Hybrid Quantum-Classical Reinforcement Learning in Latent Observation Spaces

Dániel T. R. Nagy, Csaba Czabán, Bence Bakó, Péter Hága, Zsófia Kallus, and Zoltán Zimborás

HUN REN Wigner

ERICSSON

QNL Quantum Information National Laboratory HUNGARY

National Research, Development and Innovation Office Hungary

PROGRAM FINANCED FROM THE NRDI FUND

ELTE EÖTVÖS LORÁND UNIVERSITY

GPU DAY 2024

Outline

- Motivation
- RL, PPO, QRL with PPO
- Latent-space QRL
- Numerical results
- Summary & Outlook

Motivation

- **NISQ: Noisy Intermediate-Scale Quantum Devices**
  - Today already 50-100 noisy qubits (NISQ)
  - Early versions of error correction
  - Approaching regime of potential practical quantum advantage

Motivation

- **NISQ: Noisy Intermediate-Scale Quantum Devices**
  - Today already 50-100 noisy qubits (NISQ)
  - Early versions of error correction
  - Approaching regime of potential practical quantum advantage
- **Quantum computational supremacy demonstrated on:**
  - Superconducting device by Google (2019) https://www.nature.com/articles/s41586-019-1666-5
  - Photonic
    - Xanadu, 2022: https://www.nature.com/articles/s41586-022-04725-x
    - Jiuzhang 3.0, 2023: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.131.150601

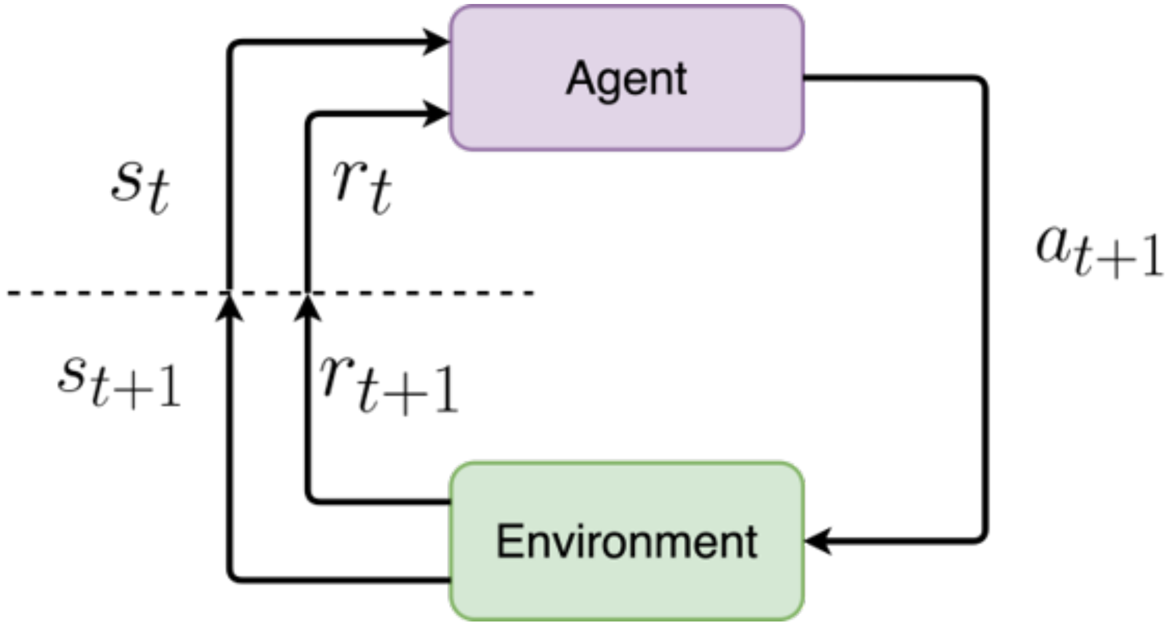
Motivation

- **NISQ-era candidates for practical quantum advantage:**
  - Simulation of quantum chemistry and many-body systems
  - Variational quantum optimization methods like QAOA
  - **Quantum Machine Learning (Includes Quantum Reinforcement Learning)**
  - **Hybrid Quantum-Classical methods enabled by classical HPC**

Motivation

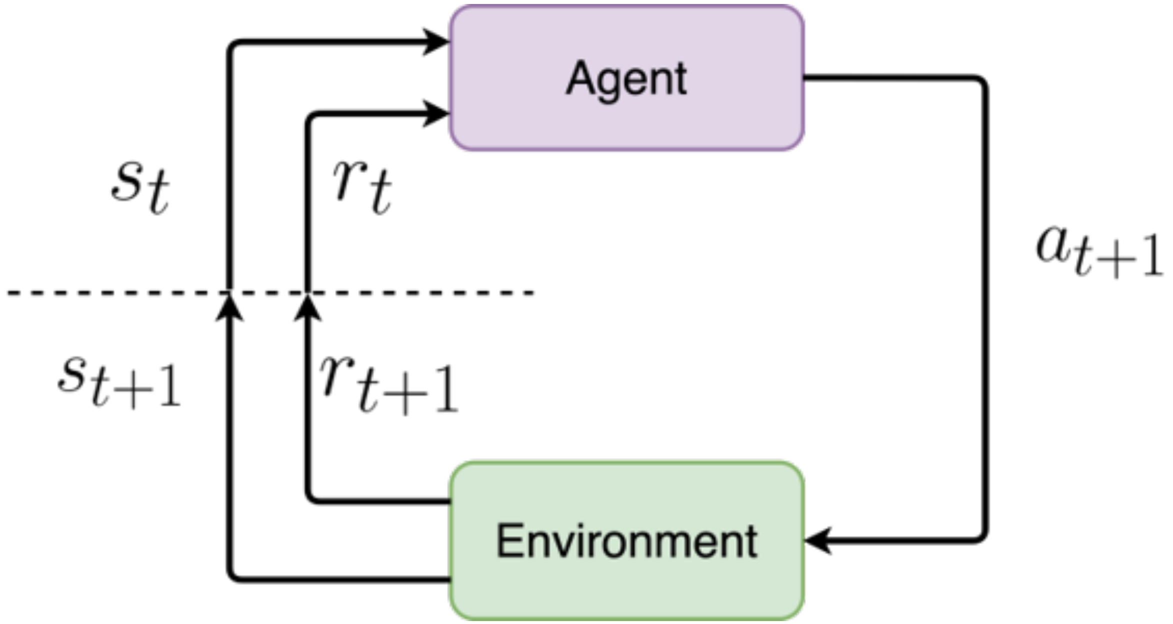
- **QRL is limited by the available QPU sizes**
  - Many RL environments have high dimensional state spaces (e.g. visual data)
  - We would need large scale QPUs to encode raw features into quantum states
  - <u>Solution: use latent features extracted by classical algorithms</u>

Reinforcement Learning



- **Reinforcement Learning (RL)** is a method designed to optimally solve a control problem in a simulated or real-world environment.
- In RL, an **Agent** is **observing the state** of the **environment** and choses **actions** accordingly.
- After the agent performs the action, the e**nvironment** returns a **reward** and the **next state.**

