Reincarnating RL: Reusing Prior Computation to Accelerate Progress

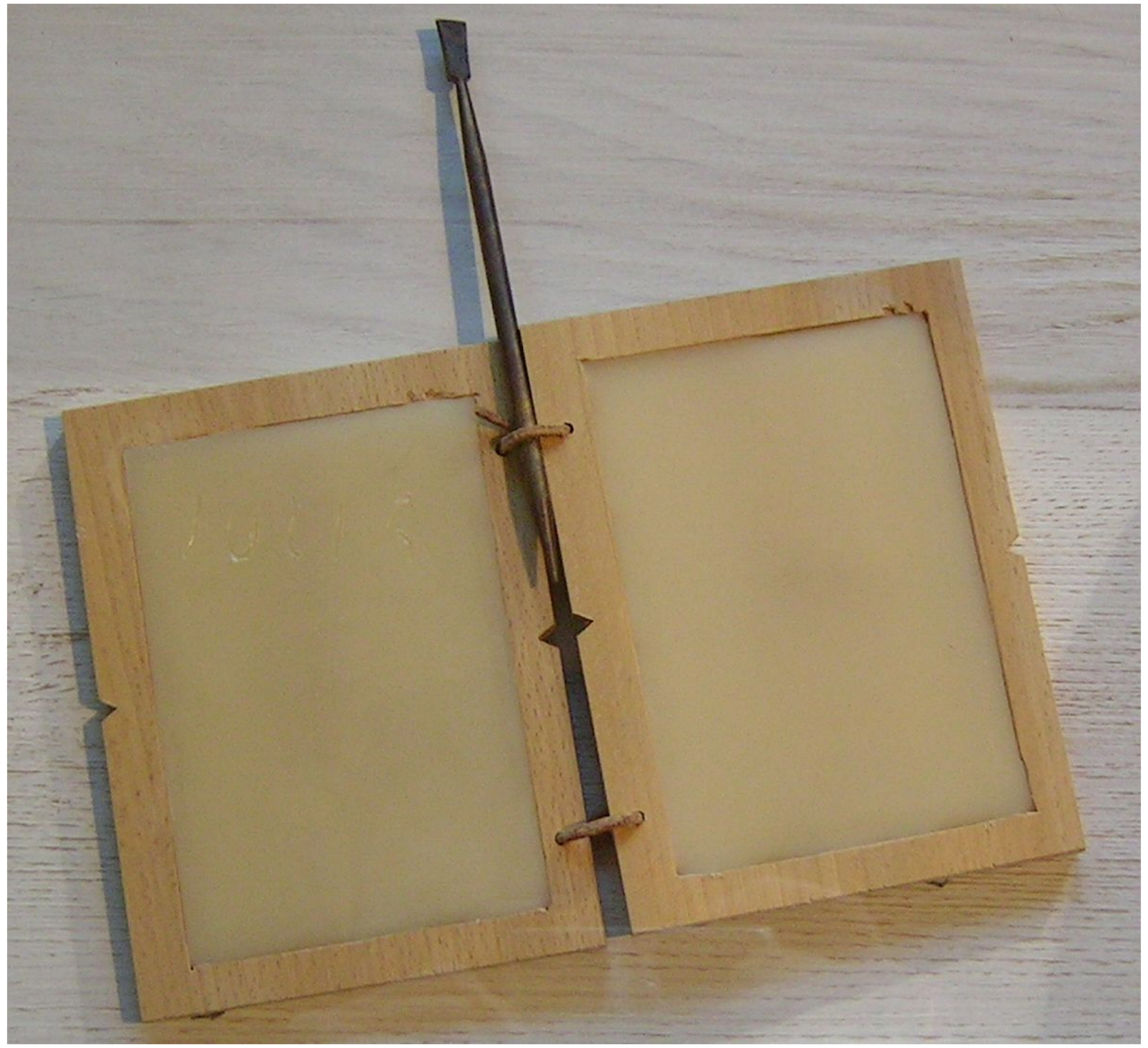
NeurIPS 2022

agarwl.github.io/reincarnating_rl

Google Research   Mila

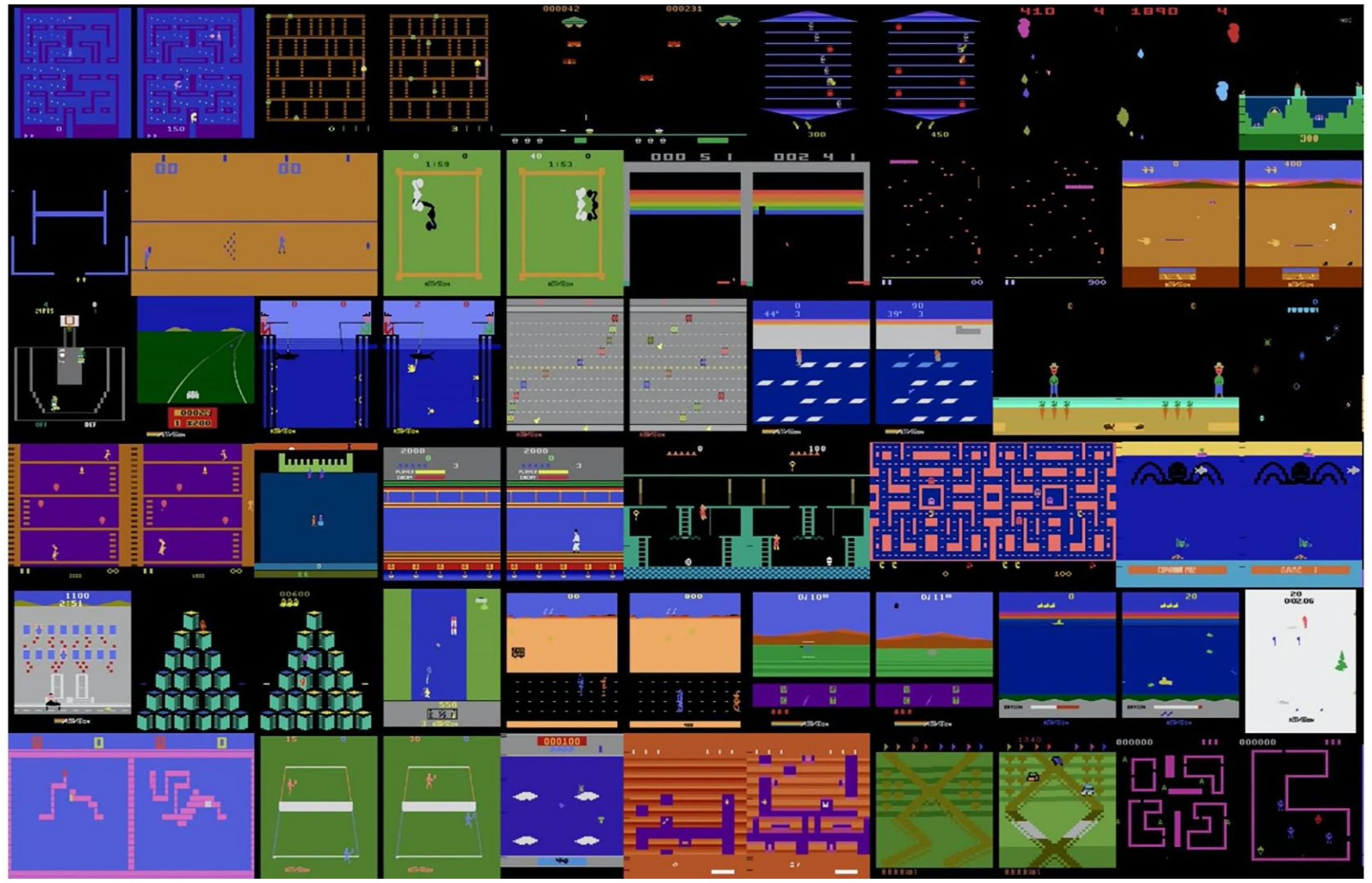
Tabula rasa Reinforcement Learning

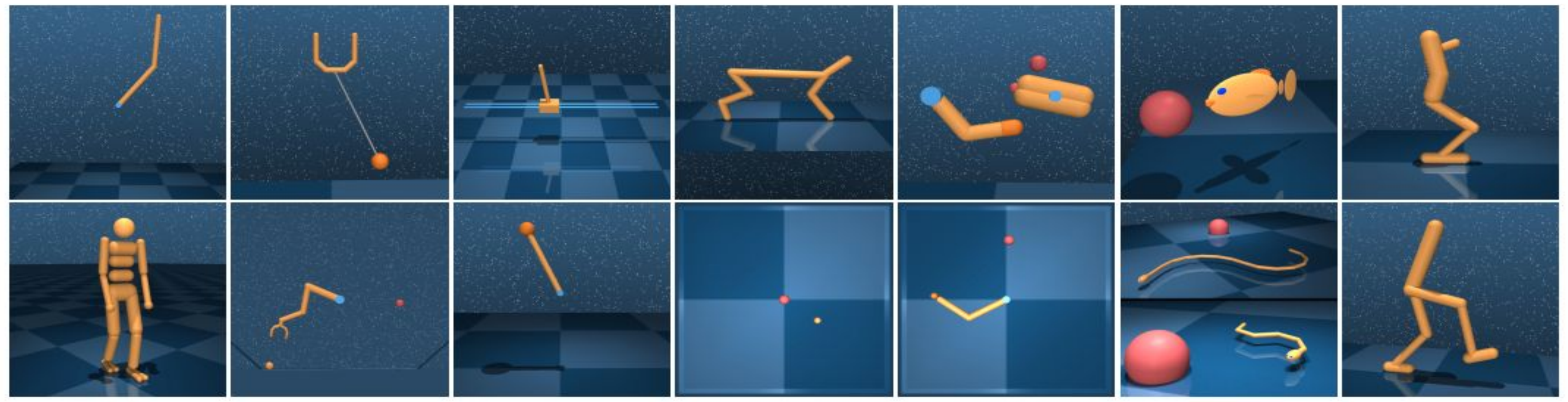


Clean or Blank state: "Learning from scratch"

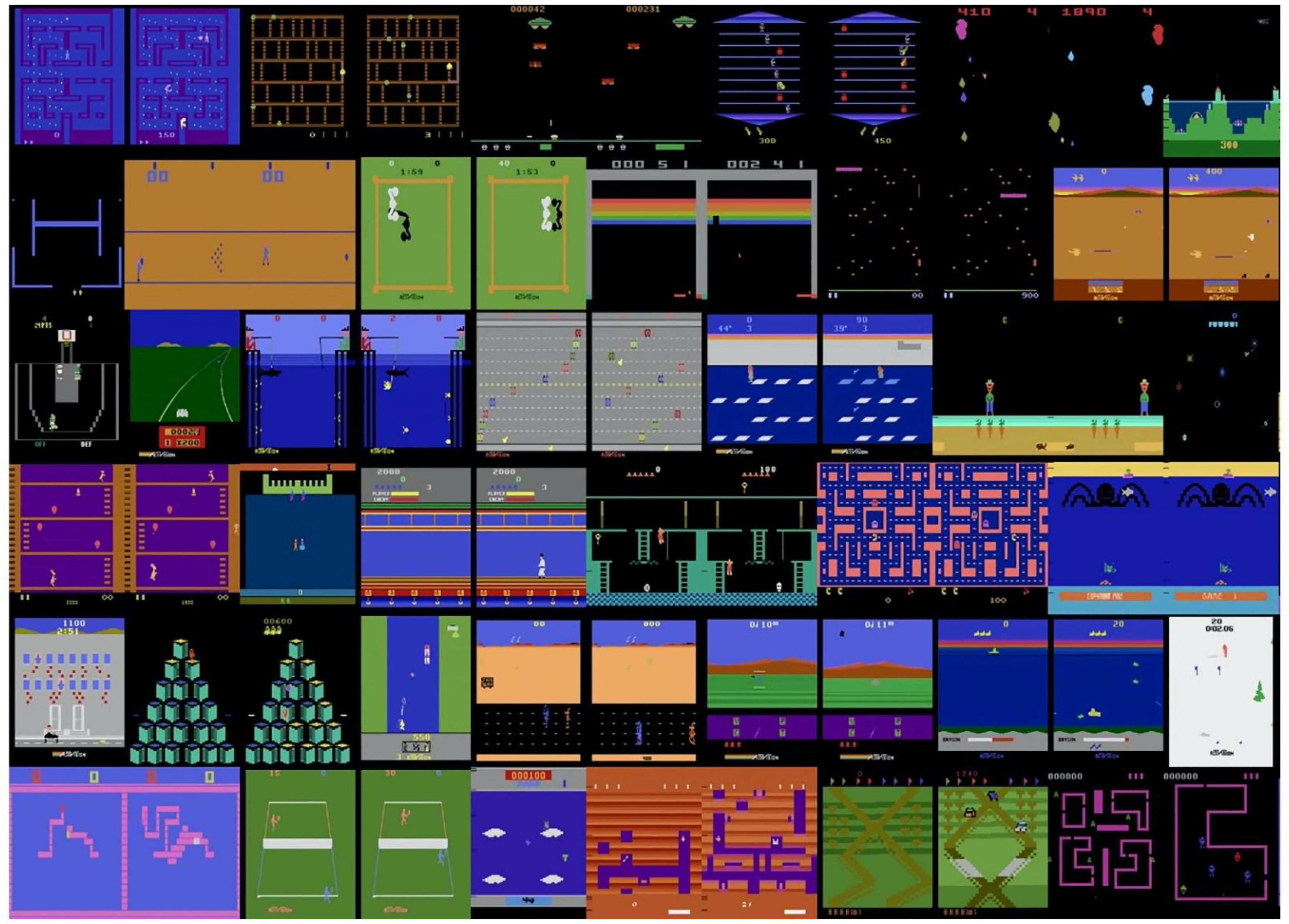
bit.ly/reincarnating_rl

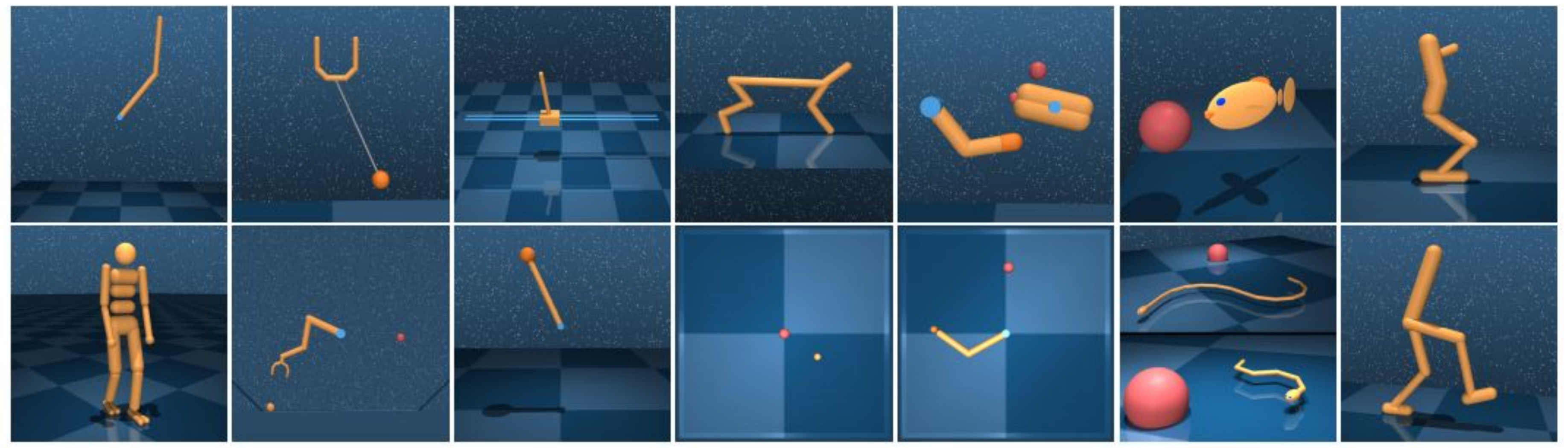
Tabula Rasa RL works for research domains.



