Abdullah Akgül

Professional Portfolio

Abdullah Akgül — Postdoctoral Researcher at the University of Southern Denmark working on sample-efficient reinforcement learning and probabilistic machine learning. Publications at NeurIPS, ICML, ICLR, and TMLR.

View Live Portfolio

CONTENTS

Featured Work

Distributional Active Inference

ICML (2026) | *First Author*

Summary: Best average rank across 19 continuous control tasks on three benchmark suites, with up to +62% sample efficiency over the next-best baseline. Integrates Active Inference into distributional RL without a world model. ICML 2026.

Links: - Paper - arXiv - Video - Scholar - View on Site

Introduction

Effective autonomous control requires two complementary capabilities: organizing raw sensory observations into compact state representations, and planning action sequences that maximize long-term reward. Reinforcement learning (RL) excels at planning but treats exploration as a secondary concern. Active Inference provides both capabilities through free-energy minimization, but prior implementations require expensive learned world models. **Distributional Active Inference (DAIF)** bridges this gap, delivering principled uncertainty-driven exploration and full return-distribution estimation in a purely model-free framework.

Problem Statement

- **RL** handles the planning side but lacks principled uncertainty quantification for exploration; standard methods add entropy bonuses or random noise without formal grounding.
- **Active Inference** provides a unified account of perception and action, but existing RL applications require a transition dynamics model, adding overhead and instability.
- **Distributional RL** (QR-DQN, IQN) tracks return distributions but does not connect uncertainty over returns to exploration; the uncertainty is computed but not acted upon.
- **Gap:** No prior method jointly delivers (1) model-free operation, (2) distributional return estimation with epistemic uncertainty, and (3) exploration driven by Expected Free Energy minimization from a single coherent framework.

Methodology

DAIF formulates quantile regression as **Bayesian quantile regression** under a Normal-Inverse-Gamma (NIG) generative model. For each (state, action, quantile) triple, a neural network outputs NIG parameters (μ, α, β), parameterizing a distribution over the quantile location and its scale:

$$\sigma \sim \mathrm{InvGam}(\alpha, \beta), \qquad G \mid \mu, \sigma, \tau \sim \mathrm{ALD}(\mu, \sigma, \tau)$$

Marginalizing out (μ, σ) yields a **closed-form training objective** (no Monte Carlo sampling). The Inverse-Gamma scale captures epistemic uncertainty; minimizing the Expected Free Energy (EFE) reduces to a distributional Bellman update with an intrinsic uncertainty-driven exploration bonus, with no world model required.

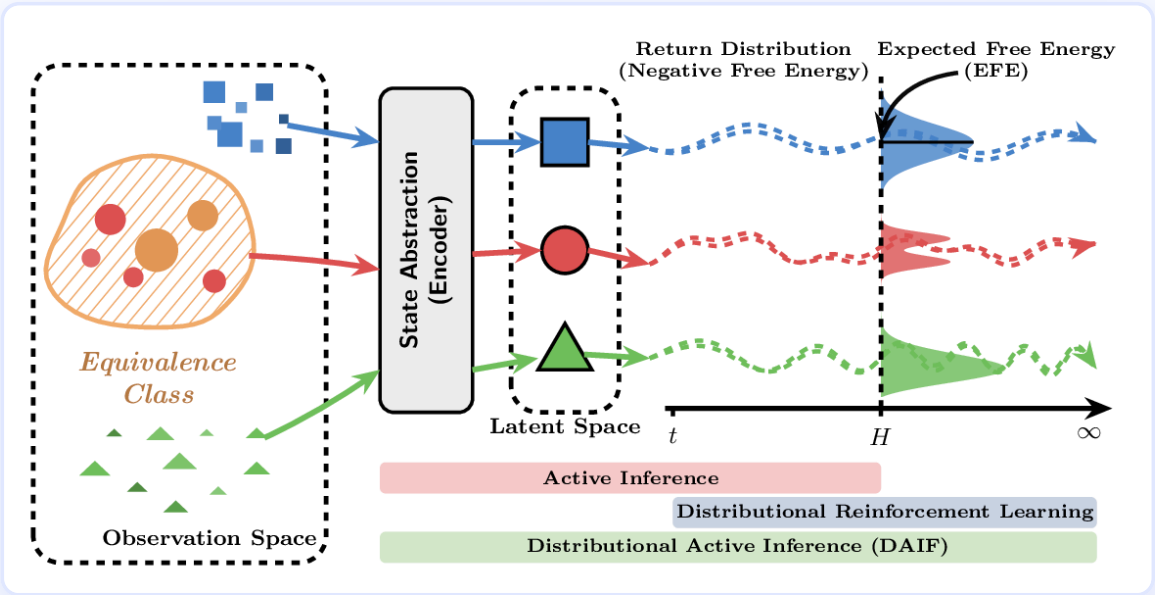


Figure 1. DAIF unifies Active Inference (state abstraction via encoder) and distributional RL (return distribution tracking). The encoder maps an equivalence class of observations to a latent state; an evidential network tracks the full return distribution as negative free energy; the Expected Free Energy (EFE) drives principled exploration without a transition dynamics model.

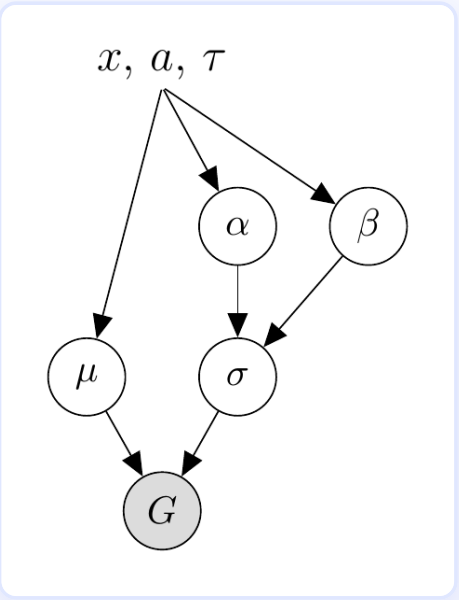


Figure 2. Generative model of DAIF. The input triple (s, a, τ) determines the NIG hyperparameters (α, β) and location mean μ. These parameterize an Inverse-Gamma prior over the scale σ, which together with μ governs the observed return G through an Asymmetric Laplace Distribution. Marginalizing out (μ, σ) yields a closed-form objective, with no world model required.

Key design choices: - Critic trained by minimizing the negative log marginal likelihood of the ALD (closed-form, no sampling noise) - TD3-style delayed policy updates and target smoothing for training stability (following DSAC [Ma et al., 2020]) - Two-critic architecture: quantile mean predictions averaged across both critics to reduce overestimation bias - Baselines: DRND, DSAC, DTD3 (distributional TD3), DrQ-v2 (pixel specialist)

Results

Evaluated across **19 continuous control tasks** from three benchmark suites (10 seeds for EvoGym/DMC, 5 seeds for DMC Vision):

- **EvoGym** — 7 morphologically diverse soft-robot locomotion and manipulation tasks [Bhatia et al., NeurIPS 2021]
- **DMC** — 7 physics-based tasks from the DeepMind Control Suite with low-dimensional state observations [Tunyasuvunakool et al., SoftwareX 2020]
- **DMC Vision** — 5 DMC tasks with raw pixel observations only

DAIF achieves the **best average ranking** on both sample efficiency (AULC) and final performance across all three suites:

Suite	Tasks	AULC rank	Final rank
EvoGym (soft robots)	7	**1.5 ± 0.7**	**1.6 ± 0.8**
DMC (state obs.)	7	**1.6 ± 0.7**	**1.5 ± 0.8**
DMC Vision (pixels)	5	**1.9 ± 1.2**	**2.0 ± 1.4**

Rank 1 = best. Lower is better.

Selected improvements over the next-best baseline (AULC metric):

Task	DAIF AULC	Next best	Improvement
EvoGym — Upstepper	**5.56 ± 0.77**	3.44 ± 0.92 (DTD3)	+62%
EvoGym — BidirectionalWalker	**7.21 ± 0.55**	4.68 ± 0.86 (DSAC)	+54%
DMC — Dog-Run	**214 ± 31**	162 ± 16 (DTD3)	+32%
DMC — Dog-Trot	**369 ± 79**	313 ± 29 (DTD3)	+18%
DMC Vision — Walker-Run	**660 ± 6**	588 ± 42 (DTD3)	+12%
DMC Vision — Quadruped-Run	**676 ± 18**	614 ± 44 (DTD3)	+10%

Figure 3. Representative benchmark environments. Left: EvoGym Catcher-v0 (soft robot catching falling objects). Center: DMC Dog-Run (high-DoF locomotion from state observations). Right: DMC Vision Quadruped-Run (locomotion from pixels only).

Conclusion

- **Unified framework:** DAIF provides the first measure-theoretic integration of model-free, distributional, and Active Inference RL.
- **No world model needed:** Active Inference's exploration benefits transfer to model-free settings via distributional Bellman updates with NIG uncertainty.
- **Consistent state-of-the-art:** Best average rank across all 19 tasks on three benchmark suites, with especially large gains on challenging locomotion (+62% on EvoGym Upstepper, +32% on DMC Dog-Run).
- **Pixel-ready:** Competitive ranking on DMC Vision without pixel-specialist tuning, confirming generality of the approach.
- **Practical:** Two-critic NIG architecture adds minimal overhead compared to standard distributional RL baselines.

