

Distributional Active Inference

Melih Kandemir

University of Southern Denmark

Department of Mathematics and Computer Science (IMADA)

kandemir@imada.sdu.dk

04-05-2026

A thought experiment

*“Der Mensch kann zwar tun, was er will,
aber er kann nicht wollen, was er will.”*

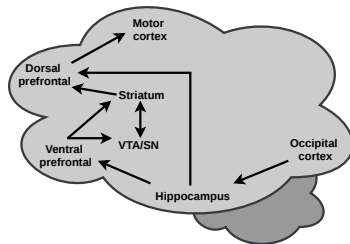
Arthur Schopenhauer, Preisschrift über die Freiheit des Willens (1839)

“One can indeed do what (s)he wants
(motor cortex / policy),
can also perceive what (s)he wants
(???),
but (s)he cannot want what (s)he wants
(basal ganglia/reward),
Neither can (s)he perceive without predicting
(???) .”

Arthur Schopenhauer, NeurIPS Keynote Speech (2026)

Cognition as a machine learning algorithm

- What's the information that is flowing?
- What is the "purpose" (Lyapunov function) of this flow?
- Which equations govern the flow?
- How does adaptation emerge from this flow?



History of the answers to how the brain may be organized:

- **Unconscious inference (conceptual)**: Stimuli are raw signals, perceptions are their generative process (Helmholtz, 1867)
- **Predictive coding (algorithmic)**: Stimuli are residuals, perceptions are error-minimizing predictions (Rao & Ballard, 1999)
- **Bayesian brain (computational)**: Stimuli are likelihoods, perceptions are posterior beliefs (Knill & Pouget, 2004)
- **Active inference (synthesis)**: Bayesian predictive coding where actions improve model fit (Friston et al., 2010)

Bayesian predictive coding

Let

$$P_W(X, Y, S) \triangleq P_D(X|Y, S)P_0(Y, S)$$

be the agent's world model with a prior belief over the latent embeddings $P_0(Y, S)$ and a decoding likelihood $P_D(X|Y, S)$. The Evidence Lower Bound (ELBO) for observation x and approximate posterior $P_Q(Y, S)$ is

$$L(x, P_W, P_Q) = \mathbb{E}_{y, s \sim P_Q} \left[\log (P_D(x|y, s)P_0(y, s)/P_Q(y, s)) \right].$$

Solve $P_Q := \arg \max_{\bar{P}_Q} L(x, P_W, \bar{P}_Q)$ assuming a prior with density

$$f_{P_0}(Y := y) \propto \exp(\mathbb{E}_{x' \sim P_W | \text{do}(Y \sim \delta_y, S \sim P_Q)} [L(x', P_W | \text{do}(Y \sim \delta_y, S \sim P_Q), P_Q)]).$$

Such a prior favors a world model that best explains its own predictions. This is where **predictive coding** fits into the picture.

The active inference prior explained and simplified

Before building the prior, decompose the approximate world model in the reverse direction

$$P_{W|\text{do}(Y, S \sim P_Q)}(X, Y, S) = P_D(X|Y, S)P_Q(Y, S) = \tilde{P}(Y, S|X)\tilde{P}(X)$$

and intervene into $\tilde{P}(X)$ by the desired state distribution $P_R(X)$. But this actually means

$$P_{\tilde{W}}(X, Y, S) \triangleq P_{W|\text{do}(Y, S \sim P_Q, X \sim P_R)}(X, Y, S) = P_Q(Y, S)P_R(X)$$

which makes intervened distribution of (Y, S) independent from X . Then the inference of $\tilde{P}(Y, S|X)$ is unnecessary! The resulting prior is

$$f_{P_0}(Y := y) \propto \exp(\mathbb{E}_{x' \sim P_D P_Q|y}[\log P_R(x')])$$

yielding the AIF objective

$$\begin{aligned} L(x, P_W, P_Q) = & \mathbb{E}_{y, s \sim P_Q}[\log P_D(x|y, s)] + \mathbb{E}_{y, s \sim P_Q}[\log P_0(s|y)] \\ & + \mathbb{E}_{x' \sim P_D P_Q}[\log P_R(x')] + \mathbb{H}[P_Q]. \end{aligned}$$