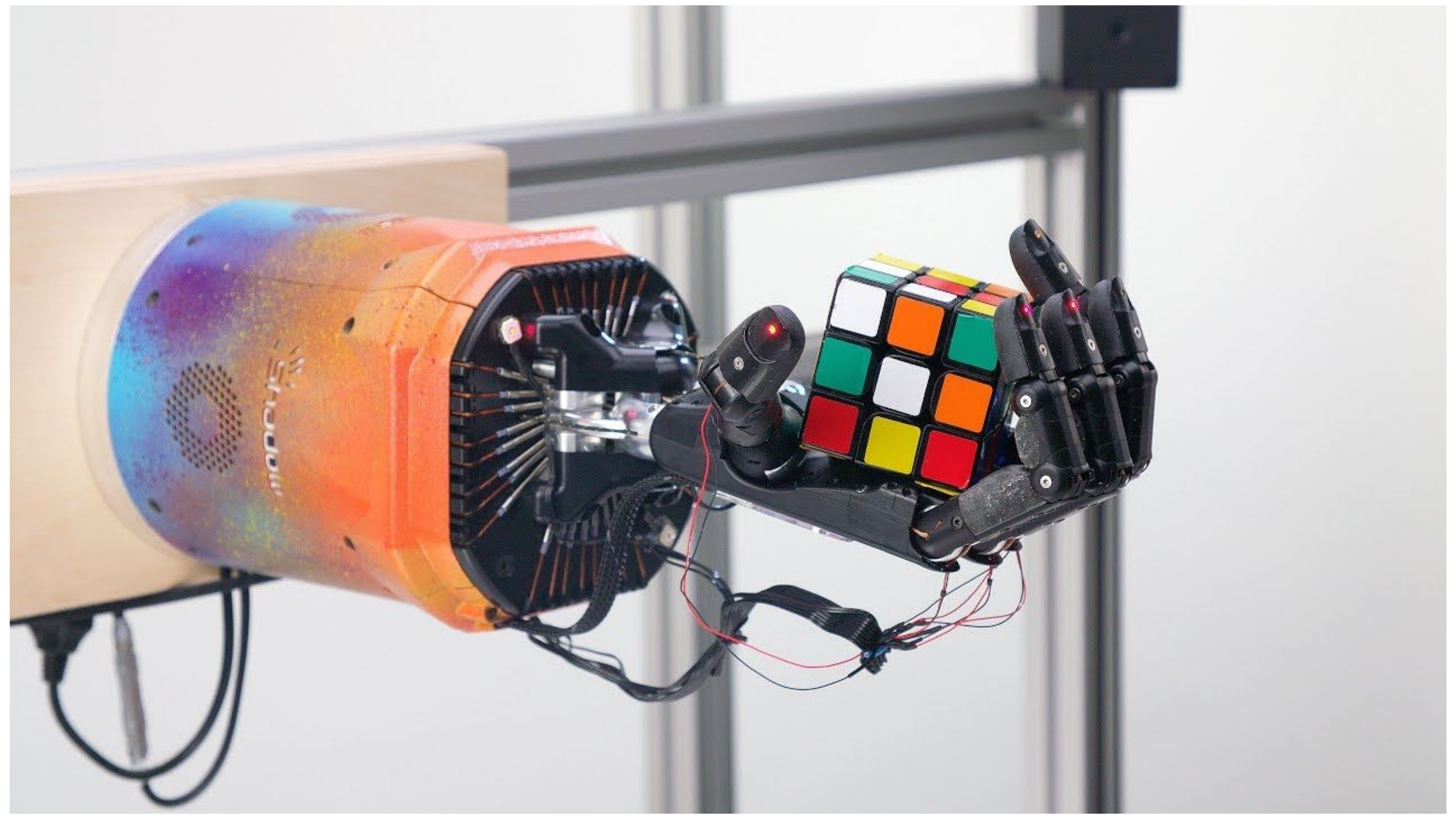


Large-scale RL problems: ~~Tabula rasa~~ workflow



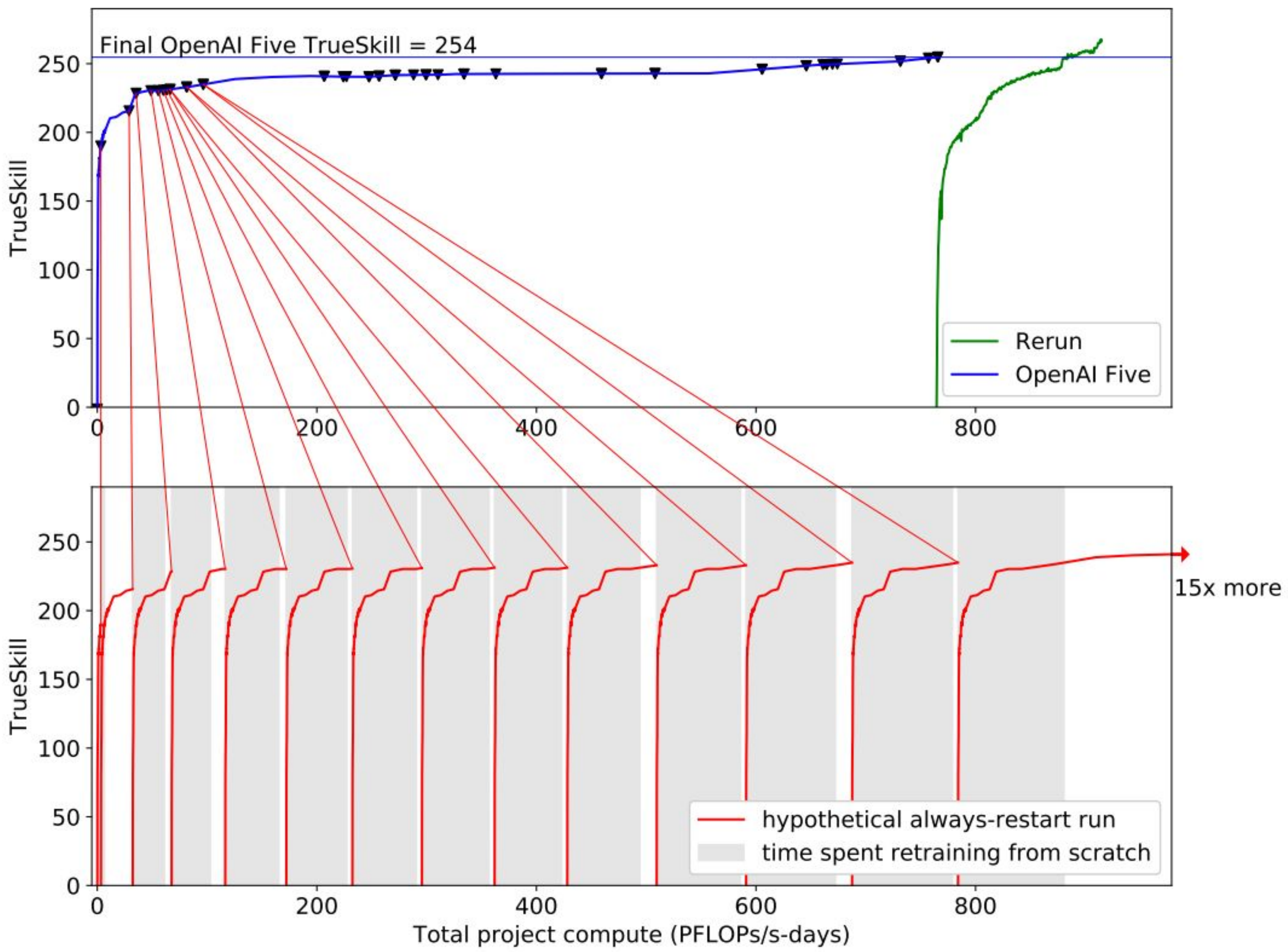


Works well here.



Not so much here.

~~Tabula rasa~~ RL Playing DOTA with large-scale RL training

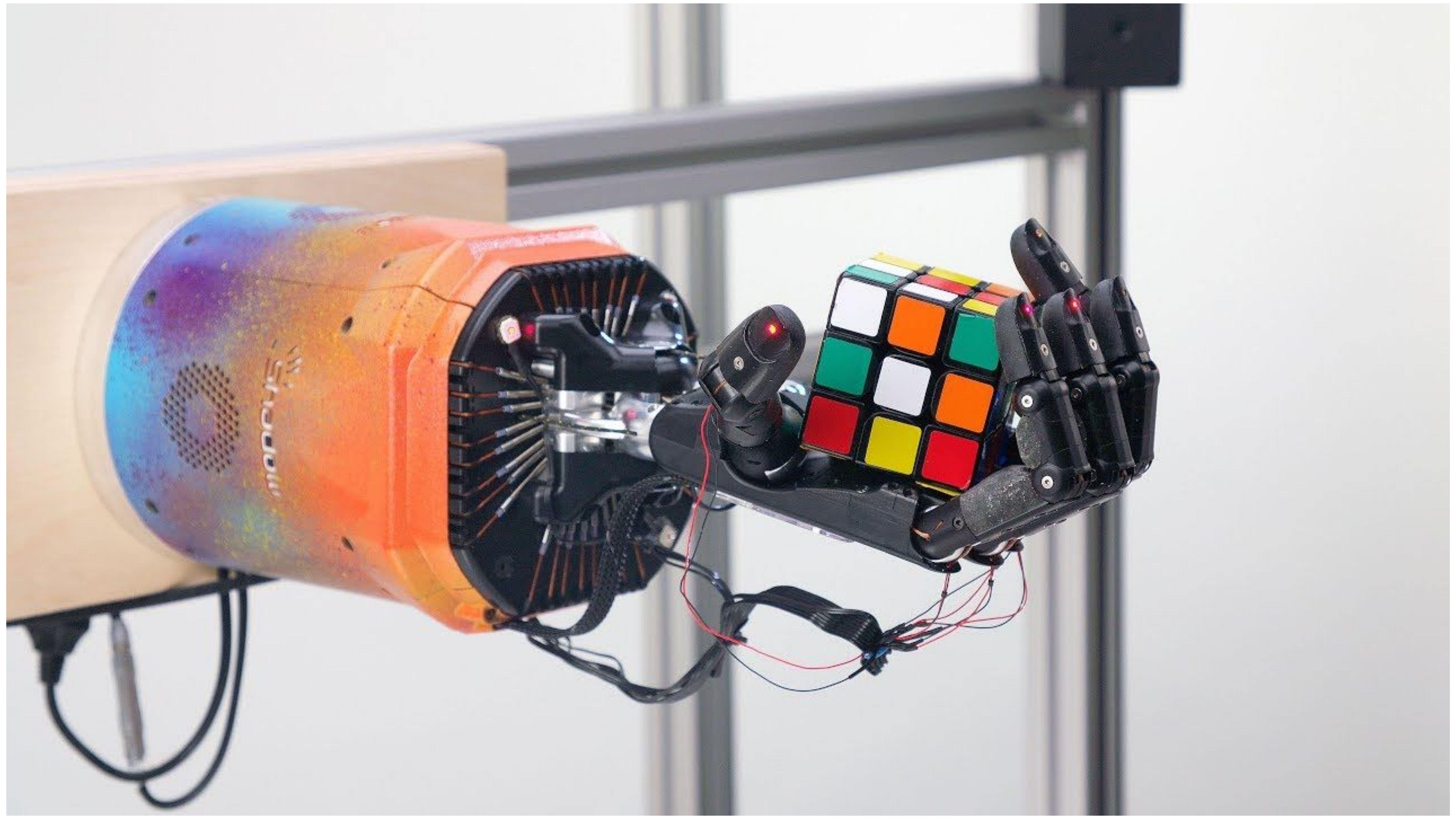


Actual learning curve (10 months)

Restarting from scratch every time (~40 months)

Berner, Christopher, et al. "Dota 2 with large scale deep reinforcement learning." *arXiv preprint arXiv:1912.06680* (2019).