Reinforcement Learning



- The goal is to train an agent which maximizes the discounted cumulative reward,

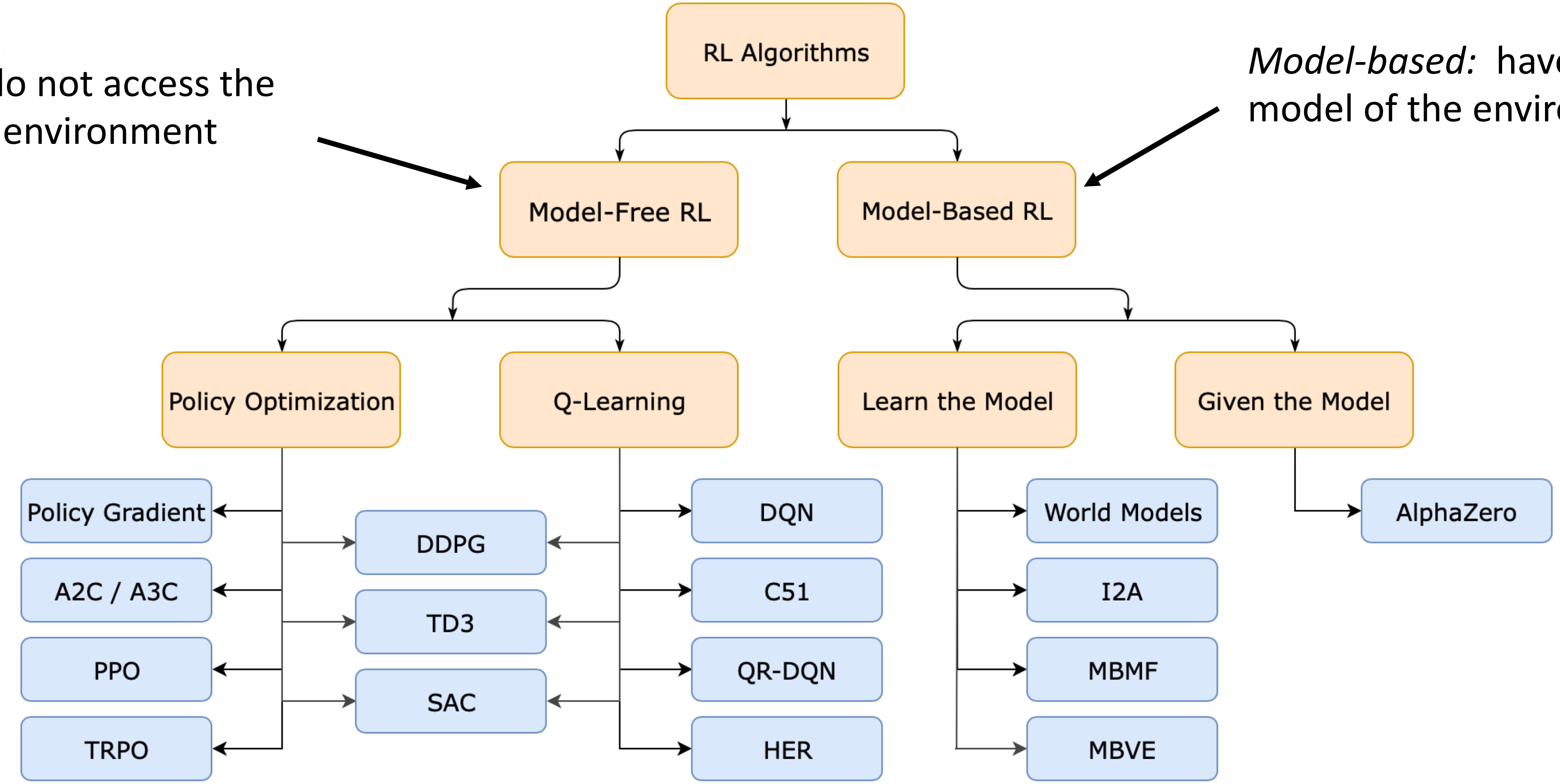
$$R = \sum_t \gamma^t r_t$$

- Such Agents are usually implemented as NNs.