Extension to an MDP

Representing the latent embedding with (S, S') where S is the representation of the current perception and S' the future perceptions, we get

$$P_W(X', S', S, \pi | X, A) \triangleq \underbrace{P_D(X' | S')}_{\text{Decoder}} \underbrace{P_M(S' | S, A)}_{\text{Latent transition kernel}} \underbrace{P_E(S | X)}_{\text{Encoder}} \underbrace{P_0(\pi)}_{\text{Policy prior}}$$

Inspired by the SSM literature, assume the approximate posterior factorization

$$P_Q(S, S', \pi) \triangleq \underbrace{P_M(S' | S, A) P_E(S | X)}_{\text{Shared with the forward model}} P_A(\pi | X).$$

The resulting AIF ELBO is

$$\begin{aligned} L(x, a, x', P_W, P_Q) = & \mathbb{E}_{S' \sim P_M P_E | x, a} [\log P_D(x' | S')] \\ & + \mathbb{E}_{x'', \pi \sim P_D P_M P_E P_A | x'} \left[\log P_R(x'' | x', \pi(x')) \right] + \mathbb{H}[P_A | x']. \end{aligned}$$

Active inference theory is great but not yet complete...

Two streams of information converge on dopamine neurons

- the **perceived** reward (excitatory)
- the **predicted** reward (inhibitory)

However, active inference leaves the following aspects unexplained

- the role of temporal differences
- the choice of the planning horizon
- the role of state abstractions

which brings about the following limitations:

- extension to Bellman backups is ad-hoc
- the construction of the latent space is arbitrary

causing the following consequences in its adoption by the RL community:

- It is always considered as a model-based approach, which is expensive.
- It doesn't help more than justifying existing approaches.

Making a genuine RL algorithm out of active inference

Expectations from a satisfactory solution:

- Incorporate temporal differences into the theory, thereby make it applicable to infinite horizons
- Clarify the role of state abstractions
- Prescribe how the uncertainty caused by the state abstraction is quantified and exploited

Model-Based Policy Iteration (MBPI)

Input: $\pi_0, i := 0$

while True **do**

Model Update:

$$P_{i+1} := \arg \max_P \{ \mathbb{E}_{x \sim \mathcal{U}(\mathcal{X})} \mathbb{E}_{x' \sim P_*(\cdot|x, \pi_i(x))} [\log f_P(x'|x, \pi_i(x))] \}$$

Policy Evaluation:

$$Q_{i+1} := \arg \min_Q \mathbb{E}_{x \sim \mathcal{U}(\mathcal{X})} \left[\left(T_{P_{i+1}}^{\pi_i} Q(x, \pi_i(x)) - Q(x, \pi_i(x)) \right)^2 \right]$$

Policy Improvement:

$$\pi_{i+1} := \arg \max_{\pi} \mathbb{E}_{x \sim \mathcal{U}(\mathcal{X})} \left[Q_{i+1}(x, \pi(x)) \right]$$

$i := i + 1$

end while

where

$$(T_P^{\pi} Q)(x, a) \triangleq R(x, a) + \gamma \mathbb{E}_{x' \sim P(\cdot|x, a)} [Q(x', \pi(x'))]$$

is the Bellman operator indexed by a specific transition kernel P .