~~Tabula rasa RL~~ Solving Rubik's cube with a robot hand

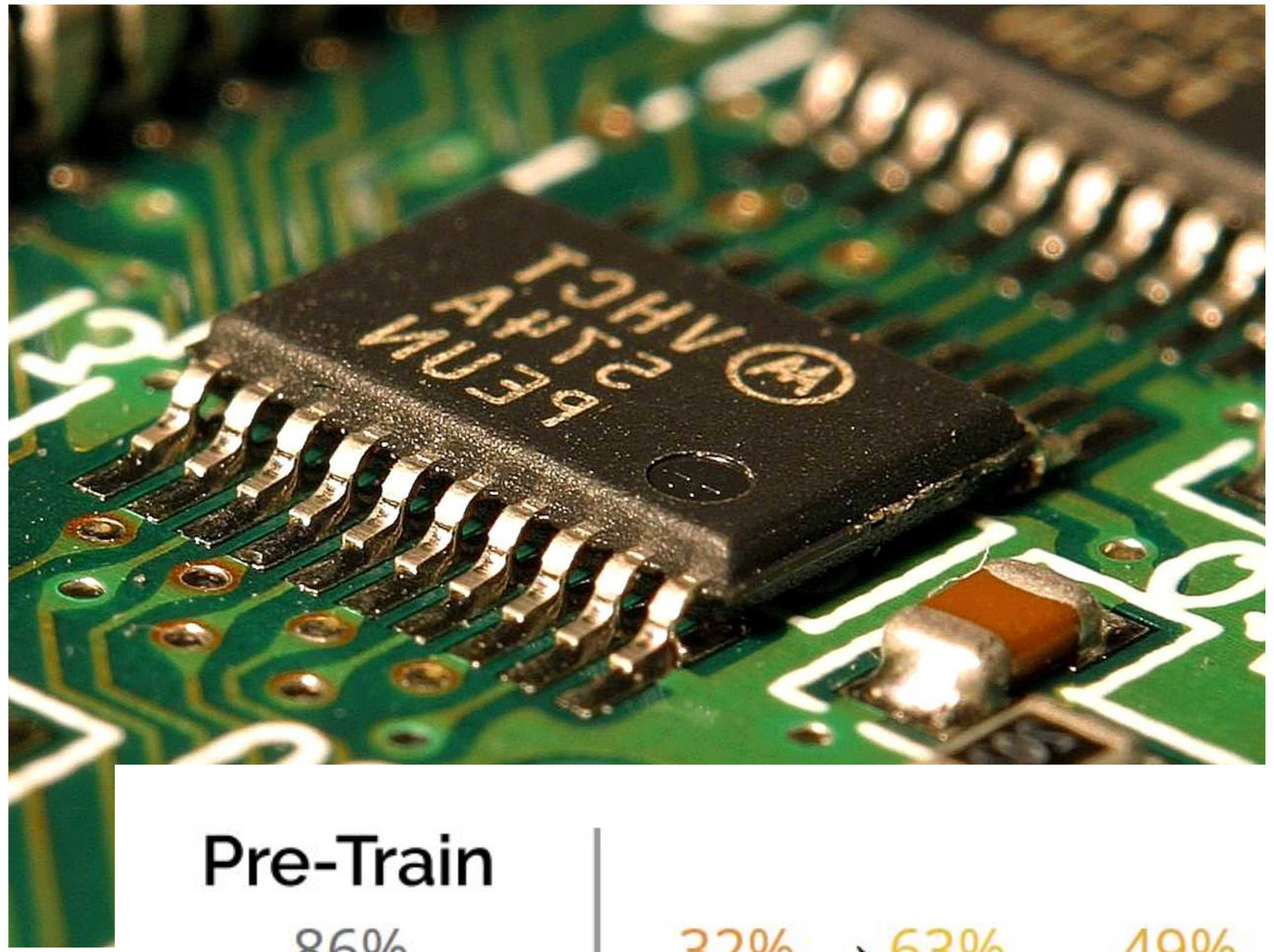


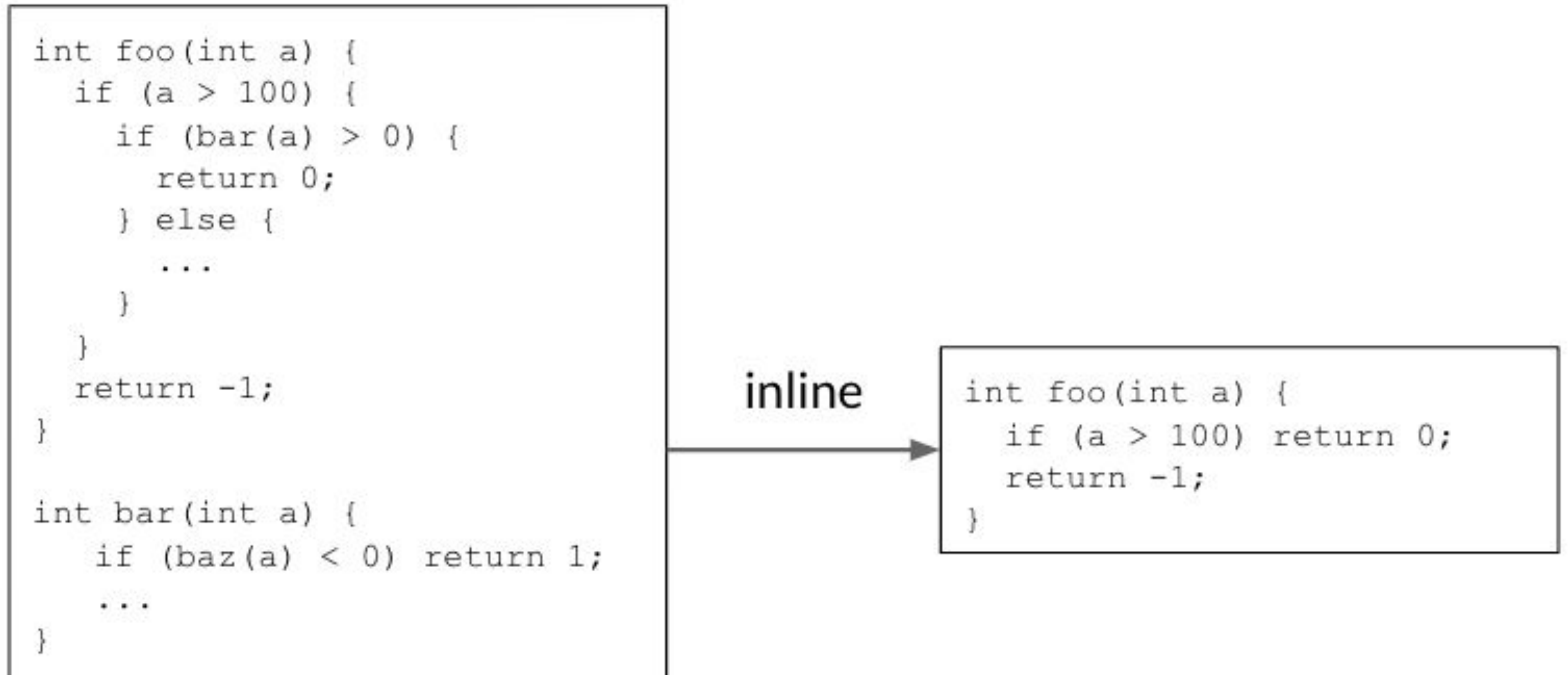
"We rarely trained experiments from scratch ..

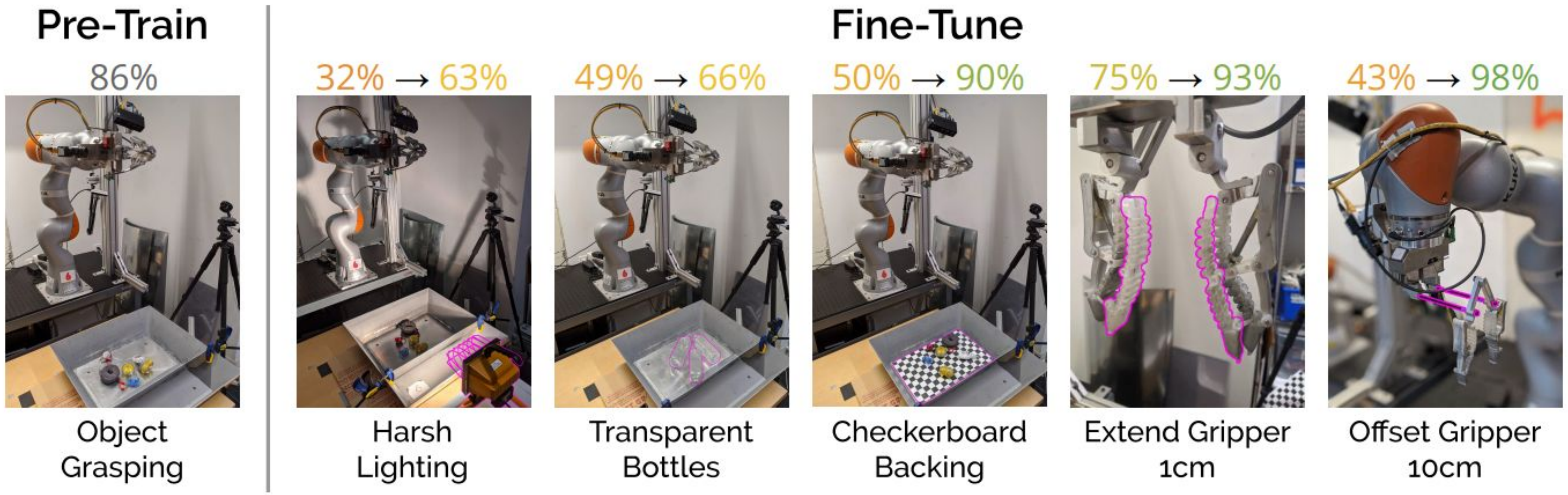
Restarting training from an uninitialized model would have caused us to lose weeks or months of training progress."

OpenAI, et al. "Solving rubik's cube with a robot hand." *arXiv preprint arXiv:1910.07113* 10 (2019).

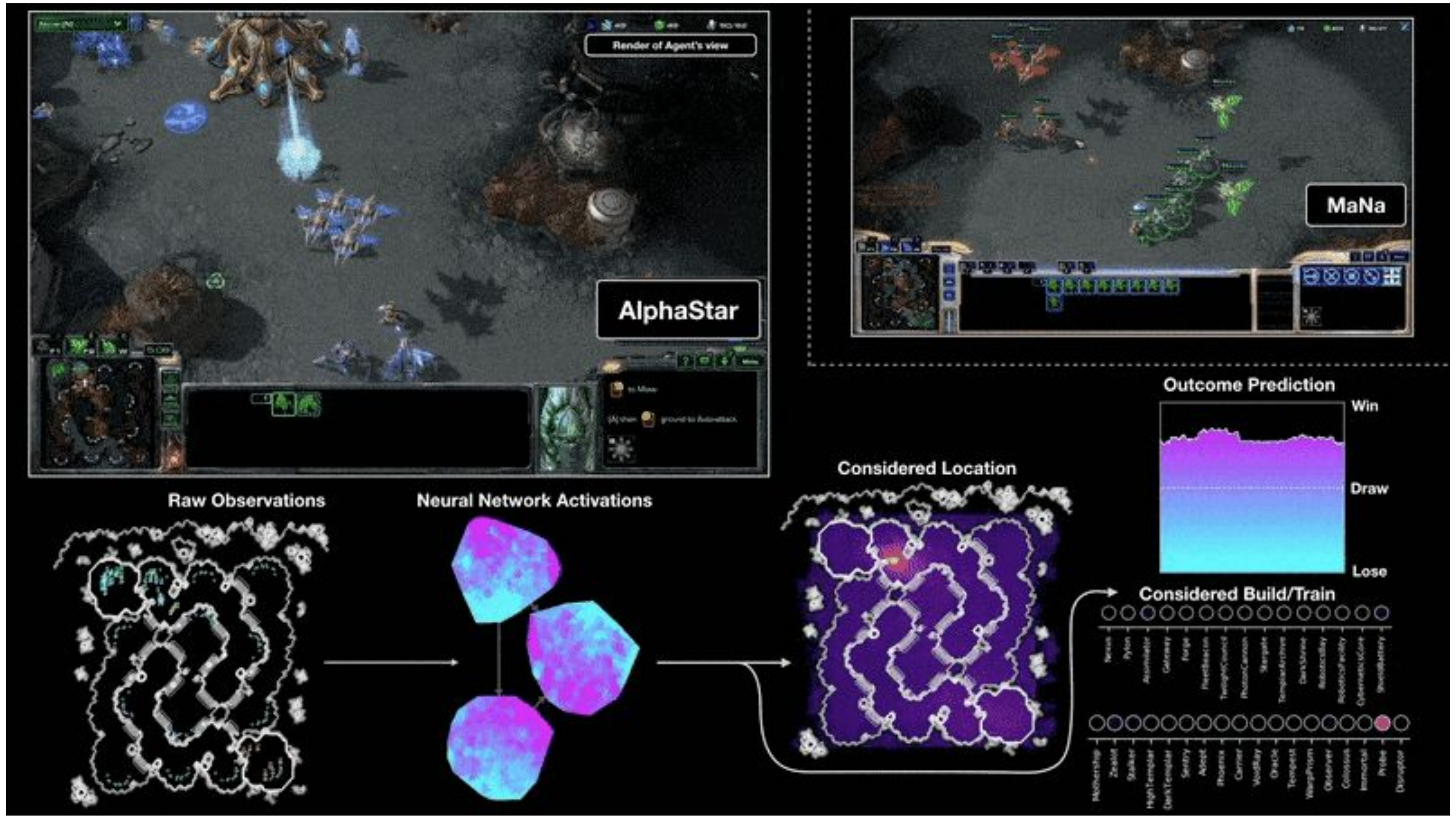
~~Tabula rasa RL~~ Fine-tuning with RL







Deep RL is computationally expensive :(

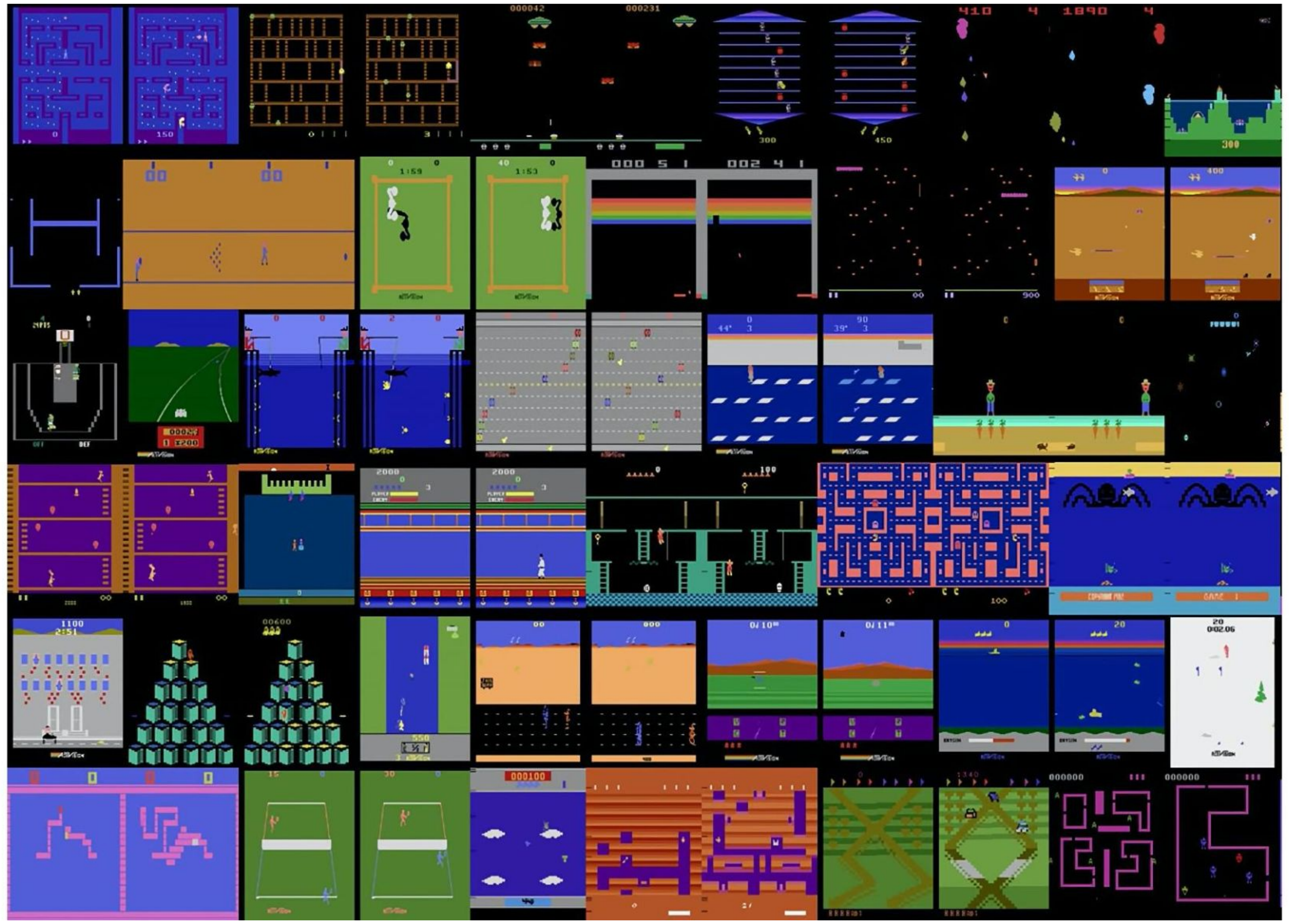


Vinyals, Oriol, et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning." *Nature* 575.7782 (2019): 350-354.

AlphaStar: Trained on several TPUs for a month. Replication would cost > **$1,000,000**.

Excludes most researchers outside resource-rich labs.

Deep RL is computationally expensive :(



Training 5 runs on 50+ Atari games for 200M frames (standard protocol) requires at least **1000+ GPU** days.

Excludes most researchers outside resource-rich labs.

What if we didn't always train RL agents from scratch for research?

Reincarnating RL: An alternative workflow

Reincarnating RL: An alternative workflow

"Prior computational work, such as learned network weights and policies, should be maximally leveraged."

Reincarnating RL: An alternative workflow

Let's say you trained an agent A_1 for a long time (e.g., days/weeks)

Experiment with better algorithms / architectures

Training another agent from scratch

(Tabula Rasa)

Reincarnating RL: An alternative workflow

Let's say you trained an agent A_1 for a long time (e.g., days/weeks) → Experiment with better algorithms / architectures

→ Training another agent from scratch

(Tabula Rasa)

→ Fine-tuning A_1

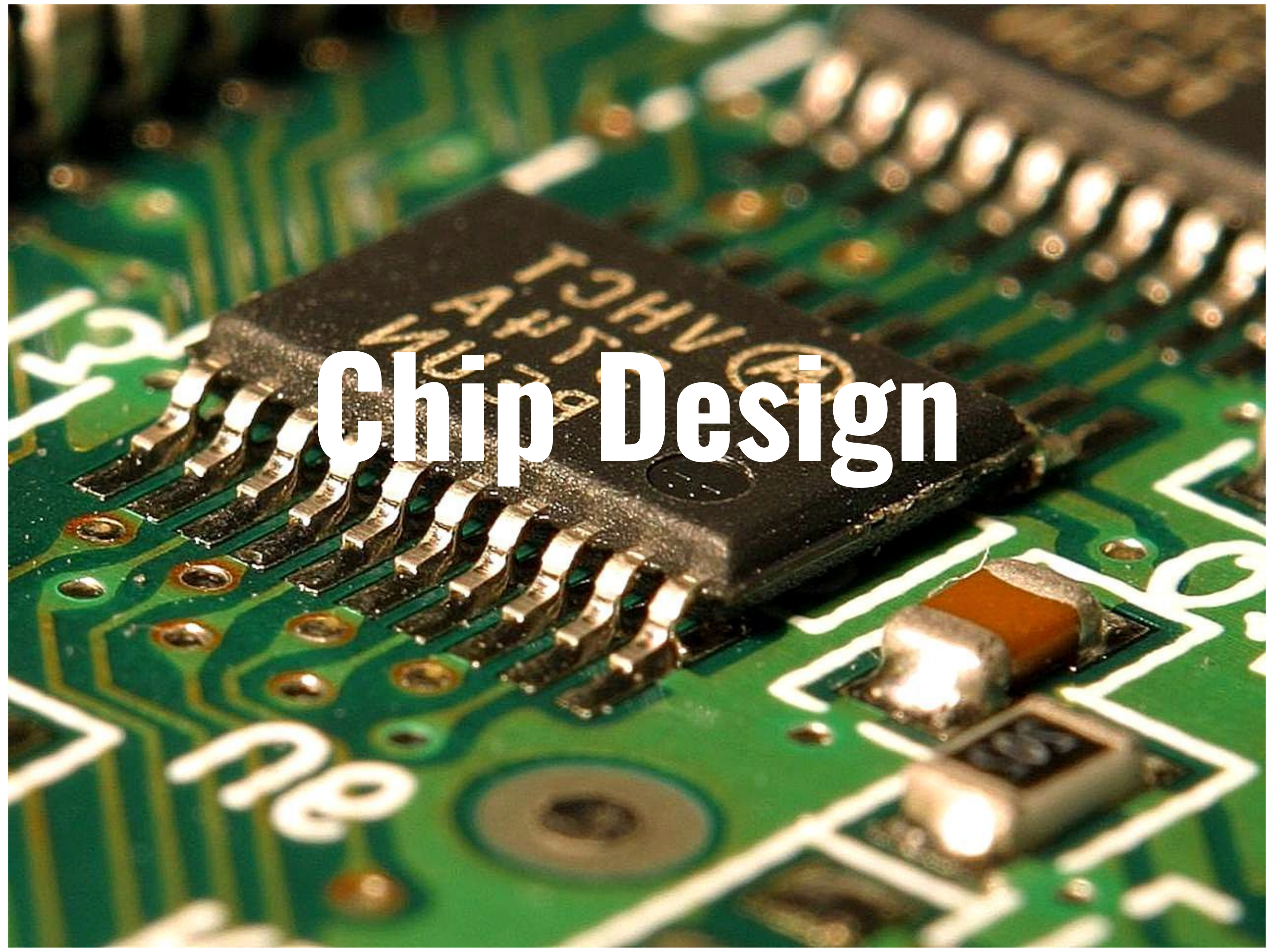
→ Transferring A_1 to another agent and training that agent further

Why Reincarnating RL?