References

1. **Akgül, A., Baykal, G., Haußmann, M., Çelikok, M. M., & Kandemir, M. (2026).** Distributional Active Inference. *ICML 2026*. arXiv:2601.20985
2. **Dabney, W., Rowland, M., Bellemare, M. G., & Munos, R. (2018).** Distributional Reinforcement Learning with Quantile Regression. *AAAI 2018*.
3. **Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018).** Implicit Quantile Networks for Distributional Reinforcement Learning. *ICML 2018*.
4. **Friston, K. J., et al. (2017).** Active Inference: A Process Theory. *Neural Computation, 29(1)*.

5. **Tunyasuvunakool, S., et al. (2020).** dm_control: Software package for physics-based simulation and reinforcement learning. *SoftwareX*.
6. **Bhatia, J., et al. (2021).** Evolution Gym: A Large-Scale Benchmark for Evolving Soft Robots. *NeurIPS 2021*.

MOMBO: Deterministic Uncertainty Propagation for Offline RL

NeurIPS (2024) | *First Author*

Summary: Best convergence rate (avg AULC rank 1.2) across all 12 D4RL offline benchmarks. Deterministic moment matching replaces Monte Carlo Bellman targets, with provably tighter suboptimality bounds. NeurIPS 2024.

Links: - Paper - arXiv - Video - Code - Scholar - View on Site

Introduction

Offline reinforcement learning (learning policies from pre-collected datasets without environment interaction) is essential for high-stakes domains where real-world exploration is costly or dangerous (healthcare, robotics, autonomous driving). The core obstacle is **distributional shift**: value estimates for actions underrepresented in the dataset become inflated, with no corrective feedback. **MOMBO** (Moment Matching Offline Model-Based Policy Optimization) identifies the root cause of training instability in model-based offline RL: high-variance Bellman targets from Monte Carlo sampling. MOMBO fixes this with deterministic moment matching, yielding provably faster convergence.

Problem Statement

- Model-based offline RL methods (MOPO, MOBILE) apply **Pessimistic Value Iteration (PEVI)**: penalize Q-value estimates by the learned dynamics model's uncertainty to keep the policy conservative about unseen state-action pairs.
- All existing PEVI methods sample a **single next state** (N=1) from the Gaussian dynamics model and evaluate the Q-network on it. A single sample is cheap but injects **high variance** into every Bellman target.
- This high variance corrupts gradient updates, slows convergence, and forces larger penalty coefficients to compensate, making model-based offline RL often slower than model-free approaches despite having access to synthetic data.
- **Theoretically:** suboptimality scales as $O(1/\sqrt{N})$ in the number of MC samples. At N=1, the bound is at its weakest; at N=1 it is also undefined in the limit, revealing a fundamental limitation.

- **Gap:** No existing method propagates next-state uncertainty analytically through the Q-network, despite this being the direct source of training instability.

Methodology

MOMBO replaces Monte Carlo sampling with **progressive moment matching**: the Gaussian next-state distribution output by the learned dynamics model is propagated through the Q-network layer by layer, analytically tracking the mean and variance of hidden activations.

Pessimistic Bellman target (exact, no sampling):

$$\widehat{\mathcal{B}}_{\text{pess}} = r + \gamma\mu_{\text{MM}} - \beta\gamma\sigma_{\text{MM}}$$

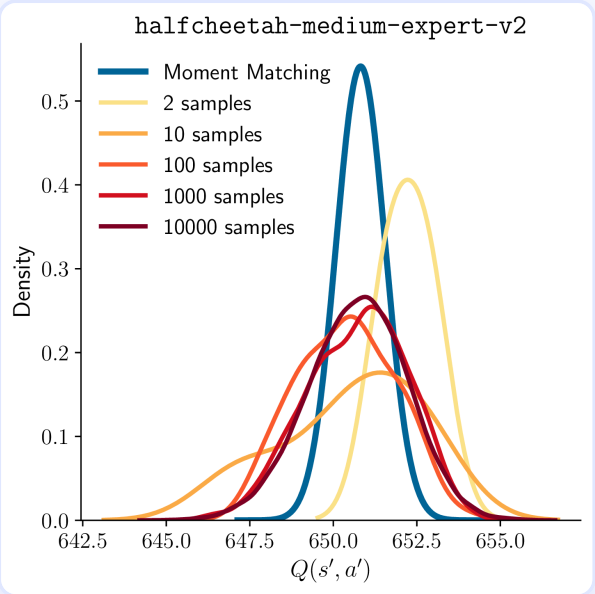


Figure 1. Moment matching versus Monte Carlo sampling on halfcheetah-medium-expert-v2. Moment matching (two forward passes) achieves sharp mean/variance estimates of the Q-value at the next state; even 10,000 MC samples fail to match this sharpness. Tighter Bellman targets reduce gradient noise and accelerate convergence throughout training.

Implementation details: - **Linear layers:** transform mean and variance analytically (exact Gaussian propagation) - **ReLU activations:** compute the first two moments via the Gaussian CDF/PDF (closed-form) - **Result:** a Normal distribution over Q-values at each next state, used directly to form the pessimistic target (mean – β × std) - Requires only two forward passes; no additional parameters or rollouts

Theoretical improvement over MC-based PEVI:

Method	Bound type	Key term
MC-based PEVI (N=1)	Probabilistic (holds w/ prob 1–δ)	Scales with $R^2_{max}/(1-\gamma)^2$
MOMBO	**Deterministic** (always holds)	Depends only on network activation constants G_l, $C_l \leq 1$

MOMBO's bound is strictly tighter: it holds without probability qualification and depends only on the network's Lipschitz structure.

Results

Evaluated on the **D4RL offline benchmark** across 12 environment-dataset combinations: halfcheetah, hopper, and walker2d × random, medium, medium-replay, and medium-expert (4 seeds). Two metrics: Normalized Reward (final policy quality) and AULC (area under the learning curve, measuring convergence speed and stability).

MOMBO achieves the **best average AULC ranking of 1.2** across all 12 settings:

Dataset type	MOPO AULC rank	MOBILE AULC rank	MOMBO AULC rank
random	2.7	2.0	**1.3**
medium	2.7	2.0	**1.3**
medium-replay	2.3	2.0	**1.7**
medium-expert	2.7	2.0	**1.3**
Overall	2.7	2.2	**1.2**

Rank 1 = best. Lower is better.

Selected AULC scores on the most practically relevant settings:

Task	MOMBO	MOBILE	MOPO
medium — hopper	**95.9 ± 2.5**	82.2 ± 7.3	37.0 ± 15.3
medium — walker2d	**84.0 ± 1.1**	79.0 ± 1.3	77.6 ± 1.3
medium-replay — hopper	**87.3 ± 2.0**	78.7 ± 4.0	81.7 ± 4.6
medium-expert — halfcheetah	**95.2 ± 0.7**	94.5 ± 1.8	77.1 ± 4.0
medium-expert — walker2d	**98.9 ± 3.3**	94.3 ± 0.9	88.3 ± 6.3

MOMBO's advantage is largest on AULC rather than final reward, directly confirming the lower-variance Bellman target hypothesis. The medium-hopper gap (95.9 vs 82.2 vs 37.0) is the most striking: MOPO's high variance under medium-quality data collapses entirely, while MOMBO stays stable.

Conclusion

- **Root cause identified:** High MC variance in Bellman targets (not model quality) is the primary source of instability in model-based offline RL.
- **Provably tighter guarantees:** MOMBO's deterministic suboptimality bound improves on probabilistic MC bounds; constants depend only on network architecture, not on reward scale or sample count.
- **Fastest convergence:** Best AULC ranking of 1.2 across all 12 D4RL settings; most striking on medium-hopper (AULC 95.9 vs 82.2 vs 37.0).
- **Minimal overhead:** Moment matching requires only two forward passes through the Q-network, with no additional parameters or MC rollouts.
- **Practically relevant:** Advantage is largest on medium-quality datasets (the norm in real applications), where MC variance is most destructive to learning stability.

References

1. **Akgül, A., Haußmann, M., & Kandemir, M. (2024).** Deterministic Uncertainty Propagation for Improved Model-Based Offline Reinforcement Learning. *NeurIPS 2024*. arXiv:2406.04088
2. **Jin, Y., Yang, Z., & Wang, Z. (2021).** Is Pessimism Provably Efficient for Offline RL? *ICML 2021*.

3. **Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., & Ma, T. (2020).** MOPO: Model-based Offline Policy Optimization. *NeurIPS 2020*.
4. **Sun, Y., et al. (2023).** Model-Bellman Inconsistency for Model-based Offline Reinforcement Learning. *ICML 2023*.
5. **Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., & Gaunt, A. L. (2019).** Deterministic Variational Inference for Robust Bayesian Neural Networks. *ICLR 2019*.
6. **Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020).** D4RL: Datasets for Deep Data-Driven Reinforcement Learning. *arXiv:2004.07219*.

EPPO: Evidential Proximal Policy Optimization

TMLR (2025) | *First Author*

Summary: State-of-the-art in non-stationary control: average rank 1.5 across 10+ environments. Evidential critic simultaneously preserves plasticity and drives directed exploration from a single probabilistic framework. TMLR 2025.

Links: - Paper - arXiv - Code - Scholar - View on Site

Introduction

Real-world control systems face continuously changing dynamics: robot joints wear and lose torque, terrain friction shifts underfoot, payloads change mid-task. On-policy RL methods like PPO are natural fits (they discard old data and learn only from fresh transitions) but fail in practice under non-stationarity. **Evidential Proximal Policy Optimization (EPPO)** integrates evidential deep learning into the PPO critic to simultaneously preserve the network's capacity to learn (plasticity) and direct exploration toward regions where dynamics have changed, from a single probabilistic framework.

