…nths ago
halfcheetah-appo.yaml	[RLlib] Refactor: All tf static graph code sho	…nths ago
halfcheetah-ppo.yaml	[RLlib] Auto-framework, retire `use_pytorch`	…ears ago
hopper-ppo.yaml	[RLlib] Auto-framework, retire `use_pytorch`	…ears ago
humanoid-ppo-gae.yaml	[RLlib] Auto-framework, retire `use_pytorch` in favor of `framework=....	2 years ago

20 lines (20 sloc) | 506 Bytes

```yaml
cartpole-appo:
    env: CartPole-v0
    run: APPO
    stop:
        episode_reward_mean: 150
        timesteps_total: 200000
    config:
        # Works for both torch and tf.
        framework: tf
        num_envs_per_worker: 5
        num_workers: 1
        num_gpus: 0
        observation_filter: MeanStdFilter
        num_sgd_iter: 6
        vf_loss_coeff: 0.01
        vtrace: false
        model:
            fcnet_hiddens: [32]
            fcnet_activation: linear
            vf_share_layers: true
```

TWO SIGMA

$ git clone https://gi…
$ cd rllib_tutorials/p…
$ jupyter-lab

…ale

RLlib

And why should I use RLlib?

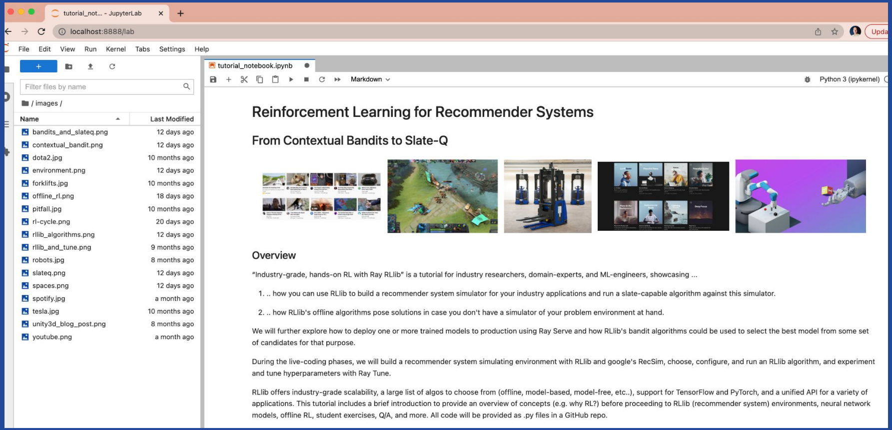
Because these companies here do!
Thx for presenting at Ray- and Production RL Summits!

RIOT GAMES
Microsoft
pathmind
SIEMENS
amazon web services
TWO SIGMA
WILD LIFE
minds.ai
Enterprise AI Consulting
QUANTUMBLACK
A MCKINSEY COMPANY
NASA

```
$ git clone https://github.com/sven1977/rllib_tutorials
$ cd rllib_tutorials/production_rl_summit_2022
$ jupyter-lab
```