- More compute and sample-efficient

Why Reincarnating RL?

- More compute and sample-efficient
- Tackle challenging problems without excessive computational resources
- Allows for continually updating/training agents



Why Reincarnating RL?

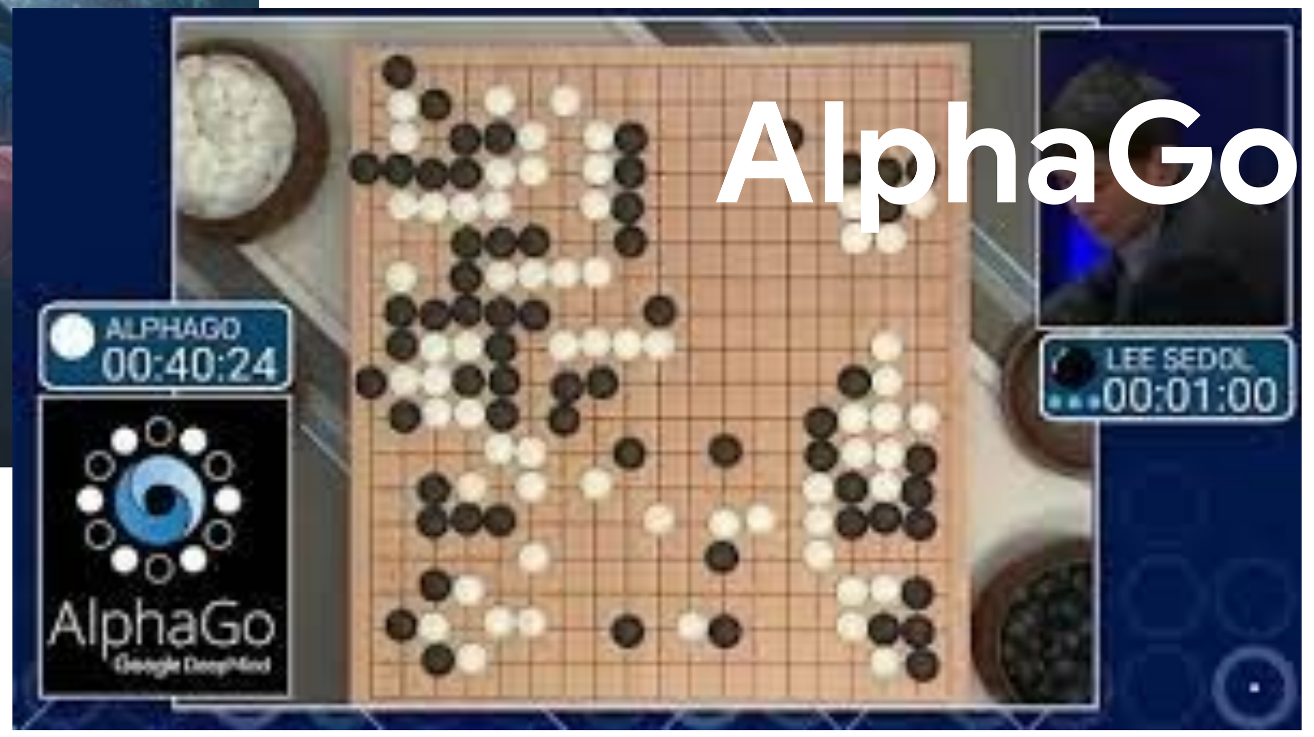
- More compute and sample-efficient
- Tackle challenging problems without excessive computational resources
- Allows for continually updating/training agents
- Suitable for real-world applications (prior computation is typically available)

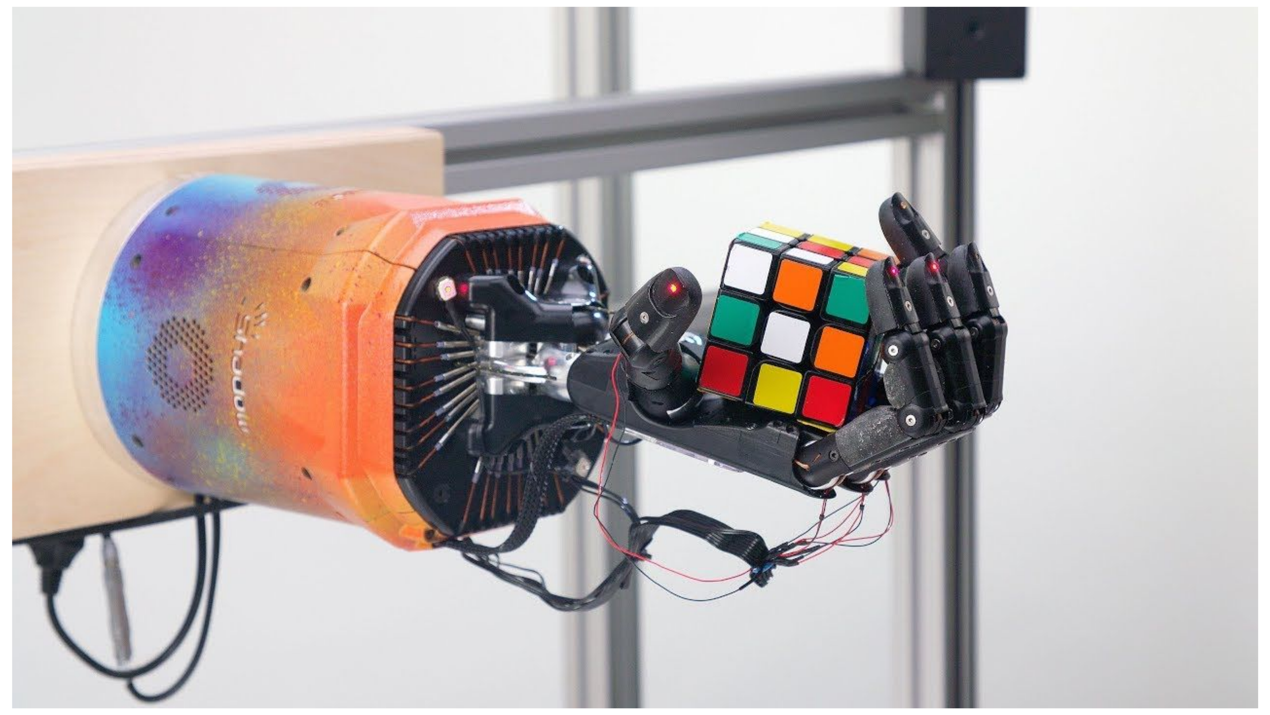
Why Reincarnating RL?

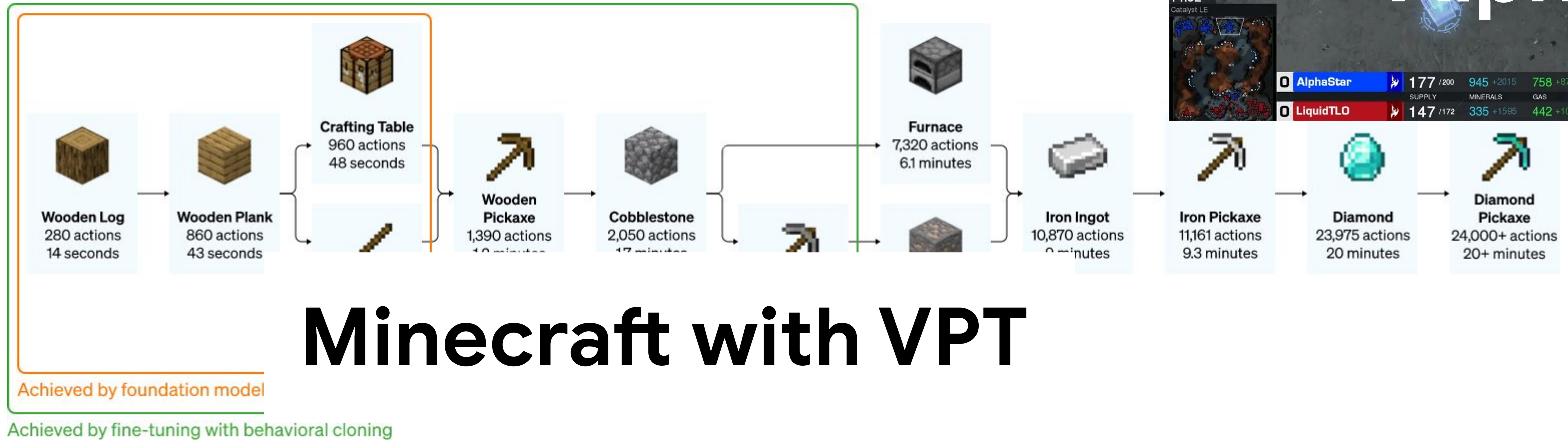
- More compute and sample-efficient
- Tackle challenging problems without excessive computational resources

Ad-hoc reincarnation strategies common in large-scale RL

Reincarnating RL ~~common~~ rare in typical papers



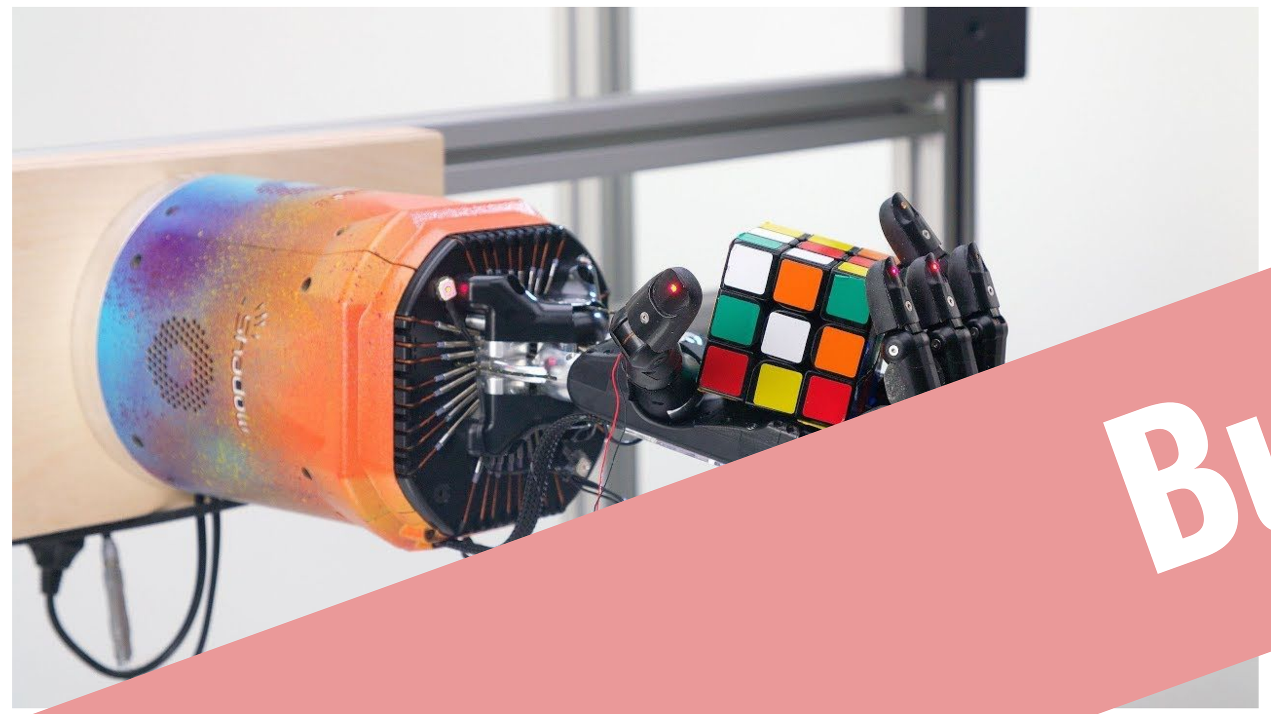


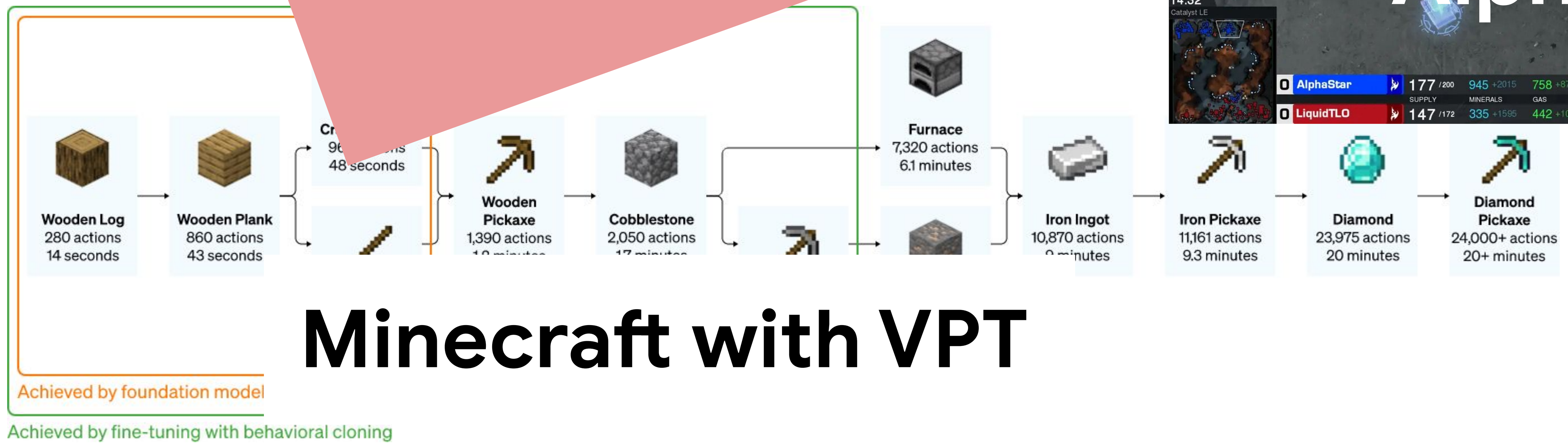


Minecraft with VPT

Ad-hoc reincarnation strategies common in large-scale RL

Reincarnating RL ~~common~~ **rare** in typical papers





Minecraft with VPT

But this doesn't have to be the case!

NEURAL INFORMATION PROCESSING SYSTEMS

ICLR

ICML
International Conference On Machine Learning

Reincarnating RL: What’s different?

- Lots of related work on imitation + RL, offline RL, transfer, LfD and so on ..
- Such papers typically don't focus on the incorporating such methods as a part of how we do RL research itself.
    - We still largely train Atari agents from scratch ..