Problem Statement

Two compounding problems prevent PPO from adapting to non-stationary environments:

- **Loss of plasticity:** Over long non-stationary training runs, critic neurons gradually saturate or go dormant. The network structurally loses the ability to update even when new data arrives. Prior plasticity fixes (PFO, CB) prevent saturation through regularization or reinitialization but provide no signal about *where* dynamics have changed.
- **Undirected exploration:** When dynamics change, the agent should actively probe the altered regions of the state space. Without a targeted uncertainty signal, exploration stays uniform and samples are wasted. Prior exploration methods (PPO_DRND) add a curiosity bonus but do nothing about plasticity.

- **Neither fix alone is sufficient:** Experiments confirm that approaches addressing only one challenge fall short; both must be solved simultaneously.
- **Gap:** No existing on-policy method provides a unified mechanism that quantifies value-function uncertainty to preserve plasticity *and* direct exploration under non-stationarity.

Methodology

EPPO replaces the scalar PPO critic with an **evidential critic**, a network outputting Normal-Inverse-Gamma (NIG) parameters (ω, ν, α, β) for each state, placing a full distribution over the value:

$$(\mu, \sigma^2) \mid s \sim \mathrm{NIG}\big(\omega(s), \nu(s), \alpha(s), \beta(s)\big)$$

This gives two types of uncertainty analytically: - **Aleatoric uncertainty** β/(α–1): irreducible noise from the environment's stochasticity - **Epistemic uncertainty** β/[ν(α–1)]: reducible uncertainty from limited data; **spikes when dynamics shift**

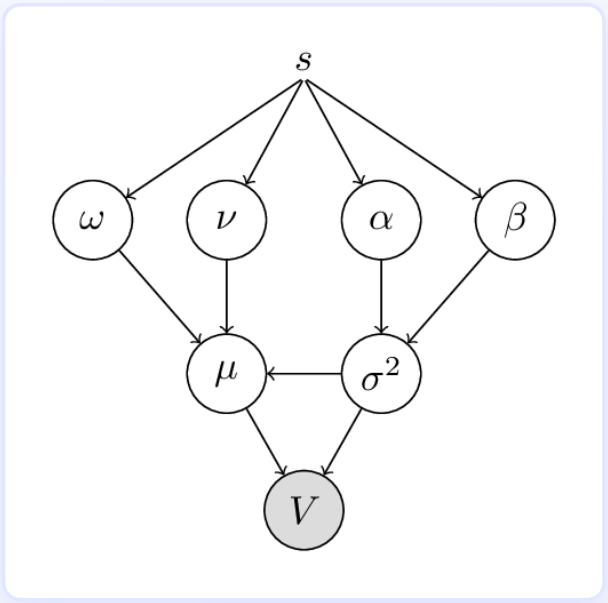


Figure 1. Generative model of EPPO's evidential critic. The state s determines the NIG hyperpriors (ω, ν, α, β) via a neural network. These parameterize a Normal-Inverse-Gamma prior over (μ, σ²), which governs the value prediction V. Marginalizing out (μ, σ²) yields a Student-t marginal likelihood used as the training objective.

Plasticity via adaptive gradient scaling. The evidential NLL loss contains an adaptive factor ζ that automatically dampens updates when approximation error Δ = |V – target| is large, preventing the runaway weight changes that saturate neurons and create dormant units.

Directed exploration via UCB advantages. Modeling V as a distribution transforms the Generalized Advantage Estimator (GAE) into a random variable. An Upper Confidence Bound converts this into directed exploration:

$$\widehat{A}_t^{\text{UCB}} = \mathbb{E}\left[\widehat{A}_t^{\text{GAE}}\right] + \kappa \sqrt{\text{Var}\left[\widehat{A}_t^{\text{GAE}}\right]}$$

Two variants differ in how GAE variance is derived: - **EPPO_cor** — propagates correlated uncertainties across the rollout - **EPPO_ind** — treats k-step advantage estimators as independent, making it more far-sighted for the same κ

Benchmark settings (novel contribution: Paralysis): EPPO introduces a challenging paralysis benchmark where specific leg joints have their torque progressively reduced to 0% and then fully restored [100→75→50→25→0→25→50→75→100]%, with 10 schemes across Ant-v5 and HalfCheetah-v5. No task identifiers are provided.

Results

Environments: Ant-v5 and HalfCheetah-v5 (MuJoCo Gymnasium v5), 500K steps per sub-task, **15 random seeds**.

Metrics: AULC (area under the learning curve, covering adaptation speed and sustained performance) and Final Return.

Baselines: PPO (foundation), PFO (plasticity via feature constraint), CB (continual backpropagation [Dohare et al., Nature 2024]), PPO_DRND (exploration [Yang et al., ICML 2024]). **EPPO_mean** (κ=0) isolates the plasticity benefit from the exploration benefit.

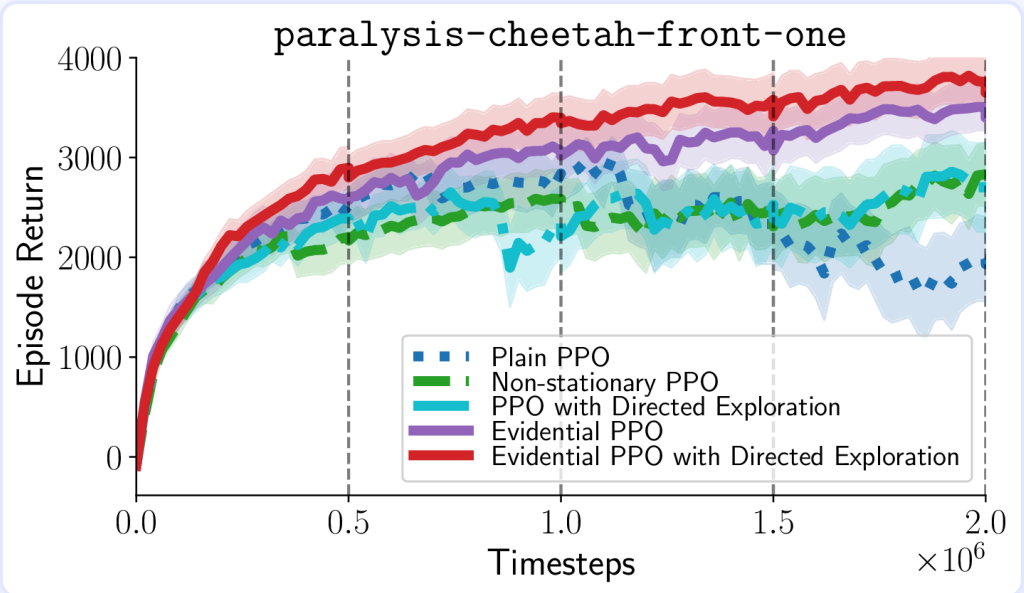


Figure 2. Episode return throughout training on HalfCheetah-v5 with progressive front-joint paralysis. Vertical dashed lines mark task changes. PPO and PFO plateau early and fail to recover. PPO_DRND gains some adaptability but degrades over time. EPPO variants continue to improve and maintain high performance across the entire training horizon.

Slippery environments (friction varied 0.5–4.0, 15 sub-tasks, averaged over 4 settings):

Model	Avg AULC	AULC rank	Avg Final Return	Final rank
PPO	2406	5.3	2475	5.5
PFO	2279	5.0	2371	5.3
CB	1956	6.5	2037	6.5
PPO_DRND	2359	4.5	2449	4.5
EPPO_mean	2658	3.5	2790	3.0
EPPO_cor	**2962**	**1.5**	**3120**	**1.5**
EPPO_ind	2908	1.8	3053	1.8

Paralysis environments (overall average, 10 schemes, 6 Ant + 4 HalfCheetah):

Model	Avg AULC	AULC rank	Avg Final Return	Final rank
PPO	2133	5.9	2310	5.8
PFO	2184	5.2	2376	5.3
CB	2091	6.2	2248	6.4
PPO_DRND	2343	3.8	2547	3.9
EPPO_mean	2643	3.1	2881	3.9
EPPO_cor	2781	2.1	3039	1.9
EPPO_ind	**2828**	**1.7**	**3074**	**1.7**

Rank 1 = best. Lower is better.

Plasticity analysis confirms EPPO variants maintain higher effective rank, higher stable rank, and fewer dormant units than all baselines throughout training ($p < 0.05$, one-sided paired t-test):

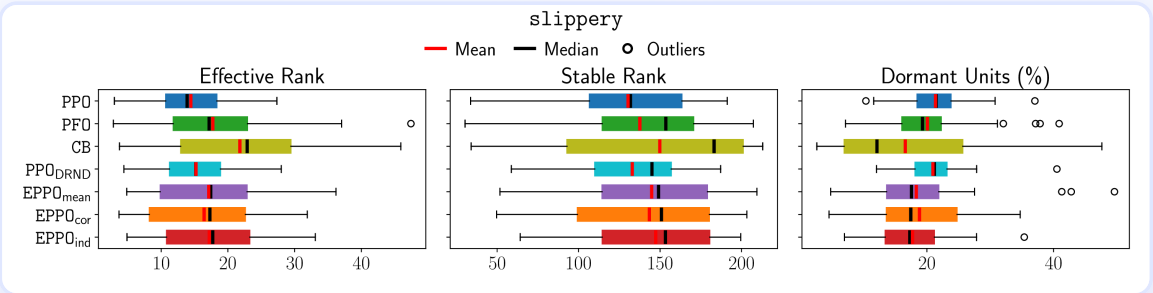


Figure 3. Plasticity metrics on slippery environments (box plots, 15 seeds × 4 settings). From left to right: effective rank, stable rank, and dormant unit percentage measured at the end of every sub-task. EPPO variants (orange/red) maintain higher ranks and fewer dormant units than all baselines, confirming evidential value learning preserves the critic's representational capacity.