Distributional RL in a nutshell

Bellman equation: Enables learning from temporal differences

$$Q^\pi(s, a) = \mathbb{E}_{s' \sim P(\cdot|s, a)}[R(s, a) + \gamma Q^\pi(s', \pi(s'))]$$

Distributional Bellman equation: Enables *risk-aware* behavior (e.g., avoiding high-variance paths even if the mean is high) and provides a richer auxiliary signal for deep learning.

$$Z^\pi(s, a) \stackrel{D}{=} \underbrace{R(s, a)}_{\text{shift}} + \underbrace{\gamma}_{\text{rescale}} \underbrace{\mathbb{E}_{s' \sim P(\cdot|s, a)}[Z^\pi(s', \pi(s'))]}_{\text{mix}}.$$

The **mix** component creates multimodality: even if future returns are simple, their mixture is complex. This makes parametric assumptions (e.g., Z^π is Normal) insufficient. To remain distribution-agnostic, common practice uses quantile regression.

We will next incorporate distributional RL into the MBPI framework!

A probability measure for infinite-horizon trajectories

Let $P_\pi(B|x) \triangleq P(B|x, \pi(x))$ be a Markov kernel induced by policy $\pi : \mathcal{X} \rightarrow \mathcal{A}$ for $B \in \sigma(\mathcal{X})$.

Define a cylinder set of length n for measurable sets $B_1, B_2, \dots, B_n \in \sigma(\mathcal{X})$ as

$$C = \{(x_1, x_2, \dots) \in \mathcal{X}^{\mathbb{N}_+} : x_1 \in B_1, x_2 \in B_2, \dots, x_n \in B_n\}.$$

According to the Ionescu-Tulcea theorem, there exists a unique probability measure $\mathbb{P}_{x_0, a_0}^{P_\pi}$ defined on the product σ -algebra $\sigma(\mathcal{X})^{\otimes \mathbb{N}_+}$ such that for every $n \in \mathbb{N}_+$ and every cylinder set C of the form $B_1 \times B_2 \times \dots \times B_n \times \mathcal{X} \times \dots$, with $B_t \in \sigma(\mathcal{X})$ for $t \geq 1$, the measure is consistent with:

$$\begin{aligned} \mathbb{P}_{x_0, a_0}^{P_\pi}(C) &= \int_{B_1} P(dx_1|x_0, a_0) \times \int_{B_2} P_\pi(dx_2|x_1) \\ &\quad \times \int_{B_3} P_\pi(dx_3|x_2) \times \dots \times \int_{B_n} P_\pi(dx_n|x_{n-1}). \end{aligned}$$

We refer to $\mathbb{P}_{x_0, a_0}^{P_\pi}$ as a **Markov process measure**.

Return distribution is a push-forward Markov process measure

Push-forward process functional: A sequence-valued map

$\mathbf{F}(\omega) = (f(x_1), f(x_2), \dots)$ where $\omega = (x_1, x_2, \dots)$ is an element of the path space $\mathcal{X}^{\mathbb{N}_+}$.

Push a Markov process measure forward with this functional:

$$\mathbf{F}_{\#} \mathbb{P}_{x_0, a_0}^{P_{\pi}} \triangleq \mathbb{P}_{x_0, a_0}^{P_{\pi}}(\mathbf{F}^{-1}(B)) = \mathbb{P}_{x_0, a_0}^{P_{\pi}}\{\omega \in \mathcal{X}^{\mathbb{N}_+} : \mathbf{F}(\omega) \in B\}$$

for any measurable set $B \in \sigma(E)^{\otimes \mathbb{N}_+}$.

The gist: The measure $(\mathbf{G}_{x_0, a_0}^{\pi})_{\#} \mathbb{P}_{x_0, a_0}^{P_{\pi}}$ corresponds precisely to the *return distribution*. For any Borel set $B \in \mathcal{B}(\mathbb{R})$, this measure can be expressed as:

$$\mathbb{P}_{x_0, a_0}^{P_{\pi}} \left(\left\{ \omega \in \mathcal{X}^{\mathbb{N}_+} : R(x_0, a_0) + \sum_{t=1}^{\infty} \gamma^t R(x_t, \pi(x_t)) \in B \right\} \right).$$

The return distributions induced by two different Markov kernels P and P' are then $(\mathbf{G}_{x_0, a_0}^{\pi})_{\#} \mathbb{P}_{x_0, a_0}^{P_{\pi}}$ and $(\mathbf{G}_{x_0, a_0}^{\pi})_{\#} \mathbb{P}_{x_0, a_0}^{P'_{\pi}}$.

