

Factored representation for fast adaptation in non-stationary and multi-domain RL

Fan Feng

Factored adaptation for non-stationary RL. Preprint; arxiv/2203.16582
AdaRL: what, where and how to adapt in transfer reinforcement learning. ICLR 2022 Spotlight

Factored Adaptation for Non-stationary RL

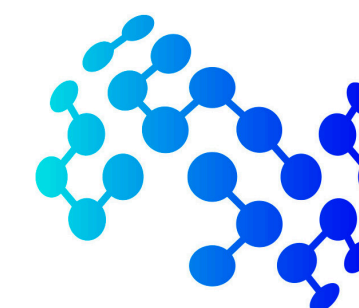
Fan Feng, Biwei Huang, Kun Zhang, Sara Magliacane



香港城市大學
City University of Hong Kong

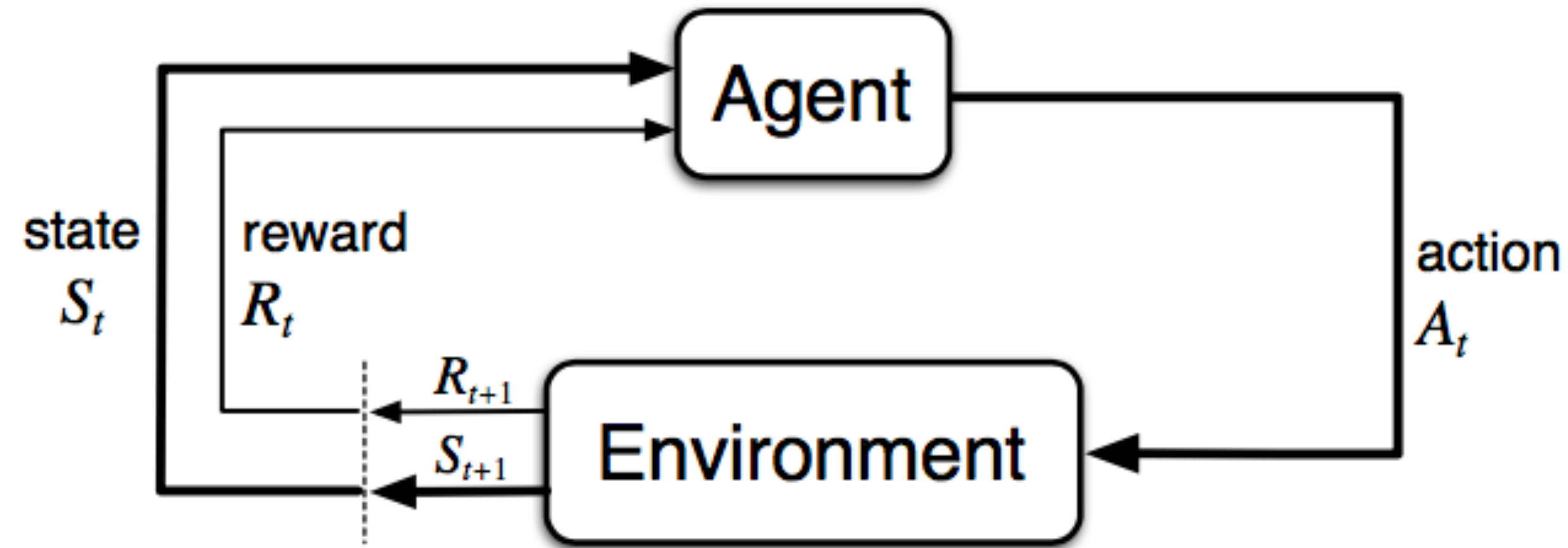


UNIVERSITY
OF AMSTERDAM



MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

Reinforcement learning (RL)



Environment (stationary):

State-transition function:

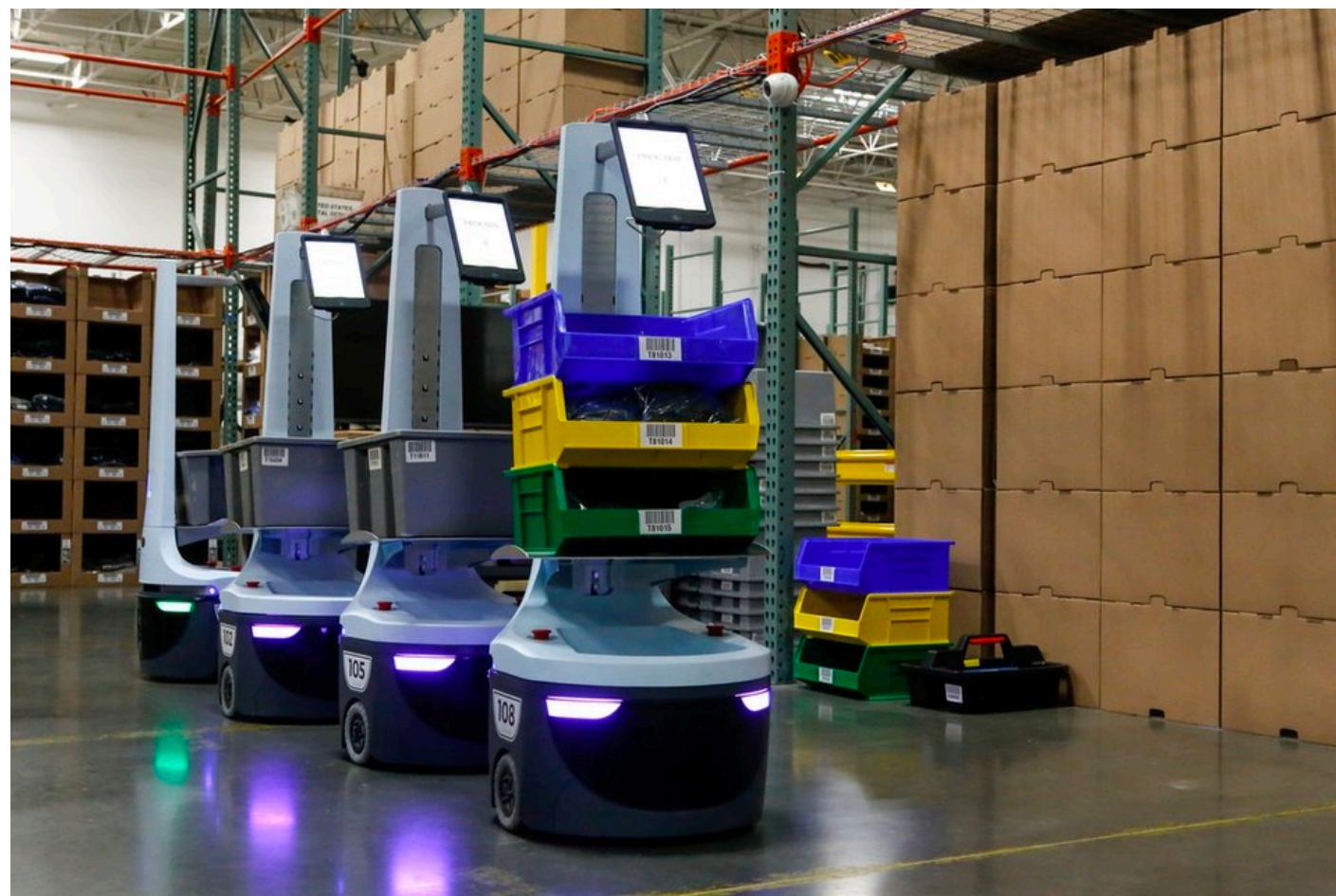
$$P_{ss'}^a = P(s' | s, a) = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} P(s', r | s, a)$$

Reward function:

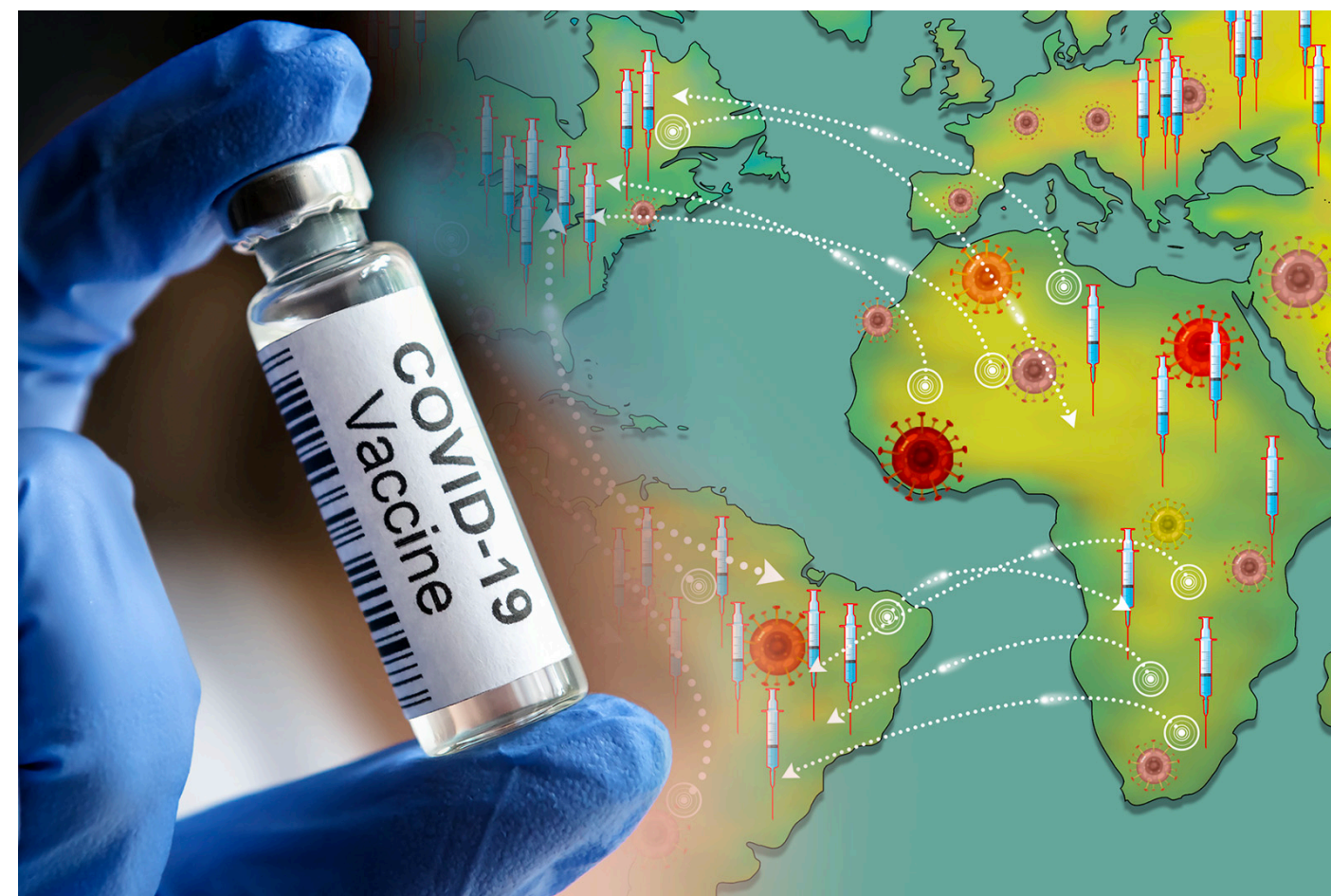
$$R(s, a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} P(s', r | s, a)$$