Reusing Prior Computation
Learned Policies
Collected Data
Pretrained Representations
Others (e.g., LLMs, Skills)
Learned Models


Reusing Prior Computation
Learned Policies
Collected Data
Pretrained Representations
Others (e.g., LLMs, Skills)
Learned Models
Policy-based Student
Value-based Student
Model-based Student


Reusing Prior Computation
Learned Policies
Collected Data
Pretrained Representations
Others (e.g., LLMs, Skills)
Learned Models


Reusing Prior Computation
Learned Policies
Collected Data
Pretrained Representations
Others (e.g., LLMs, Skills)
Learned Models
Policy-based Student
Value-based Student
Model-based Student

A quick primer on RL

Markov Decision Process (MDP)

S - Set of States

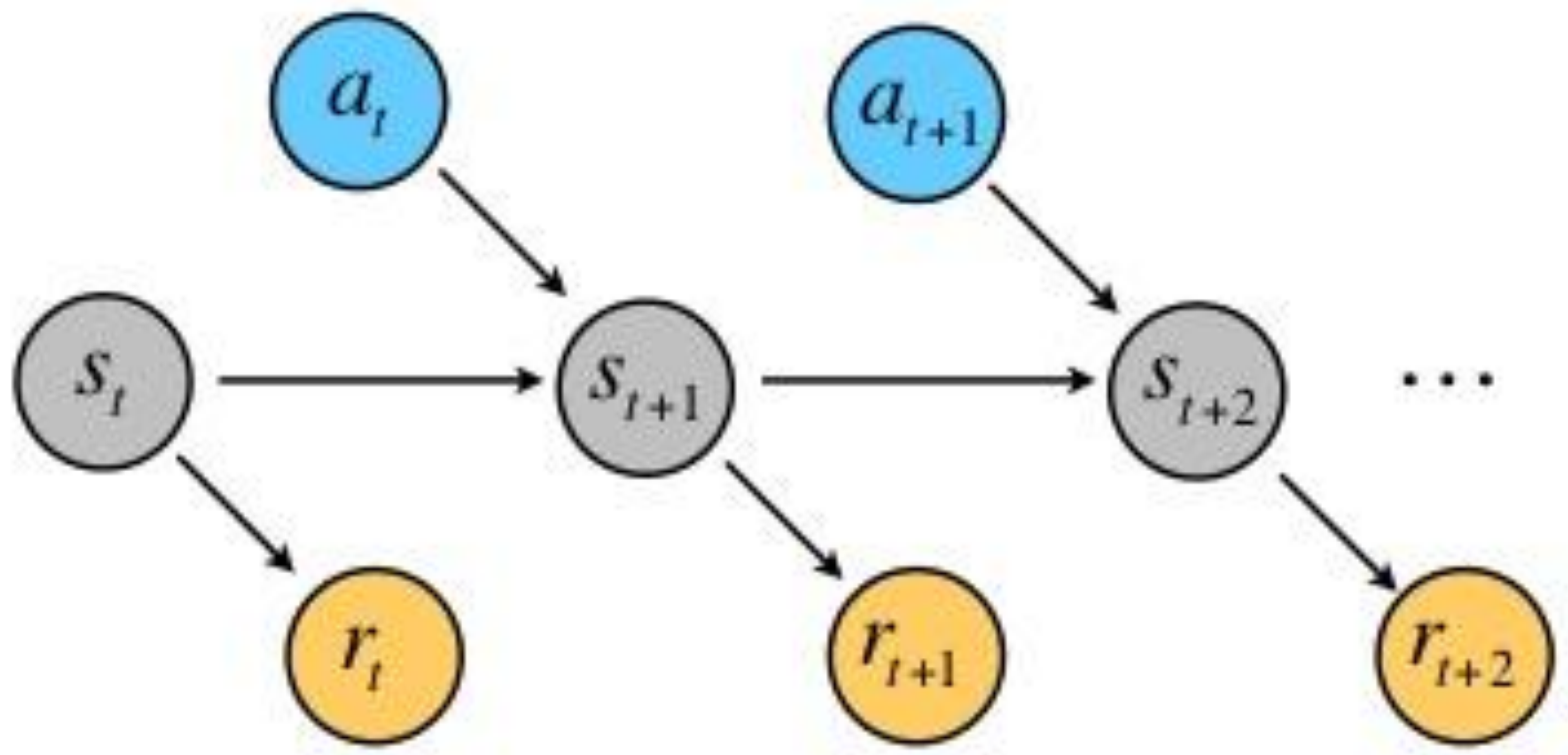
A - Set of Actions

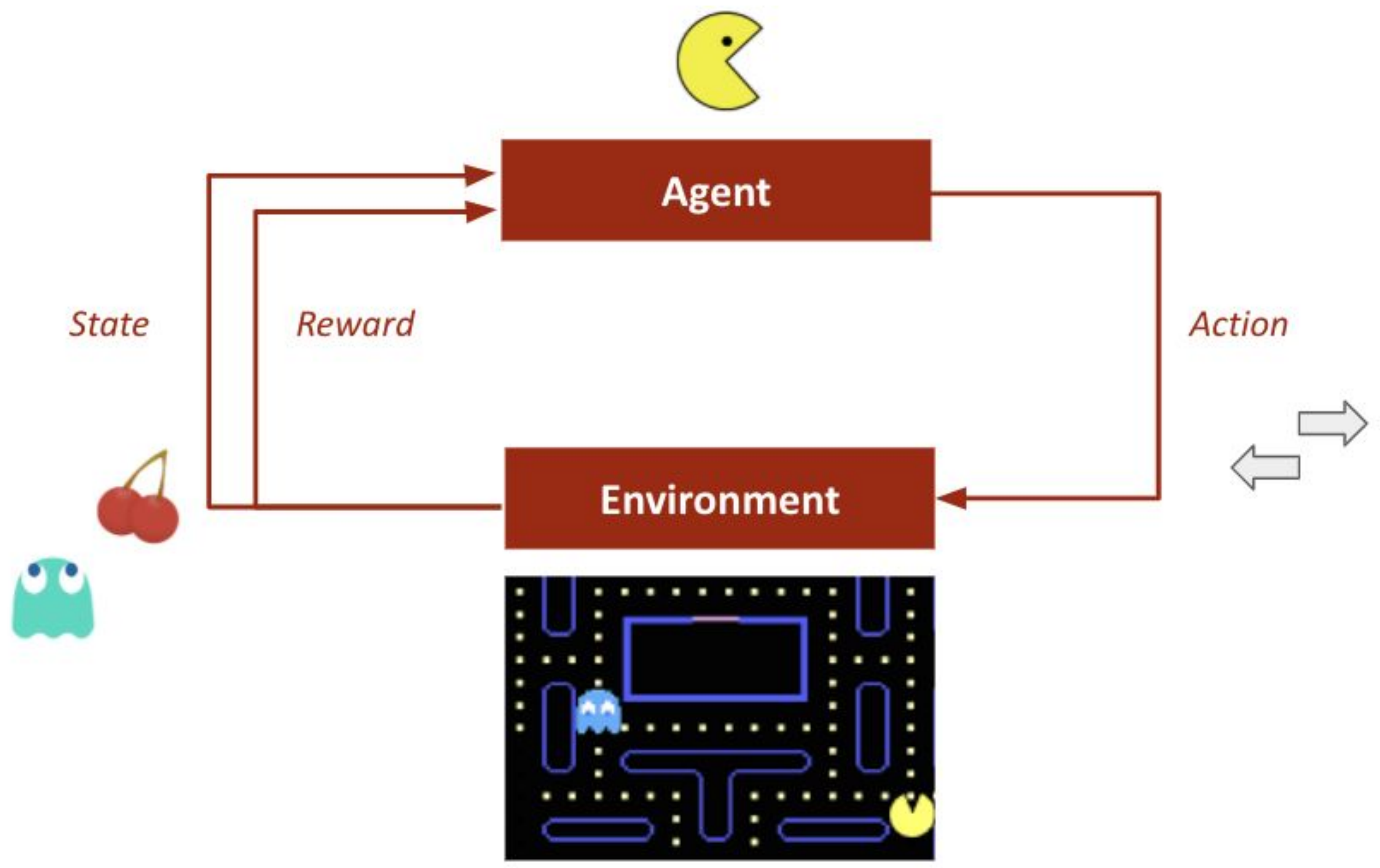
$\Pr(s' \mid a, s)$ - Transitions

α - Starting State Distribution

γ - Discount Factor

$r(s)$ - Reward [or $r(s,a)$]





Goal: $\max_{\pi} \mathbb{E}_{\pi}\left[\sum_t \gamma^t r(s_t, a_t)\right]$

$$s_t \sim P(\cdot \mid s_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|s_t)$$

A quick primer on RL

How good is a state-action pair?

The Q-function at state s and action a, is the expected cumulative reward from taking action a in state s and then following the policy π. Formally,

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_t \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a, s_t \sim P(\cdot|s_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|s_t)\right]$$

Bellman Optimality Equation

$$Q^*(s,a) := \max_{\pi} Q^{\pi}(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right]$$

Solving for the optimal policy

Q-learning: Use a function approximator to estimate the Q-function, *i.e.*

$$Q(s,a;\theta) \approx Q^*(s,a)$$

function parameters (weights)

If the function approximator is a deep neural network => Deep Q-learning!

Case Study: Policy to Value Reincarnating RL (PVRL)

$$\pi_\Phi(a|s) \longrightarrow Q_\theta(s, a)$$

Existing suboptimal teacher policy

Value-based Student (e.g., DQN, SAC)

Transfer an existing policy to a (more) sample-efficient value-based student agent.

Policy to Value Reincarnating RL (PVRL)

$$\pi_{\Phi}(a|s) \longrightarrow Q_{\theta}(s, a)$$

Suboptimal Teacher

Value-based Student

Desiderata

- **Teacher-agnostic**
  - Student shouldn't be constrained by teacher's architecture and algorithm

Policy to Value Reincarnating RL (PVRL)

$\pi_{\Phi}(a|s)$ → $Q_{\theta}(s, a)$

Suboptimal Teacher

Value-based Student

Desiderata

- Teacher-agnostic
- **Weaning off teacher**
    - Undesirable to maintain teacher dependency for successive reincarnations

Policy to Value Reincarnating RL (PVRL)

$$\pi_\Phi(a|s) \longrightarrow Q_\theta(s, a)$$

Suboptimal Teacher

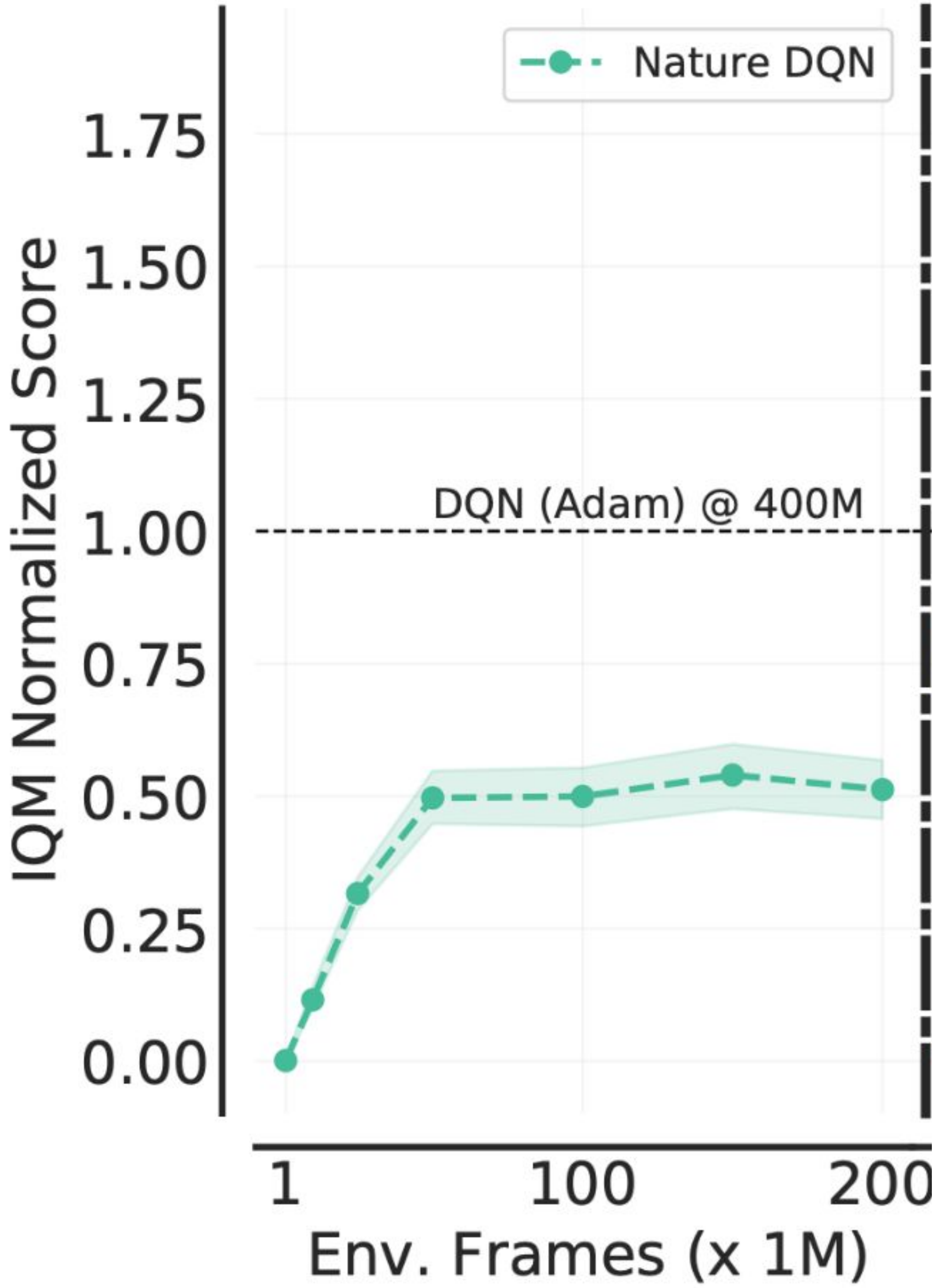
Value-based Student

Desiderata

- Teacher-agnostic
- Weaning off teacher
- **Compute Efficient**
  - Reincarnation should be cheaper than training from scratch