anyscale

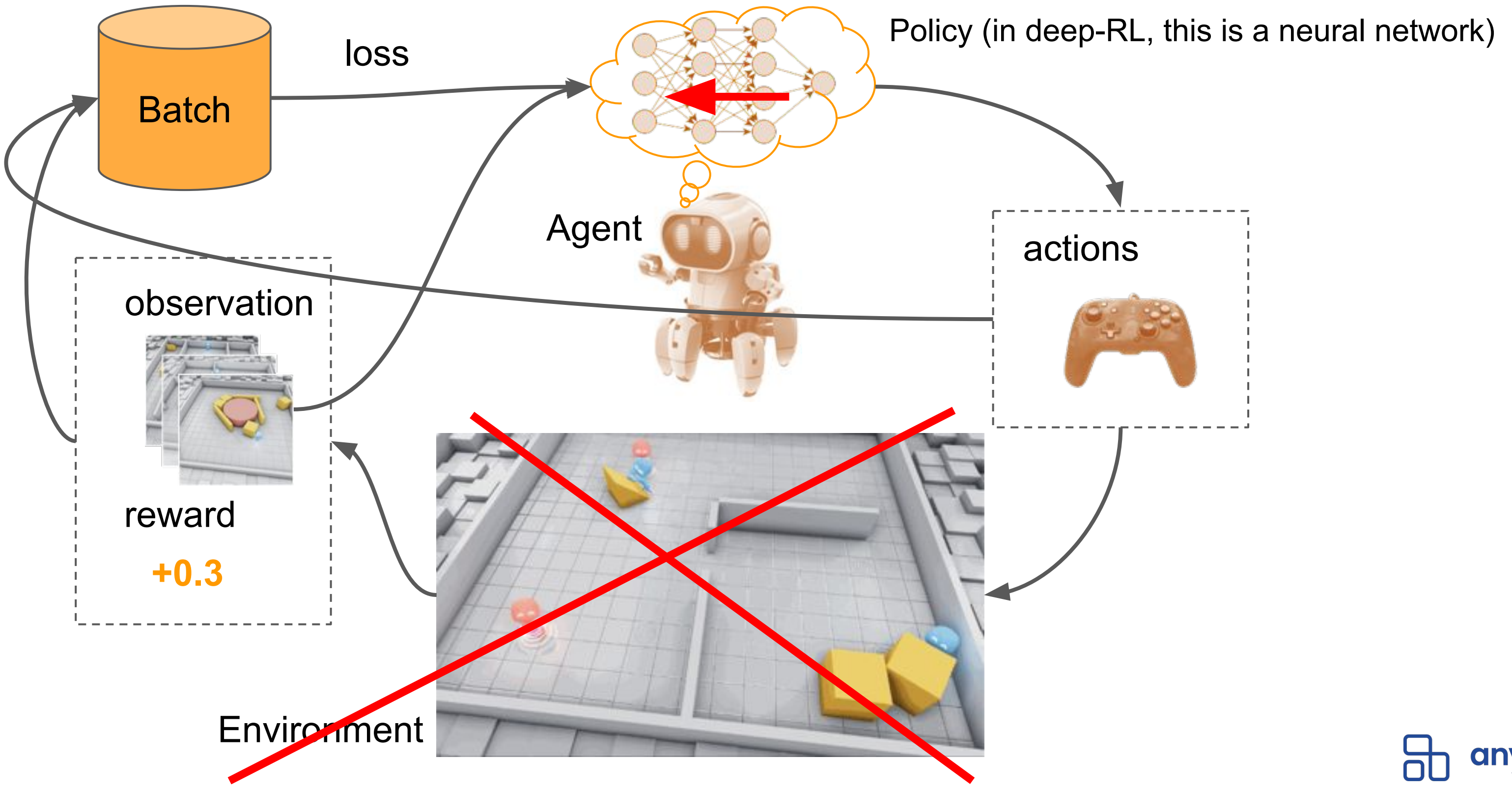
5 min Break :)
Then … moving to our Jupyter Notebook



anyscale

What’s Offline Reinforcement Learning?