Reinforcement Learning

Model-free: do not access the model of the environment

Model-based: have access to a model of the environment



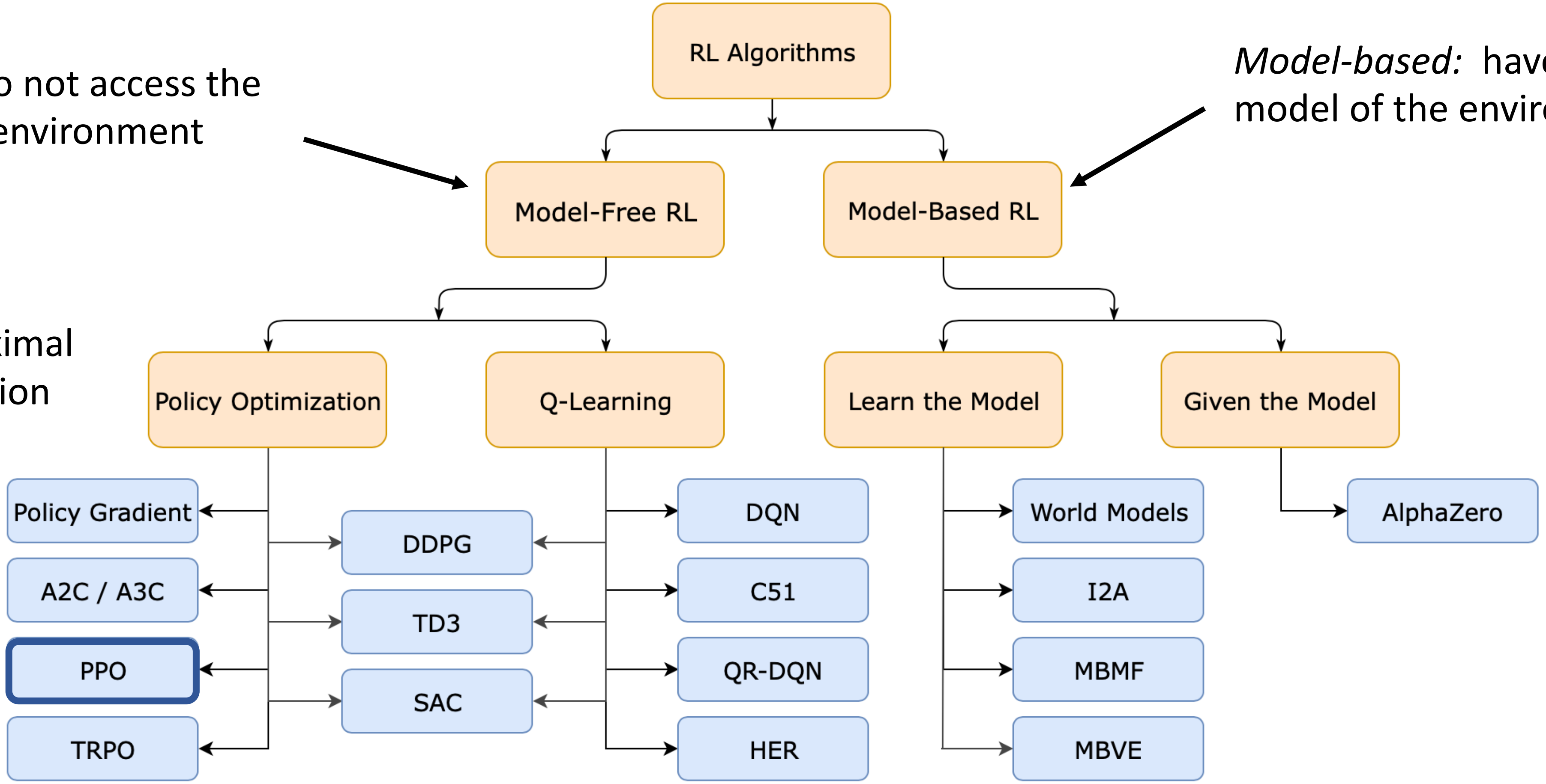
Source: https://spinningup.openai.com/

Reinforcement Learning

Model-free: do not access the model of the environment

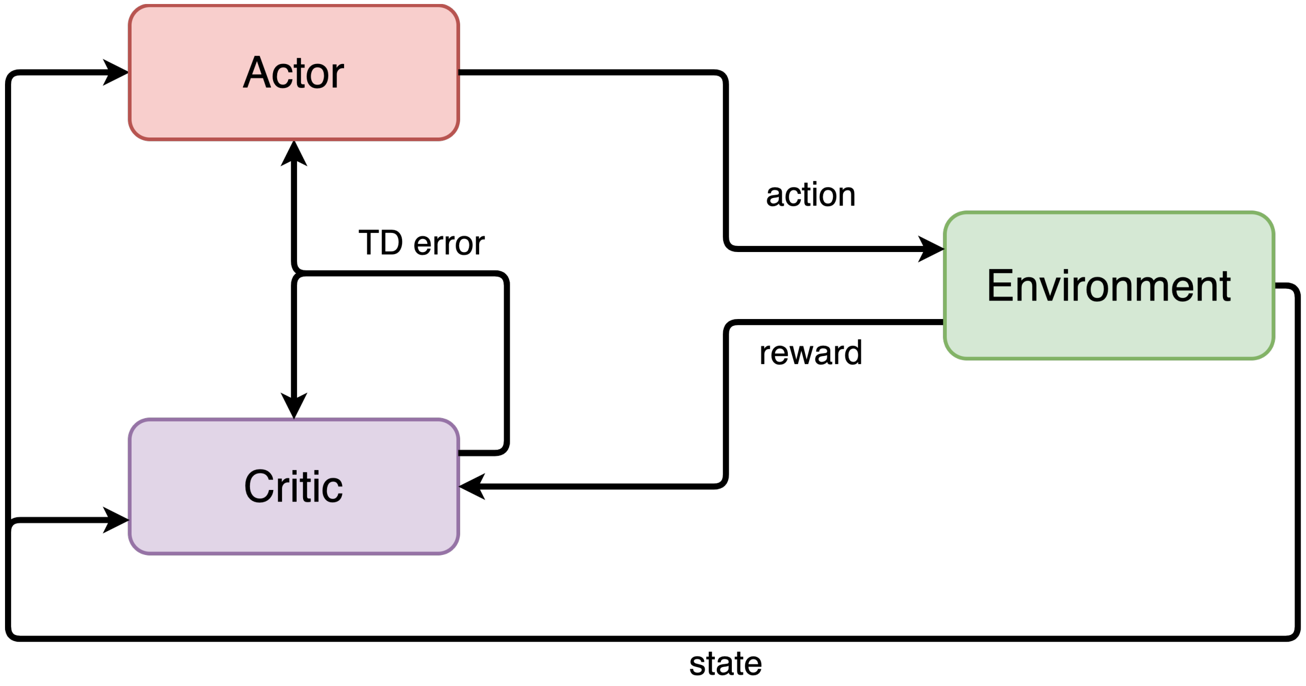
Model-based: have access to a model of the environment

We will use Proximal Policy Optimization



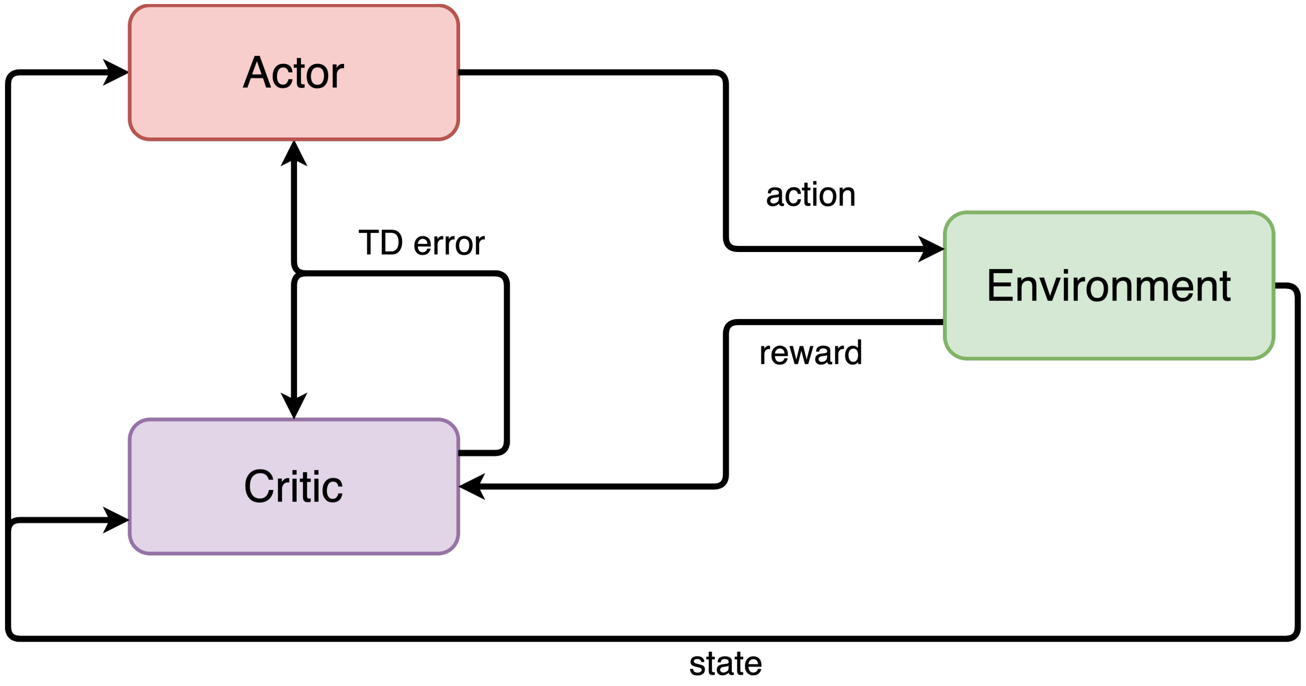
Source: https://spinningup.openai.com/

Proximal Policy Optimization (PPO)



- PPO is a model-free method using two function approximators:
  an **Actor** and a **Critic**.
- The Actor choses an action according to a policy π.
- The Critic calculates the estimated value of the state.

