Forecasting in Offline Reinforcement Learning for Non-stationary Environments

Suzan Ece Ada
Boğaziçi University
University of Tübingen

Georg Martius
University of Tübingen

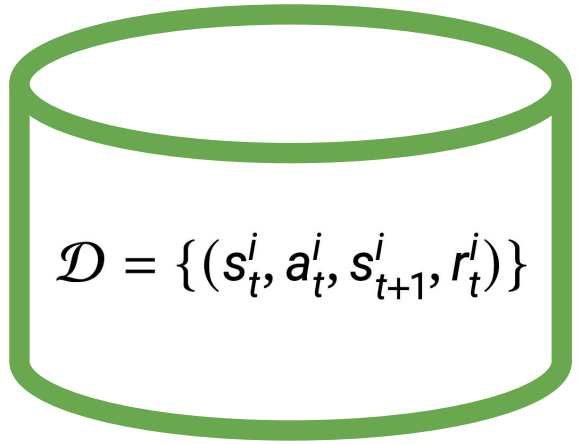
Emre Uğur
Boğaziçi University

Erhan Öztop
Osaka University
Özyeğin University

Offline Reinforcement Learning (RL)

- Learning a policy from previously collected datasets without real-world interaction.
- The dataset $\mathcal{D} = \{(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}'_i, r_i)\}$ consists of MDP tuples generated by a behavior policy $\pi_\beta(\boldsymbol{a}|\boldsymbol{s})$.
- Avoids expensive and risky data collection.

Aim: Improve upon the behavior policy used for data collection.



🚫 **Assumes stationarity or full observability during training and at test time.**

Non-stationary Environments

Each component of the underlying MDP or POMDP can evolve over time.

[Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives., 2022]

$$\left(\mathcal{S}(t), \mathcal{A}(t), \mathcal{T}(t), \mathcal{R}(t), \mathbf{x}(t), \mathcal{O}(t)\right)$$

Standard Offline RL Fails with Non-stationary Perturbations

🚫 Standard offline RL assume stationarity or full observability during training and at test time.

Robust offline RL addresses settings with **perturbed states at test-time** (e.g., **Gaussian or adversarial noise**).
The real world is non-stationary.

How can we handle non-stationary perturbations?

We need test-time adaptation **without re-training** and presupposing any specific pattern of **future non-stationarity during training**.

Image generated by Gemini

S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Forecasting in Offline Reinforcement Learning for Non-stationary Environments

Forecasting in Non-stationary Offline RL (FORL) tackles the foundational challenge of additive episodic biases in the observation function.

✨ AI Assistants (LLMs)	💊Healthcare & Finance	🦾Industrial Robots
Semantic or contextual bias	Data may be withheld (privacy/regulations).	Sensors drift. Robots require daily calibration offsets.

Standard **noise-driven** or **parametric** state-estimation techniques, which typically rely on **smoothly varying or randomly perturbed functions**, cannot reliably identify **persistent, episode-wide offsets** that are **not available after episode terminates**.

S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

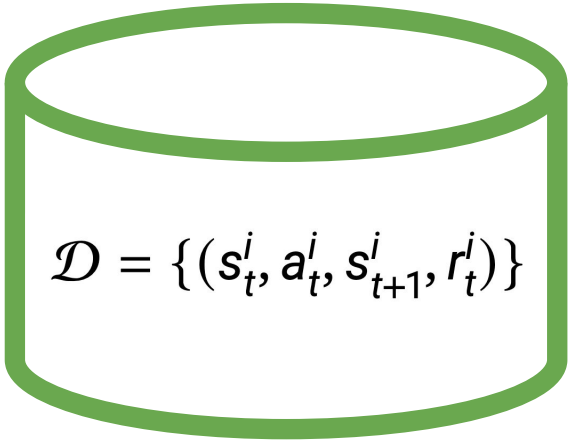
Problem Statement

Training

Offline dataset $\mathcal{D} = \{(s_t^i, a_t^i, s_{t+1}^i, r_t^i)\}$ from a stationary environment.

No knowledge of how offsets evolve in the future during training.

Train an offline RL policy π_ϕ and a FORL *diffusion model* (**FORL-DM**) using $\mathcal{D}$.

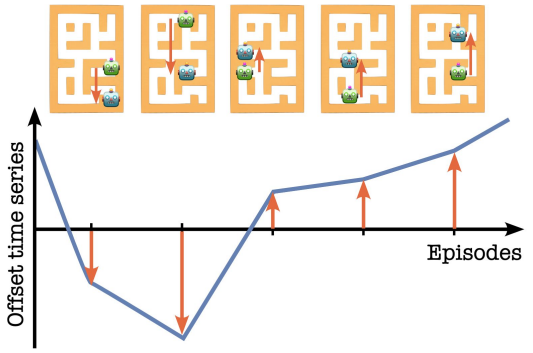


Test Environment

At test time, the agent faces an infinite sequence of POMDPs $\{\hat{\mathcal{M}}_j\}_{j=1}^{\infty}$.
Each POMDP $\mathcal{M}_j$ is described by a 7-tuple [Kaelbling et al., 1998]:

$$\hat{\mathcal{M}}_j = (\mathcal{S}, \mathcal{A}, \mathcal{O}_j, \mathcal{T}, \mathcal{R}, \rho_0, \mathbf{x}_j),$$

where $\mathcal{S}$, $\mathcal{A}$, transition function $\mathcal{T}$ and the reward function $\mathcal{R}$ remain identical to the training MDP.
x is the **deterministic** observation function ($\mathbf{x} : \mathcal{S} \to \mathcal{O}$)
[Bonet, 2012, Khetarpal et al., 2022] that evolves.