Conclusion

- **Joint solution is necessary:** Baselines addressing only plasticity (PFO, CB) or only exploration (PPO_DRND) both underperform; EPPO's unified probabilistic framework is what enables consistent adaptation.
- **Evidential value learning preserves plasticity:** Higher effective rank, stable rank, and fewer dormant neurons compared to all baselines, verified across 15 seeds and two benchmark types.
- **UCB exploration amplifies gains:** The exploration bonus guides the agent toward states where dynamics have shifted (+42–44% AULC on HalfCheetah paralysis schemes vs. next-best baseline).
- **Novel benchmark:** The Paralysis environment (progressive joint torque reduction) provides a harder and more realistic non-stationarity test than standard friction-varying benchmarks.
- **State-of-the-art on both metrics:** Best AULC rank of 1.5 on slippery and 1.7 on paralysis environments, averaged across 10+ non-stationary scenarios.

References

1. **Akgül, A., Baykal, G., Haußmann, M., & Kandemir, M. (2025).** Overcoming Non-stationary Dynamics with Evidential Proximal Policy Optimization. *Transactions on Machine Learning Research (TMLR)*. arXiv:2503.01468
2. **Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017).** Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.
3. **Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015).** High-Dimensional Continuous Control Using Generalized Advantage Estimation. *ICLR 2016*.
4. **Amini, A., Schwarting, W., Soleimany, A., & Rus, D. (2020).** Deep Evidential Regression. *NeurIPS 2020*.

5. **Dohare, S., et al. (2024).** Loss of Plasticity in Deep Continual Learning. *Nature*.
6. **Yang, K., et al. (2024).** Exploration and Anti-Exploration with Distributional Random Network Distillation. *ICML 2024*.
7. **Todorov, E., Erez, T., & Tassa, Y. (2012).** MuJoCo: A physics engine for model-based control. *IROS 2012*.

CDDP: Continual Learning of Multi-modal Dynamics

L4DC (2024) | *First Author*

Summary: Learns new dynamical modes sequentially without catastrophic forgetting or mode labels, outperforming parameter-transfer baselines on 4 out of 5 datasets. Neural episodic memory with a Dirichlet Process prior for automatic mode discovery. L4DC 2024.

Links: - Paper - arXiv - Code - Scholar - View on Site

Introduction

No prior method could learn a dynamical system's behavioral modes sequentially without either catastrophically forgetting earlier ones or requiring explicit mode labels at test time — two constraints that the standard continual learning fix of parameter transfer (Variational Continual Learning) cannot simultaneously satisfy for multi-modal dynamics. **CDDP (Continual Dynamic Dirichlet Process)** solves both by replacing parameter transfer with a neural episodic memory and a Dirichlet Process prior on attention weights, enabling automatic mode discovery and zero-forgetting transfer within a Bayesian State-Space Model.

*This work was the second contribution of my **Master's thesis** at Istanbul Technical University, published at L4DC 2024.*

Problem Statement

- **Bayesian State-Space Models (BSSMs)** can fit a single dynamical mode well but are not designed for continual multi-modal learning.
- The standard continual learning fix, **Variational Continual Learning (VCL)**, transfers posterior parameters from one task to the next as the new prior. For classification, this works; for dynamics, it fails because the shared parameter space cannot simultaneously represent modes with fundamentally different transition structures.
- VCL also requires knowing which mode is active at test time, a strong and often unrealistic assumption.

- **Catastrophic forgetting:** adapting to a new mode overwrites representations of earlier ones when only parameter transfer is used.
- **Gap:** No prior method handles continual learning of sequential tasks with unknown, multi-modal dynamics without explicit mode labels or per-task network heads.

Methodology

CDDP augments a BSSM with two key components: a **neural episodic memory** of mode descriptors, and a **Dirichlet Process (DP) prior** on attention weights.

Memory-gated transition kernel: Given a context window of observations y_{1:C}, an encoder maps them to a query. Attention weights over R memory slots are computed via cosine similarity; the top-matching descriptor is retrieved and injected into the state transition kernel as an additional input, with no parameter transfer between tasks.

$$w_r\left(y_{1:C}, m_r\right) = \frac{e^{\langle m_r,\, e_\lambda\left(y_{1:C}\right)\rangle}}{\sum_{j=1}^{R} e^{\langle m_j,\, e_\lambda\left(y_{1:C}\right)\rangle}}$$

After observing a new task, memory is updated by a convex interpolation: high-similarity slots absorb the new mode; low-similarity slots are left largely unchanged, **preserving old knowledge without parameter transfer**.

Dirichlet Process prior (automatic mode discovery): The mixture weight π follows a GEM (stick-breaking) distribution with concentration α_0. Small α_0 concentrates mass on a few slots; large α_0 spreads mass broadly. The model never needs to be told how many modes exist.

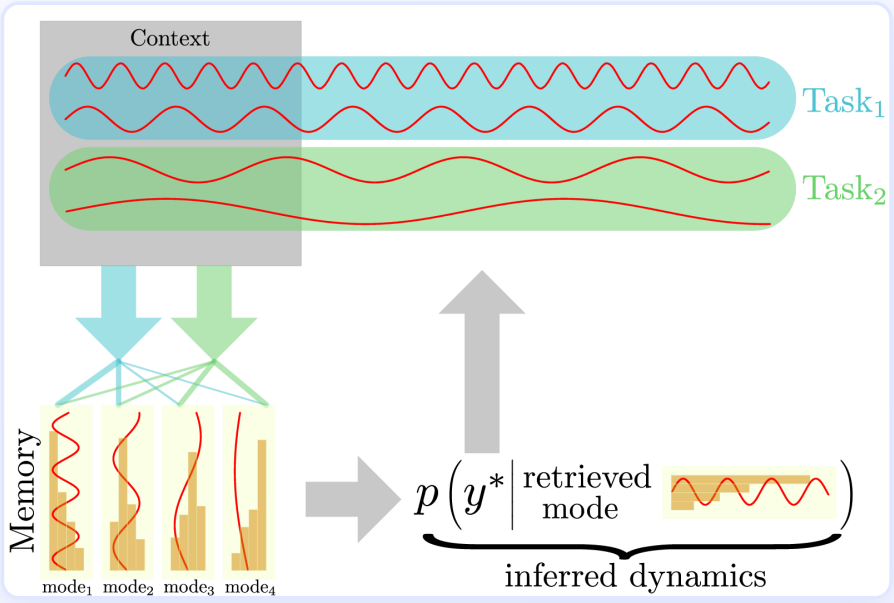


Figure 1. CDDP overview. During inference, the context sequence $y_{1:C}$ is encoded and matched against stored memory descriptors via cosine similarity. The Dirichlet Process prior induces a sparse attention distribution over R memory slots. The retrieved descriptor is injected into the transition kernel to predict $y_{C+1:T}$. Memory slots are updated online, with no parameter transfer between tasks.

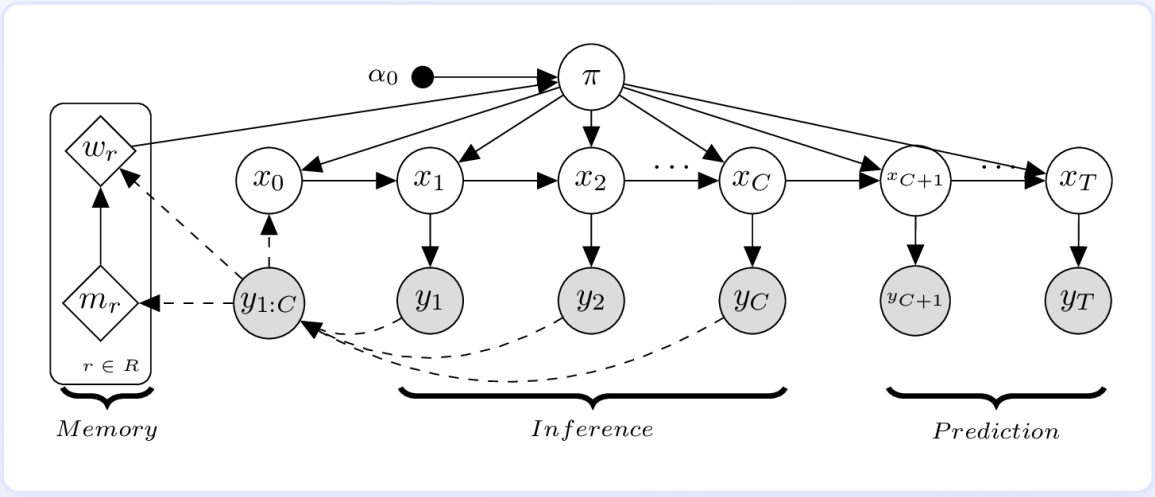


Figure 2. Graphical model of CDDP. Left: R memory slots and their attention weights. Center: context observations encoded and matched to memory; the Dirichlet Process–weighted descriptor initializes the latent state. Right: the latent chain propagates forward and emits predictions $y_{C+1:T}$.

Training: the variational objective (ELBO) includes a KL term that aligns learned attention weights with the DP prior, encouraging sparse and interpretable mode assignments.

Results

Evaluated on **3 synthetic** and **2 real-world** multi-modal trajectory datasets, each structured as a continual learning sequence. The model sees tasks one at a time; no mode labels are given at test time.