Real-world Systems are usually non-stationary

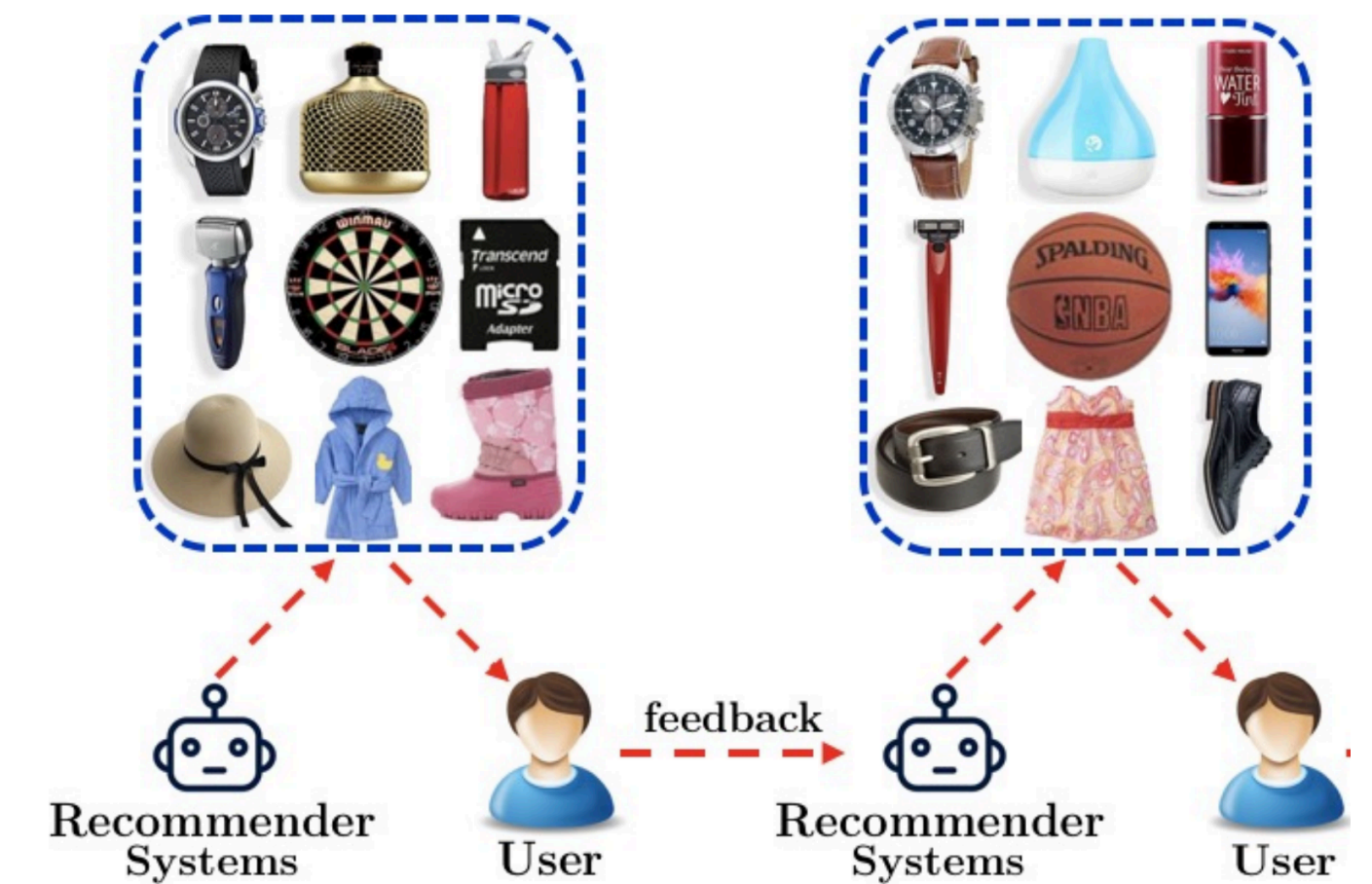
Robot learning



AI for Healthcare



Recommendation systems

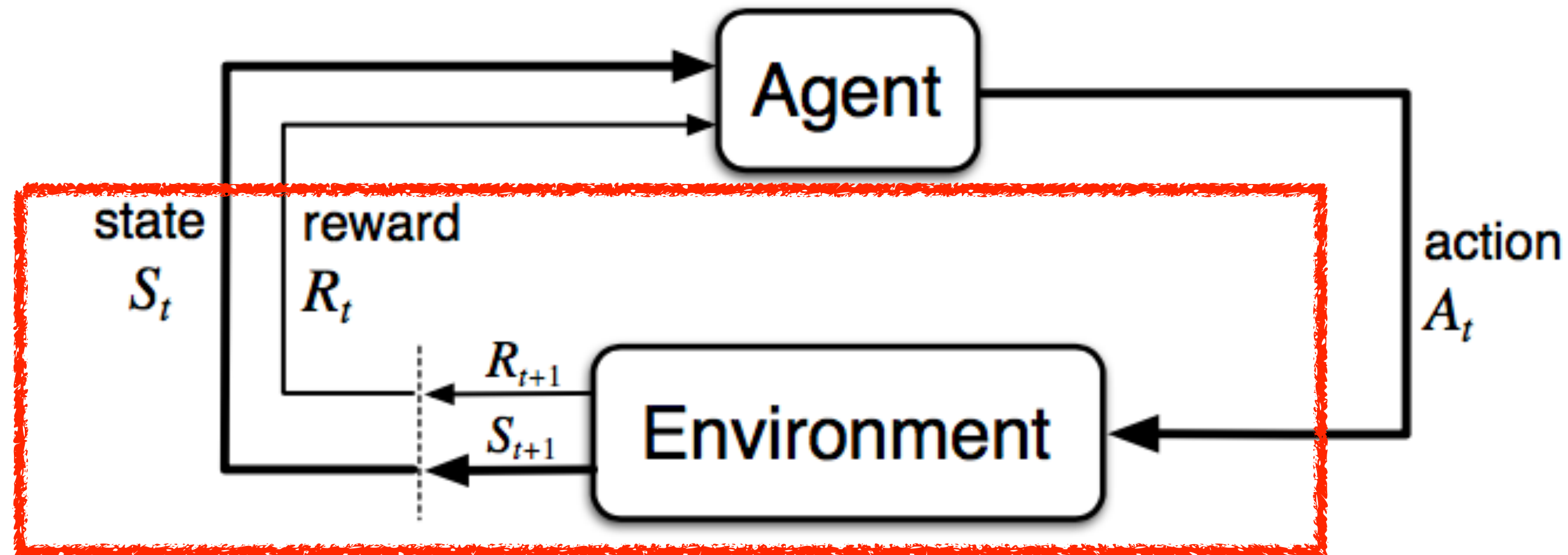


Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., & Finn, C. (2018). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *ICLR 2018*

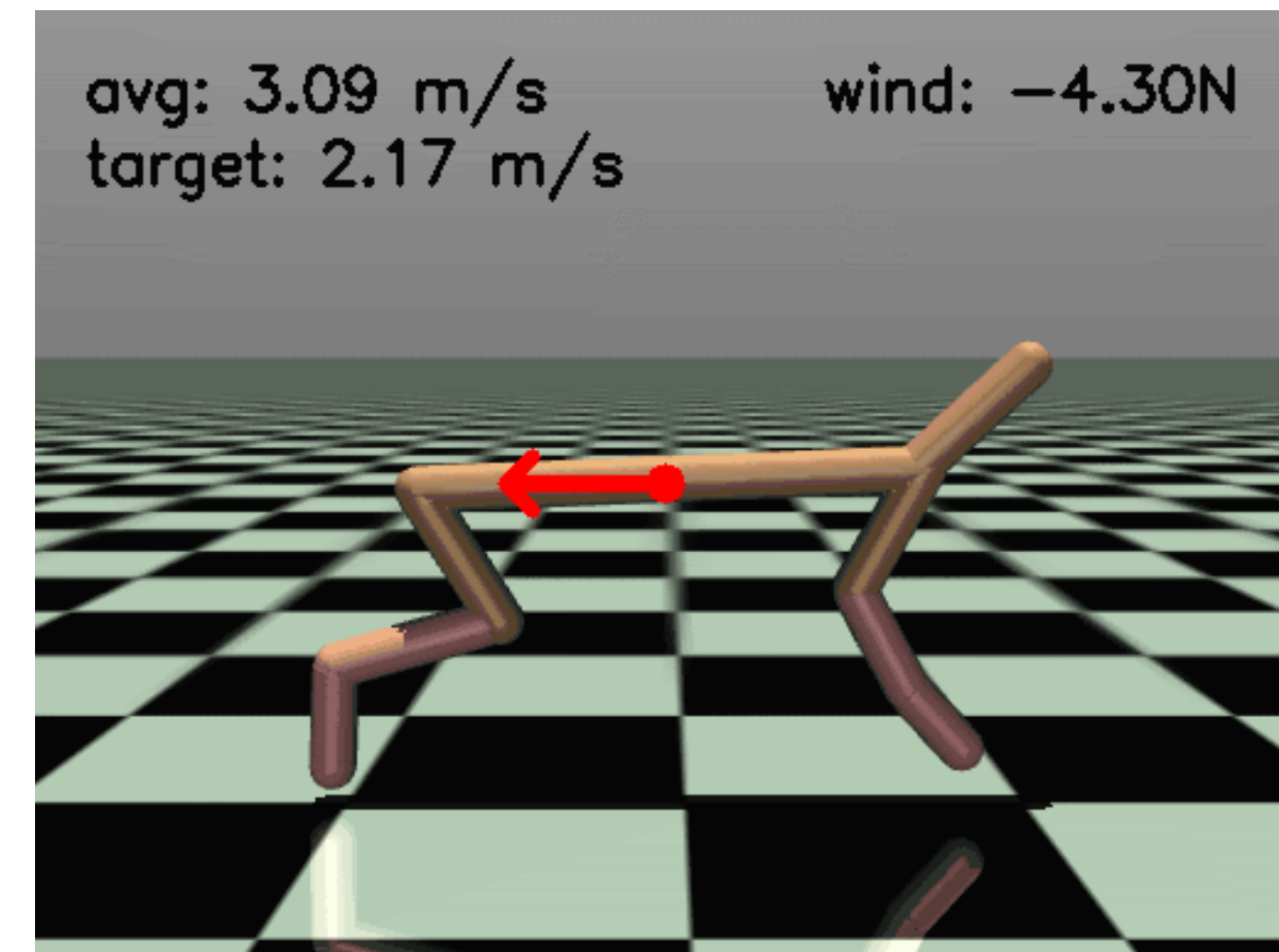
Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., & Celi, L. A. (2019). Guidelines for reinforcement learning in healthcare. *Nature medicine*

Zhao, X., Xia, L., Zhang, L., Ding, Z., Yin, D., & Tang, J. (2018). Deep reinforcement learning for page-wise recommendations. *ACM Conference on Recommender Systems 2018*.