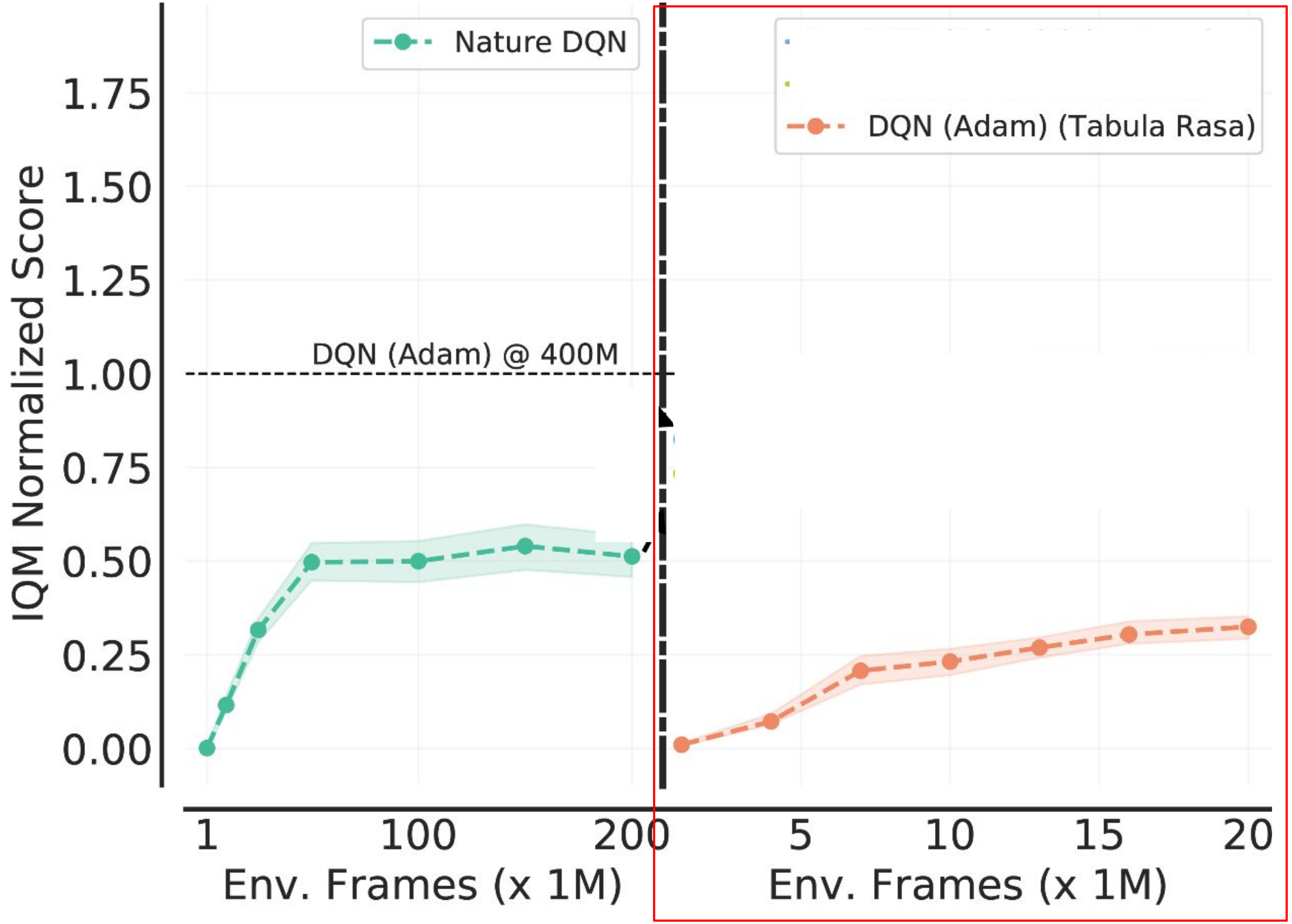
Reincarnating RL as a Research Workflow

Reincarnation on ALE



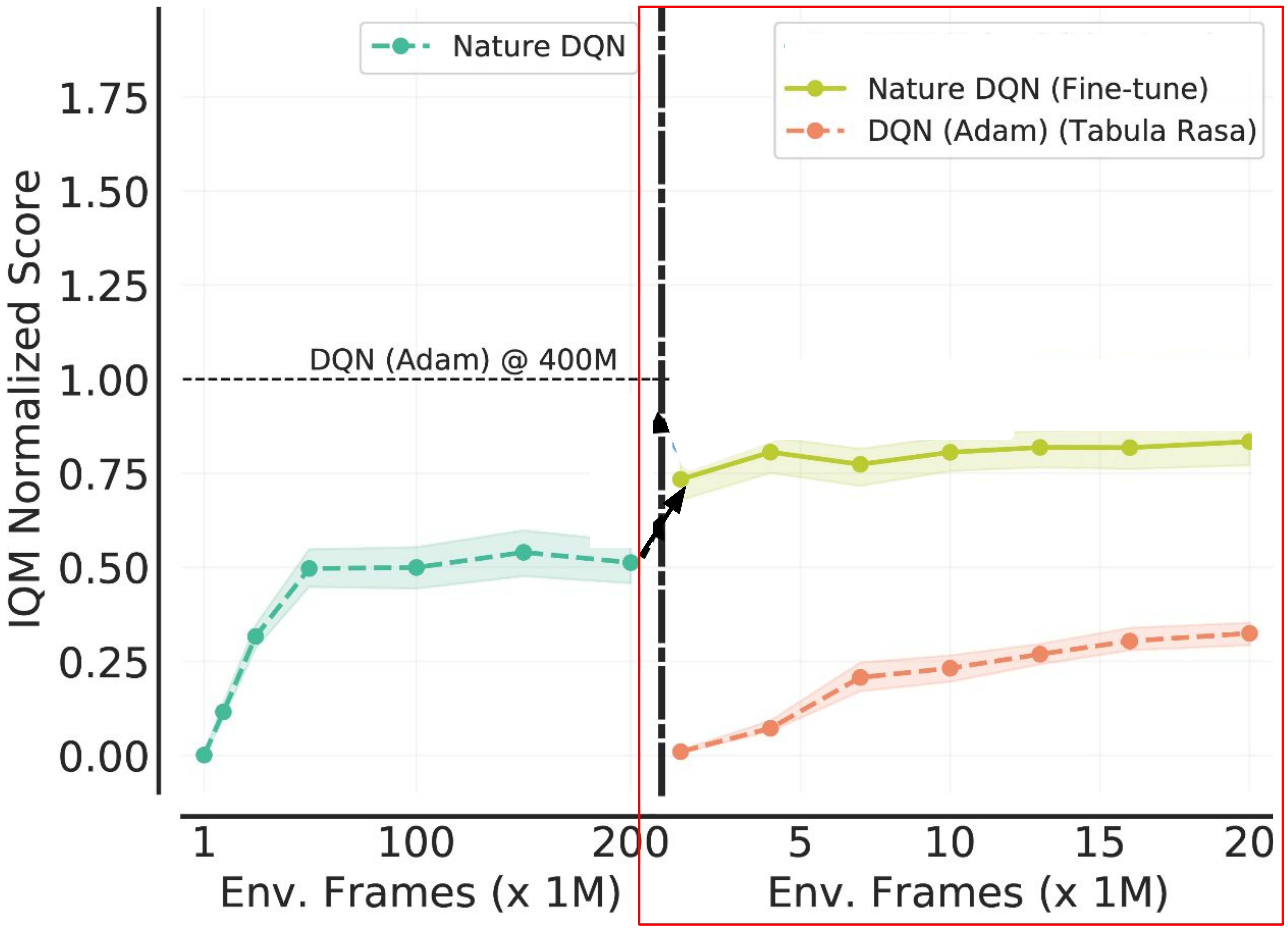
Let's assume we have access to the Nature DQN trained by Mnih et. al. (2015)

Switching optimizer to Adam



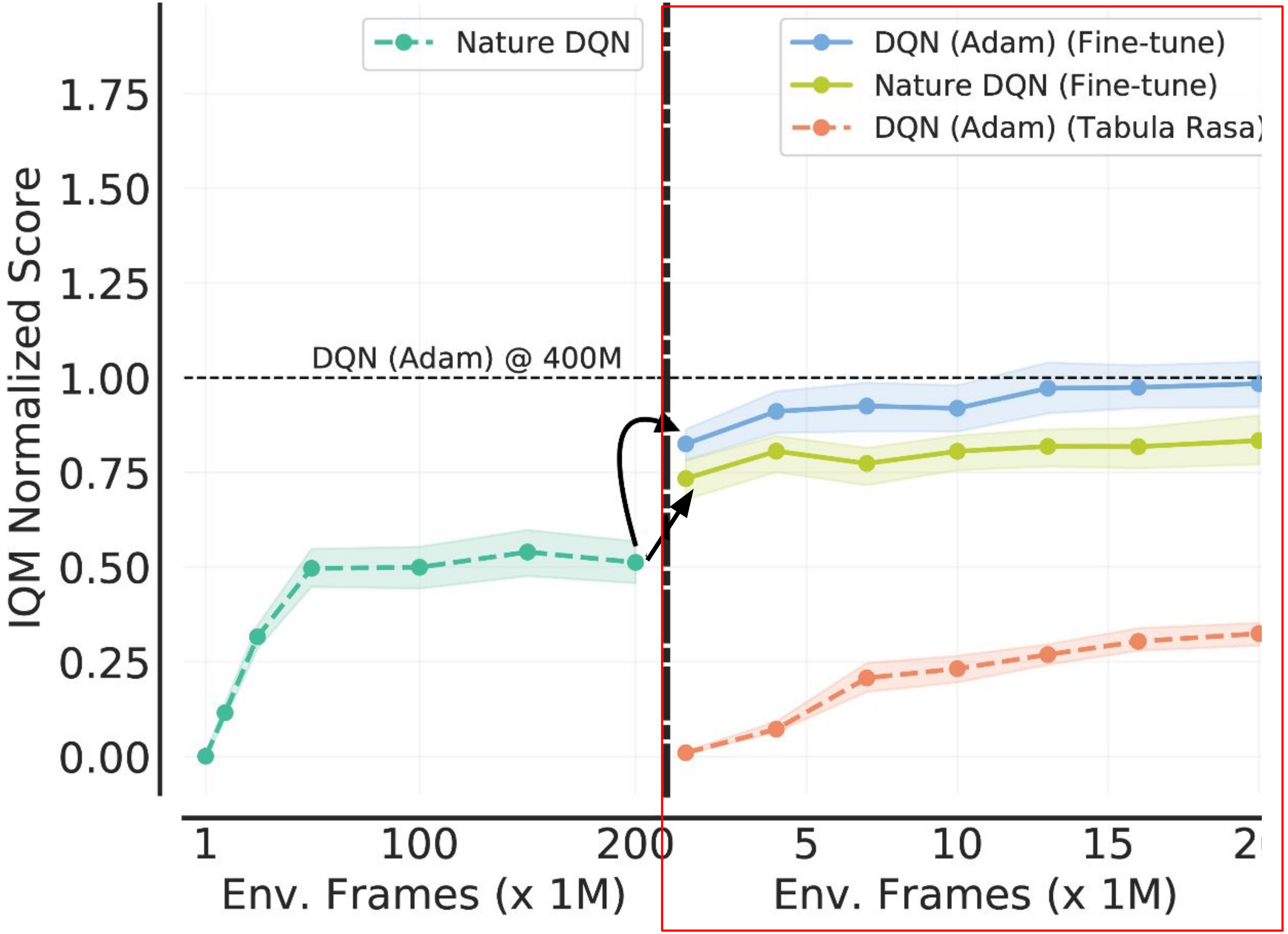
DQN (Adam) seems to be better than Nature DQN.

DQN (Adam) vs. Fine-tuning Nature DQN



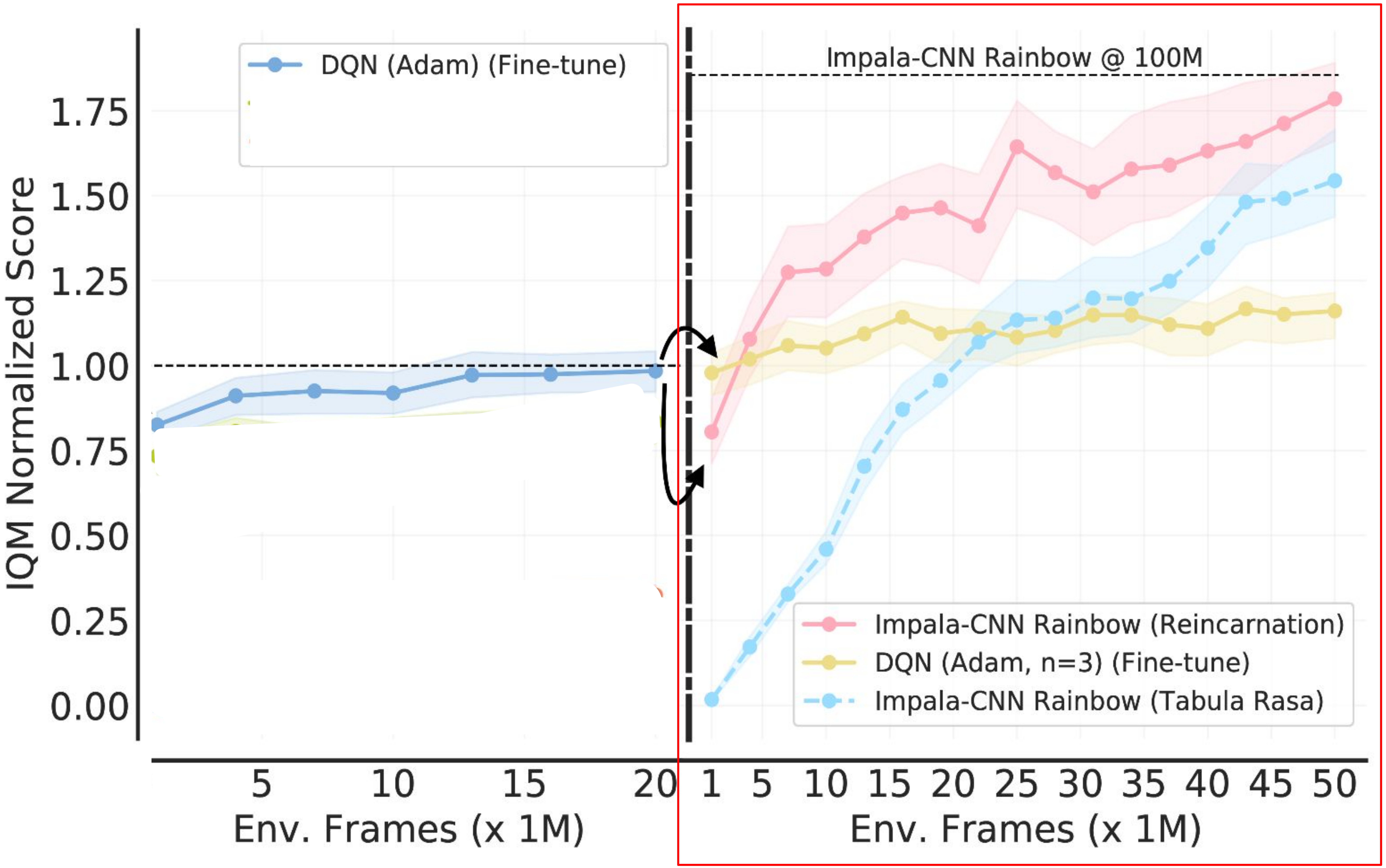
Fine-tuning DQN significantly improves performance.

Reincarnating DQN (Adam) via Fine-Tuning



Similar results to DQN (Adam) trained from scratch for 400M frames in few hours of training rather than a week!

Reincarnating a Different Architecture / Algorithm



Saved 50M frames or 1 day of GPU training!

Recap: Policy to Value Reincarnating RL (PVRL)

$$\pi_\Phi(a|s) \longrightarrow Q_\theta(s,a)$$

Suboptimal Teacher

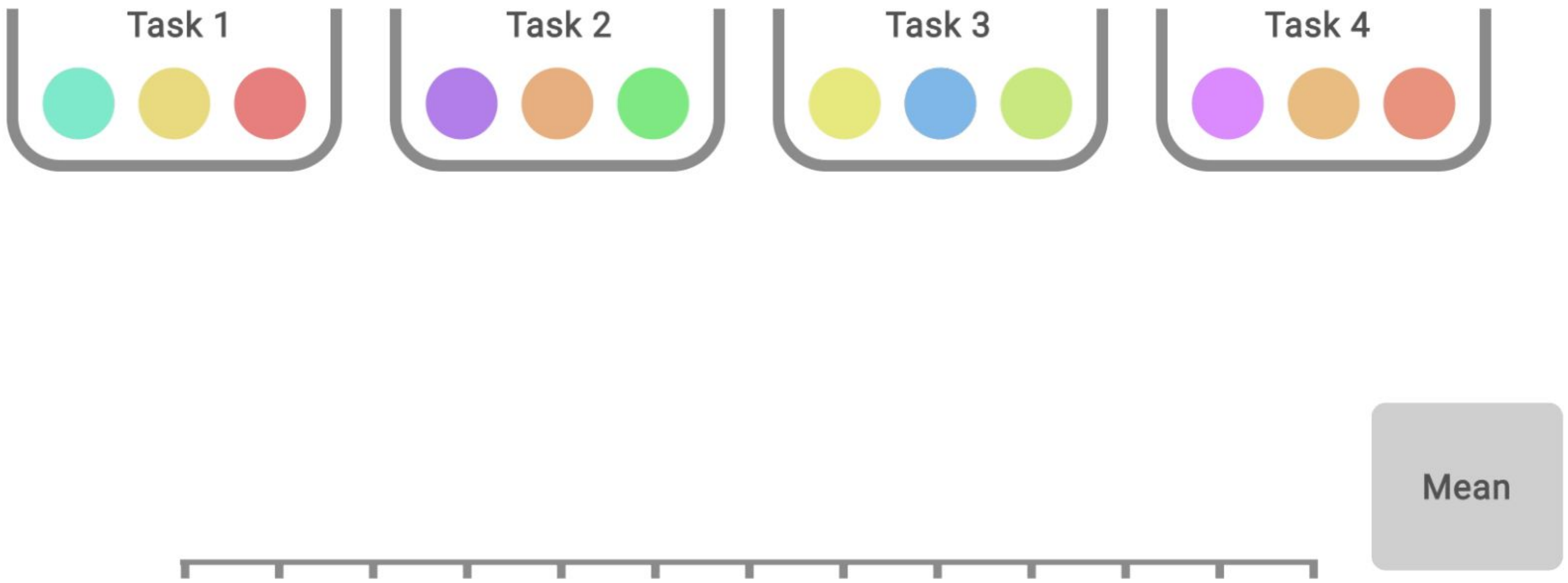
Value-based Student

Desiderata

- Teacher-agnostic
- Weaning off teacher
- Compute Efficient

PVRL: Experimental Setup

- Interactive teacher policy: DQN trained for 400M frames (**7 days on a single GPU**)
    - Also assume access to replay data of the teacher
- Transfer a student DQN using 10M frames (a few hours)
- 10 Atari games with sticky actions (for stochasticity)
- Evaluation: Interquartile Mean [1]



[1] For more details, see **Agarwal, Rishabh et al. Deep RL at the Edge of the Statistical Precipice.** NeurIPS 2021 (Outstanding Paper).