Datasets

These datasets are not standard ML benchmarks; they are drawn from dynamical systems and human motion capture, each presenting a distinct type of multi-modal behavior:

Sine Waves — 1D oscillations $y_t = A \sin(2\pi f t)$ with 5 amplitude levels $A \in \{3, 6, 9, 12, 15\}$ and 3 frequency levels $f \in \{\frac{2}{3}, 1, \frac{4}{3}\}$, yielding 15 modes across 5 tasks. The simplest benchmark; modes differ in scale and oscillation rate.

Lotka-Volterra — classic predator-prey ordinary differential equation (ODE):

$$\frac{dx_t}{dt} = \alpha x_t - \beta x_t y_t, \qquad \frac{dy_t}{dt} = \delta x_t y_t - \gamma y_t$$

Eight modes generated by varying the biological parameters $(\alpha, \beta, \gamma, \delta)$ across 4 tasks. Sequence length 25, step size Δt = 0.4. Each mode produces qualitatively different oscillatory dynamics between prey (x) and predator (y) populations.

Lorenz Attractor — chaotic 3D system with sensitive dependence on initial conditions:

$$\frac{dx_t}{dt} = \sigma(y_t - x_t), \qquad \frac{dy_t}{dt} = x_t(\rho - z_t) - y_t, \qquad \frac{dz_t}{dt} = x_t y_t - \beta z_t$$

Twelve modes from different parameter triples (σ, ρ, β) across 4 tasks. Sequence length 50, step size Δt = 0.01. This is the hardest synthetic benchmark: neighboring trajectories diverge exponentially, making mode identification from a short context window especially challenging.

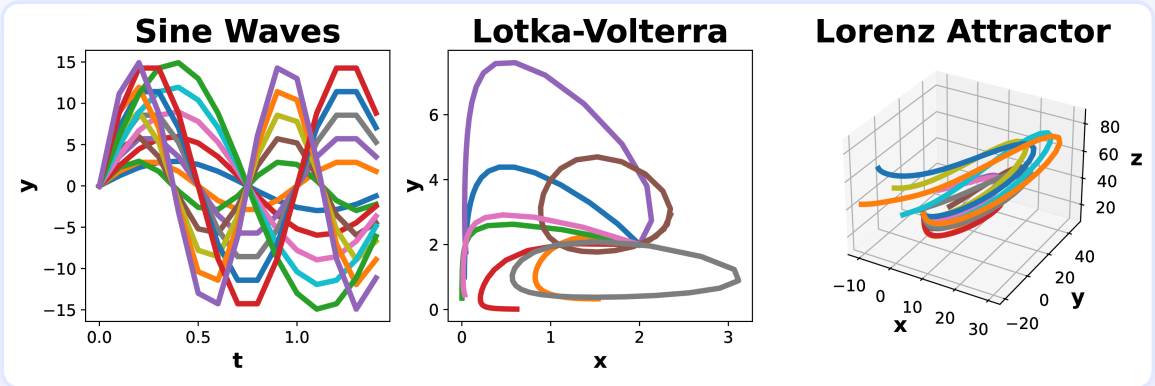


Figure 3. Examples from the three synthetic datasets. Left: Sine Waves with different amplitude/frequency combinations (1D). Center: Lotka-Volterra predator-prey trajectories showing distinct oscillation cycles for different parameter settings (2D). Right: Lorenz Attractor chaotic trajectories projected from 3D, where each color corresponds to a different parameter mode.

Libras Movement — 2D hand-movement trajectories from 15 classes of Brazilian Sign Language (LIBRAS), captured via video at 45 frames per sequence. 5 tasks, 15 modes, 180 train / 180 test sequences. Modes correspond to distinct sign gestures with different spatial extents and trajectories.

Character Trajectories — 3-attribute stylus-pen trajectories (x position, y position, pen tip force) for 20 English characters, subsampled to length 109. 5 tasks, 20 modes, 1422 train / 1436 test sequences. The most challenging real-world dataset: 20 distinct character shapes with shared stroke primitives require fine-grained mode discrimination.

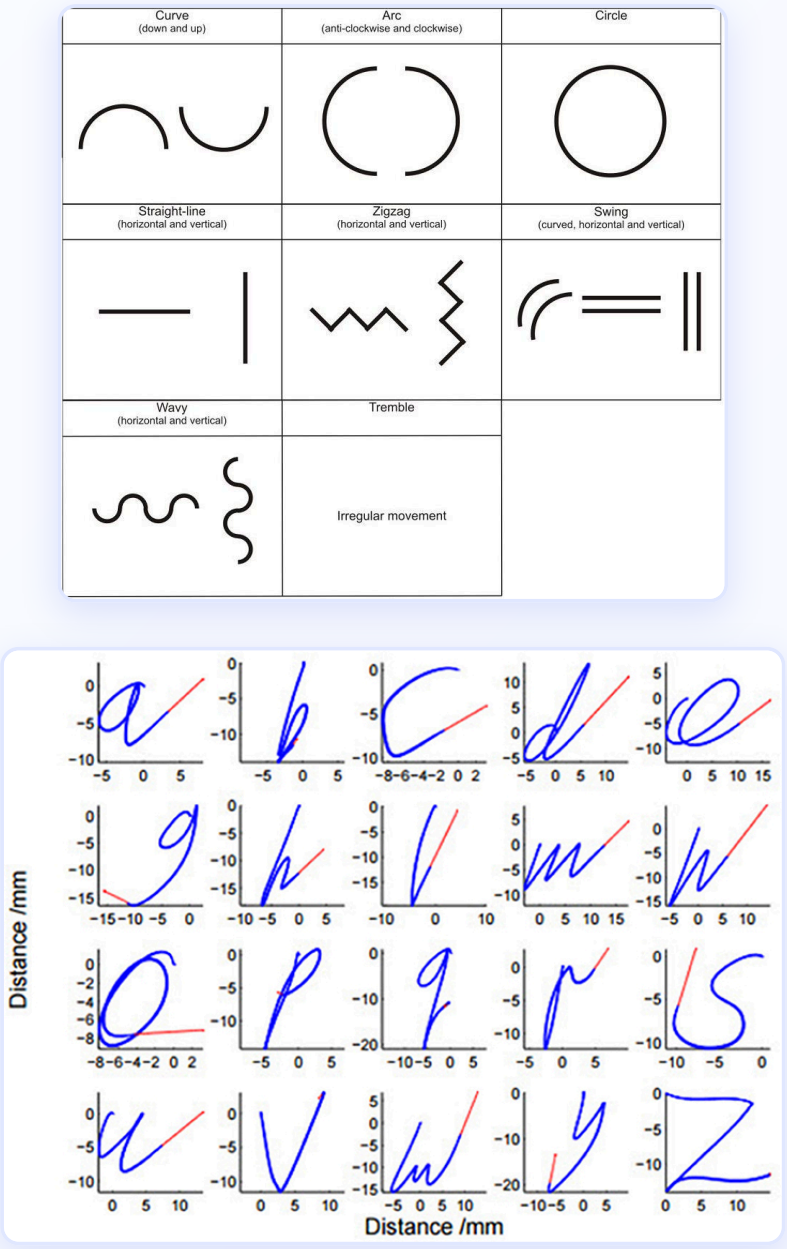


Figure 4. Real-world datasets. Left: hand-movement trajectories from the Libras dataset (Brazilian Sign Language), with each panel showing a different sign class. Right: stylus-pen character trajectories for selected English alphabet letters; modes correspond to distinct characters whose strokes share low-level primitives.

Dataset summary:

Type	Dataset	Tasks	Modes	Seq. Length	Attributes
Synthetic	Sine Waves	5	15	15	1
Synthetic	Lotka-Volterra	4	8	25	2
Synthetic	Lorenz Attractor	4	12	50	3
Real-world	Libras	5	15	45	2
Real-world	Character Trajectories	5	20	109	3

Quantitative Results

Metrics: AUC of NMSE and NLL plotted against tasks learned, averaged over 10 repetitions. Lower is better for both. - **NMSE** (Normalized MSE): prediction error relative to signal magnitude - **NLL** (Negative Log-Likelihood): calibration quality of the predictive distribution

Main results — AUC NMSE ↓ and AUC NLL ↓ (mean ± SE, 10 seeds). Lower is better.

Dataset	VCL-BSSM NMSE	CDDP NMSE	VCL-BSSM NLL	CDDP NLL
Sine Waves	1.00 ± 0.04	**0.91 ± 0.03**	3.57 ± 0.09	**3.50 ± 0.09**
Lotka-Volterra	**0.58 ± 0.04**	0.60 ± 0.06	1.50 ± 0.05	**1.32 ± 0.08**
Lorenz Attractor	0.26 ± 0.00	**0.24 ± 0.01**	4.42 ± 0.04	**4.35 ± 0.06**
Libras	**0.14 ± 0.00**	**0.14 ± 0.00**	-0.37 ± 0.02	**-0.39 ± 0.04**
Character Trajectories	0.87 ± 0.04	**0.64 ± 0.01**	0.14 ± 0.02	**-0.19 ± 0.03**

CDDP wins 4/5 on NMSE and 5/5 on NLL. Largest gain on Character Trajectories: −26% NMSE, NLL drops from 0.14 to −0.19 (better calibration).