The push-forward contraction lemma

Define the distributional counterpart of the generalized Bellman operator with a specific Markov kernel P as follows:

$$\mathbf{T}_P^\pi \eta_{x,a} \triangleq R(x,a) + \gamma \mathbb{E}_{x' \sim P(\cdot|x,a)} [\eta_{x',\pi(x')}].$$

Lemma

Let $P_\pi, \bar{P}_\pi, P_\pi^*$ be Markov kernels induced by a fixed policy π and $\mathbf{F}$ be a push-forward process functional, then the distributional Bellman operator $\mathbf{T}_{P_*}^\pi$ is a contraction with respect to $\bar{\mathcal{W}}_p$

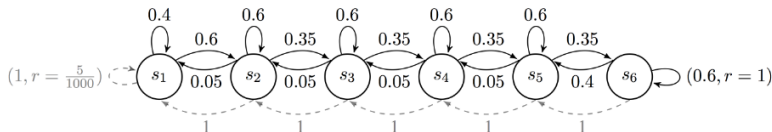
$$\bar{\mathcal{W}}_p \left(\mathbf{T}_{P_*}^\pi \mathbf{F}_\# \mathbb{P}^{P_\pi}, \mathbf{T}_{P_*}^\pi \mathbf{F}_\# \mathbb{P}^{\bar{P}_\pi} \right) \leq \gamma \bar{\mathcal{W}}_p \left(\mathbf{F}_\# \mathbb{P}^{P_\pi}, \mathbf{F}_\# \mathbb{P}^{\bar{P}_\pi} \right).$$

where $\bar{\mathcal{W}}_p(P, \bar{P}) \triangleq \sup_{x,a} \left(\inf_{\nu \in \Gamma(P, \bar{P})} \mathbb{E}_{(x,x') \sim \nu} [|x - x'|^p] \right)^{1/p}$.

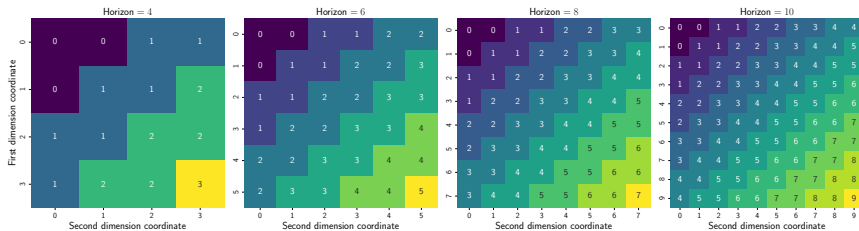
Key implications of the lemma

- There exists $\hat{P}_\pi \in \mathcal{P}_\mathcal{X}$ that satisfies $\bar{\mathcal{W}}_p(\mathbf{T}_{P_*}^\pi \mathbf{F}_\# \mathbb{P}^{\hat{P}_\pi}, \mathbf{F}_\# \mathbb{P}^{\hat{P}_\pi}) = 0$ for a sufficiently large set of admissible Markov kernels $\mathcal{P}_\mathcal{X}$.
- There may be multiple Markov process measures almost surely equal to the fixed point $\mathbf{F}_\# \mathbb{P}_{x_0, a_0}^{\hat{P}_\pi}$ after being pushed forward with $\mathbf{F}$.
- Distributional RL outperforms MBRL if capturing the push-forward of the Markov process measure with the return functional is sufficient for decision making. The equivalence of this measure within a set of transition kernels brings sample efficiency when it can be exploited by the learning algorithm.