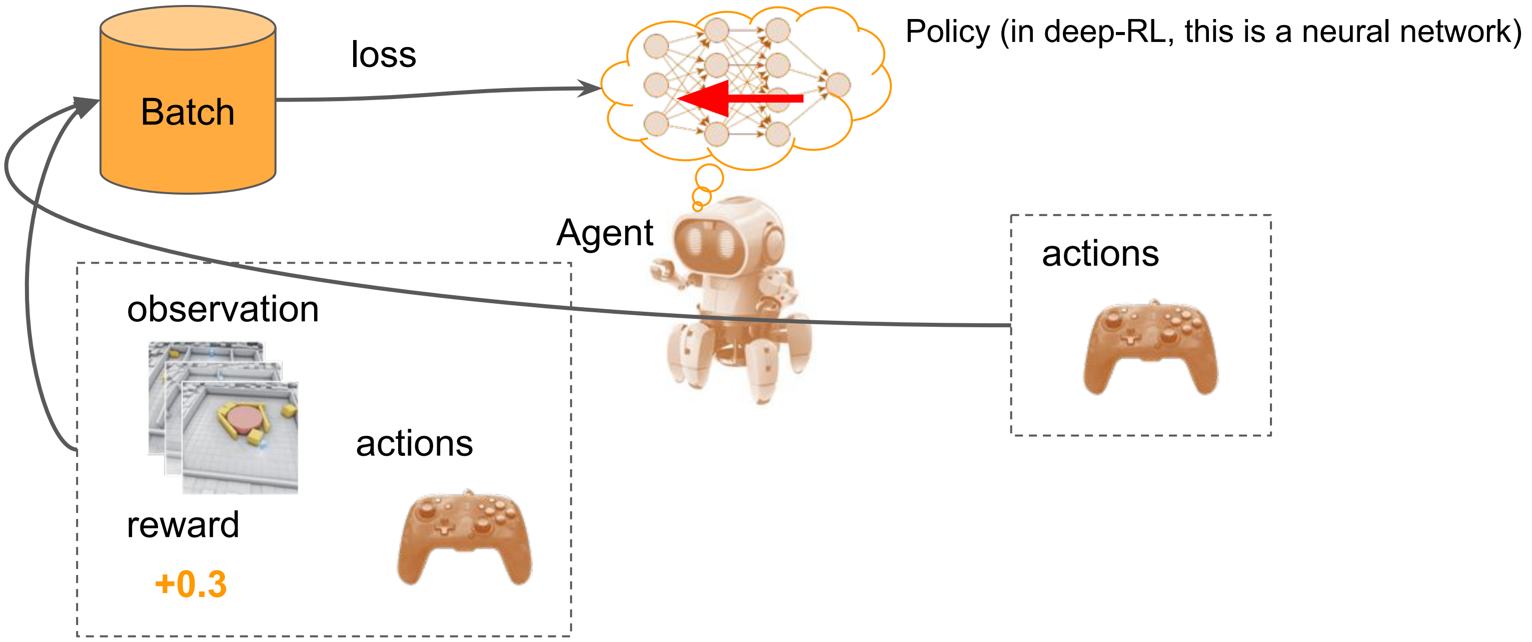
Aka: “Batch RL”



anyscale

What's Offline Reinforcement Learning?

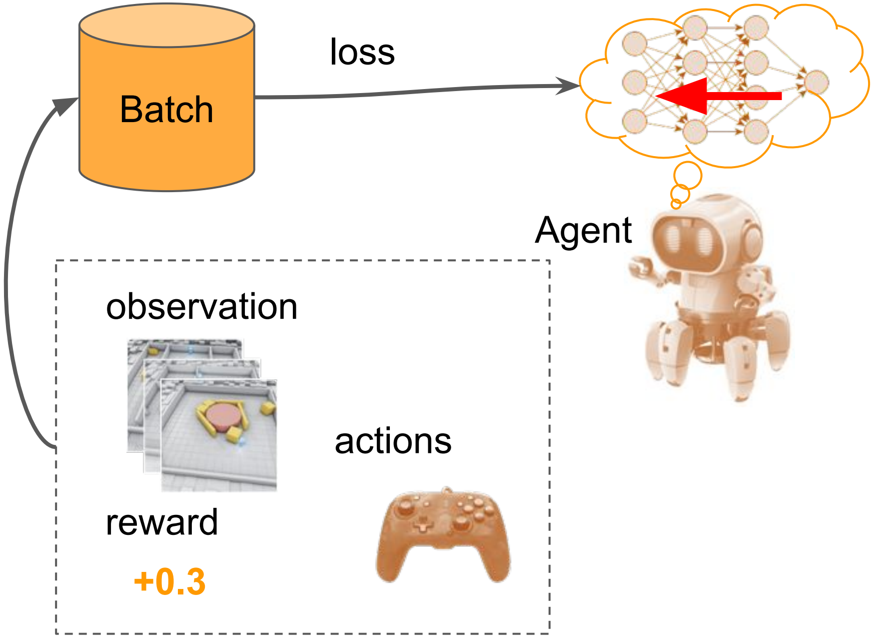
Aka: Batch RL



Historic data recorded in the past (e.g. a JSON file!).

What’s Offline Reinforcement Learning?

Aka: Batch RL



Policy (in deep-RL, this is a neural network)

Historic data recorded in the past (e.g. a JSON file!).

<u>2 ways of doing this:</u>

- Behavioral Cloning (BC), aka: “Imitation learning”: loss = $-\log(p(a_h))$
  - Don’t care about rewards.
  - Pure SL: Observations=inputs; actions=labels
- Offline RL: Don’t only imitate the historic policy, but also try to improve over it.
  - Rewards are needed for computing losses.