Proximal Policy Optimization (PPO)



- The Critic receives the reward and calculates the temporal difference error, which is used to update both Actor and Critic networks.

Proximal Policy Optimization (PPO)

$s_t\,,\ a_t\,, r_t$ are the state, action & reward at timestep t.

$\boxed{\pi_\theta(\cdot|s)}$ is the policy, where theta are the tunable parameters.

$r_t(\theta) = \pi_\theta / \pi_{\theta_{\text{old}}}$ is the ratio of the new and old policies.

$V^\pi(s)$ is the value function used by the Critic.

$\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l}$ is the estimated advantage with $\delta_t = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$

The advantage function estimates the extra reward that could be obtained by the agent by taking that particular action.

Proximal Policy Optimization (PPO)

Critic Loss:

$$\mathcal{L}^{VF} = \mathop{\mathbb{E}}_{t}\left[\left(V^{\pi}(\mathbf{s}_t) - V^{\pi}_{\mathrm{targ}}(\mathbf{s}_t)\right)^2\right]$$

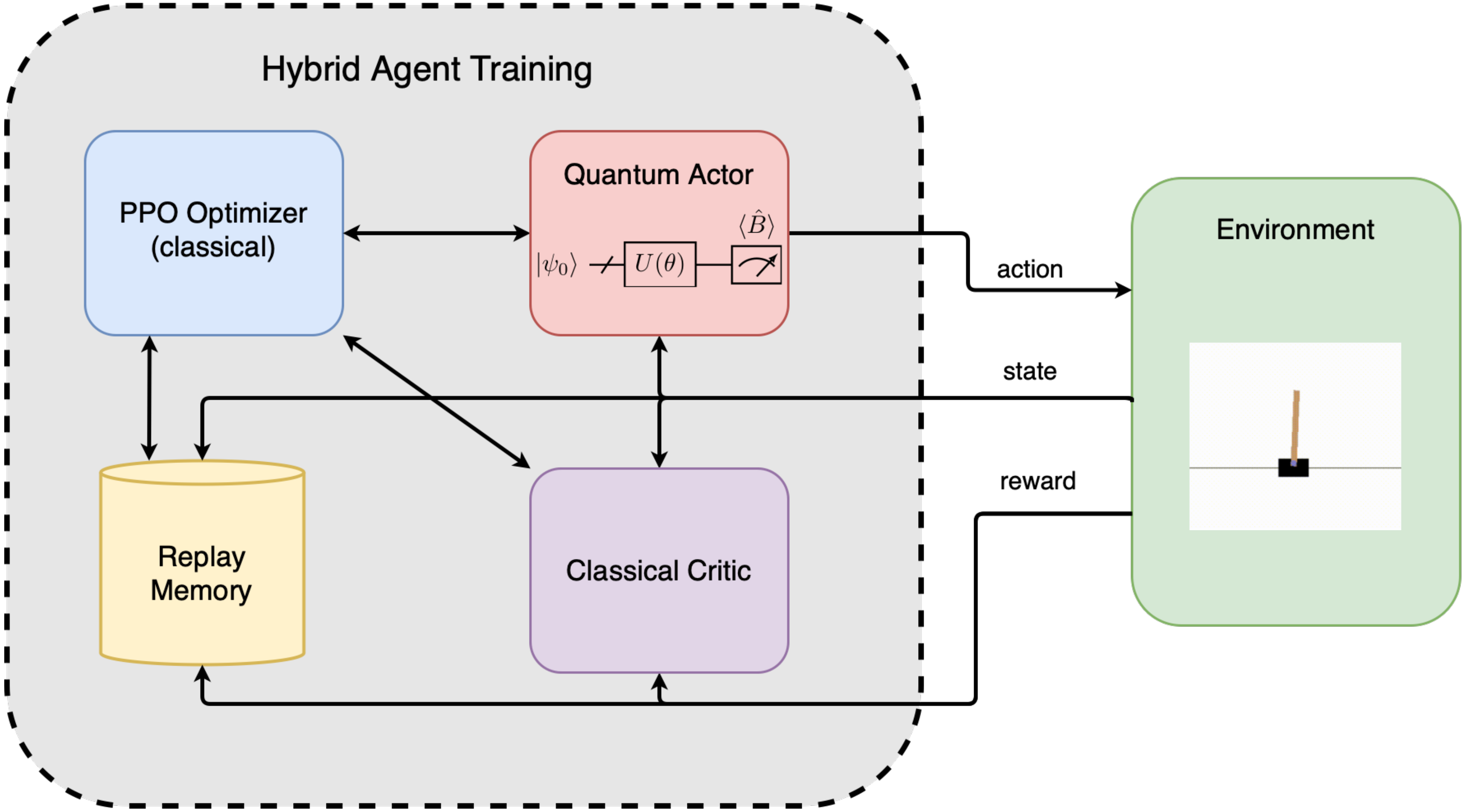
Clipped Surrogate Objective:

$$\mathcal{L}^{\mathrm{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}\left(r_t(\theta), \epsilon\right)\hat{A}_t\right)\right]$$

PPO Objective:

$$\boxed{\mathcal{L}^{\mathrm{PPO}} = \mathcal{L}^{\mathrm{CLIP}}(\theta) + c_1 S\left[\pi_\theta\right] + c_2 \mathrm{Reg}(\theta)}$$