PVRL: Closely Related Methods

Adapting existing approaches:

- **Rehearsal:** Replaying Teacher Samples
- **Pretraining:** Offline RL on Teacher Data
- **Kickstarting:** On-policy Distillation + Q-learning
- **DQfD:** Learning from teacher demonstrations
- **JSRL**: Improving data collection using teacher

PVRL on ALE: DQN (Adam) @ 400M → DQN

QDagger: A simple PVRL baseline

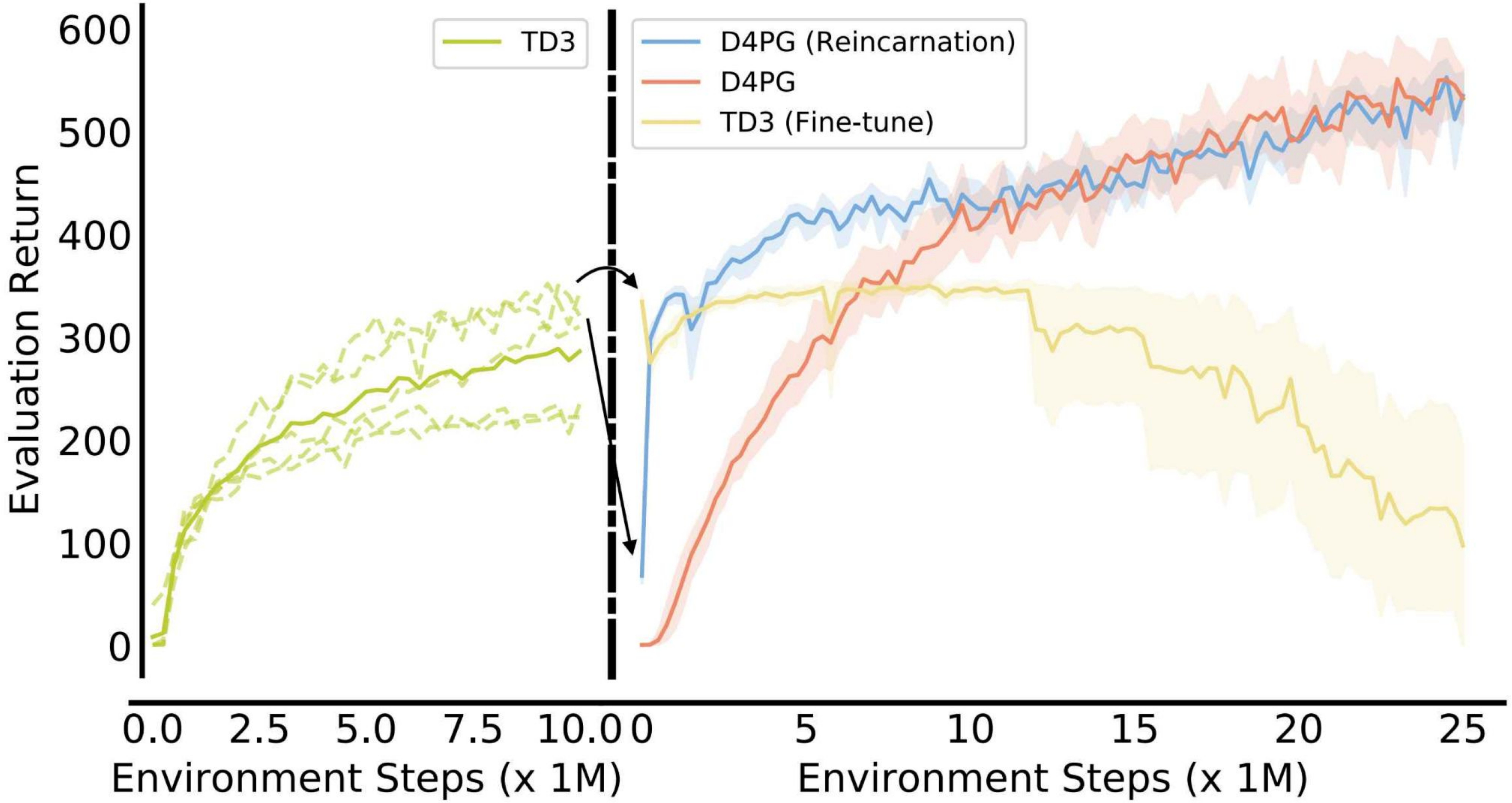
$$\mathcal{L}_{QDagger}(\mathcal{D}) = \underbrace{\mathcal{L}_{TD}(\mathcal{D})}_{\text{Q-learning loss}} + \lambda_t \mathbb{E}_{s\sim\mathcal{D}}\Big[\sum_a \underbrace{\pi_T(a|s)\log\pi(a|s)}_{\text{On-policy distillation}}\Big]$$

Decaying coefficient to wean off the teacher.

Combine Q-learning with Dagger. Phases:

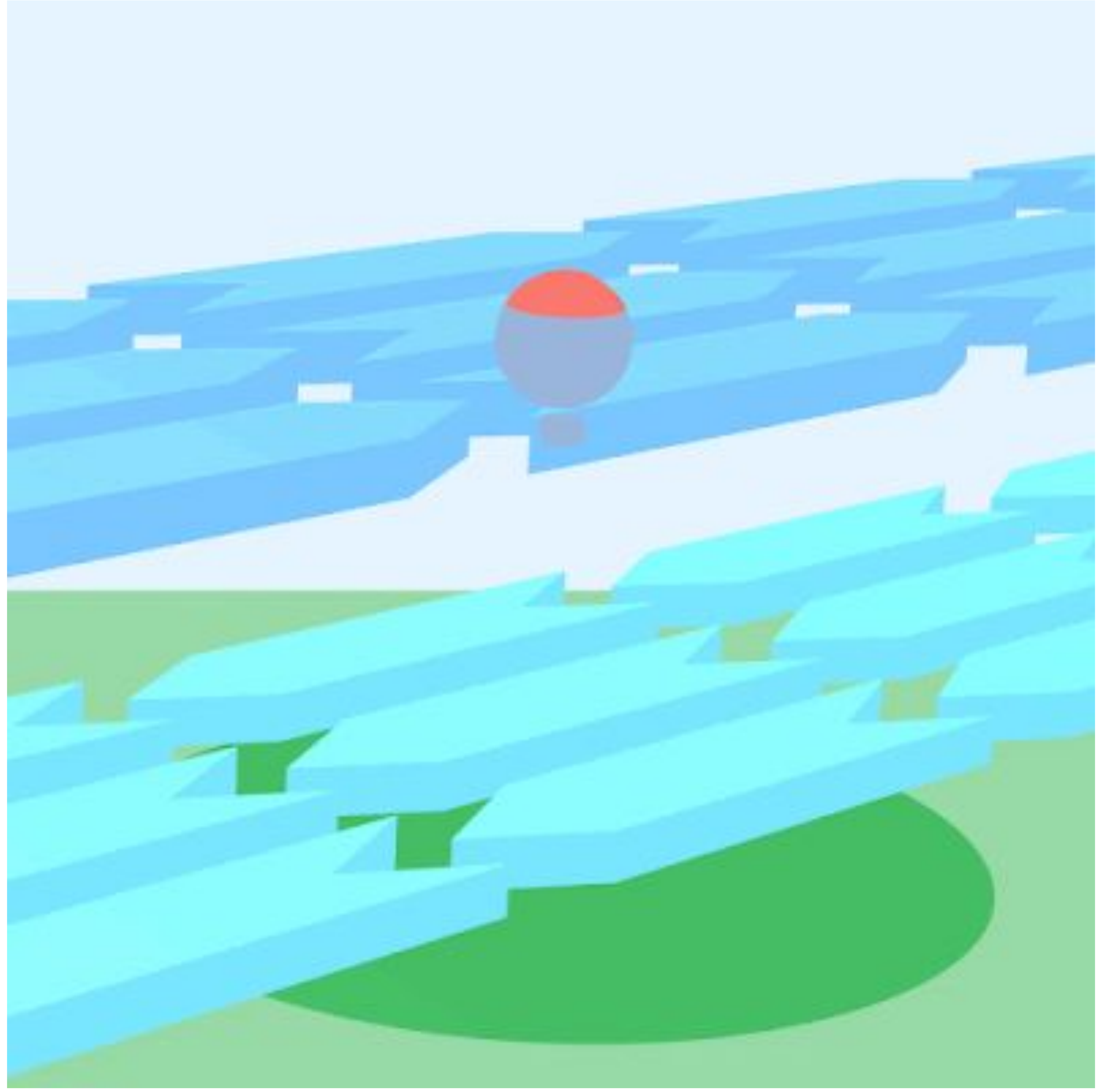
- (Offline) Pretrain on Teacher data
- (Online) Train on self-collected data.

Reincarnation on a difficult control task: Humanoid Run



Saved 10M frames
(10-12 hours on a V100)

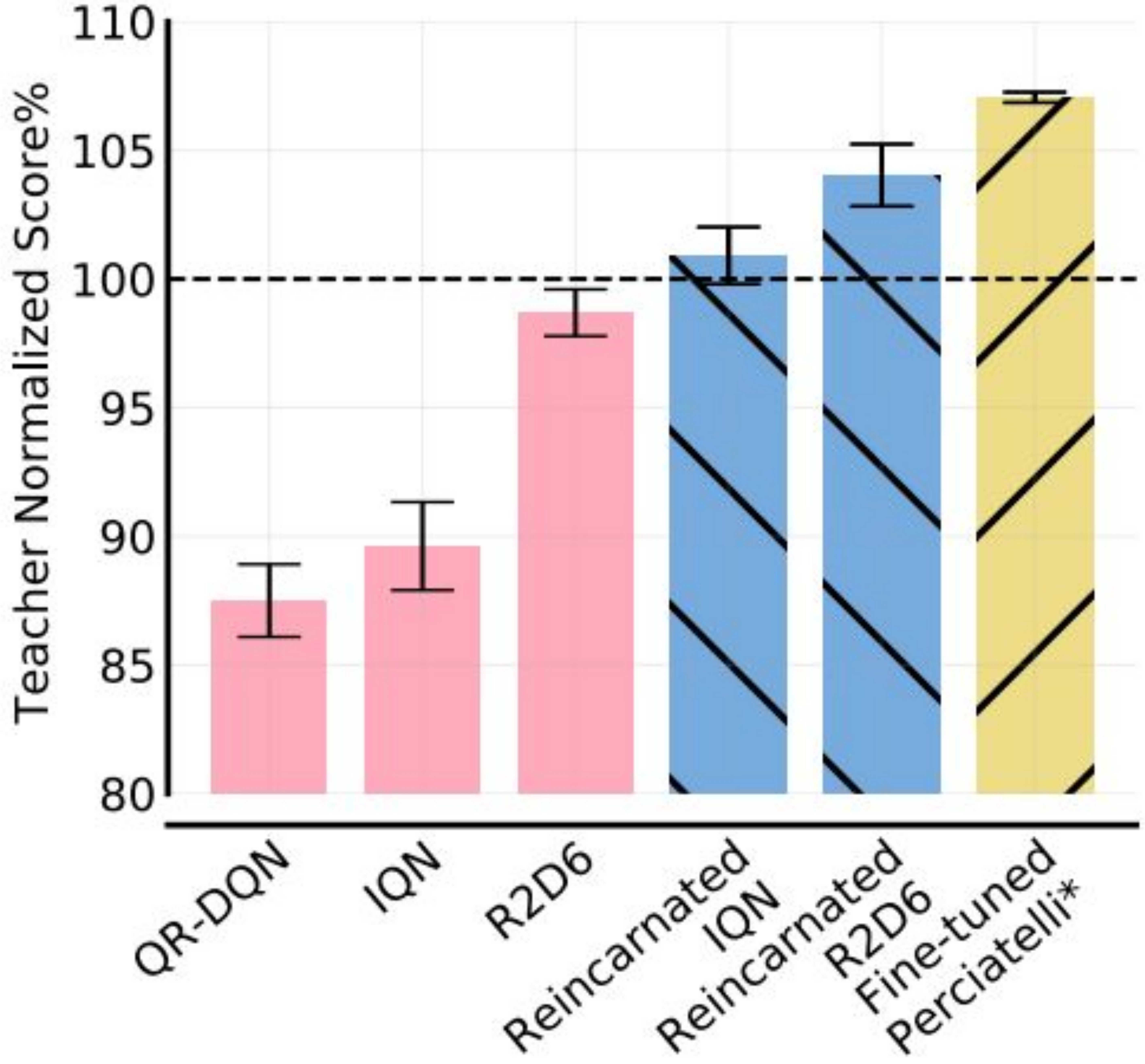
Reincarnation on Balloon Learning Environment (BLE)



- Access to the existing agent **trained for a month with distributed RL**.
- Given access to finite compute (10-12 hours on a TPU-v2), how much progress can be made?