Non-stationary RL



Time-varying environments



Stationary

Non-stationary

State-transition function:

$$P_{ss'}^a = P(s' | s, a) = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

Reward function:

$$R(s, a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

State-transition function:

$$P_{ss'}^a = P(s' | s, a, \theta^s) = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a, \theta_t^s]$$

Reward function:

$$R(s, a, \theta^r) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a, \theta_t^r]$$

**How to capture the non-stationarity
and locate the exact changes
explicitly?**

Factored Non-stationary MDP (FN-MDP)

- **Bayesian Dynamic Network** \mathcal{G} over all dimensions of states, actions, rewards and changing factors;

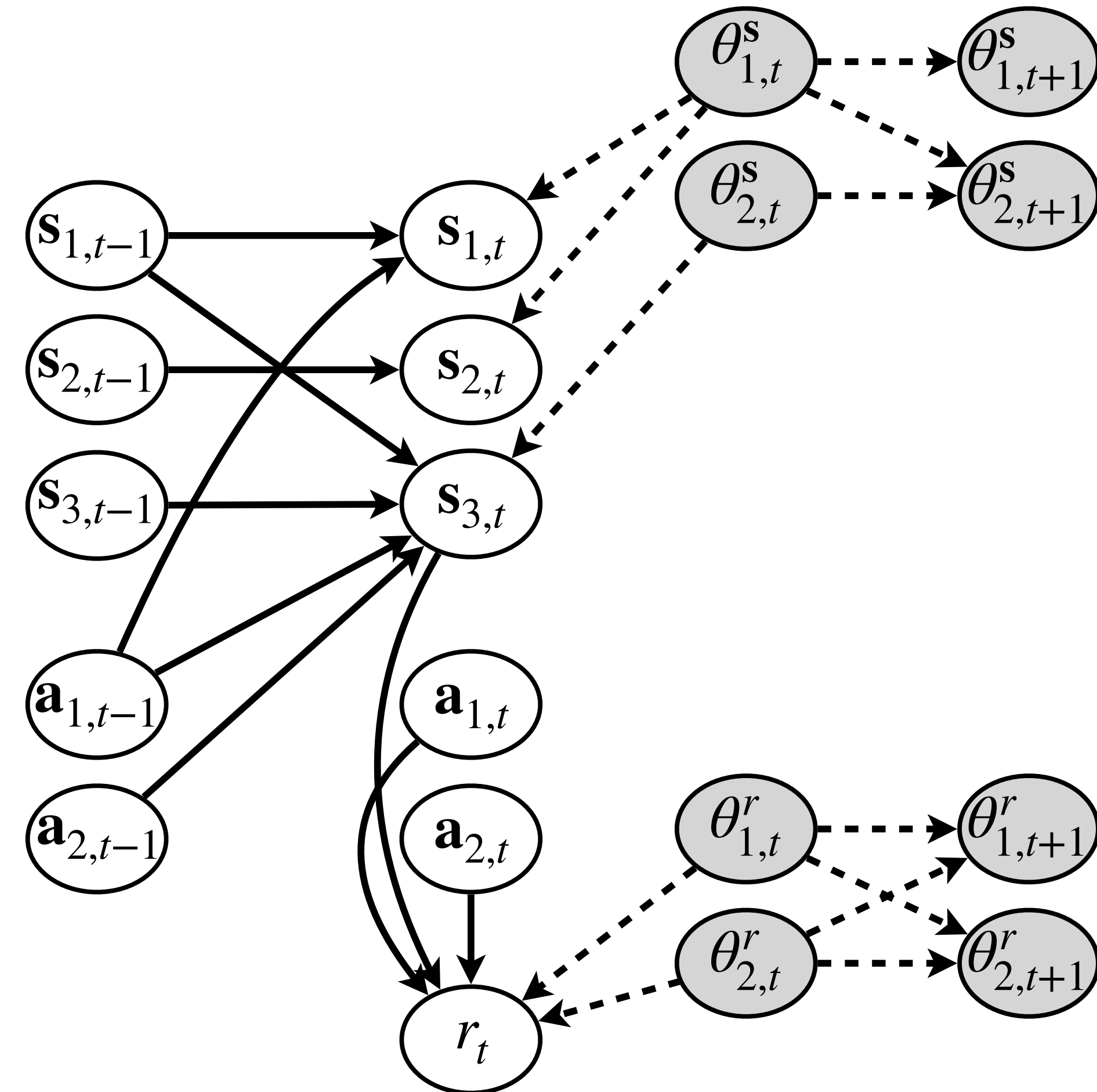
- State transition: $\mathbb{P}_s(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \boldsymbol{\theta}_t^s) = \prod_{i=1}^d \mathbb{P}_s(s_{i,t} | pa(s_{i,t}))$

- Reward functions: $\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t, \boldsymbol{\theta}_t^r) = \mathcal{R}(pa(r_t))$

- Markov properties of change factors

$$\mathbb{P}_{\theta^s}(\boldsymbol{\theta}_t^s | \boldsymbol{\theta}_{t-1}^s) = \prod_{j=1}^p \mathbb{P}_{\theta^s}(\theta_{j,t}^s | pa(\theta_{j,t}^s))$$

$$\mathbb{P}_{\theta^r}(\boldsymbol{\theta}_t^r | \boldsymbol{\theta}_{t-1}^r) = \prod_{k=1}^q \mathbb{P}_{\theta^r}(\theta_{k,t}^r | pa(\theta_{k,t}^r))$$



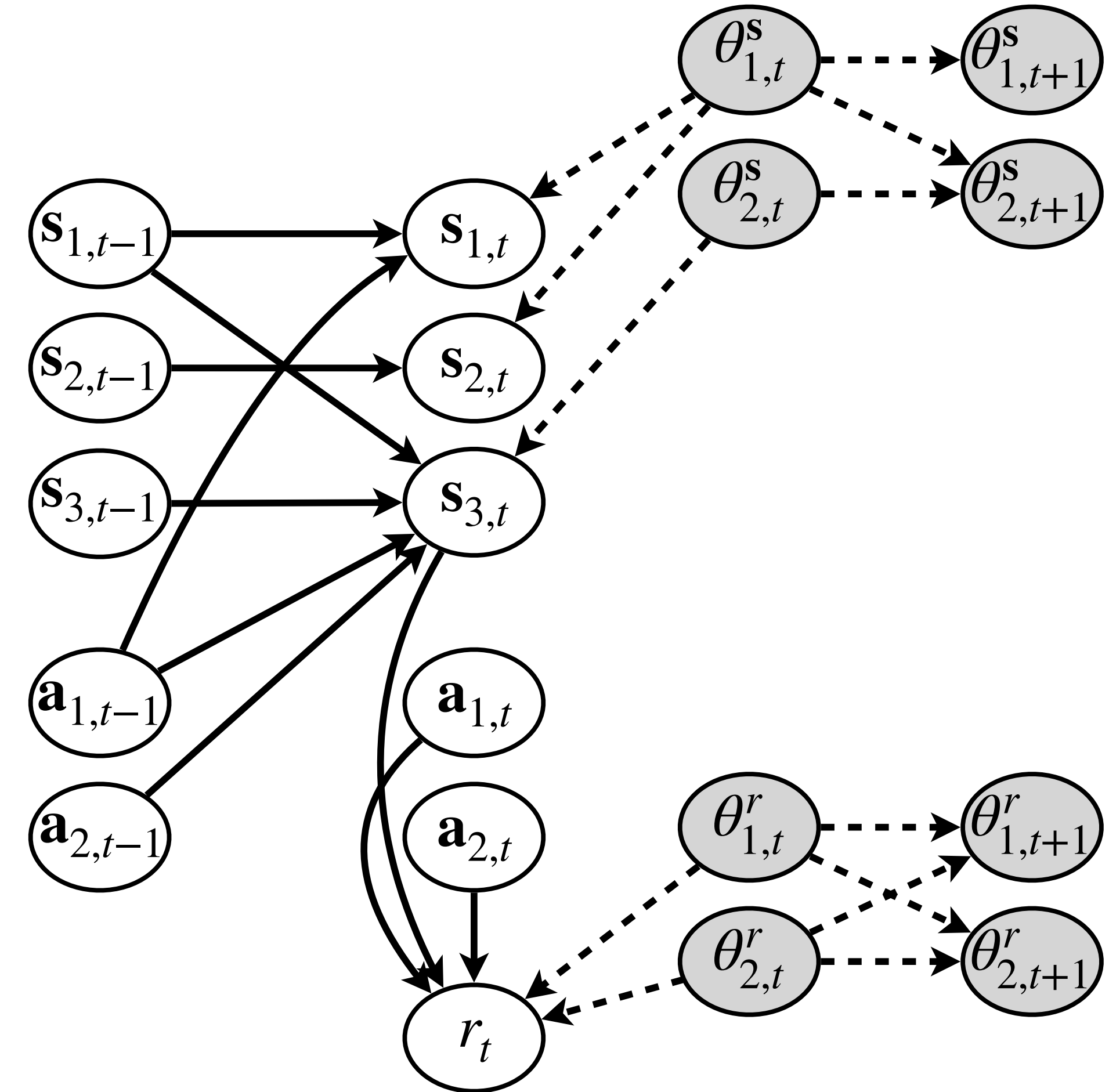
Structural Equations to model FN-MDP

$$s_{i,t} = f_i \left(\mathbf{c}_i^{s \rightarrow s} \odot \mathbf{s}_{t-1}, \mathbf{c}_i^{a \rightarrow s} \odot \mathbf{a}_{t-1}, \mathbf{c}_i^{\theta^s \rightarrow s} \odot \boldsymbol{\theta}_t^s, \epsilon_{i,t}^s \right)$$

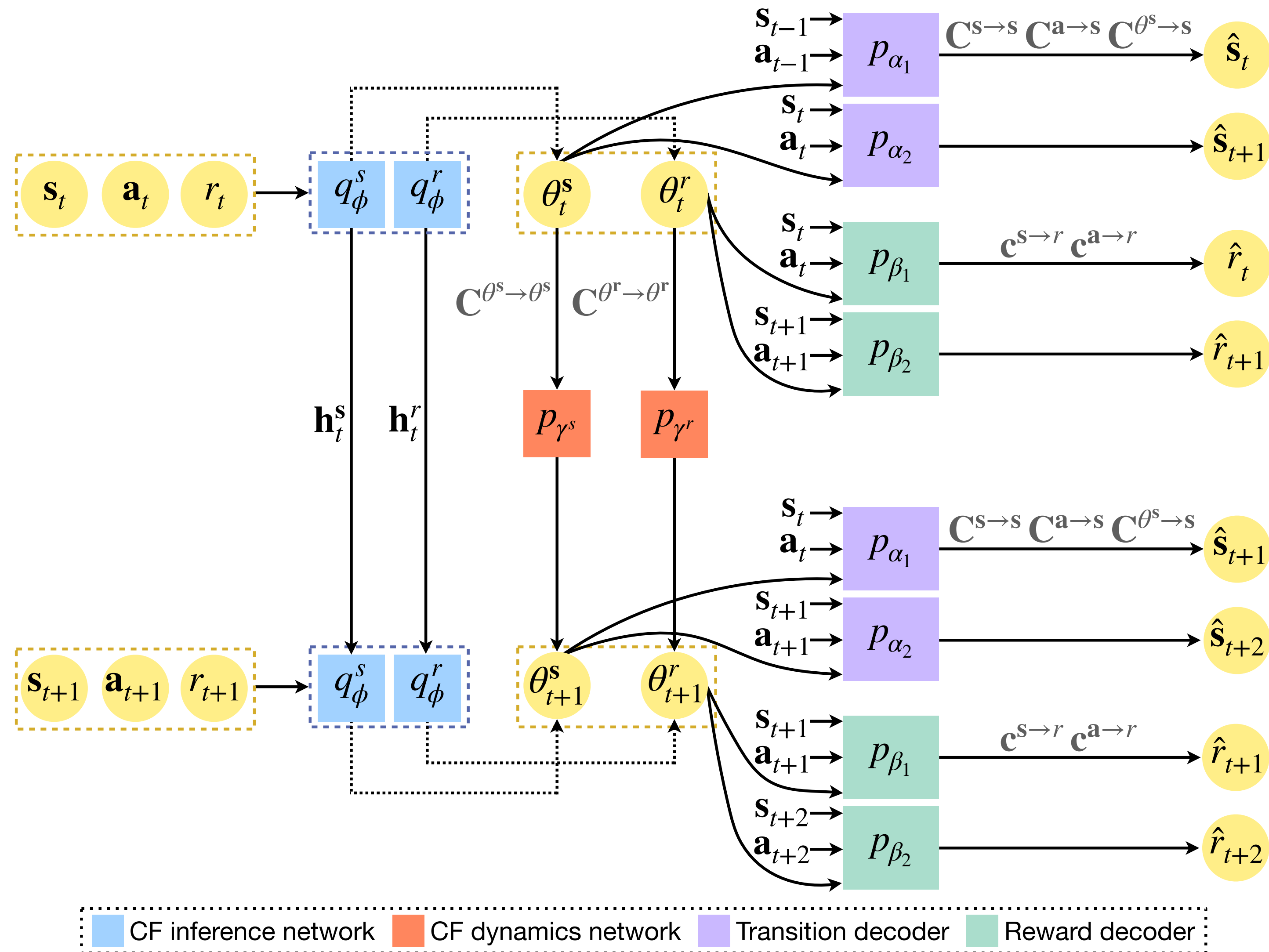
$$r_t = h \left(\mathbf{c}^{s \rightarrow r} \odot \mathbf{s}_t, \mathbf{c}^{a \rightarrow r} \odot \mathbf{a}_t, \boldsymbol{\theta}_t^r, \epsilon_t^r \right)$$

$$\theta_{j,t}^s = g^s \left(\mathbf{c}_j^{\theta^s \rightarrow \theta^s} \odot \boldsymbol{\theta}_{t-1}^s, \epsilon_t^{\theta^s} \right)$$

$$\theta_{k,t}^r = g^r \left(\mathbf{c}_k^{\theta^r \rightarrow \theta^r} \odot \boldsymbol{\theta}_{t-1}^r, \epsilon_t^{\theta^r} \right)$$