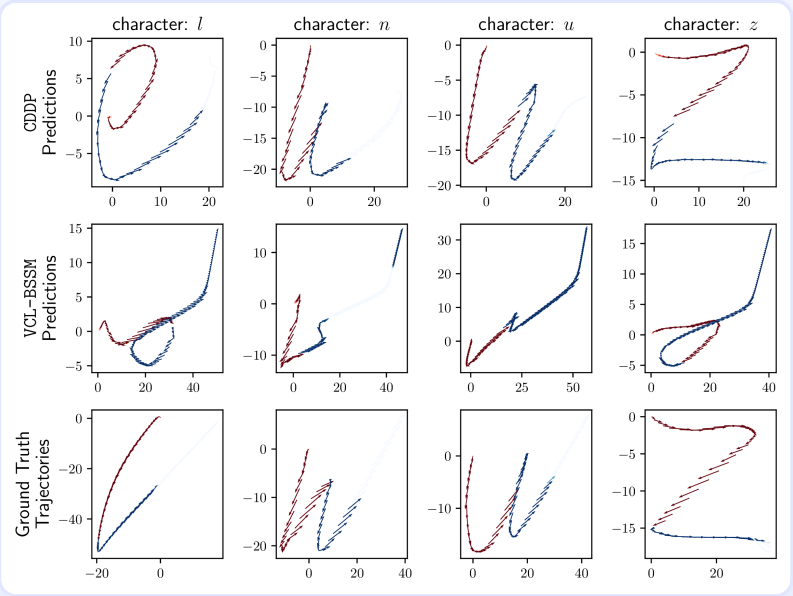


Figure 3. CDDP predictions on Character Trajectories. Black = context window $y_{1:C}$ (observed); colored = prediction $y_{C+1:T}$. CDDP correctly identifies the character mode from the short context and produces accurate trajectories across all 20 classes, without being told which character is being written.

Ablation on Sine Waves confirms that both learned memory content and the absence of parameter transfer are necessary for best performance. Fixed-initialization variants degrade monotonically with initialization magnitude; adding parameter transfer to CDDP does not help and slightly hurts NLL.

Conclusion

- **First study** on continual learning of multi-modal dynamical systems, introducing both the problem formulation and the associated continual learning risk objective.
- **VCL-BSSM** introduced as a strong parameter-transfer baseline for practitioners adapting continual classification methods to dynamics.
- **Memory beats parameter transfer:** CDDP outperforms VCL-BSSM in 4/5 datasets on NMSE and 5/5 on NLL; memory preserves structure that shared parameters cannot represent simultaneously.
- **No mode labels required:** the DP prior discovers the number of active modes automatically from data.
- **Broad applicability:** the framework applies directly to weather forecasting (features transferred across climates), autonomous driving (adapting across countries), and model-based RL (handling environment changes from agent actions or external factors).

References

1. **Akgül, A., Unal, G., & Kandemir, M. (2024).** Continual Learning of Multi-modal Dynamics with External Memory. *Proceedings of the 6th Annual Learning for Dynamics and Control Conference (L4DC 2024)*. arXiv:2203.00936
2. **Nguyen, C. V., Li, Y., Bui, T. D., & Turner, R. E. (2018).** Variational Continual Learning. *ICLR 2018*.
3. **Rangapuram, S. S., et al. (2018).** Deep State Space Models for Time Series Forecasting. *NeurIPS 2018*.
4. **Sethuraman, J. (1994).** A constructive definition of Dirichlet priors. *Statistica Sinica*.
5. **Graves, A., Wayne, G., & Danihelka, I. (2014).** Neural Turing Machines. *arXiv:1410.5401*.
6. **Kirkpatrick, J., et al. (2017).** Overcoming catastrophic forgetting in neural networks. *PNAS*.

Evidential Turing Processes

ICLR (2022) | *2nd Author*

Summary: The only model achieving top-tier performance on calibration, class overlap, and OOD detection simultaneously across five real-world benchmarks. External memory unifies global and local uncertainty in a single principled framework. ICLR 2022.

Links: - Paper - arXiv - Video - Code - Scholar - View on Site

Introduction

No single probabilistic classifier had simultaneously achieved model calibration, faithful class-overlap quantification, and reliable out-of-distribution detection — what this work formalizes as **total calibration**. Existing Bayesian approaches address these challenges in isolation: **Parametric Bayesian Models (PBMs)** such as Bayesian Neural Networks (BNNs) capture model-level uncertainty well but poorly represent local class ambiguity; **Evidential Bayesian Models (EBMs)** such as Evidential Deep Learning (EDL) quantify class overlap well but lack the global structure needed for out-of-distribution detection. **Evidential Turing Processes (ETP)** unifies both through the **Complete Bayesian Model (CBM)** framework — provably the minimal structure required for total calibration — realized via a **Turing Process**: a stochastic process with external memory that accumulates in-domain evidence without requiring a held-out context set at test time.

In this work (2nd author), I designed and conducted all experiments and wrote the experimental section of the paper.

Problem Statement

The paper formally defines **Total Calibration** as the simultaneous competence of a discriminative predictor in three distinct tasks:

- **Model misfit**: how closely the model's predicted class distribution matches the true class-conditional distribution, measured by **Negative Log-Likelihood (NLL)**. Reducible by seeing more training data; signals systematic errors in model fit.
- **Class overlap**: how faithfully the model reflects the irreducible ambiguity at decision boundaries (where two classes genuinely overlap), measured by **Expected Calibration Error (ECE)**. Cannot be eliminated with more data; requires a local uncertainty mechanism.
- **Domain mismatch**: the ability to detect inputs drawn from a distribution unseen during training (out-of-distribution detection), measured by the **Area Under the ROC Curve (AUROC)**. Requires a global uncertainty signal that builds up over the training set.

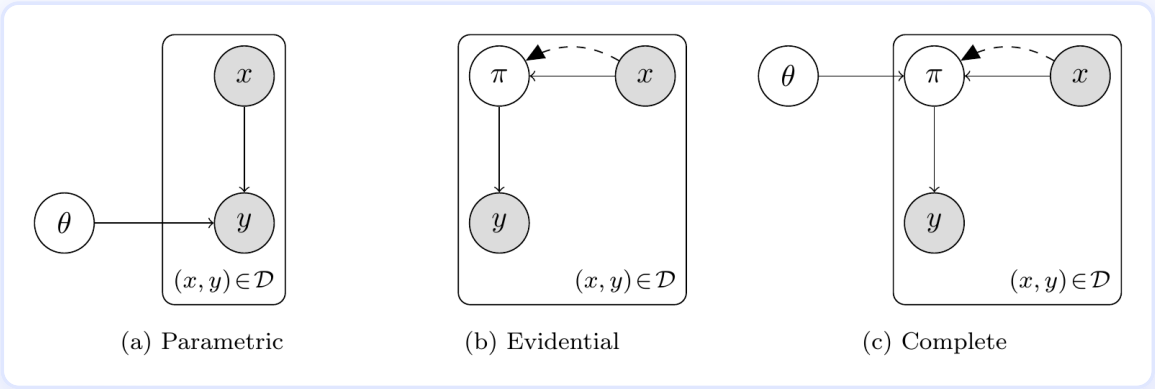
The key observation: PBMs maintain a posterior over a global parameter θ; this variance shrinks with data and drives good NLL and AUROC, but poorly captures per-sample class ambiguity. EBMs maintain an observation-specific prior over class probabilities π; this captures class overlap via ECE, but has no global variable to detect domain shift. No prior method achieves all three simultaneously.

Methodology

Complete Bayesian Models (CBM)

The CBM framework introduces a **global** parameter θ (from PBM) alongside a **local** class-probability variable π (from EBM). Their joint generative model is:

$$\Pr\left[y \mid \pi\right]\ p(\pi \mid \theta, x)\ p(\theta)$$



(a) Parametric (b) Evidential (c) Complete

Figure 1. Plate diagrams of the three Bayesian modeling approaches. Shaded nodes are observed; open circles are latent; the diamond node M is a deterministic external memory. Solid arrows are generative dependencies; dashed arrows are amortized inference paths. (a) Parametric Bayesian Models (PBM): a single global latent θ modulates all predictions; its posterior shrinks with data size, enabling domain-shift detection but not local class-overlap quantification. (b) Evidential Bayesian Models (EBM): a local latent π is inferred per input x via amortized inference; it captures class overlap but has no global variable to detect distribution shift. (c) Complete Bayesian Models (CBM): combines both θ and π, inheriting the favorable uncertainty decomposition properties of each approach.

The CBM variance decomposition contains all three required uncertainty components:

$$\mathrm{Var}\big[y \mid x\big] = \underbrace{\mathrm{Var}_{p(\theta|\mathcal{D})}\big[\mathbb{E}_{p(\pi|x,\theta)}\big[\mathbb{E}\big[y|\pi\big]\big]\big]}_{\text{Reducible model uncertainty}} + \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}\big[\mathrm{Var}_{p(\pi|\theta,x)}\big[\mathbb{E}\big[y|\pi\big]\big]\big]}_{\text{Irreducible model uncertainty}} + \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}\big[\mathbb{E}_{p(\pi|x,\theta)}\big[\mathrm{Var}\big[y|\pi\big]\big]\big]}_{\text{Data uncertainty}}$$

The first term recovers the reducible model uncertainty of PBMs (handles model misfit and domain mismatch); the second and third recover the irreducible and data uncertainty of EBMs (handles class overlap). **A CBM is the minimal structure that guarantees total calibration.**

Turing Processes and the ETP

Equipping the CBM's empirical prior $p(\pi \mid \theta, x)$ with a mechanism that *learns to be accurate* on individual samples requires two ingredients: (1) a global random variable connecting samples across the dataset (as in Neural Processes [Garnelo et al., 2018]), and (2) a memory that accumulates evidence from context data without needing to store that context at test time (as in Neural Turing Machines [Graves et al., 2014]).

The **Turing Process** is a formally defined stochastic process combining both: its prior $p_M(\theta)$ has free parameters M (the memory), and conditioning on a context set $\mathcal{D}_C$ updates the memory parameters via an explicit rule r: $M' = r\big(M, \mathcal{D}_C\big)$. Unlike Neural Processes, no context is required at prediction time, since the memory has absorbed the context during training.