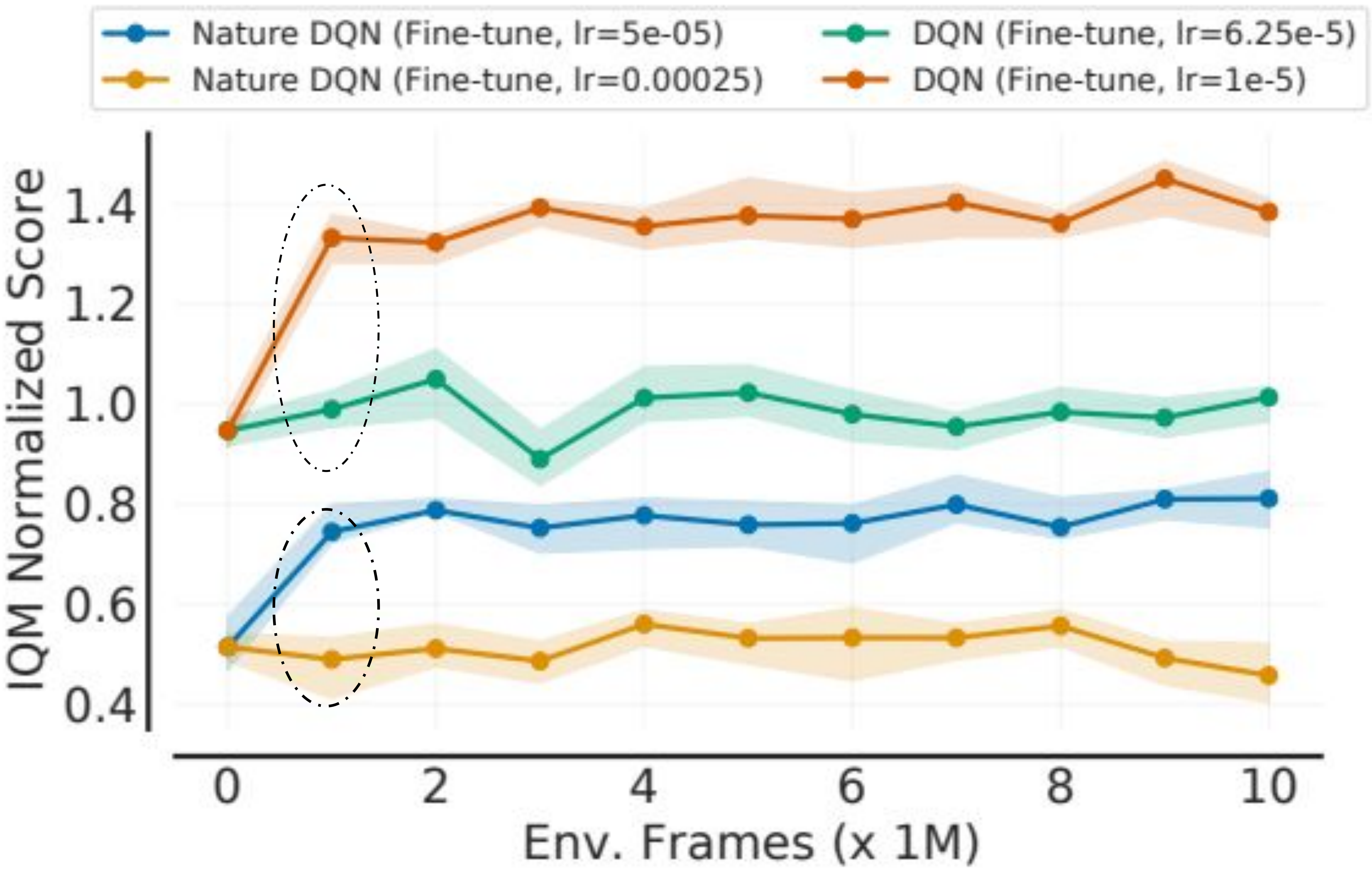
[1] Bellemare, Marc G., et al. "Autonomous navigation of stratospheric balloons using reinforcement learning." *Nature* 588.7836 (2020): 77-82.
[2] The Balloon Learning Environment. https://ai.googleblog.com/2022/02/the-balloon-learning-environment.html

Reincarnation on BLE

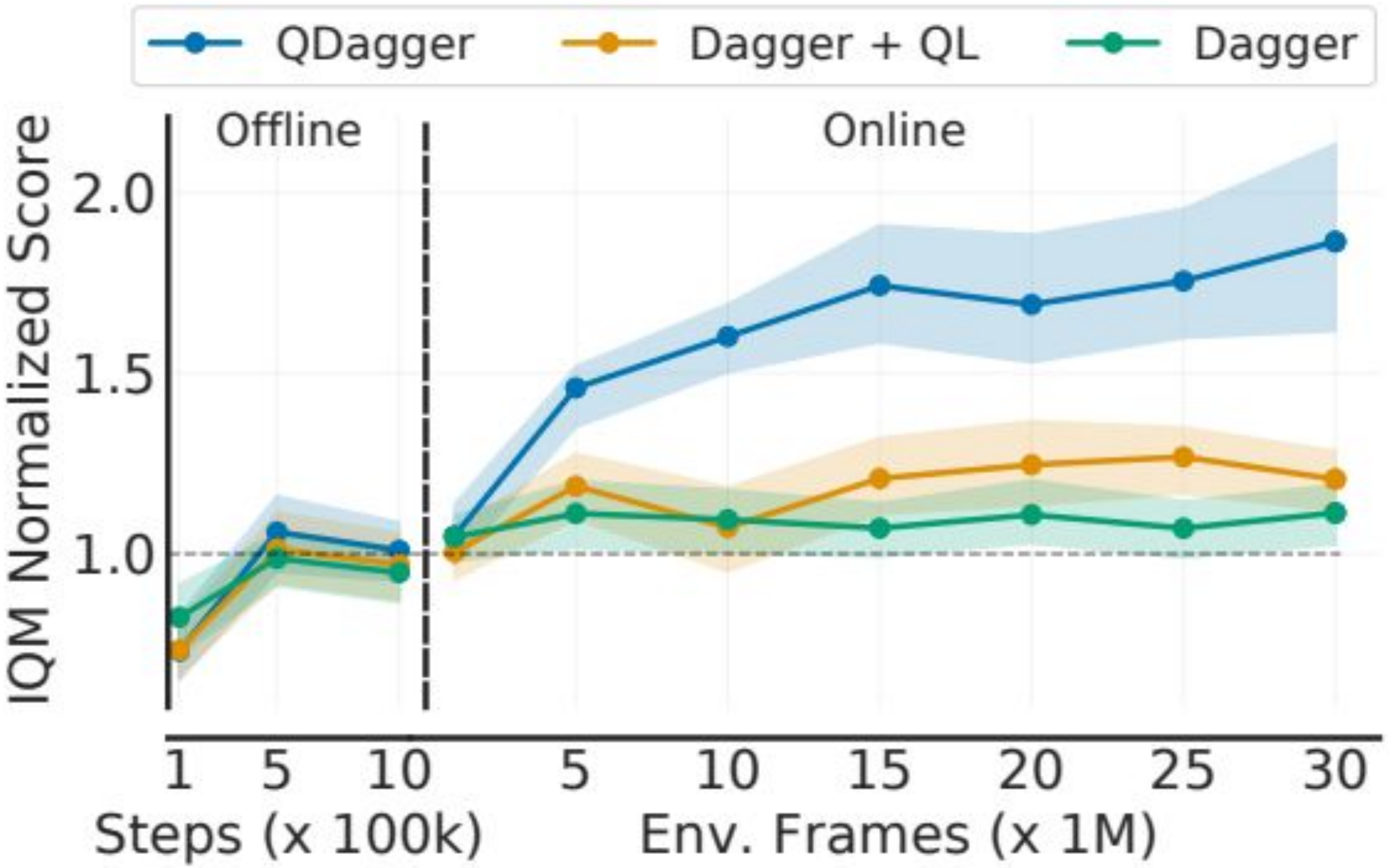


Considerations in Reincarnating RL

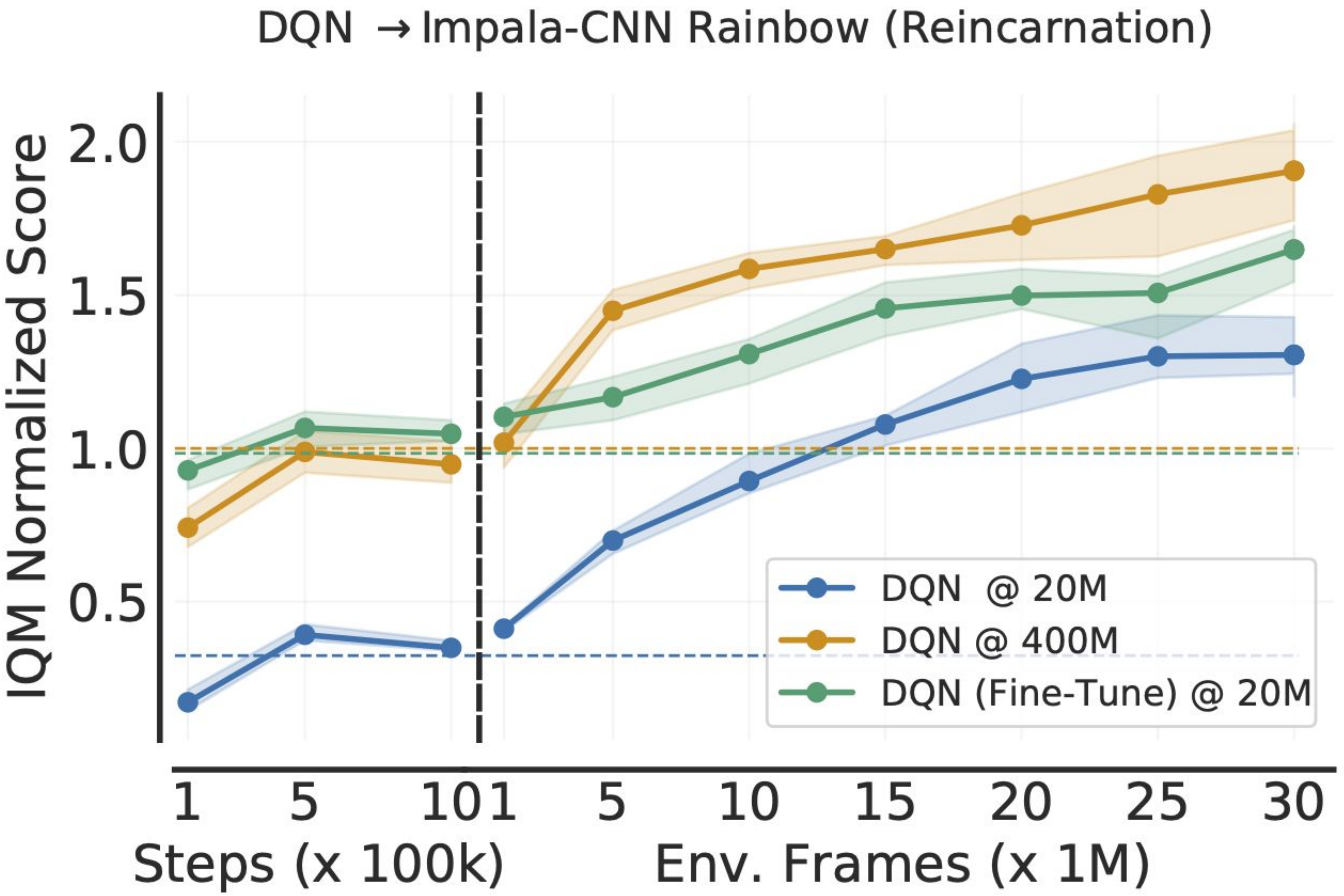
Fine-tuning for Reincarnation



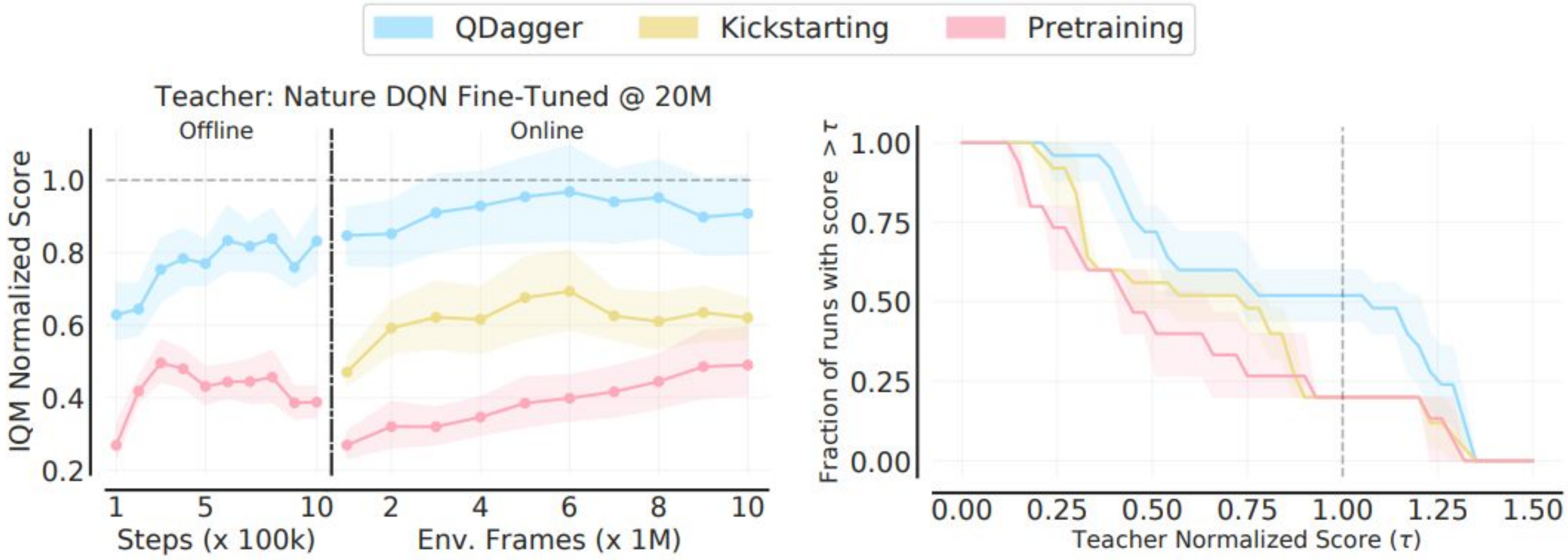
Reincarnation vs Distillation



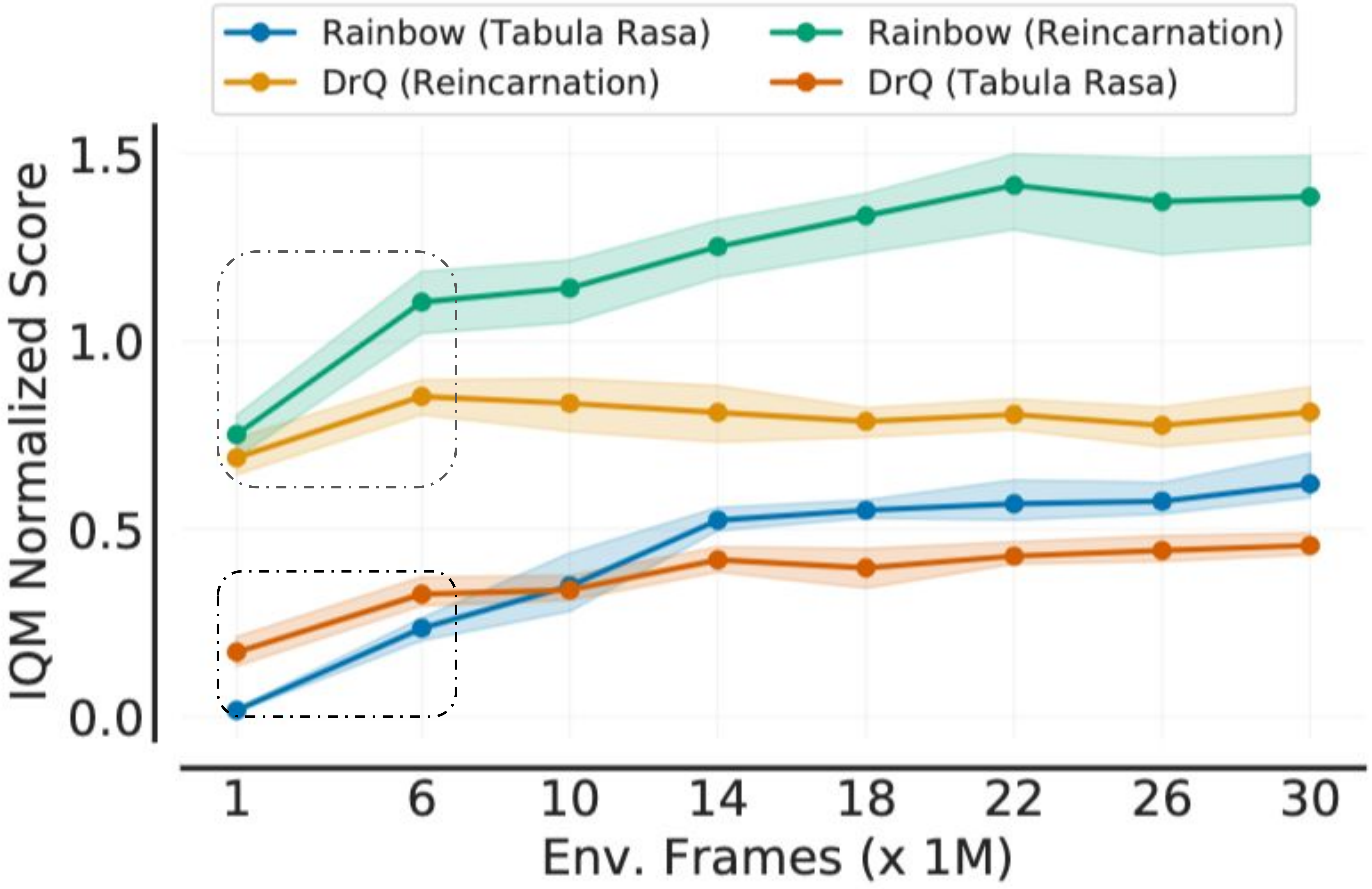
Dependence of Prior Computation



Reproducibility: Algorithmic Ranking is consistent.



Benchmarking Differences with Tabula Rasa



"If I have seen further than others, it is by standing upon the shoulders of giants."

– Sir Isaac Newton