Learn the FN-MDP



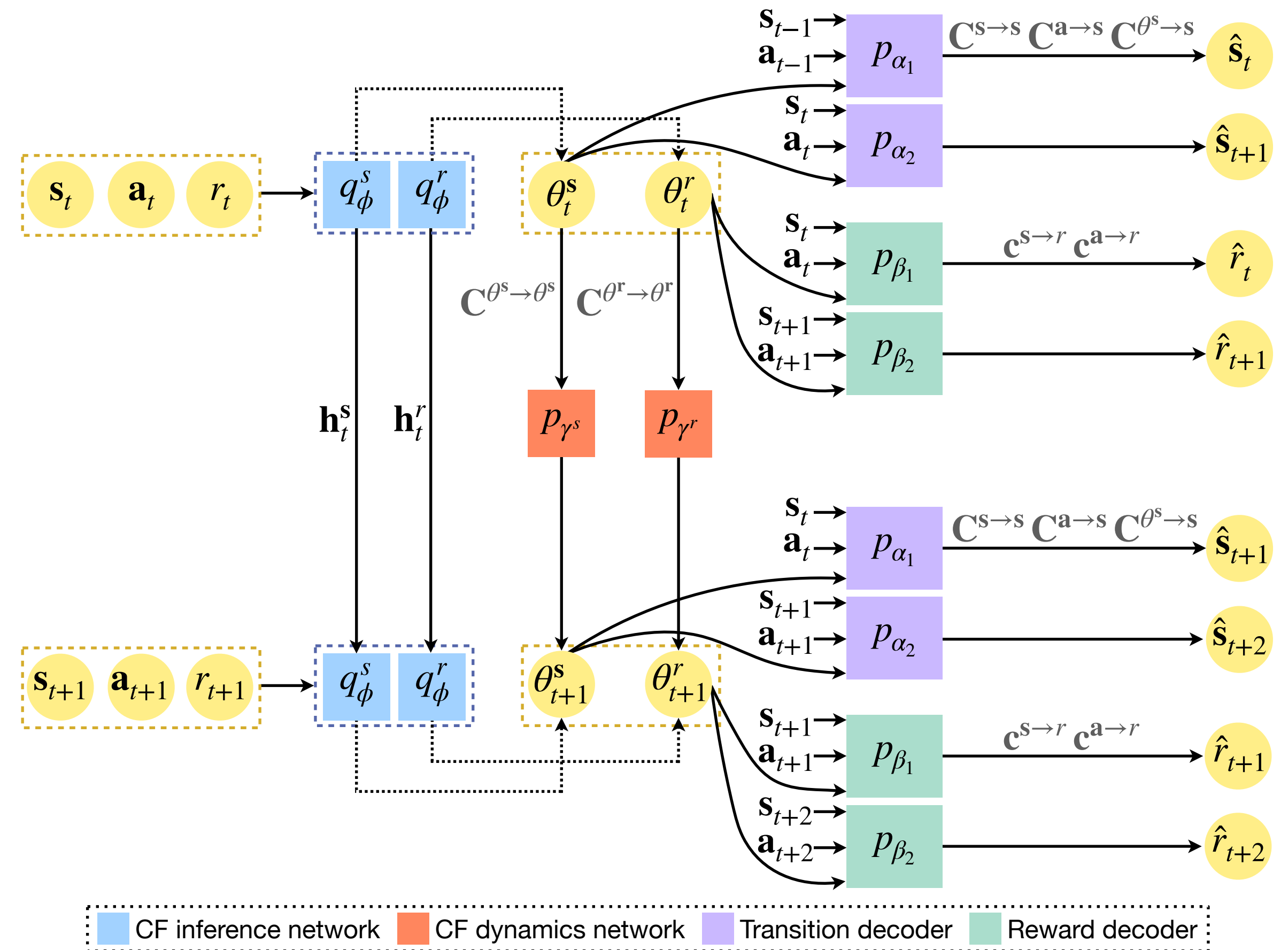
Learn the FN-MDP

CF inference network

- Infer the changes on dynamics:

$$q_{\phi^s}(\theta_t^s | s_t, \mathbf{a}_t) = q_{\phi^s}(\theta_t^s | s_t, \mathbf{a}_t, r_t, \mathbf{h}_{t-1}^s)$$
- Infer the changes on rewards:

$$q_{\phi^r}(\theta_t^r | s_t, \mathbf{a}_t, r_t) = q_{\phi^r}(\theta_t^r | s_t, \mathbf{a}_t, r_t, \mathbf{h}_{t-1}^r)$$



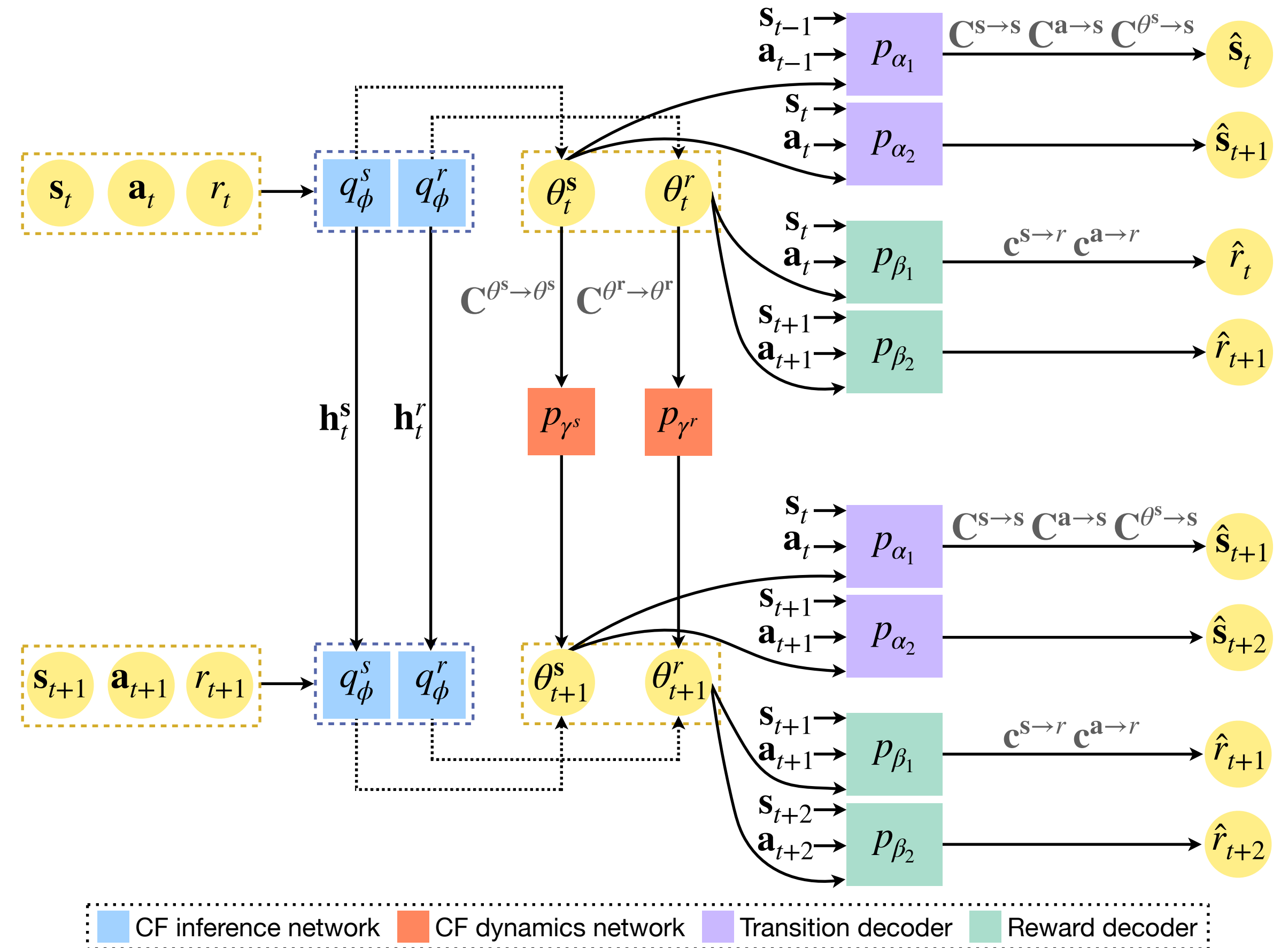
Learn the FN-MDP

CF dynamics network

- Learn the Markov properties of change factors

- $p_{\gamma^s}(\theta_{t+1}^s \mid \theta_t^s, \mathbf{C}^{\theta^s \rightarrow \theta^s})$

- $p_{\gamma^r}(\theta_{t+1}^r \mid \theta_t^r, \mathbf{C}^{\theta^r \rightarrow \theta^r})$



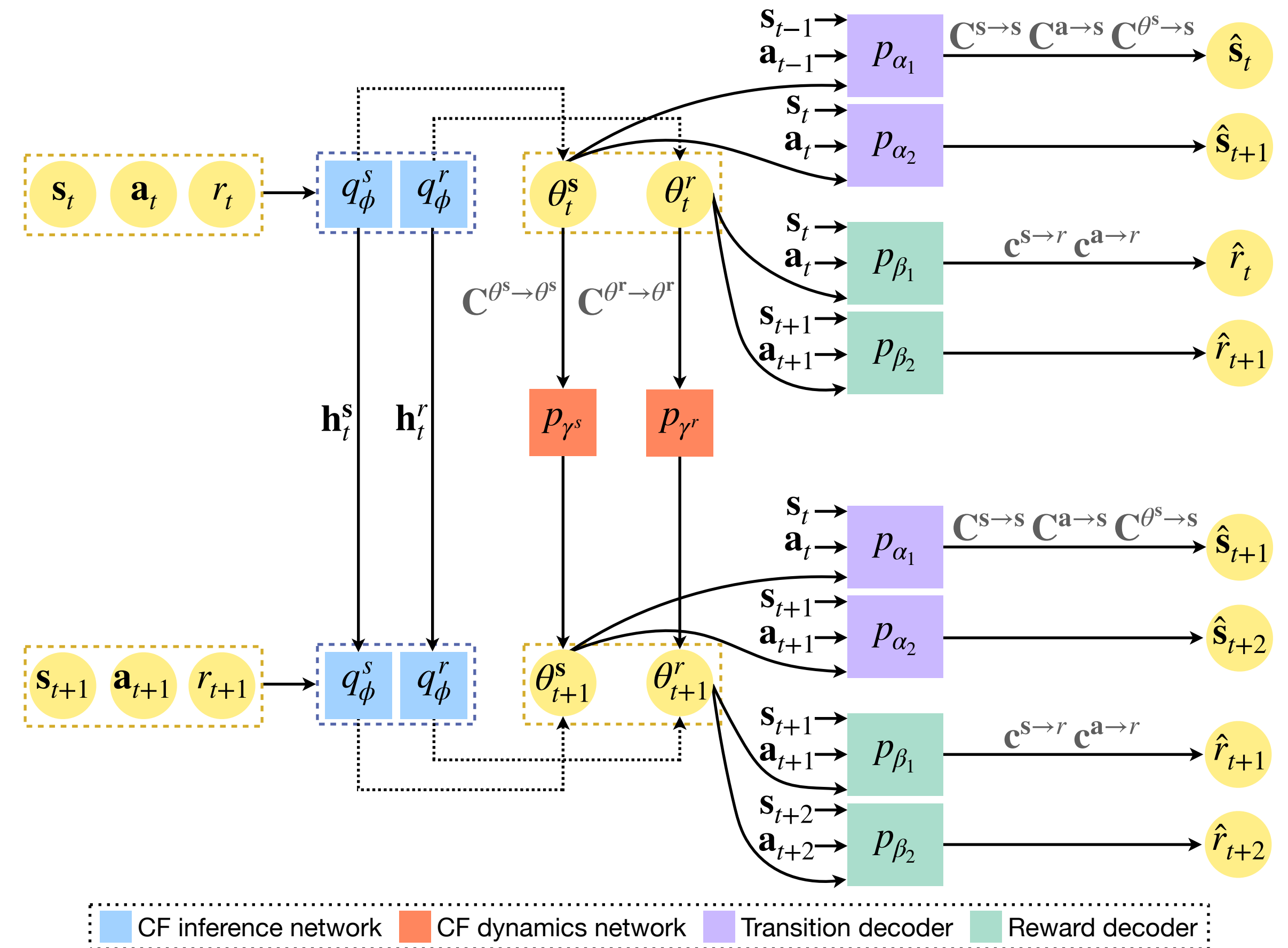
Learn the FN-MDP

Transition decoders

- Reconstruct the current states and predict the future states

$$\mathcal{L}_{\text{rec-dyn}} = \sum_{t=1}^{T-2} \mathbb{E}_{\theta_t^s \sim q_\phi} \log p_{\alpha_1} (s_t | s_{t-1}, a_{t-1}, \theta_t^s; C^{\cdot \rightarrow s})$$