B. Bonet, **Deterministic POMDPs revisited,**
Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra, **Planning and acting in partially observable stochastic domains**
Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup, **Towards continual reinforcement learning: A review and perspectives.**

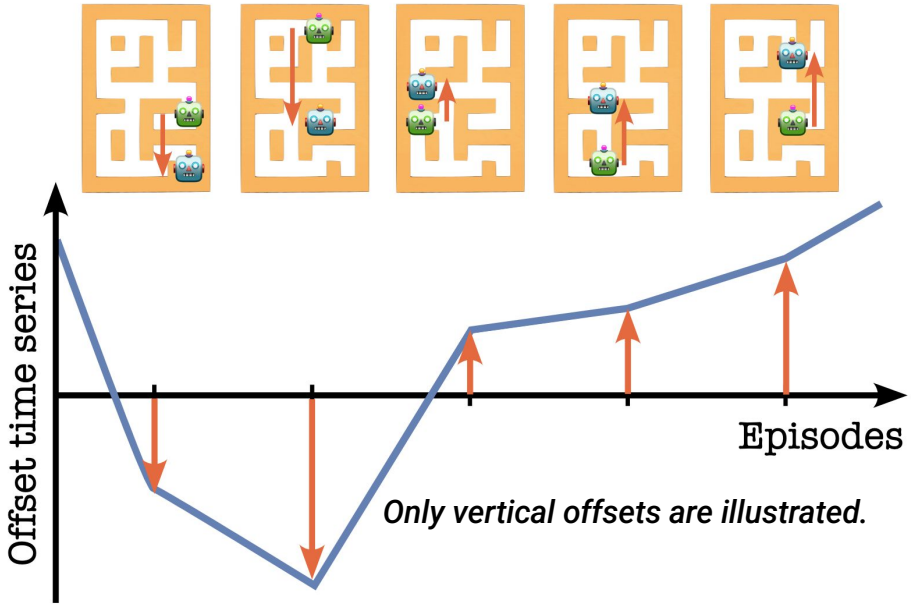
S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Test Environment: Time-varying Unknown Process

The non-stationarity is driven by a time-varying, unknown process:

- Each **episode** j is characterized by an **unknown** offset b^j added to the true state: $o_t = s_t + b^j$.
- The initial state distribution ρ_0 is uniform, providing no information about the offset.
- The offsets originate from **real-world time-series** $(\ldots, b^{j-1}, b^j, b^{j+1}, \ldots)$
- The agent only sees the sequence of biased observations $\{o_t\}$.

The agent must act effectively from the onset of each episode, despite being "lost."



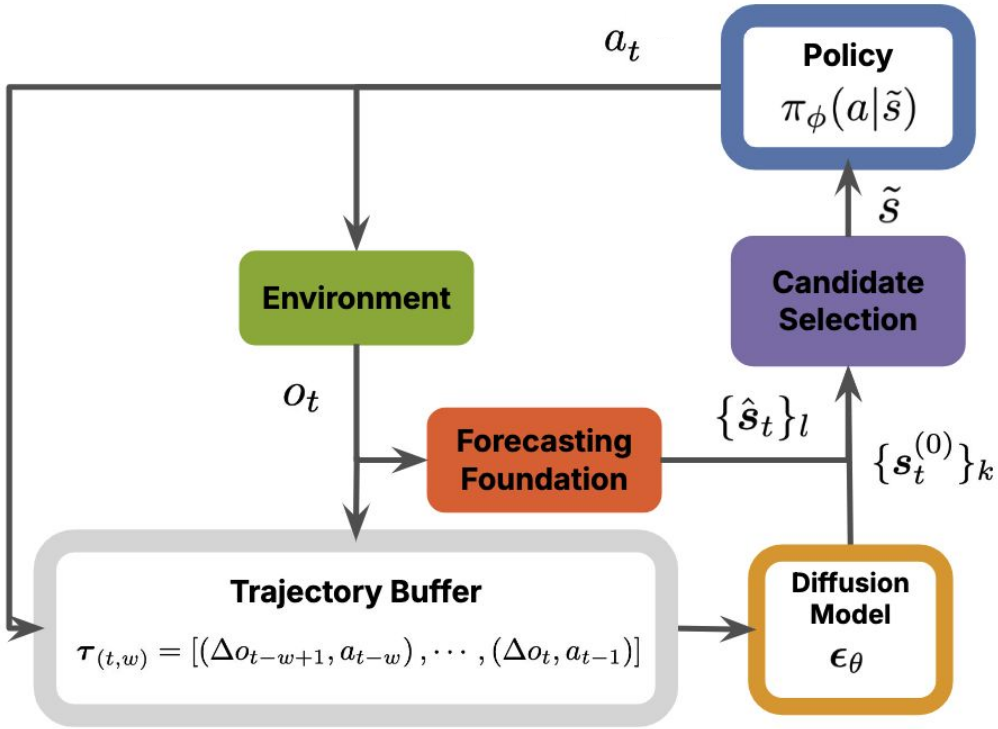
(↓,↑) **Ground truth offsets are hidden throughout the evaluation episodes.**

S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Overview of FORL Framework at Test-time

Forecast offsets over next *P* episodes (zero-shot).