Example: Latent RiverSwim



Vanilla RiverSwim (Osband et al., 2013)



The equivalence classes when RiverSwim governs the embedding space

The mechanics of state abstractions

Definition

A mapping $K : \mathcal{S} \rightarrow \mathcal{P}_{\mathcal{X}}$ is said to be an **L -Lipschitz Markov kernel** if (i) for every $B \in \sigma(\mathcal{X})$, $s \mapsto K(s, B)$ is measurable and (ii) for all $s_1, s_2 \in \mathcal{S}$: $\mathcal{W}_p(K(s_1, \cdot), K(s_2, \cdot)) \leq L \cdot |s_1 - s_2|$. The action of the kernel K on a measure $P \in \mathcal{P}_{\mathcal{S}}$, denoted KP , is the measure on $\mathcal{X}$ defined by:

$$(KP)(B) \triangleq \int_{\mathcal{S}} K(s, B) P(ds), \quad \forall B \in \sigma(\mathcal{X}).$$

Definition

Let $F : \mathcal{X} \rightarrow \mathcal{S}$ be a measurable transformation, and let $K : \mathcal{S} \rightarrow \mathcal{P}_{\mathcal{X}}$ be an L -Lipschitz Markov kernel. We define the **composite kernel operator** $(KF) : \mathcal{X} \rightarrow \mathcal{P}_{\mathcal{X}}$ pointwise as $K(x, \cdot) = K(F(x), \cdot)$. For any measure $P \in \mathcal{P}_{\mathcal{X}}$, the action of the operator KF is given by:

$$(KF P)(B) \triangleq \int_{\mathcal{X}} K(F(x), B) P(dx), \quad \forall B \in \sigma(\mathcal{X}).$$

... and why/when state abstractions expedite adaptation

Lemma

Let $F : \mathcal{X} \rightarrow \mathcal{S}$ be an M -Lipschitz transformation and $K : \mathcal{S} \rightarrow \mathcal{P}_{\mathcal{X}}$ be an L -Lipschitz Markov kernel. Then the composite kernel operator $(KF) : \mathcal{X} \rightarrow \mathcal{P}_{\mathcal{X}}$ is an $(L \cdot M)$ -Lipschitz Markov kernel.

Theorem

Let $P_{\pi}, \bar{P}_{\pi}, P_{\pi}^* \in \mathcal{P}_{\mathcal{X}}$ and $\mathbf{F}$ be a push-forward process functional. Let P_D be an L_D -Lipschitz Markov kernel and $S : \mathcal{X} \rightarrow \mathcal{S}$ be a L_E -Lipschitz continuous measurable map from $(\mathcal{X}, \sigma(\mathcal{X}))$ to $(\mathcal{S}, \sigma(\mathcal{S}))$. Then for any $x_0, a_0 \in \mathcal{A}, \mathcal{X}$

$$\begin{aligned} \bar{\mathcal{W}}_p \left(\mathbf{T}_{P_*}^{\pi} \mathbf{F}_{\#} \mathbb{P}_{x_0, a_0}^{(P_D S) P_{\pi}}, \mathbf{T}_{P_*}^{\pi} \mathbf{F}_{\#} \mathbb{P}_{x_0, a_0}^{(P_D S) \bar{P}_{\pi}} \right) \\ \leq \gamma \cdot L_E \cdot L_D \cdot \bar{\mathcal{W}}_p \left(\mathbf{F}_{\#} \mathbb{P}_{x_0, a_0}^{P_{\pi}}, \mathbf{F}_{\#} \mathbb{P}_{x_0, a_0}^{\bar{P}_{\pi}} \right). \end{aligned}$$

Key message: Efficient compression in the embedding space $\Rightarrow$ Small $L_E \Rightarrow$ Reduced contraction modulus $\Rightarrow$ Faster convergence

Equipping distributional RL with an embedding space

Goal: Redefine $\mathbf{G}_{\#} \mathbb{P}_{x_0, a_0}^{(P_D S) P_{\pi}}$ in terms of an *encoding process* defined on the latent space and a *decoding process* that maps back to the observation space.