Evidential Turing Processes (ETP) instantiate this design as a CBM:

$$p\big(y, \mathcal{D}_T, \pi, \theta\big) = p(w) \quad \underbrace{p_M(Z)}_{\text{External memory}} \prod_{(x,y)\,\in\,\mathcal{D}_T} \big[p\big(y|\pi\big) \underbrace{p(\pi \mid Z, w, x)}_{\text{Input-specific prior}}\big]$$

The global parameters $\theta = \{w, Z\}$ split into a neural network weight w and a memory $Z = \{z_1, \ldots, z_R\}$ parameterized by $M = (m_1, \ldots, m_R)$. Each input x is encoded and matched to memory via **attention**, yielding input-specific Dirichlet hyperparameters for π. Memory cells are updated by a convex interpolation rule that writes new uncertainty information (both the ground-truth label and the model's current soft prediction) into cells weighted by their similarity to the current input, without gradient-based optimization of M.

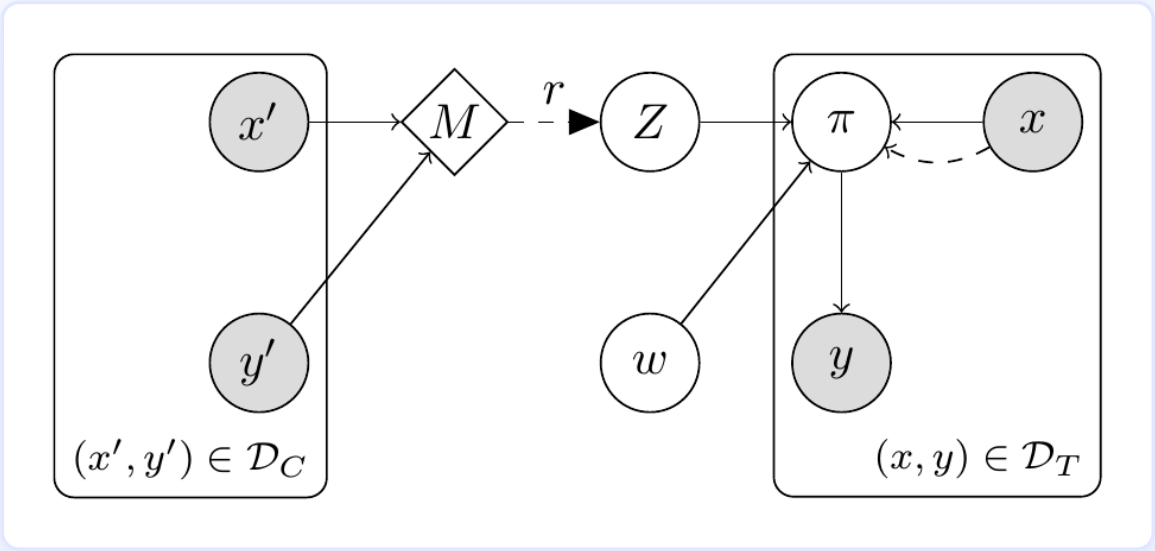


Figure 2. ETP generative model (plate diagram). Left plate: context observations (x′, y′) ∈ D_C update the memory M via the explicit rule r, which parameterizes the global latent Z (diamond = deterministic node). Right plate: for each target (x, y) ∈ D_T, the global Z and network weights w jointly determine the input-specific prior over local variable π via an attention mechanism, which then generates the class label y. The dashed arrows mark the amortization and memory-update paths; no context set is needed at test time.

Ablation structure: Deactivating ETP components one at a time recovers established baselines. Removing the external memory and update rule recovers the **Evidential Neural Process (ENP)** (a novel surrogate for the Attentive Neural Process [Kim et al., 2019]); further removing the local variable π recovers the standard **Neural Process**; removing the global Z recovers **EDL**; removing both π and Z recovers a **BNN**.

Model	Local π	Global Z	Memory M	Rule r	Context at test time
ETP (target)	✓	✓	✓	✓	×
ENP (surrogate)	✓	✓	×	×	×
EDL	✓	×	×	×	×
BNN	×	×	×	×	×

Results

Datasets: Five real-world classification benchmarks covering text and images, each with a designated out-of-distribution dataset for AUROC evaluation:

Domain data	Architecture	OOD data
IMDB sentiment	LSTM	Random tokens
Fashion MNIST	LeNet-5	MNIST
SVHN	LeNet-5	CIFAR10
CIFAR10	LeNet-5	SVHN
CIFAR100	ResNet-18	TinyImageNet

Results are averages over **10 random seeds**. Baselines are BNN, EDL, and ENP (curated as the strongest possible ablation). ETP is the only method that consistently ranks among top performers across all three total-calibration metrics on every dataset.

Total Calibration Results (mean ± std, 10 seeds)

Test error ↓ — model fit:

	IMDB	Fashion	SVHN	CIFAR10	CIFAR100
BNN	**16.4 ± 0.6**	**7.9 ± 0.1**	7.9 ± 0.1	**15.3 ± 0.3**	30.2 ± 0.3
EDL	38.3 ± 13.3	8.6 ± 0.1	7.3 ± 0.1	18.5 ± 0.2	45.2 ± 0.4
ENP	50.0 ± 0.0	**7.9 ± 0.2**	**6.7 ± 0.1**	**14.8 ± 0.2**	39.0 ± 0.3
ETP	**15.8 ± 1.3**	**7.9 ± 0.2**	**6.9 ± 0.1**	**15.3 ± 0.2**	**29.2 ± 0.3**

ECE ↓ — class overlap calibration:

	IMDB	Fashion	SVHN	CIFAR10	CIFAR100
BNN	14.4 ± 0.4	6.7 ± 0.0	6.5 ± 0.0	5.5 ± 0.3	15.2 ± 0.0
EDL	41.1 ± 2.6	3.7 ± 0.2	4.0 ± 0.1	9.0 ± 0.2	**5.3 ± 0.4**
ENP	**0.8 ± 1.6**	6.0 ± 0.2	10.7 ± 0.2	7.2 ± 0.3	39.7 ± 0.4
ETP	**3.1 ± 0.4**	**2.6 ± 0.2**	**2.6 ± 0.1**	**2.7 ± 0.1**	6.6 ± 0.1

NLL ↓ — negative log-likelihood:

	IMDB	Fashion	SVHN	CIFAR10	CIFAR100
BNN	0.47	0.65	0.71	0.50	1.78
EDL	0.66	0.37	0.34	0.72	2.24
ENP	0.69	0.34	0.33	0.50	2.52
ETP	**0.37**	**0.29**	**0.26**	**0.46**	**1.36**

ETP achieves the best NLL on all five datasets, a result no other method matches.

AUROC ↑ — out-of-distribution detection:

	IMDB	Fashion	SVHN	CIFAR10	CIFAR100
BNN	**60.9 ± 4.2**	75.9 ± 2.3	86.2 ± 0.5	**84.1 ± 1.3**	97.2 ± 0.5
EDL	**55.1 ± 5.1**	77.5 ± 2.0	90.9 ± 0.3	79.2 ± 0.7	89.6 ± 0.3
ENP	**53.7 ± 5.7**	**88.9 ± 1.0**	**92.4 ± 0.4**	**81.4 ± 0.8**	**100.0 ± 0.1**
ETP	**59.1 ± 5.1**	**90.0 ± 0.9**	90.0 ± 0.4	**82.1 ± 0.6**	99.6 ± 0.1

Bold entries are within three standard deviations of the best result. ETP is the only method that never catastrophically fails on any metric: EDL collapses on IMDB accuracy (38.3% error) and NLL universally; ENP collapses on IMDB (50% error, no better than chance) and NLL; BNN fails at ECE across the board. ETP is the only model that simultaneously achieves top-tier NLL on all five datasets while remaining competitive on ECE and AUROC.

Robustness Against Gradual Domain Shift

Models are evaluated on corrupted variants of three datasets (FMNIST-C, CIFAR10-C, SVHN-C) using **19 corruption types** (e.g., motion blur, fog, pixelation) at **five severity levels**. The ECE metric is tracked at each corruption level.

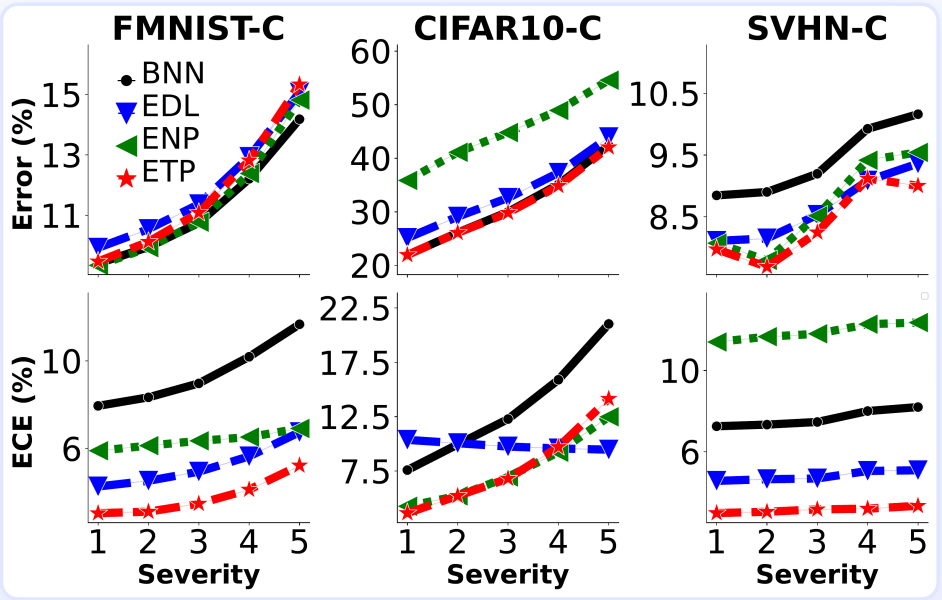


Figure 3. ECE averaged across 19 corruption types at five severity levels on Fashion MNIST (left), CIFAR10 (center), and SVHN (right). ETP maintains the lowest ECE across nearly all datasets and distortion levels, demonstrating that its calibration advantage holds under gradual distributional shift, not only on clean test sets.