- The observations are processed by both the trajectory buffer and the time-series **forecasting foundation** module.
- Observation changes and actions sampled from the **policy**, $(\Delta o_t, a_t)$ are stored in the trajectory buffer.
- The **diffusion model** generates candidate states $\{\boldsymbol{s}_t^{(0)}\}_k$ conditioned on $\boldsymbol{\tau}_{(t,w)}$.
- The **candidate selection** module then generates the estimated $\tilde{s}_t$.

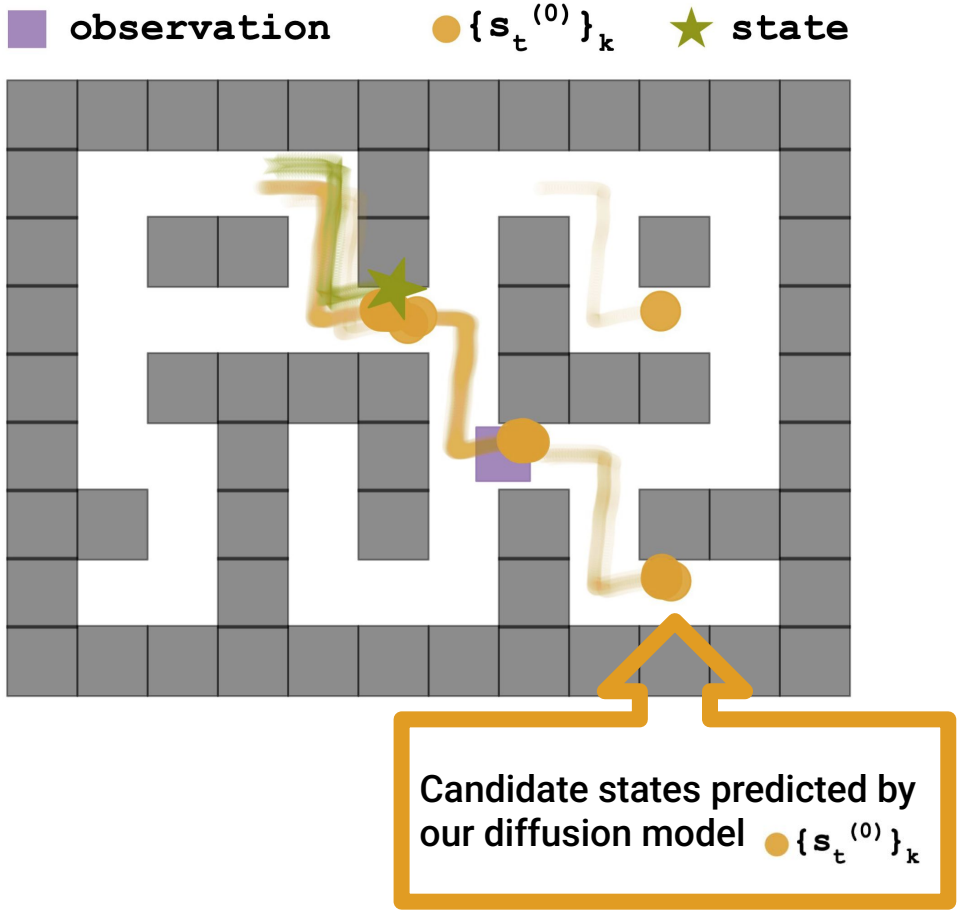


S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Candidate State Generation: Diffusion Model

How can an agent infer its location when the state is subject to a large unknown offset?

- An agent observes a sequence of its own actions and the resulting changes in its (biased) observations.
- This action-effect history, $(\Delta o, a)$, is **invariant** to the offset b^j.



S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Candidate State Generation: Diffusion Model

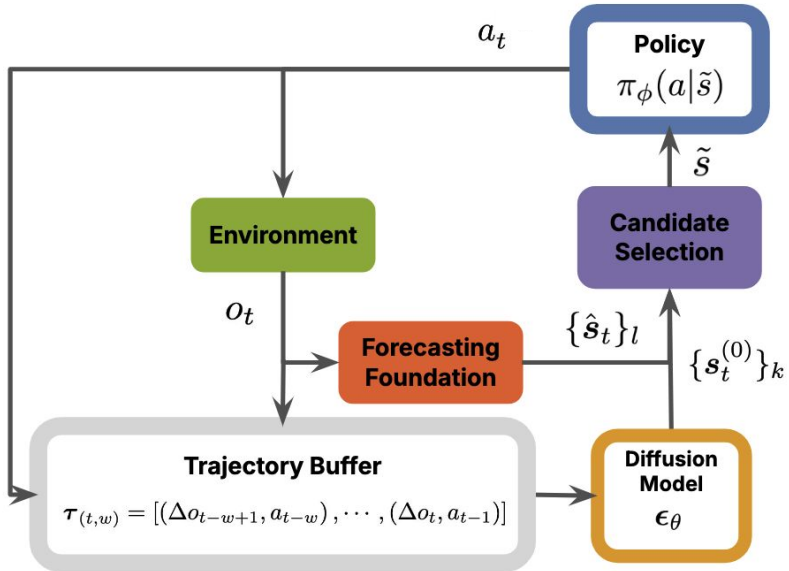
We model the **multi-modal distribution** of plausible states using a conditional diffusion model.