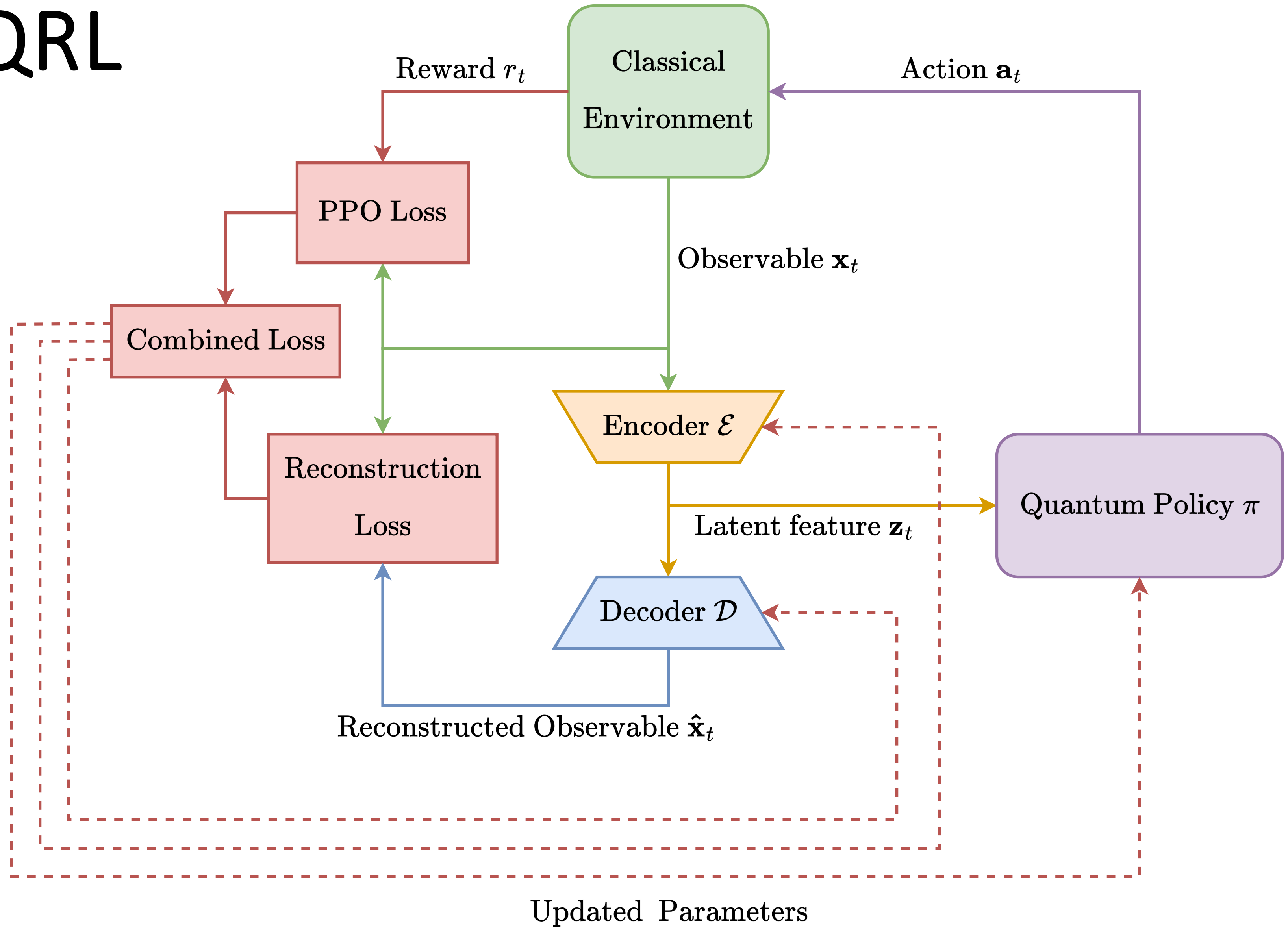
Quantum Reinforcement Learning with PPO



- Substitute the classical policy with a QNN
- Encode states into q-states, compute actions from measurements
- The rest of the system is classical
- Optimize the QNN policy parameters via gradient descent

Latent-space QRL

- As mentioned: environments often have high dimensional observables
- We can not encode the full observable into the initial state of a QNN
- We use a classical AE for feature extraction, and encode latent features
- The classical AE is trained together with the quantum agent

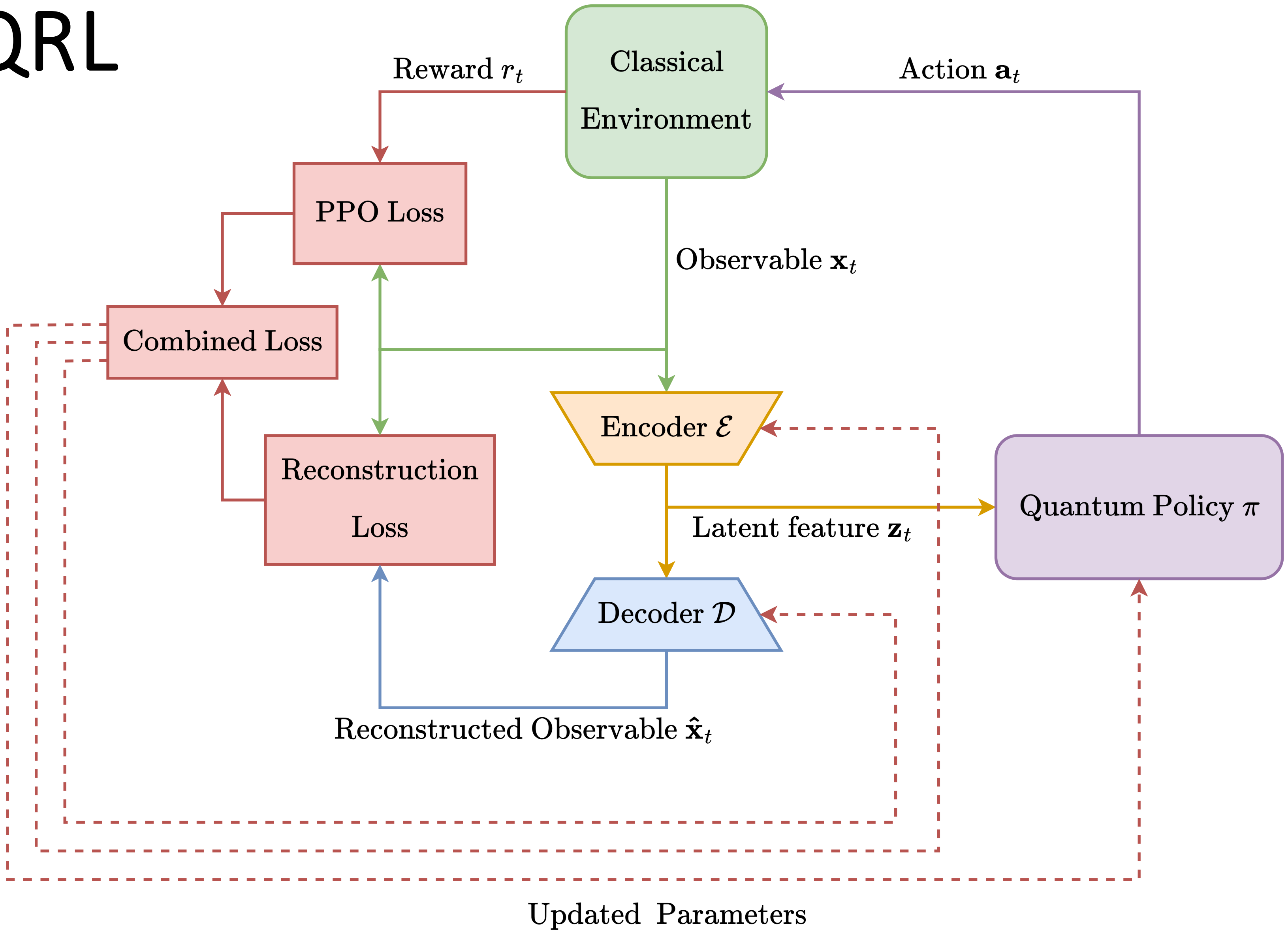


Latent-space QRL

- We optimize the hybrid system via a combined loss function:

$$\mathcal{L}^{(PPO+AE)} = \mathcal{L}^{(PPO)} + c_{\text{ae}}\mathcal{L}^{(AE)}$$