$$\mathcal{L}_{\text{pred-dyn}} = \sum_{t=1}^{T-2} \mathbb{E}_{\theta_t^s \sim q_\phi} \log p_{\alpha_2} (s_{t+1} | s_t, a_t, \theta_t^s)$$



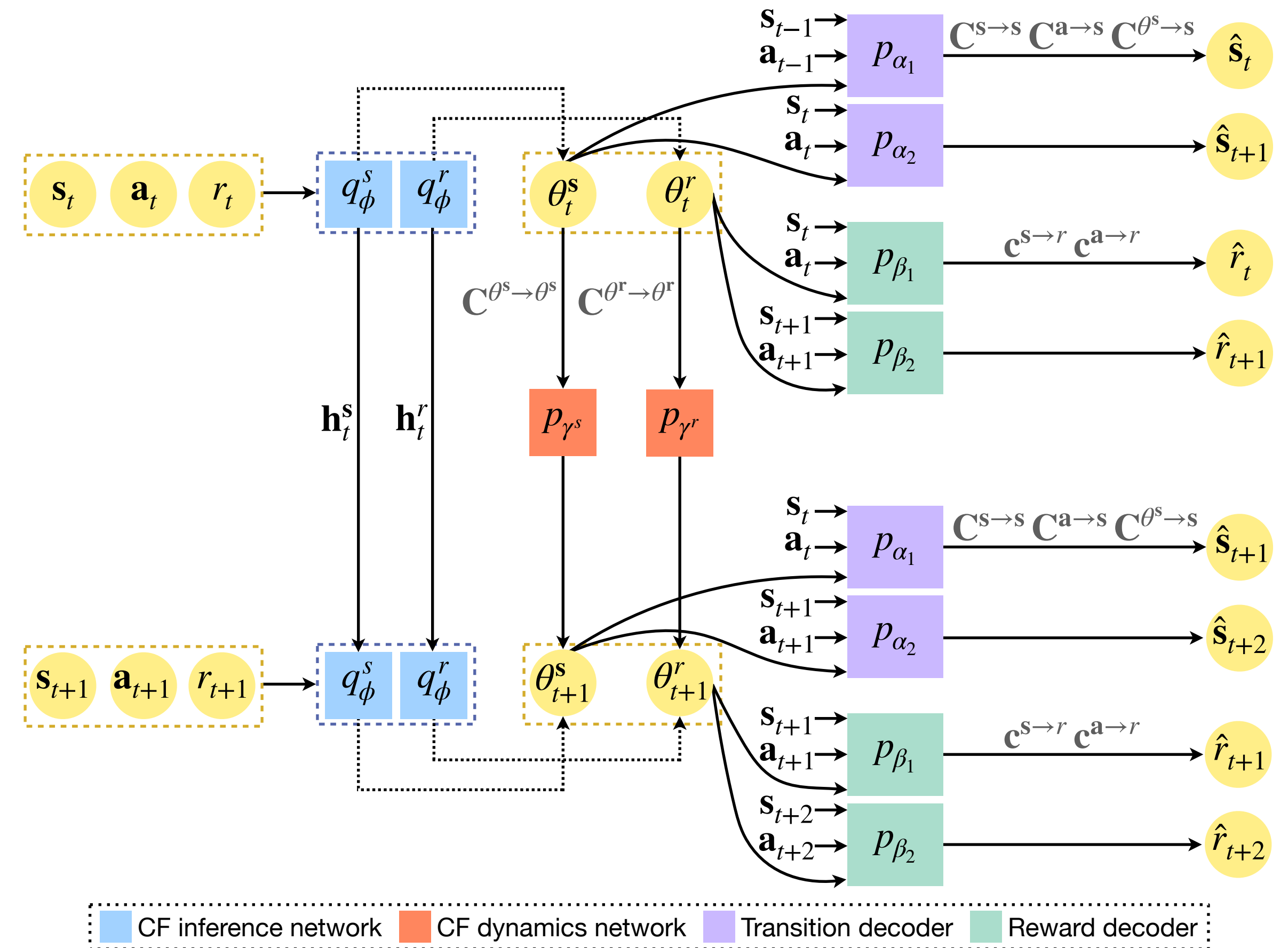
Learn the FN-MDP

Reward decoders

- Reconstruct the current rewards and predict the future rewards

$$\mathcal{L}_{\text{rec-rw}} = \sum_{t=1}^{T-2} \mathbb{E}_{\theta_t^r \sim q_\phi} \log p_{\beta_1}(r_t | s_t, a_t, \theta_t^r; \mathbf{c}^{s \rightarrow r}, \mathbf{c}^{a \rightarrow r})$$

$$\mathcal{L}_{\text{pred-rw}} = \sum_{t=1}^{T-2} \mathbb{E}_{\theta_t^r \sim q_\phi} \log p_{\beta_2}(r_{t+1} | s_{t+1}, a_{t+1}, \theta_t^r)$$



Policy Learning with FN-MDP

- With the learned FN-MDP, we can identify the minimal and sufficient sets of states and change factors for efficient policy learning

$$\tau_{\psi} \left(\mathbf{a}_t \mid \mathbf{s}_t^{\min}, q_{\phi} \left(\boldsymbol{\theta}_t^{\min} \mid \boldsymbol{\tau}_{0:t} \right) \right)$$

- To identify the minimal and sufficient sets:

$$s_{i,t} \in \mathcal{S}^{\min} \iff s_{i,t} \rightarrow \dots \rightarrow r_{t+\tau} \text{ for } \tau \geq 1$$

$$\theta_i \in \boldsymbol{\theta}^{\min} \iff \theta_i \rightarrow \dots \rightarrow r_{t+\tau} \text{ for } \tau \geq 1$$

Evaluation Objectives

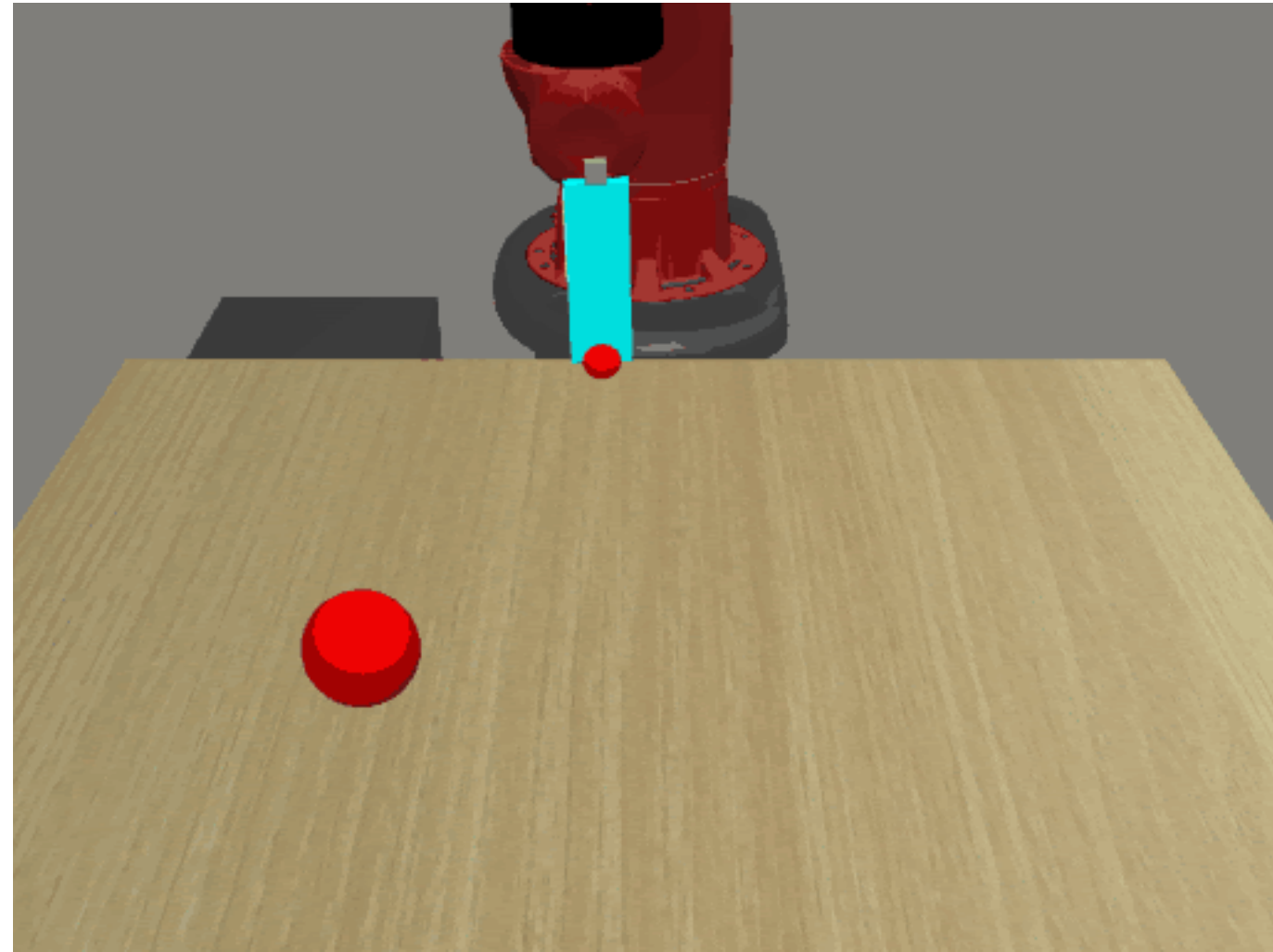
- Can the framework improve the learning performances under non-stationary RL?
- Can the framework learn a more compact latent representation against other baselines?
- Can the framework deal with a more general set of non-stationarity in RL?