1. Define the **agent's recent history** as a trajectory $\tau_{(t,w)}$ of the past w action-effect Δo_t pairs:

$$\tau_{(t,w)} = [(\Delta o_{t-w+1}, a_{t-w}), \ldots, (\Delta o_t, a_{t-1})]$$

2. Train the conditional diffusion model FORL-DM to generate plausible states s_t conditioned on this history using the **stationary offline RL dataset**.

Goal: Learn $p(\boldsymbol{s}_t \mid \tau_{(t,w)})$



At test time, this model acts as a powerful inference engine, generating a set of k plausible state candidates $\{\boldsymbol{s}_t^{(0)}\}_k$ from the agent's real-time, biased observations and actions.

S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Forecasting Foundation Model

We use a probabilistic, zero-shot **foundation model** (Lag-Llama [Rasul et al., 2023]) to forecast future offsets based on past history.

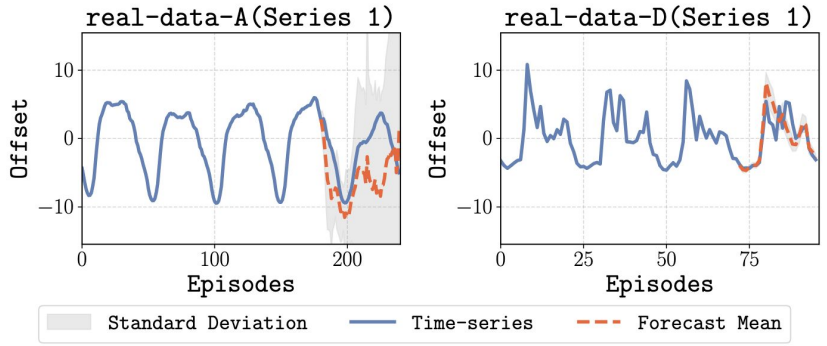
- **Input:** A context window of C past, known offsets:

$$\{b^{j-C}, \ldots, b^{j-1}\}$$

- **Output:** A probabilistic forecast of I samples for the next P episodes:

$$\{\hat{b}_I^j, \ldots, \hat{b}_I^{j+P-1}\}$$

The model is **univariate**, and each dimension of the offset vector b^j is forecast independently.



Rasul, Kashif, et al. **"Lag-llama: Towards foundation models for time series forecasting."**

Candidate Selection: Dimension-wise Closest Match (DCM)

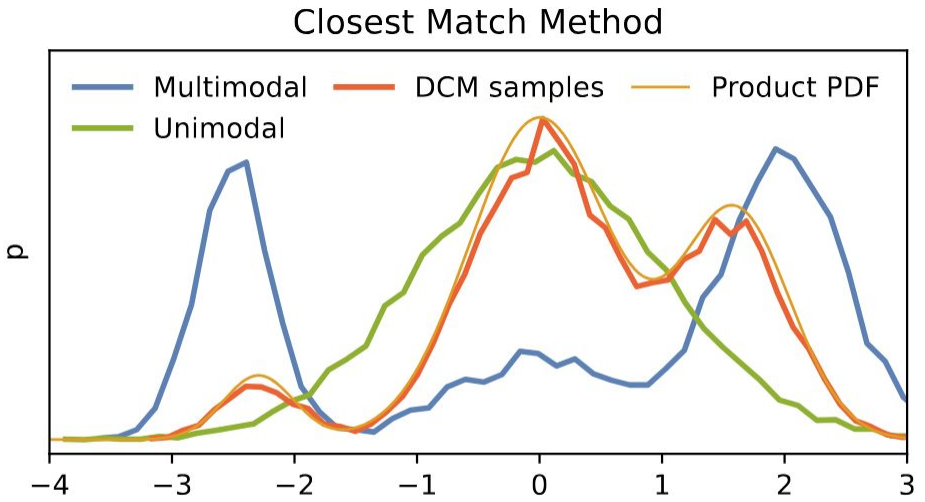
DCM is a simple and light-weight approach to adaptively fuse samples from 2 sources.

Let $\mathcal{D}_{\text{Diffusion}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, $\mathcal{D}_{\text{Forecaster}} = \{\mathbf{y}_1, \ldots, \mathbf{y}_l\}$, where $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^n$. Then DCM constructs $\mathbf{z} \in \mathbb{R}^n$ by

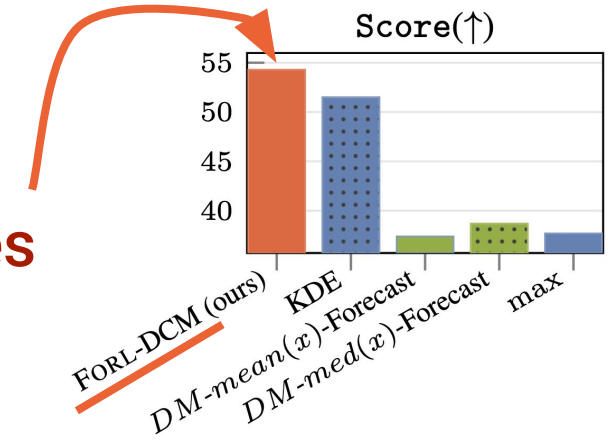
$$z_d = y_{j^*(d),d} \quad \text{where} \quad j^*(d) = \arg\min_j \left(\min_i \left| x_{i,d} - y_{j,d} \right| \right),$$

where $d = 1 \ldots n$ are the dimensions.