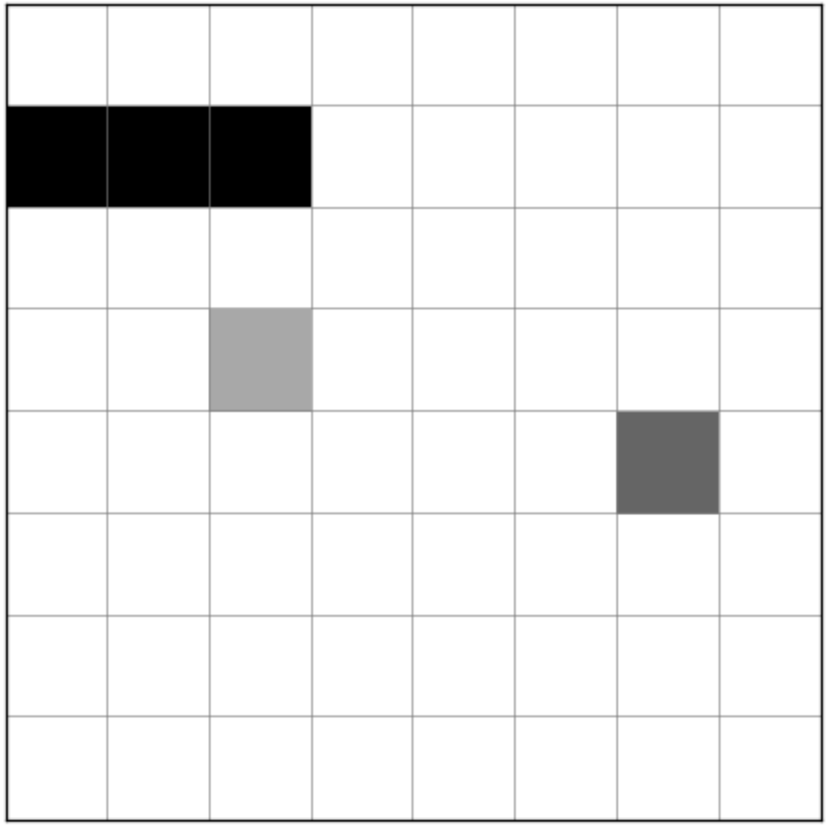
- Optionally, the AE can be pre-trained



Numerical experiments

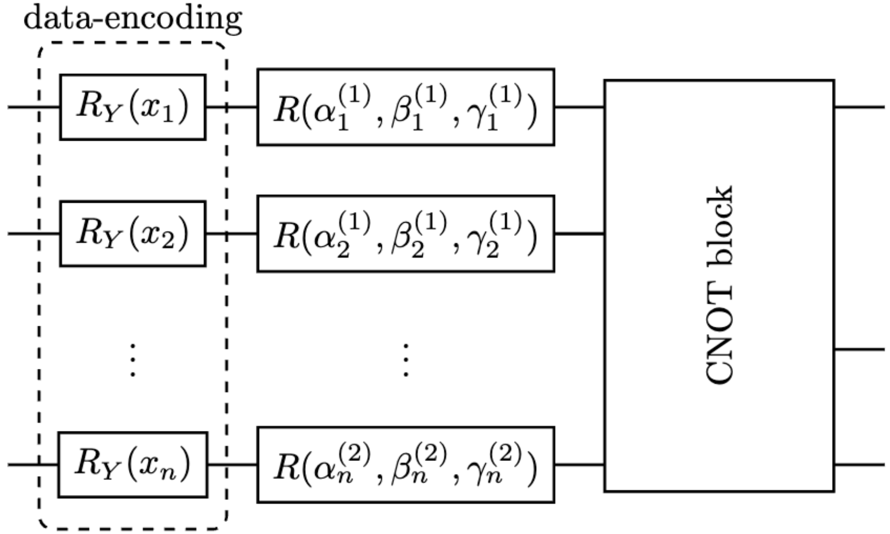
- We tested this approach with various configurations:
  - Three environments: Cartpole-v1, Acrobot-v1 and Maze-v0
  - Various AE sizes, and various number of QNN layers
  - Both qubit-based and photonic QNNs
  - “Cold started” AEs versus pre-trained Aes
  - Compared with fully classical baselines