Encoding process: Push $P_{x_0, a_0}^{P_{\pi}}$ forward through the sequence map $\mathbf{S}(\omega) = (S(x_1), S(x_2), \dots)$, which leads to a new measure $\mathbf{S}_{\#} \mathbb{P}_{x_0, a_0}^{P_{\pi}}$.

Decoding process: The kernel P_D induces a conditional path measure $\mathbb{P}^{P_D | \mathbf{s}}$.

Return distribution on the embedding space:

$$(\mathbf{G}_{x_0, a_0}^{\pi})_{\#} \mathbb{P}^{P_D | \mathbf{s}}(B) \triangleq \mathbb{P}^{P_D | \mathbf{s}} \left(\left\{ \omega \in \mathcal{X}^{\mathbb{N}_+} : R(x_0, a_0) + \sum_{t=1}^{\infty} \gamma^t R(x_t, \pi(x_t)) \in B \right\} \right).$$

Complete return distribution after the auto-encoding process:

$$(\mathbf{G}_{x_0, a_0}^{\pi})_{\#} \mathbb{P}_{x_0, a_0}^{(P_D S) P_{\pi}}(B) = \int_{\mathcal{S}^{\mathbb{N}_+}} (\mathbf{G}_{x_0, a_0}^{\pi})_{\#} \mathbb{P}^{P_D | \mathbf{s}}(B) \mathbf{S}_{\#} \mathbb{P}_{x_0, a_0}^{P_{\pi}}(d\mathbf{s}).$$

This is a *composite measure* and we will exploit this fact!

The implied policy evaluation objective

Apply the Wasserstein Auto-Encoder trick to get a trainable upper bound:

$$\begin{aligned} J(S) &\triangleq \mathbb{E}_{x \sim \mathcal{U}(\mathcal{X})} [\mathcal{W}_2^2((\mathbf{G}_{x_0, a_0}^\pi)_{\#} \mathbb{P}_{x_0, a_0}^{(P_D S) P_\pi}, \mathbf{T}_{P_*}^\pi(\mathbf{G}_{x', \pi(x')}^\pi)_{\#} \mathbb{P}_{x_0, a_0}^{(P_D S) P_\pi})] \\ &\leq \mathbb{E}_{x \sim \mathcal{U}(\mathcal{X})} \left[\mathbb{E}_{\mathbf{s} \sim \mathbf{S}_{\#} \mathbb{P}_{x_0, a_0}^{P_\pi}} \left[\mathbb{E}_{\tau \in \mathcal{U}(0,1)} \left[\left(\left(F_{(\mathbf{G}_{x_0, a_0}^\pi)_{\#} \mathbb{P}^{P_D | \mathbf{s}}}^{-1}(\tau) - F_{\mathbf{T}_{P_*}^\pi(\mathbf{G}_{x', \pi(x')}^\pi)_{\#} \mathbb{P}^{P_D | \mathbf{s}}}^{-1}(\tau) \right)^2 \right) \right] \right] \right]. \end{aligned}$$

The upper bound is indexed only by $\mathbf{S}_{\#} \mathbb{P}_{x_0, a_0}^{P_\pi}$. Then search directly the space of encoding measures

$$\mathcal{Z}_{x,a}^\pi \triangleq \{Z_{x,a} : Z_{x,a} = \mathbf{S}_{\#} \mathbb{P}_{x_0, a_0}^{P_\pi}, P_\pi \in \mathcal{P}_\pi\}.$$

Push-forward Policy Iteration (PPI)