For each dimension d, we choose the sample from $\mathcal{D}_{\text{Forecaster}}$ that has a closest sample in $\mathcal{D}_{\text{Diffusion}}$.



Distribution of samples produced by DCM Histograms for 10k samples



Highest performance compared to standard approaches

No hyperparameters & No fallback mechanism

S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

Experimental Setup and Baselines

FORL is plug and play for different offline RL algorithms.

FORL-π vs. baselines:

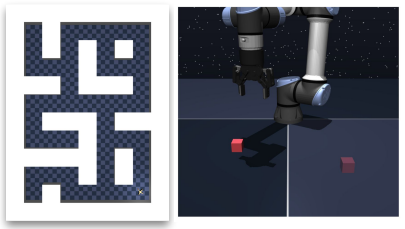
$\pi \in \{$ DQL [Wang et al., 2023], TD3BC [Fujimoto,2021], RORL [Yang,2022], IQL [Kostrikov et al., 2021], FQL [Park et al., 2025b] $\}$

- π
- π-Lag[Rasul et al.2023]
- $\pi^{(*)}$-DMBP[Yang et al.,2024]-Lag[Rasul et al.2023]

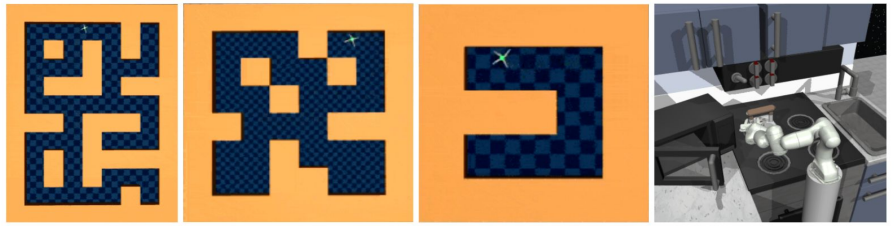
(*) $\pi \in$ {DQL, TD3BC, RORL, IQL}

Offline RL Datasets

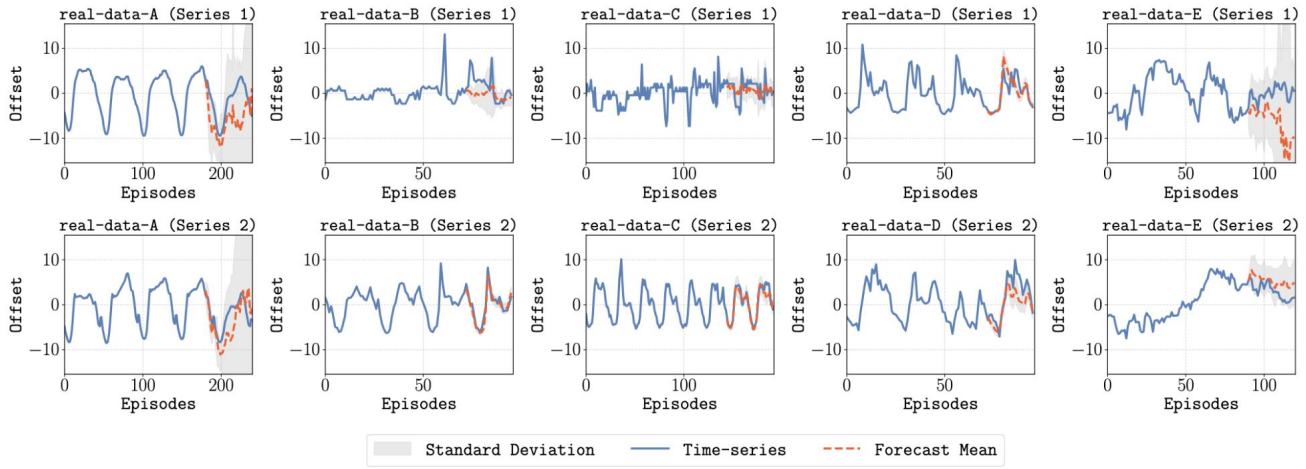
OGBench [Park et al., 2025a]



D4RL [Fu et al., 2020]