Evaluation

Baselines

- Tracking non-stationarity via inferring the latent parameters via meta-learning on a set of tasks: [TRIO](#) (Poiani et al., IJCAI'21), [VariBAD](#) (Zintgraf et al., ICLR'21);
- Learn the latent parameters to capture the non-stationary components: [LILAC](#) (Xie et al., ICML'21), [ZeUS](#) (Sodhani et al., ArXiv'22)

Evaluation Benchmarks

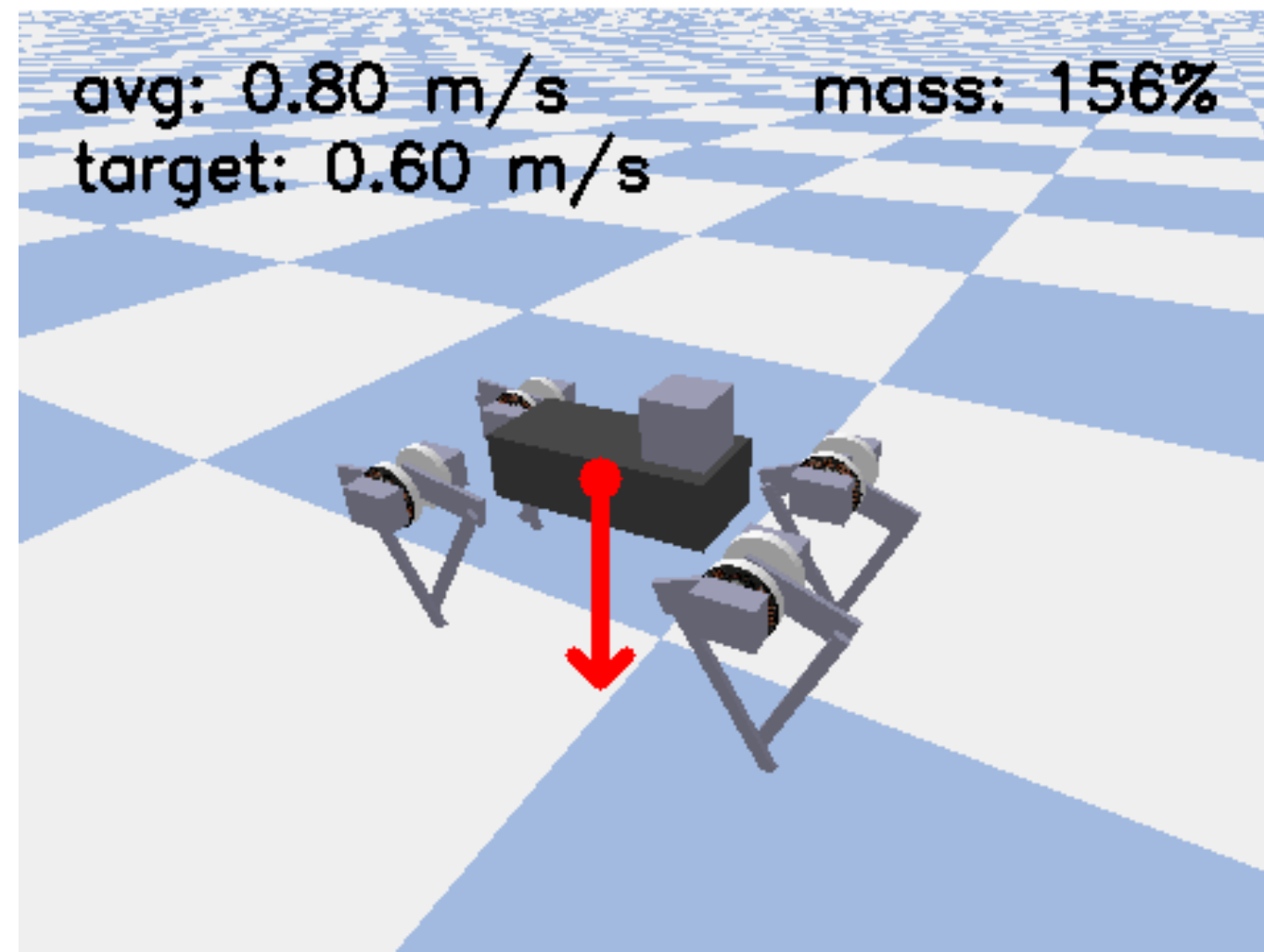


Sawyer

**Dyn
changes**

**Reward
changes**

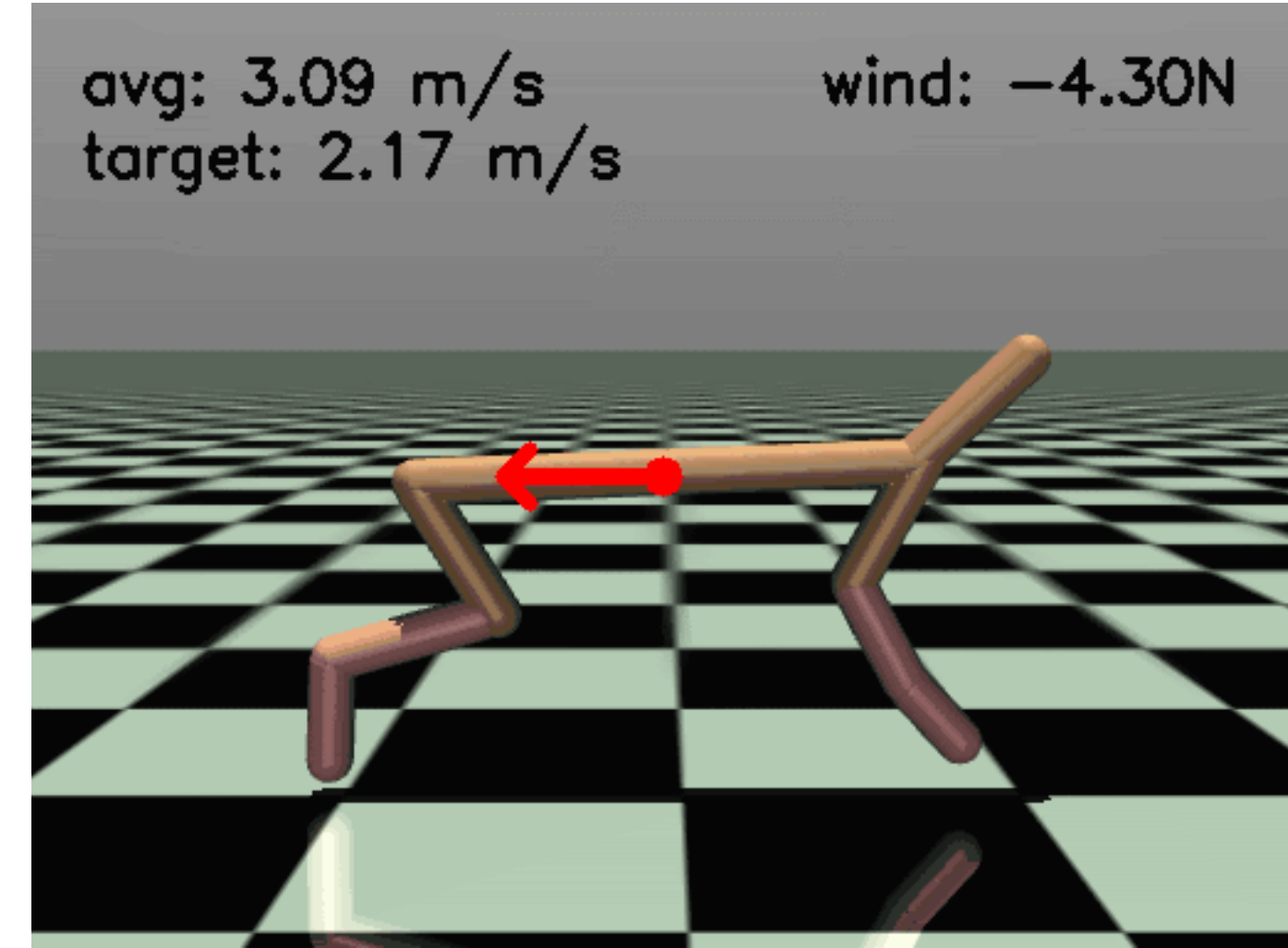
Target location



Mini-taur

Gravity

Target speed



Half-Cheetah

Wind speed

Target speed

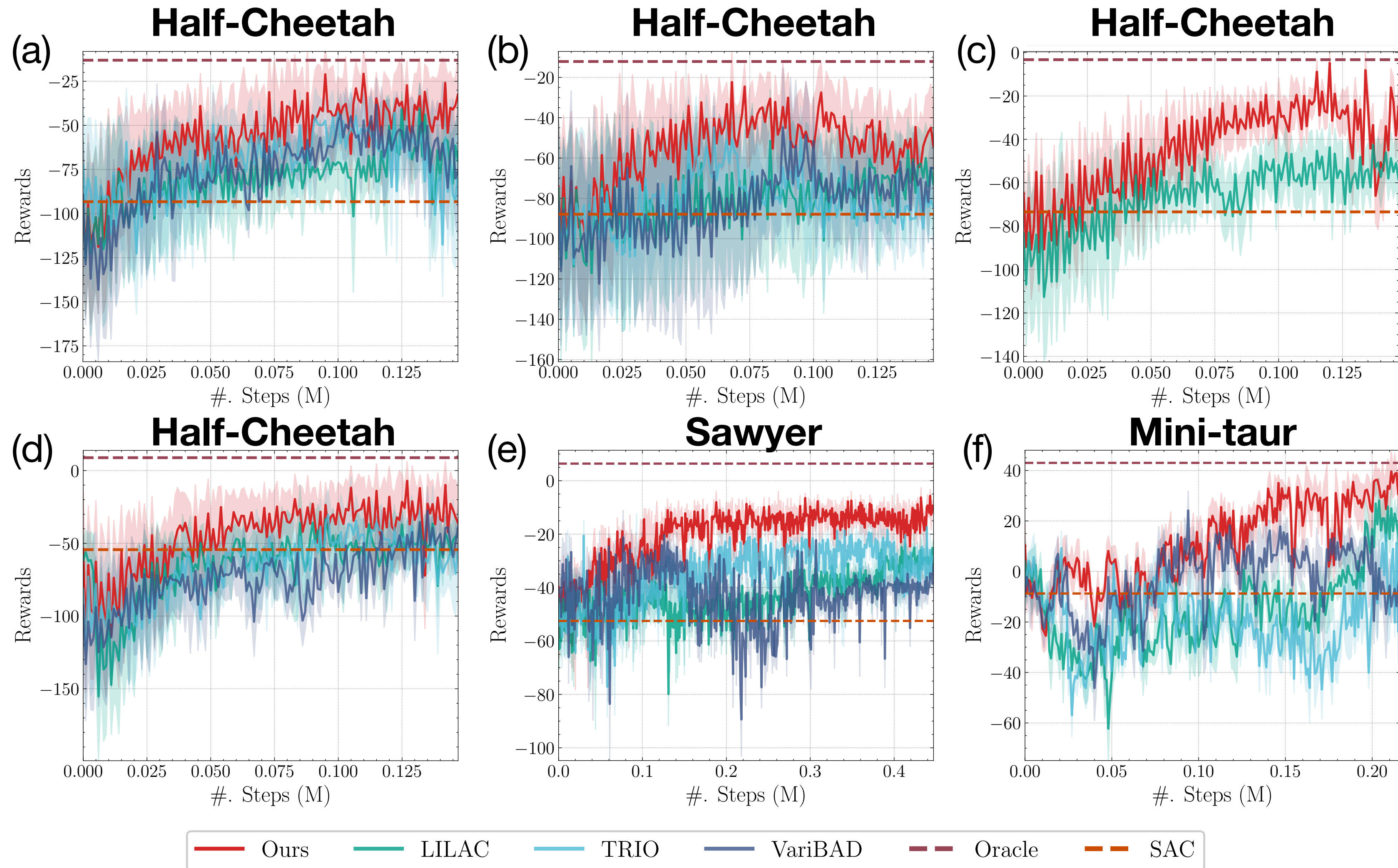
Evaluation

A more general non-stationary setting

	LILAC	TRIO	VariBAD	ZeUS	Ours
Continuous changes	✓				✓
Discrete changes	✓	✓	✓	✓	✓
Multiple changes				✓	✓
Learning from pixels				✓	✓