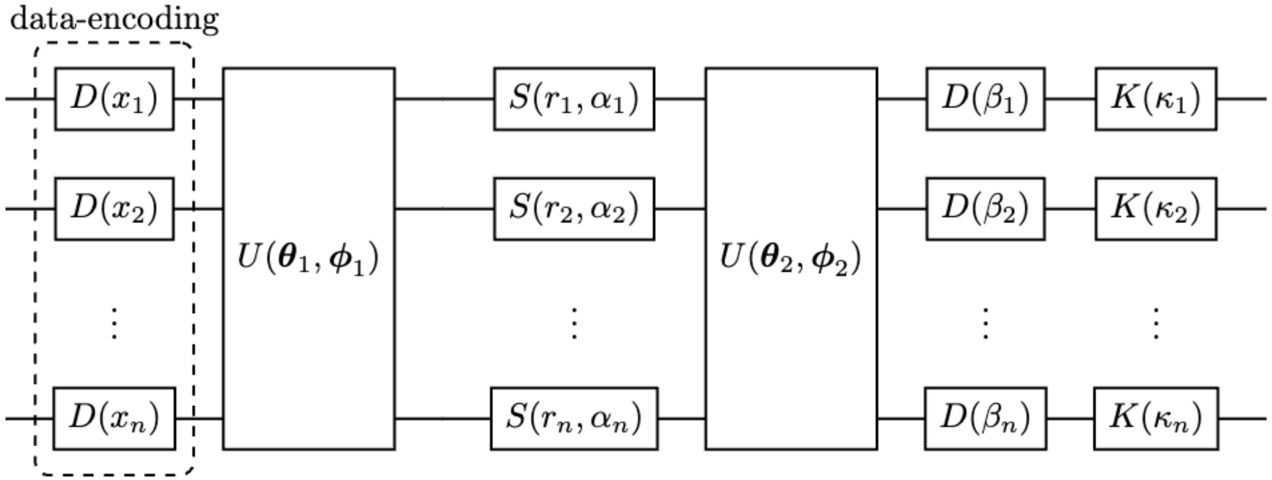
Numerical experiments



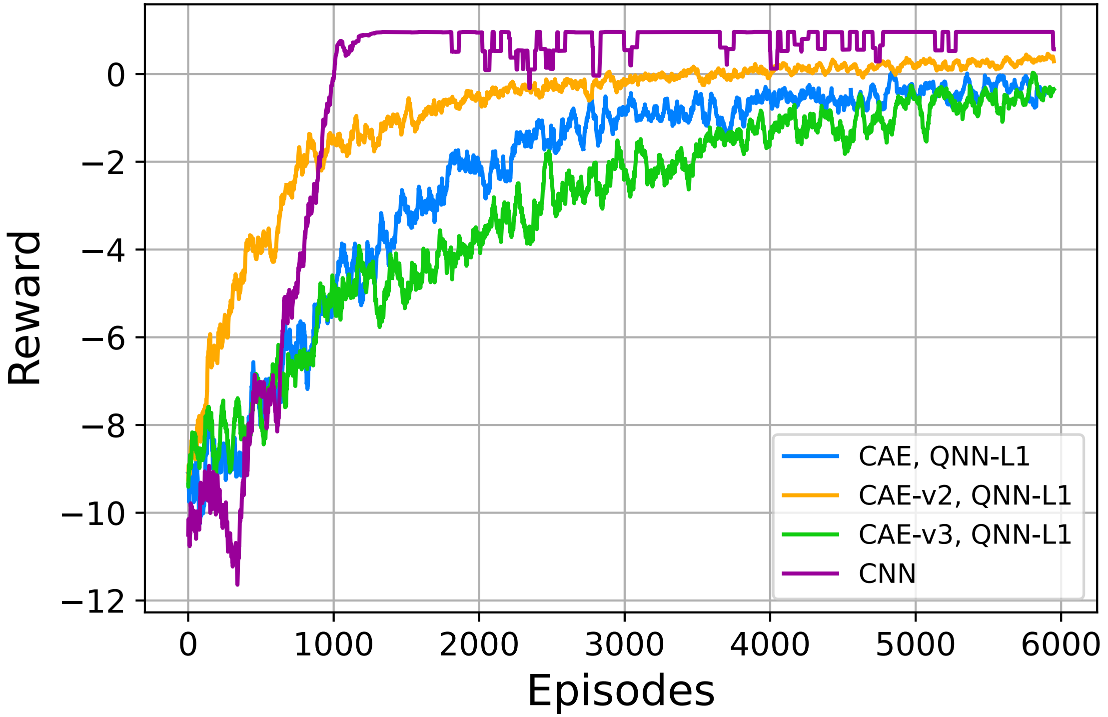
- Maze-v0 environment
- 48x48 grayscale image
- 4 possible actions: up, down, left, right
- The agent starts at a random cell