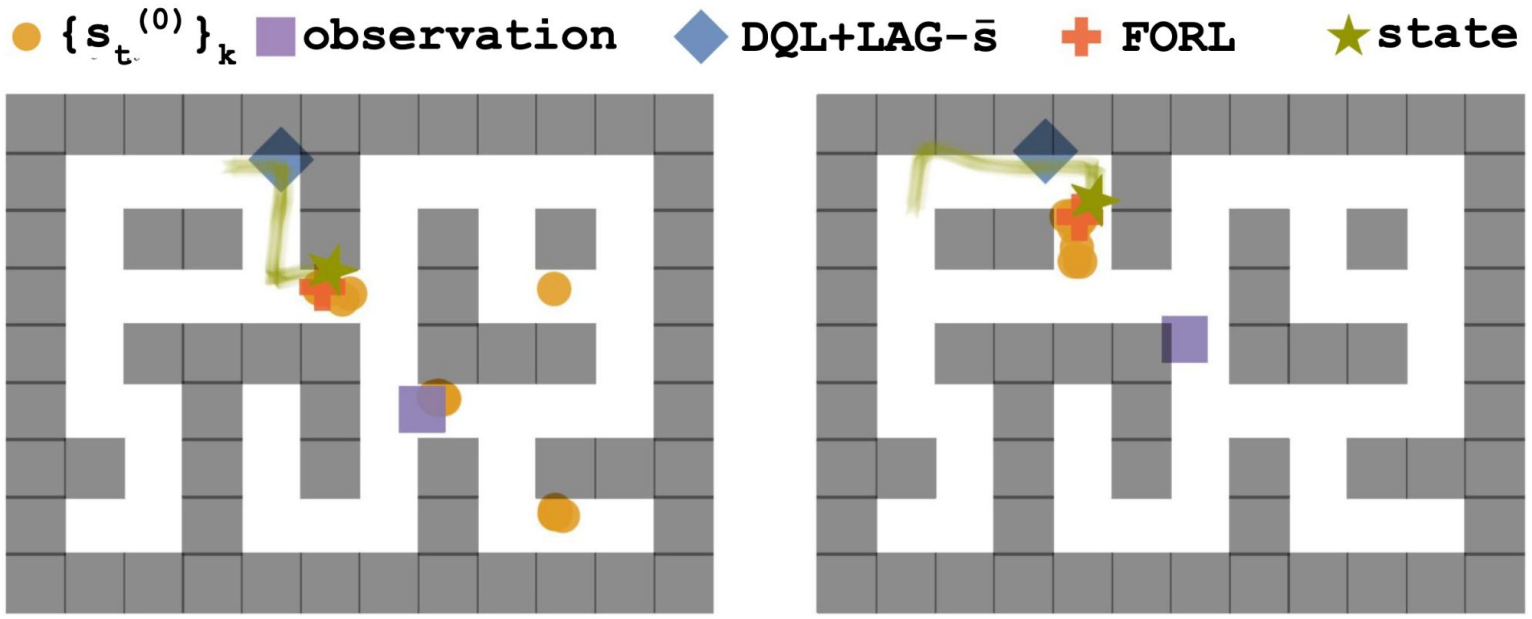
Time-series offsets from energy and finance data GluonTS [Alexandrov et al.,2020]



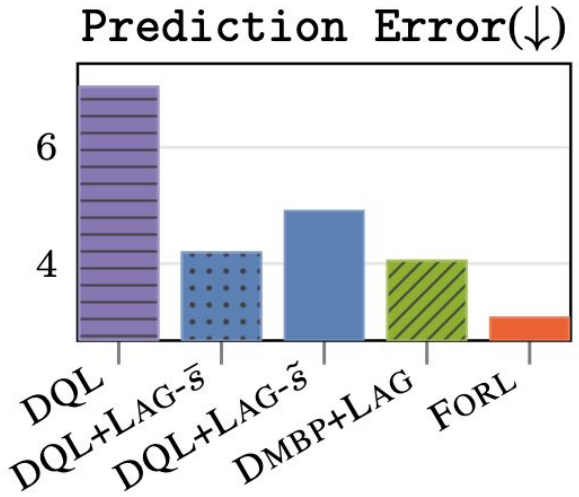
S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

State Prediction Error

Visualization of predicted states and true state ★
FORL is the closest to the true state