Input: $\pi_0, i := 0$

while True **do**

for $x \in \mathcal{X}$ **do**

$$Z_{x,a}^{i+1} := \arg \min_{Z \in \mathcal{Z}_{x,a}^{\pi_i}} \mathbb{E}_{\tau \sim \mathcal{U}(0,1)} \mathbb{E}_{\mathbf{s} \sim Z_{x,a}} \left[\right.$$

$$\left. \left(F_{(\mathbf{G}_{x,a}^{\pi_i})_{\#} \mathbb{P}^{P_D | \mathbf{s}}}^{-1}(\tau) - F_{\mathbf{T}_{P_*}^{\pi_i}(\mathbf{G}_{x',\pi_i(x')}^{\pi_i})_{\#} \mathbb{P}^{P_D | \mathbf{s}}}^{-1}(\tau) \right)^2 \right]$$

$$\pi_{i+1}(x) := \arg \max_{a \in \mathcal{A}} \mathbb{E}_{\mathbf{s} \sim Z_{x,a}^{i+1}} \left[(\mathbf{G}_{x,a}^{\pi_i})_{\#} \mathbb{P}^{P_D | \mathbf{s}}(\mathbf{s}) \right]$$

$$i := i + 1$$

end for

end while

Active inference as a PPI instance

Exploit the fact that in the AIF ELBO, P_M and P_E always appear together in a marginalization process

$$\begin{aligned} L(x, a, x', P_W, P_Q) = & \mathbb{E}_{S' \sim P_M P_E | x, a} [\log P_D(x' | S')] \\ & + \mathbb{E}_{x'', \pi \sim P_D P_M P_E P_A | x'} [\log P_R(x'' | x', \pi(x'))] + \mathbb{H}[P_A | x']. \end{aligned}$$

Denote their marginal by

$$P_{S|x,a}(S') \triangleq \mathbb{E}_{s \sim P_E | x} [P_M(S' | s, a)].$$

Define the next state S' as the infinitely-long trajectory of future latent states s simulated on the world model after taking action a at state x and following policy π afterwards. Let ω be the corresponding states in the observation space. Then construct the desired state distribution as the return of this trajectory:

$$\mathbb{E}_{\omega \sim P_D P_M P_E | x, a} [(\mathbf{G}_{x,a}^\pi)_\#(\omega)] = \mathbb{E}_{\omega \sim P_D P_{S|x,a}} [(\mathbf{G}_{x',\pi(x')}^\pi)_\#(\omega)].$$

Whenever the decomposition $P_{S|x,a} = S_\# P(X' | x, a)$ holds, **AIF becomes a push-forward policy iteration instance!**

Quantile regression on the embedding space

Approach: Re-cast the quantile regression problem as MLE and model the parameter as a state-action amortized random variable.

Standard quantile regression:

$$\hat{g}_\tau \triangleq \arg \min_{g_\tau} \mathbb{E}_{X,Y \sim P} [\ell_\tau(Y - g_\tau(X))]$$

for a predictor g_τ where $\ell_\tau(u) \triangleq (|u| + (2\tau - 1)u)/2$. Its solution is equivalent to the MLE of the mean of the Asymmetric Laplace Distribution (ALD)

$$f_\tau(Y|X; g_\tau, \sigma_\tau) = \frac{\tau(1 - \tau)}{\sigma_\tau} \exp \left(-\frac{\ell_\tau(Y - g_\tau(X))}{\sigma_\tau} \right).$$

Set $\mu_\tau \triangleq g_\tau$ and treat both μ_τ and σ_τ as random variables that follow a prior distribution with state and action dependent hyperparameters.