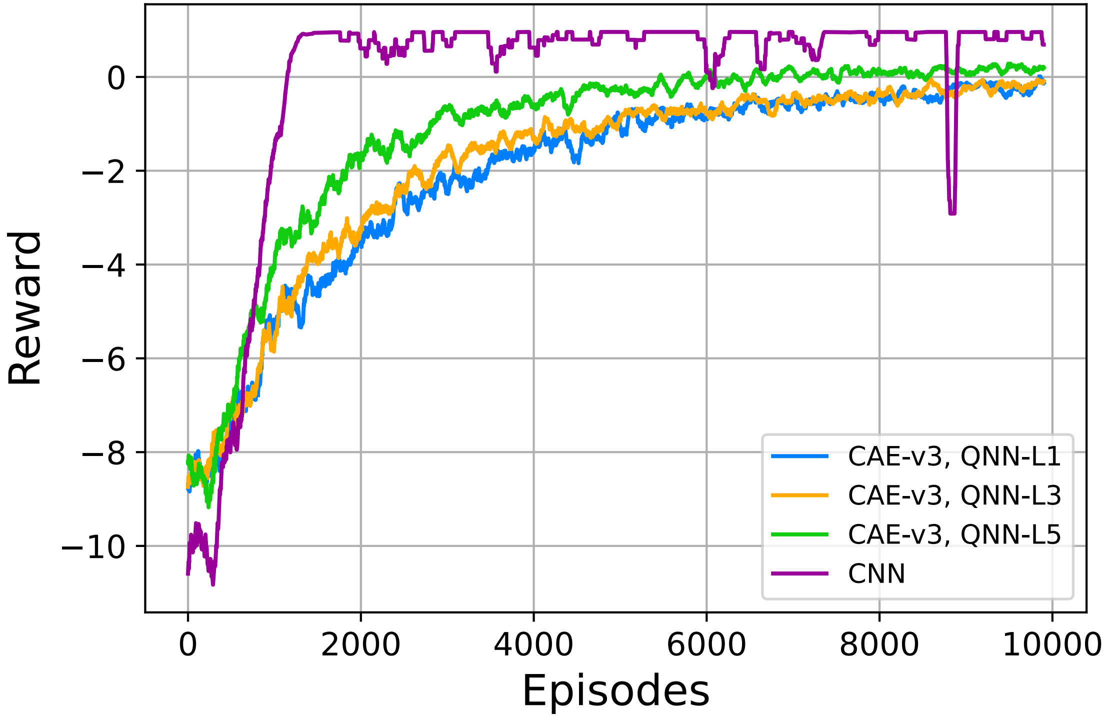
Numerical experiments





Numerical experiments

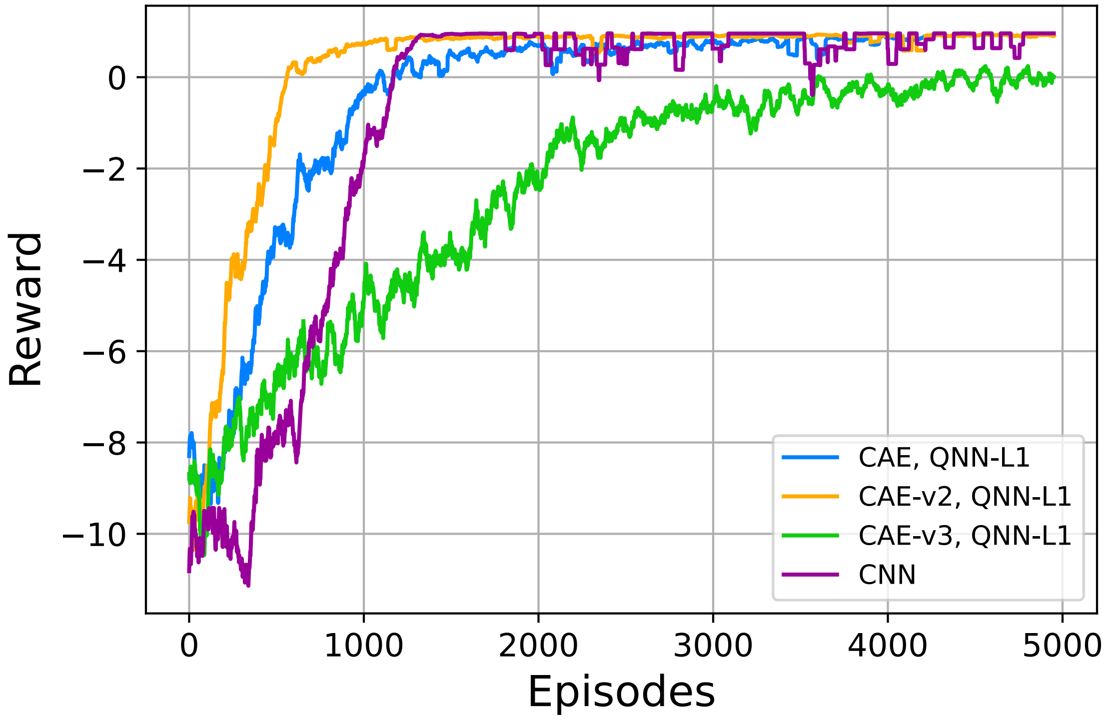


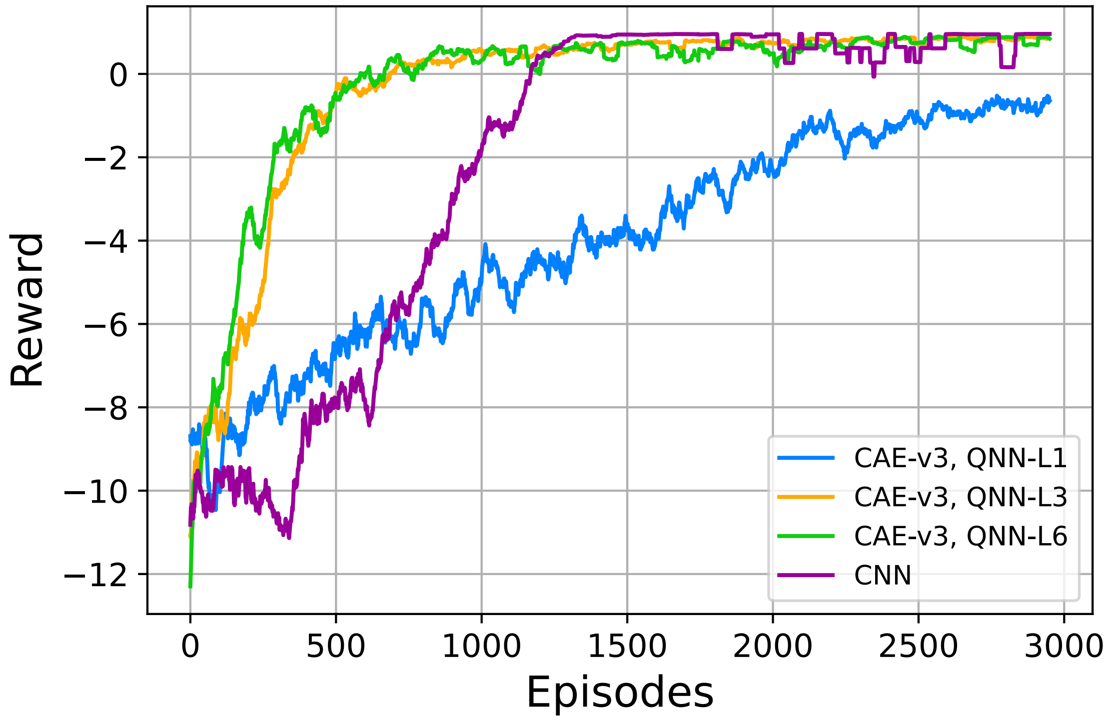


Qubit-based results simulated with Pennylane. Left: Comparing different AE sizes with 1-layer QNN; Right: Comparing different QNN layer count with the smallest AE.

Each curve is a smoothed average over five agents run in parallel

Numerical experiments





Photonic results simulated with Piquasso. Left: Comparing different AE sizes with 1-layer QNN;
Right: Comparing different QNN layer count with the smallest AE.

Each curve is a smoothed average over five agents run in parallel

Conclusions

- We demonstrated that the AE+QNN method enables the application of QRL for high dimensional environments.
- We showed that the joint training of a classical AE and a QRL agent is necessary for convergence.
- We see a tradeoff between AE size and QNN layer count
- We conclude that in some cases, the AE + QNN method can outperform the fully classical approach in terms of parameter count, however this needs further research

Outlook

- A manuscript is in progress with more details.
- Further tasks:
  - Introduce a quantum critic alongside the quantum policy
  - Try to find the best value for the c_ae coefficient

Thank You

HUN REN | Wigner

ERICSSON

Quantum Information National Laboratory HUNGARY

National Research, Development and Innovation Office Hungary

PROGRAM FINANCED FROM THE NRDI FUND

ELTE Eötvös Loránd University

GPU DAY 2024

References

[1] Dunjko, V., Taylor, J.M., Briegel, H.J.: Advances in quantum reinforcement learning. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 282–287 (2017)

[2] Hoof, H., Chen, N., Karl, M., Smagt, P., Peters, J.: Stable reinforcement learning with autoencoders for tactile and visual data. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3928–3934 (2016)

[3] Killoran, N., Bromley, T. R., Arrazola, J. M., Schuld, M., Quesada, N., & Lloyd, S. (2019). Continuous-variable quantum neural networks. Phys. Rev. Research, 1, 033063. doi:10.1103/PhysRevResearch.1.033063

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, May 2015. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

[5] Proximal policy optimization algorithms
J Schulman, F Wolski, P Dhariwal, A Radford, O Klimov - arXiv preprint arXiv:1707.06347, 2017

[6] Nagy, D., Tabi, Z., Hága, P., Kallus, Z., and Zimborás, Z., "Photonic Quantum Policy Learning in OpenAI Gym arXiv:2108.12926.

Suppplimentary information

Environment	Name	platform	QNN params
CartPole-v1	QNN-l1	qubit	6
CartPole-v1	QNN-l3	qubit	18
CartPole-v1	QNN-l6	qubit	32
CartPole-v1	QNN-l1	qumode	14
CartPole-v1	QNN-l3	qumode	42
CartPole-v1	QNN-l6	qumode	84
CartPole-v1	MPL	classic	114
Acrobot-v1	QNN-l1	qubit	9
Acrobot-v1	QNN-l3	qubit	27
Acrobot-v1	QNN-l6	qubit	54
Acrobot-v1	QNN-l1	qumode	28
Acrobot-v1	QNN-l3	qumode	84
Acrobot-v1	QNN-l6	qumode	168
Acrobot-v1	MPL	classic	163
Maze-v0	QNN-l1	qubit	24
Maze-v0	QNN-l3	qubit	72
Maze-v0	QNN-l5	qubit	120
Maze-v0	QNN-l1	qumode	94
Maze-v0	QNN-l3	qumode	282
Maze-v0	QNN-l6	qumode	564
Maze-v0	CNN	classic	81140

Environment	Name	Hidden Sizes	Encoder Params	Decoder Params
CartPole-v1	AE-h0	-	10	12
CartPole-v1	AE-h4	4	30	32
Acrobot-v1	AE-h0	-	21	24
Acrobot-v1	AE-h4	4	43	46

Name	Platform	Filter Sizes	Pooling size	Encoder Params	Decoder Params
CAE	qumode	2, 2, 4, 4	2	504	679
CAE-v2	qumode	2, 4, 8, 8	2	1414	2057
CAE-v3	qumode	2, 2	4	172	221
CAE	qubit	2, 2, 4, 4	2	578	751
CAE-v2	qubit	2, 4, 8, 8	2	1560	2075
CAE-v3	qubit	2, 2	4	210	257