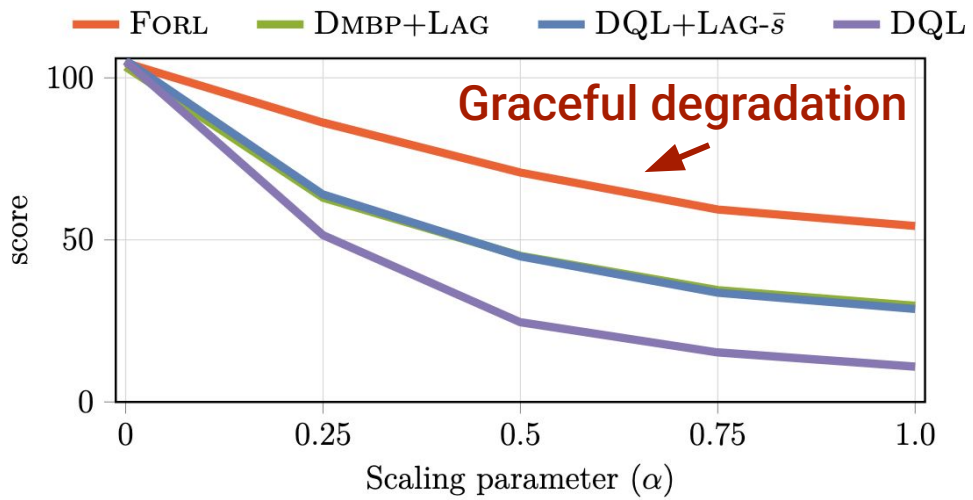


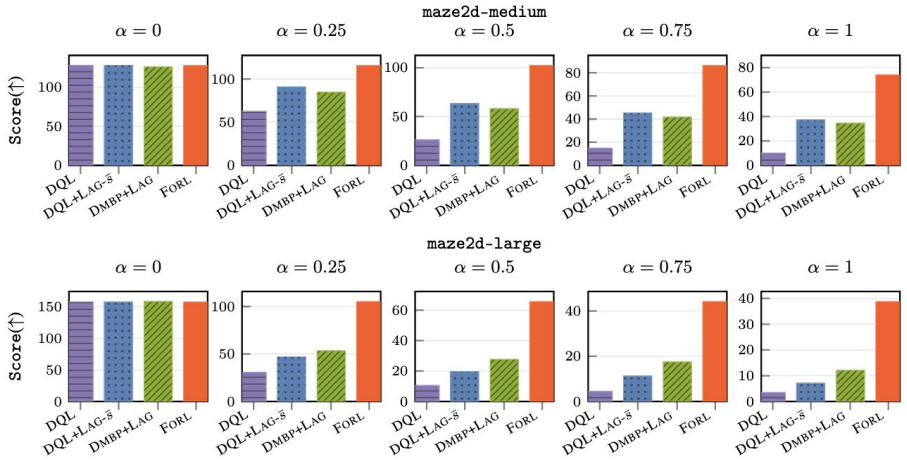
FORL has the lowest prediction error

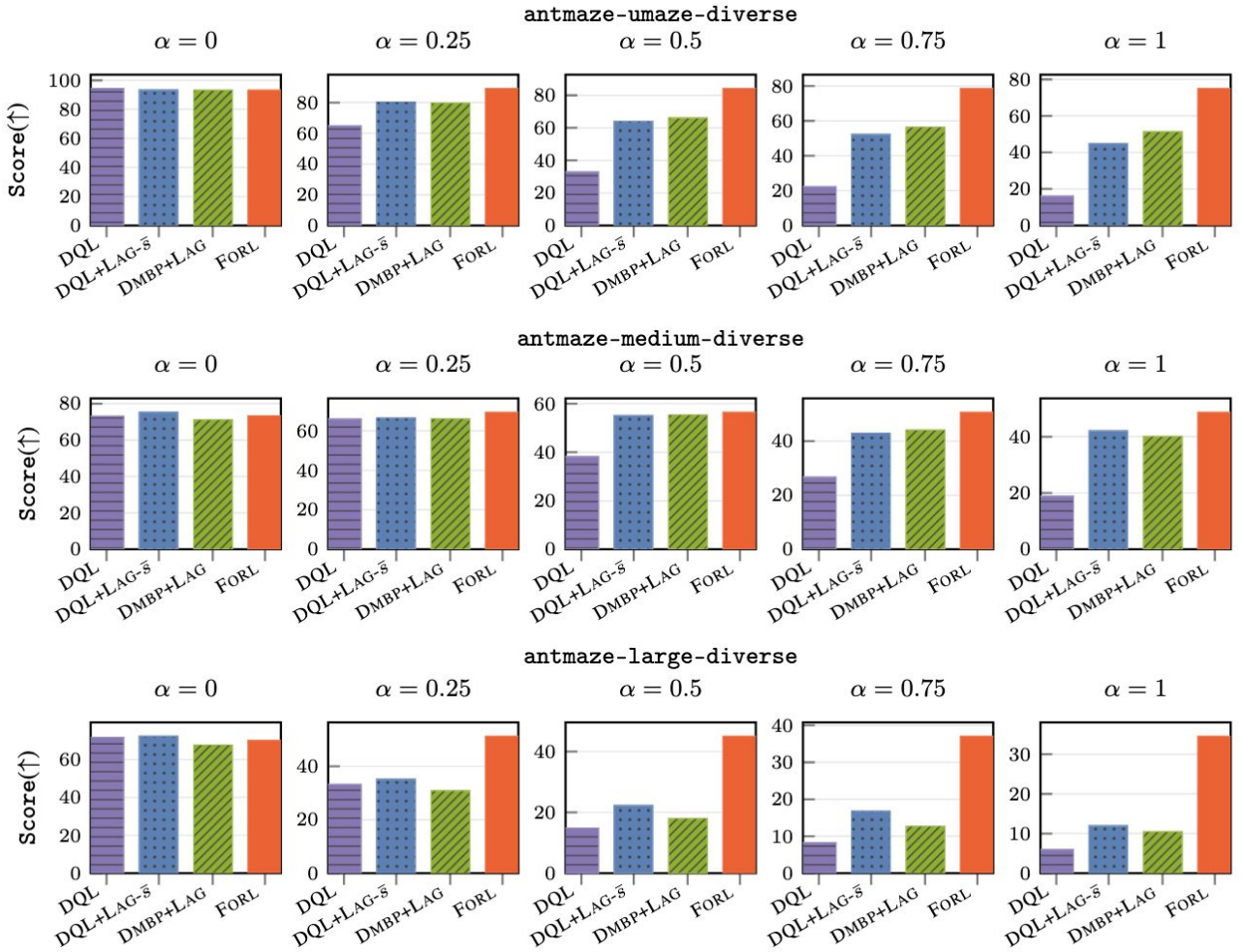


S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

How gracefully does performance degrade as the offset magnitude α is scaled from 0 (no offset) → 1?