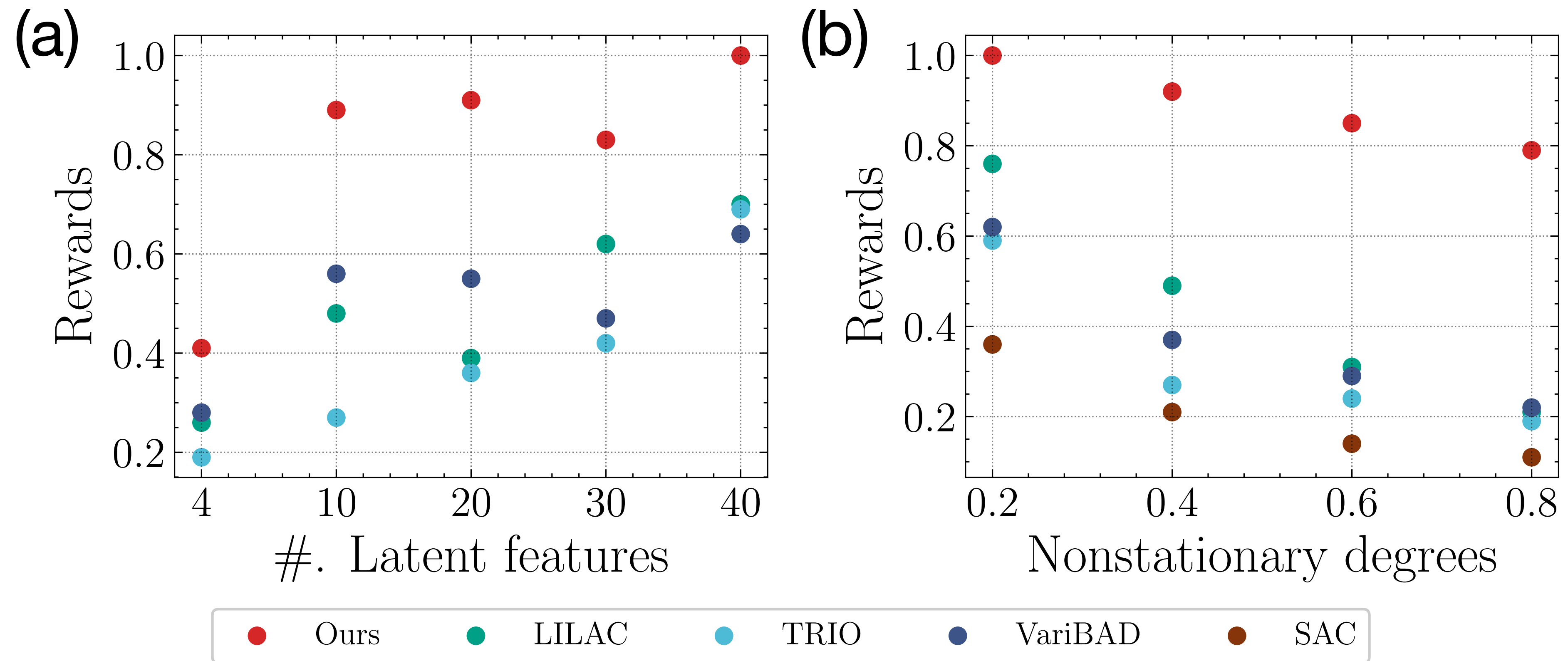
Evaluation

Learning curves



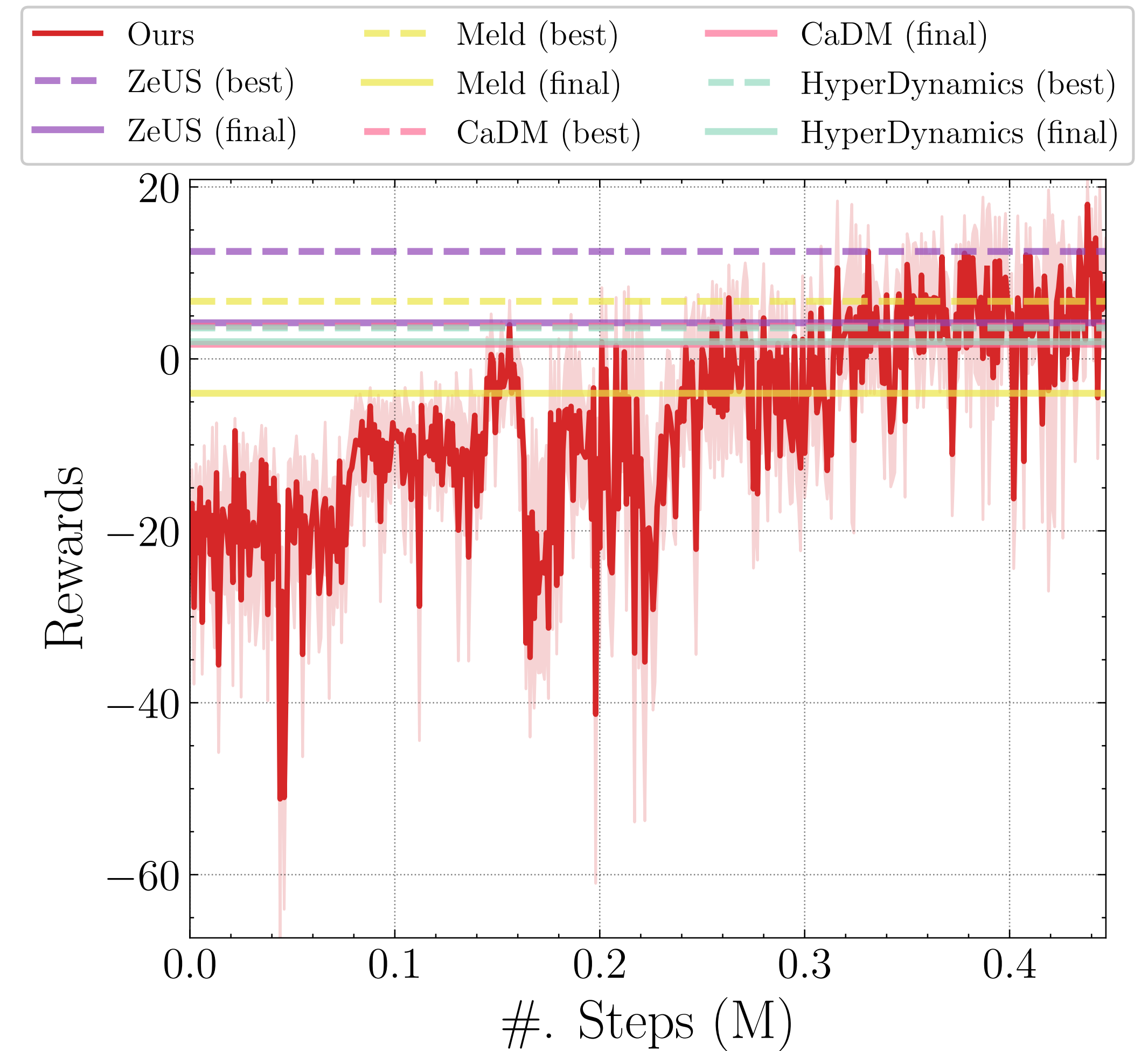
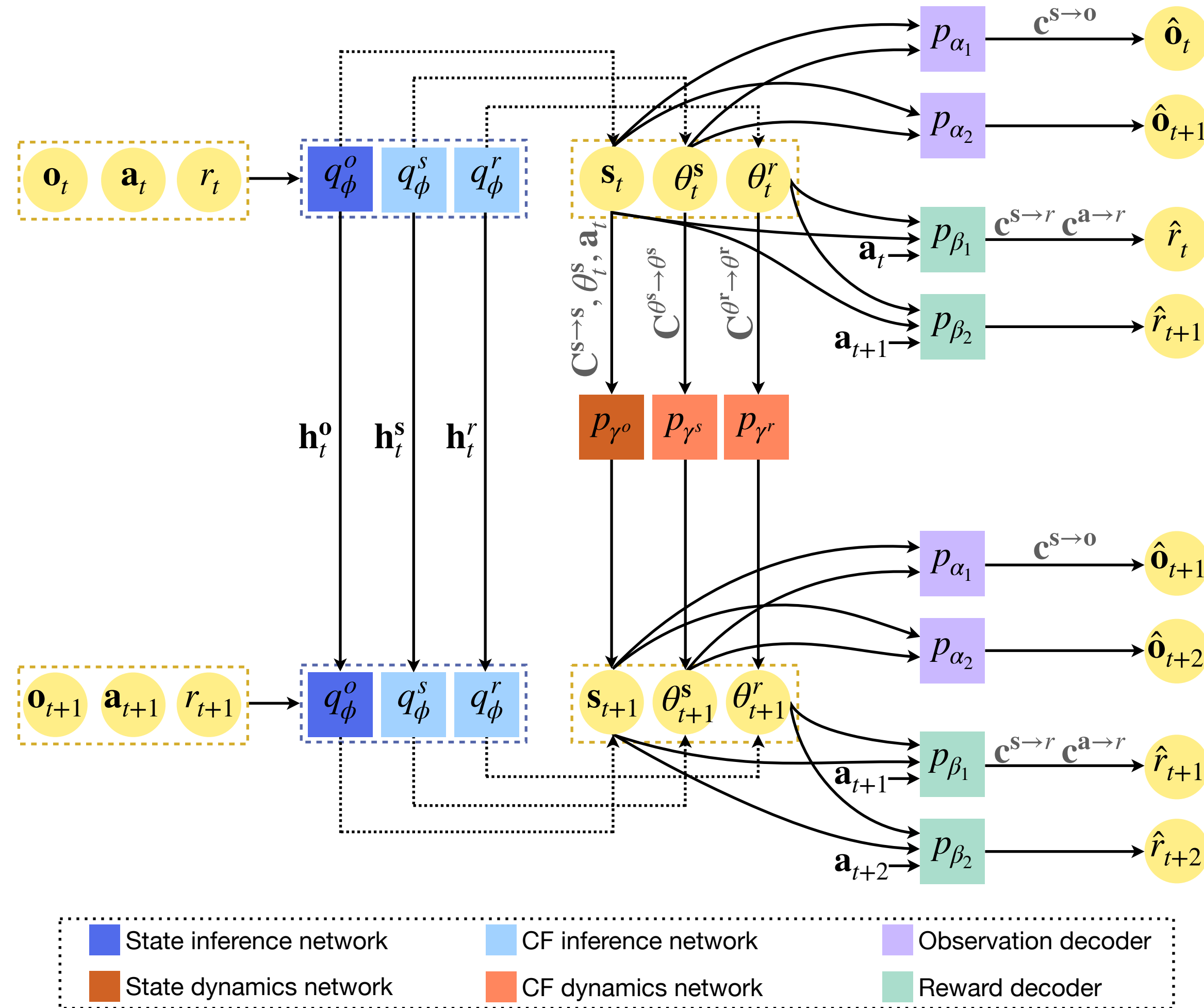
Evaluation