End output: The DAIF algorithm

Input: P_A

$x = \text{env.reset}()$

while True **do**

$\pi \sim P_A(\cdot|x), a := \pi(x)$

$x' := \text{env.step}(a)$

$D := D \cup (x, a, x')$

repeat

$(x, a, x') \sim D, \tau \sim \mathcal{U}(0, 1), \pi \sim P_A(\cdot|x)$

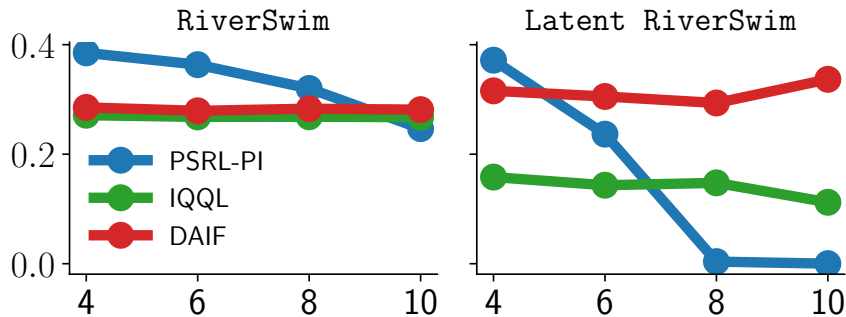
$\phi := \arg \max_{\phi'} \left\{ \mathbb{E}_{\mu_{\tau}, \sigma_{\tau}^2 \sim E_{\phi'}(x, a, \tau)} \left[\log f_{\tau} \left(R(x, a) + \gamma \mathbb{E}_{\mu'_{\tau} \sim E_{\phi'}(x', \pi(x'), \tau)} [\mu'_{\tau}] | \mu_{\tau}, \sigma_{\tau}^2 \right) \right] \right\}$

$P_A := \arg \max_{P'_A} \left\{ \mathbb{E}_{\pi \sim P'_A(\cdot|x)} \left[\mathbb{E}_{\mu_{\tau} \sim E_{\phi'}(x, \pi(x), \tau)} [\mu_{\tau}] \right] \right\} + \mathbb{H}[P'_A] \}$

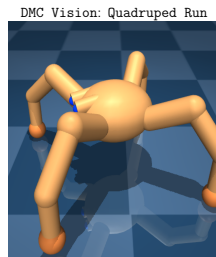
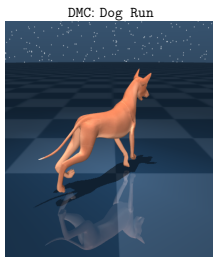
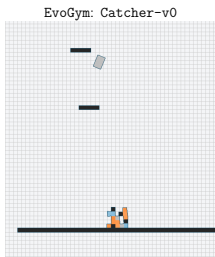
until end of training epoch

end while

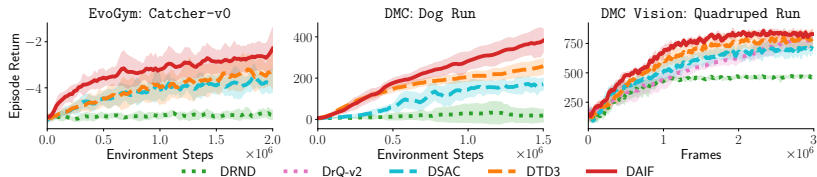
Proof of principle: DAIF works when it should



SotA race standing: DAIF cracks many hard nuts



Continuous control environments



Results

What's next?

- **Finite-sample analysis:** How many environment interactions does DAIF need to find the optimal policy?
- **Multi-objective distributional RL:** Can we learn the empirical return distribution of a multi-dimensional reward signal using few particles?
- **Measure-theoretic RL:** Can we reconstruct the whole RL theory on measures and push-forwards?
- **Foundation policies on Markov process measures:** Can we learn the essential properties of environment dynamics on big interaction data so that we can read out the optimal policy of any reward push-forward?

The ADIN Lab says thank you!



Melih Kandemir
Lab Head



Mert Çelikok
Assistant Professor



Sai Nagarajan
Assistant Professor



Abdullah Akgül
Postdoc



Manuel Haussmann
Postdoc



Gülcin Baykal Can
Postdoc



Magnus Boock
PhD Fellow



Robin Faier
PhD Fellow



ADIN Lab



ObjectRL