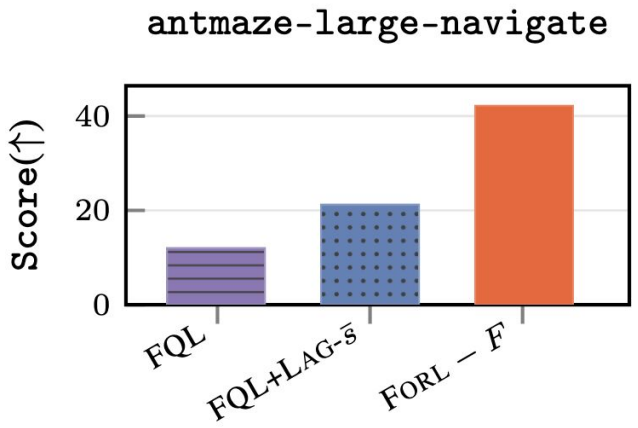


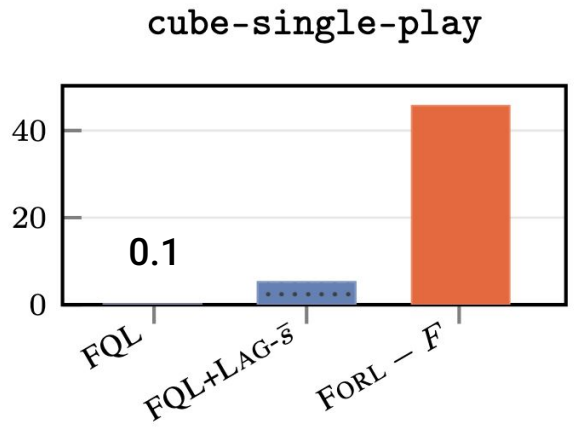


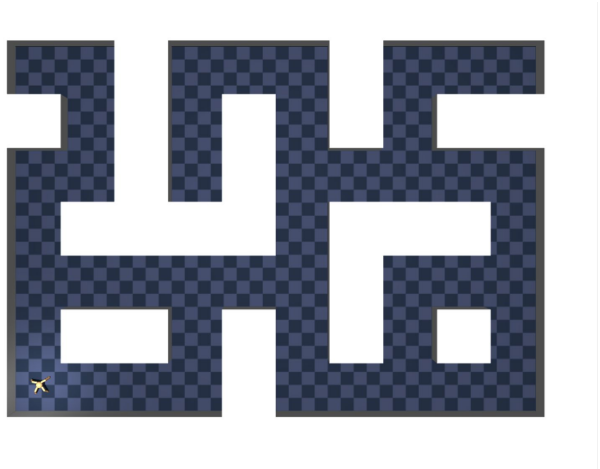


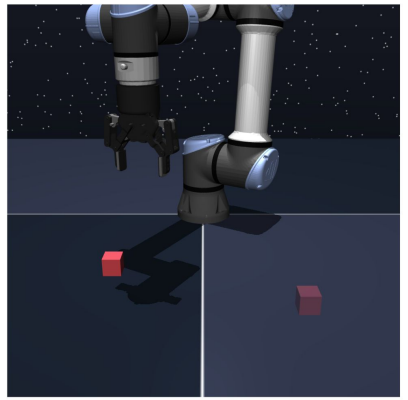
S. E. Ada, G. Martius, E. Ugur, E. Oztop, **"Forecasting in Offline Reinforcement Learning for Non-stationary Environments"**, ***NeurIPS2025 (Spotlight)***

OGBench: FORL Outperforms the Baselines









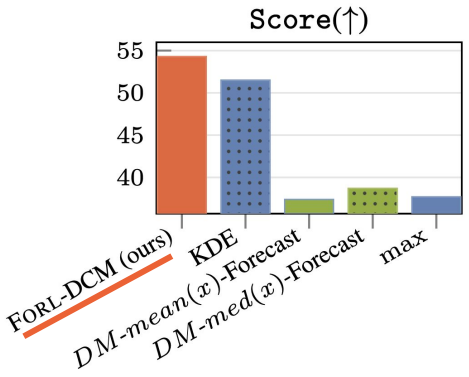
*Park, Seohong, Qiyang Li, and Sergey Levine. "**Flow q-learning.**" ICML 2025*

Park, Seohong and Frans, Kevin and Eysenbach, Benjamin and Levine, Sergey, **OGBench: Benchmarking Offline Goal-Conditioned RL,ICLR2025**

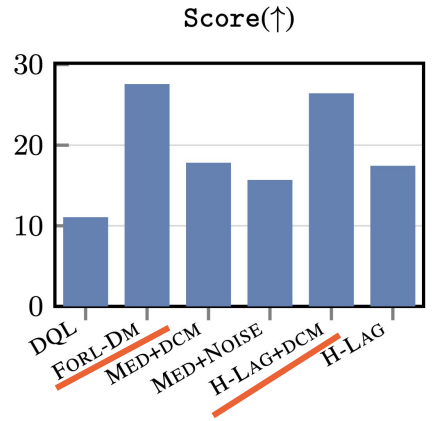
Thank you! We invite you to our project web-page!

🔗https://sites.google.com/view/forecastrl

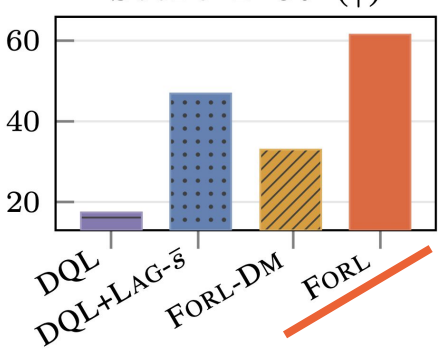
Dimension Wise Closest Match



What if we do not have access to the past offsets?



What if the offset changes are "in-episode" every f=50 timesteps?



Can FORL serve as a plug-and-play module for different offline-RL algorithms without retraining?

Suzan Ece Ada
Boğaziçi University
University of Tübingen

Georg Martius
University of Tübingen

Emre Uğur
Boğaziçi University

Erhan Öztop
Osaka University
Özyeğin University