Compactness and robustness



Evaluation

Learning from raw pixels

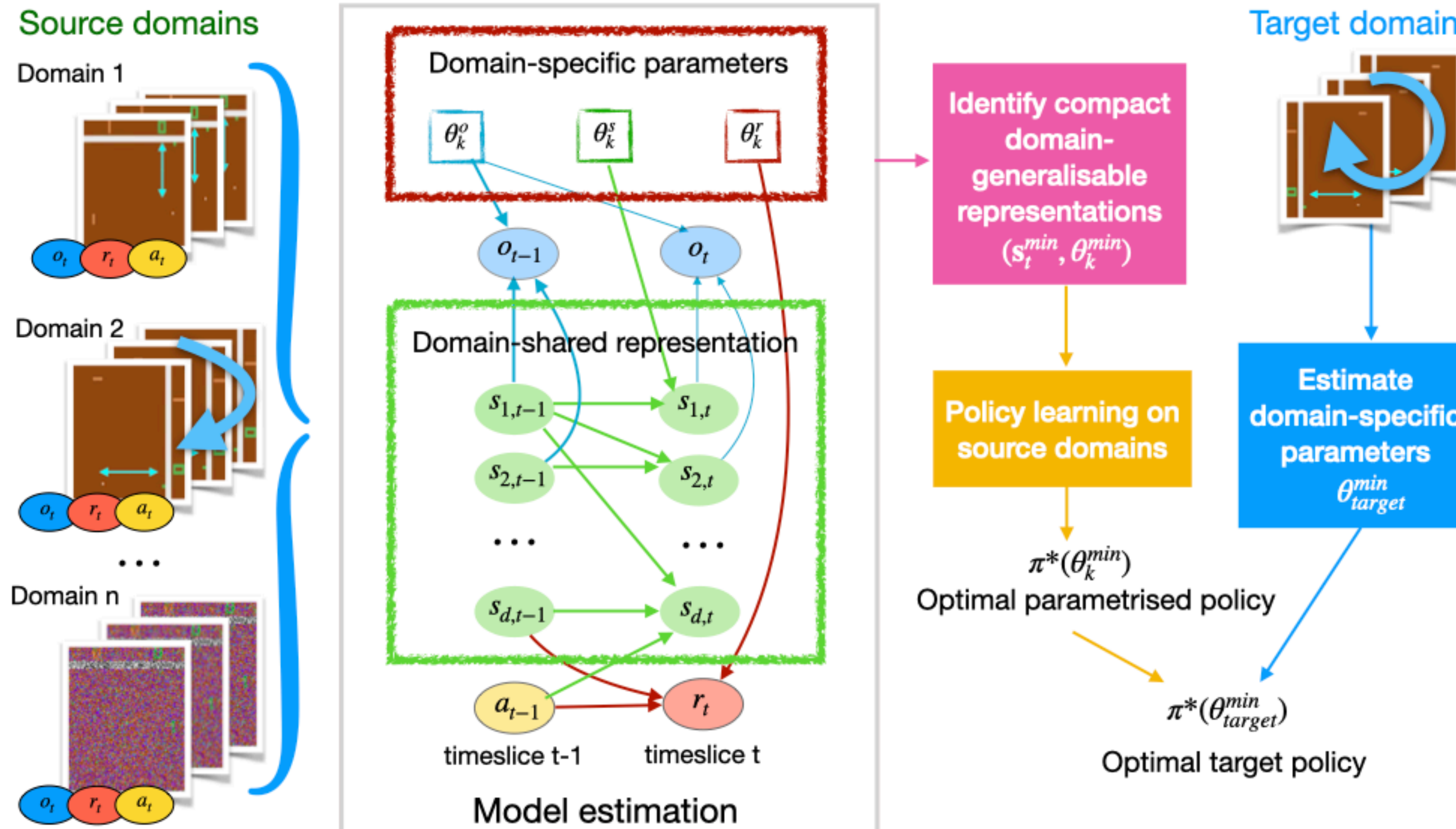


AdaRL: what, where and how to adapt in transfer reinforcement learning

Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, Kun Zhang

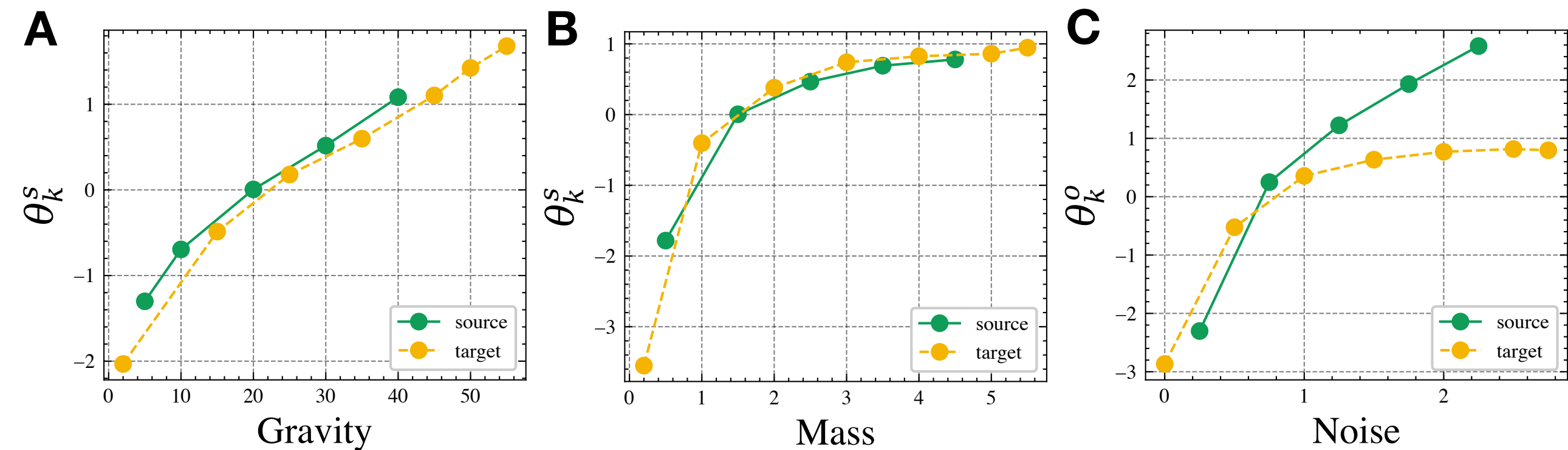
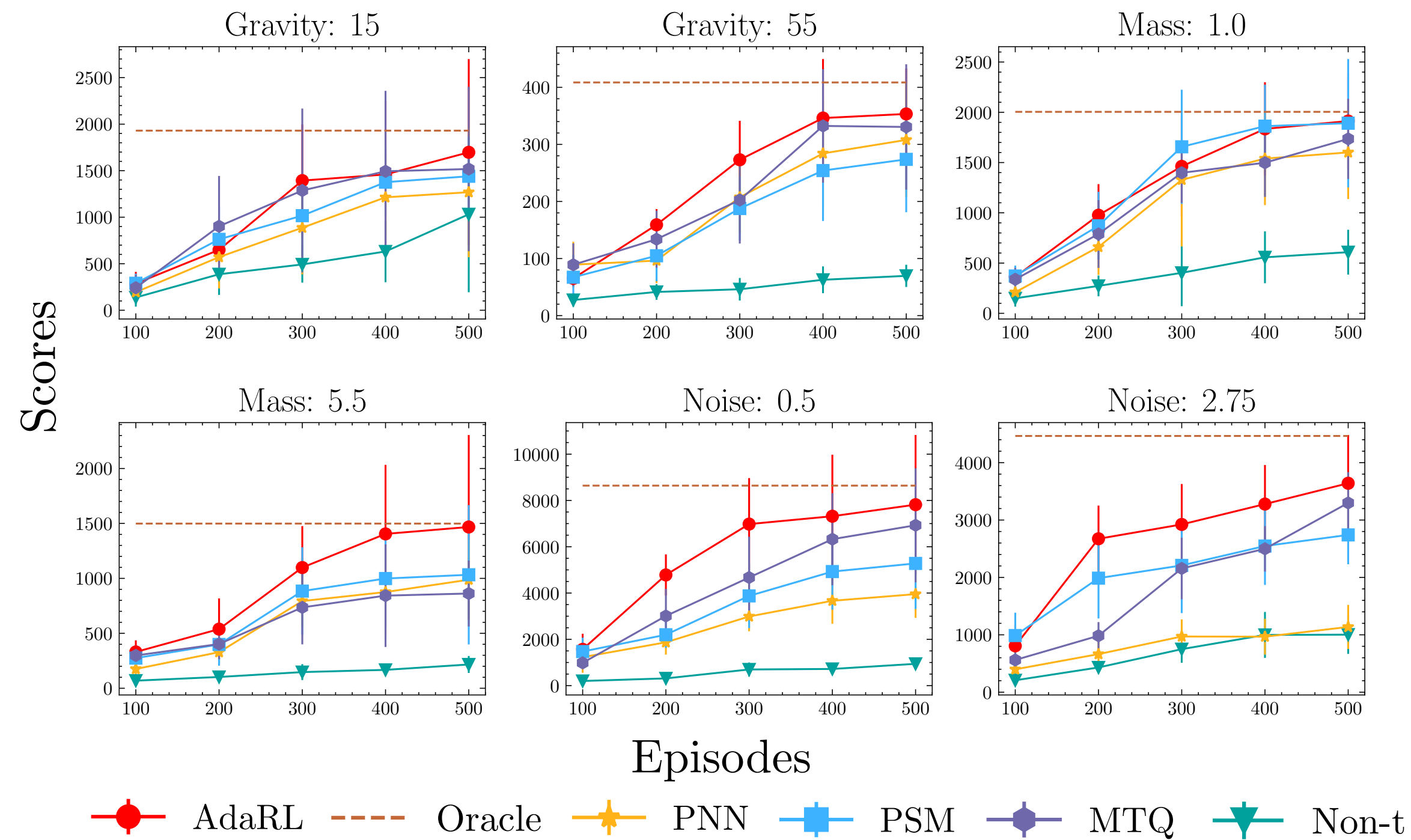


System Diagram



Experiments

- Cartpole (gravity, mass, and noise on observation)
- Pong (Orientation, Size, color, noise and reward functions)



Conclusions

- We propose a general non-stationary RL framework that learns a **factored** representation for adapting to changes in dynamics and reward functions.
- We formalise a unified framework that can handle multi-domain and non-stationary settings;
- Results outperform the state of the art on rewards, compactness of the latent space representation and robustness to different change factors.

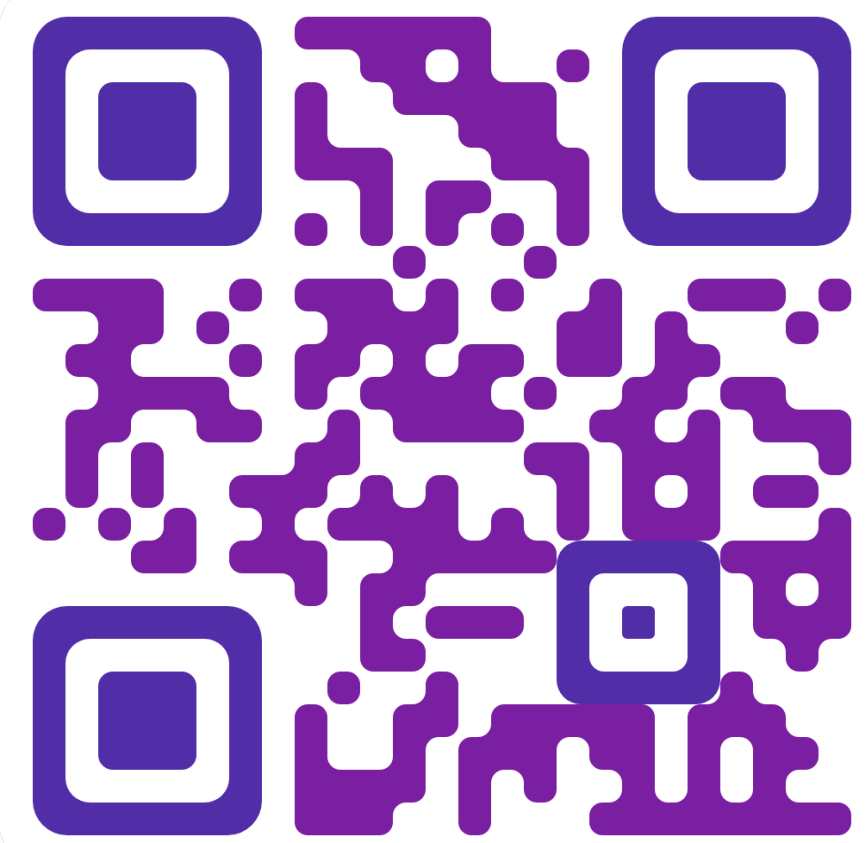
Thanks!

Q&A

Contact: ffeng1017@gmail.com



AdaRL paper



FANS-RL paper



Code