Computational Cost

Measured on CIFAR10, wall-clock time per epoch: ETP (10.8 ± 0.1 s), BNN (8.2 ± 0.1 s), EDL (8.6 ± 0.1 s), ENP (9.5 ± 0.2 s). ETP incurs a ~14% overhead versus the next-closest baseline (ENP), a modest cost relative to the consistent gain across all three calibration axes.

Conclusion

- **Total Calibration formalized:** Model misfit, class overlap, and domain mismatch are defined as three formally distinct uncertainty types, each with a designated metric (NLL, ECE, AUROC), providing a rigorous evaluation protocol for uncertainty-aware classifiers.
- **CBM as a unifying theory:** The Complete Bayesian Model framework shows analytically that a model combining a global (PBM-style) and a local (EBM-style) random variable inherits the favorable uncertainty decomposition of each, a strictly necessary structure for total calibration.
- **Turing Process as a practical realization:** The external memory mechanism accumulates uncertainty evidence during training without requiring a context set at test time, making ETP suitable for standard (non-meta-learning) classification pipelines.
- **Unique simultaneous performance:** ETP is the only method among all compared approaches that consistently ranks among top performers on all three calibration metrics across all five real-world datasets.
- **Corruption robustness:** The ECE advantage holds under 19 types of data corruption at five severity levels, confirming that ETP's calibration improvements reflect a genuine improvement in uncertainty structure rather than overfitting to clean-domain statistics.

References

1. **Kandemir, M., Akgül, A., Haussmann, M., & Unal, G. (2022).** Evidential Turing Processes. *International Conference on Learning Representations (ICLR 2022)*. Code
2. **Graves, A., Wayne, G., & Danihelka, I. (2014).** Neural Turing Machines. *arXiv:1410.5401*.
3. **Garnelo, M., Rosenbaum, D., Maddison, C. J., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., & Eslami, S. M. A. (2018).** Neural Processes. *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*.
4. **Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, S. M. A., Rosenbaum, D., Vinyals, O., & Teh, Y. W. (2019).** Attentive Neural Processes. *ICLR 2019*.
5. **Sensoy, M., Kaplan, L., & Kandemir, M. (2018).** Evidential Deep Learning to Quantify Classification Uncertainty. *NeurIPS 2018*.
6. **Malinin, A., & Gales, M. (2018).** Predictive Uncertainty Estimation via Prior Networks. *NeurIPS 2018*.

7. **Hendrycks, D., & Dietterich, T. (2019).** Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *ICLR 2019*.

iS-QL: Bridging Target-free and Target-based Reinforcement Learning

ICLR (2026) | *4th Author*

Summary: Closes the 10–60% performance gap between target-free and target-based RL by sharing all parameters except the final linear head — matching target-based stability at near target-free memory cost across Atari, DMC, and language modeling. ICLR 2026.

Links: - Paper - arXiv - Code - Scholar - View on Site

Introduction

The target-free vs. target-based choice in deep RL had no middle ground: target networks stabilize training but double Q-network memory, while target-free methods cut memory at the cost of a 10–60% performance drop on standard benchmarks. No prior work escaped this binary. **iS-QL** (iterated Shared Q-Learning) resolves it by sharing all network parameters except the final linear output layer between the online and target sides — delivering target-based stability at near target-free memory cost across five distinct RL settings.

In this collaborative work (4th author), I designed and conducted the offline language model experiments — specifically the evaluation of iS-ILQL on the Wordle task using a GPT-2 backbone.

Problem Statement

- **Target networks** (Mnih et al., 2015) stabilize training by decoupling the regression target from the changing online network. They are critical for large architectures and are shown to matter even for methods originally designed without them.
- The cost is **doubled memory footprint** for Q-networks, limiting usable network size on constrained hardware (edge devices, high-dimensional inputs, mixture-of-experts critics).
- **Target-free methods** avoid the extra memory but suffer severe performance drops: a 10–60% AUC gap relative to their target-based counterparts in standard benchmarks.
- **Gap:** No prior work escapes this binary choice. The question is whether a hybrid architecture can achieve target-based stability with target-free memory usage.

Methodology

iS-QL uses a **single Q-network with K+1 linear output heads**, sharing all parameters in the backbone (convolutional or MLP body) while keeping the heads lightweight and separate.

Architecture (Figure 1):

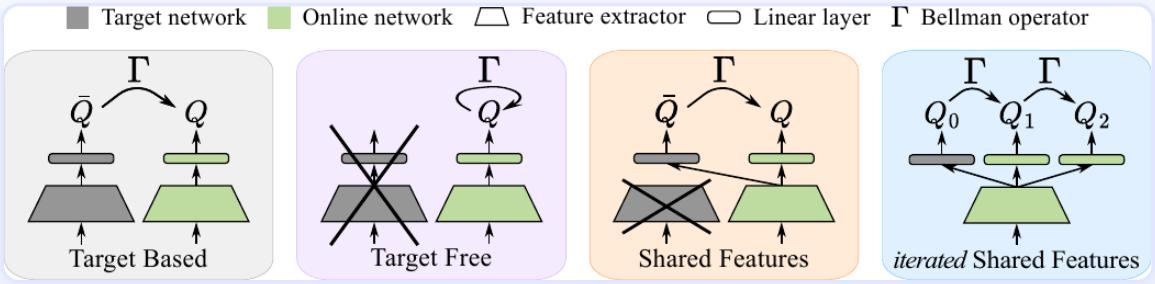


Figure 1. Conceptual comparison of target-based, target-free, shared features, and iterated shared features (iS-QL). In the shared-features variant, only the last linear layer is duplicated as the target; the backbone is shared with the live online network. iS-QL extends this with K+1 heads forming a chain of consecutive Bellman iterations; each head is trained to approximate the Bellman update of the previous one. From Vincent et al., ICLR 2026.

Key ideas:

- Let ω denote the shared backbone parameters and $\omega_0, \omega_1, \ldots, \omega_K$ the K+1 head parameters. Define $\theta_k = (\omega, \omega_k)$.
- Head ω_0 is **never updated by gradient descent**; it plays the role of the target network.
- The training loss sums K temporal-difference objectives in a chain:

$$\mathcal{L}^{\text{iS-QL}}(\theta) = \sum_{k=1}^{K} \mathcal{L}^{\text{TD}}(\theta_k, \theta_{k-1})$$

where head k–1 provides regression targets for head k (stop-gradient applied).

- Every T steps, heads are **cyclically shifted**: $\omega_k \leftarrow \omega_{k+1}$ for k = 0, …, K–1. This propagates learned values backward through the chain and refreshes the frozen head ω_0 with a recent snapshot of ω_1, exactly as DQN's hard target update, but only for a tiny linear layer.
- Learning K consecutive Bellman iterations in parallel improves sample efficiency beyond simply sharing the backbone.

Why it works: three mechanisms analysed in the paper:

Mechanism	Target-free	iS-QL K=1	Target-based
Gradient alignment with target-based	low	**high**	—
Target churn (instability of regression targets)	high	**intermediate**	zero
Feature srank (representational capacity)	low	**higher**	moderate

Variants evaluated:

- **iS-DQN** — discrete online RL (Atari)
- **iS-CQL** — discrete offline RL (Atari)
- **iS-SAC** — continuous online RL (DeepMind Control Suite)
- **iS-ILQL** — offline language RL (Wordle, GPT-2 small backbone)
- **iS-Stream Q(λ)** — streaming RL (no replay buffer, no batch updates)

Results

All AUC scores are normalized by the target-based approach (= 100); higher is better. Results use IQM with 95% stratified bootstrap intervals.

Online Discrete Control — Atari

Evaluated on 15 Atari games with CNN+LayerNorm:

Method	Normalized AUC	Parameters vs target-based
TF-DQN (target-free)	90%	~50%
TB-DQN (target-based)	100%	100%
iS-DQN K=9	**106%**	**~50%**

iS-DQN K=9 **outperforms the target-based approach by 6%** while using approximately half its parameters. Without LayerNorm, where target-free suffers a 60% performance drop, iS-DQN K=1 already cuts this gap to 18%, by storing only one lightweight linear head. Results on the IMPALA architecture confirm the trend: iS-DQN fully closes the performance gap as K increases.

Offline Discrete Control — Atari

Evaluated on 10 Atari games with IMPALA+LayerNorm and CQL loss (10% of DQN dataset):

Method	Performance gap vs target-based
TF-CQL (target-free)	−26%
iS-CQL K=9	**−6%**

iS-CQL shrinks the offline performance gap from 26% to 6%.

Online Continuous Control — DeepMind Control Suite

Evaluated on 7 hard DMC tasks with SAC+SimbaV2+BatchNorm:

- iS-SAC K=1 **fully recovers** the performance drop of the target-free approach.
- Reduces **total parameter count by 49%** (SimbaV2 uses a large critic; only the linear head is duplicated).

Offline Language Modeling — Wordle

Evaluated with Implicit Language Q-Learning (ILQL) on the Wordle word-guessing game using GPT-2 small (264M parameters total):

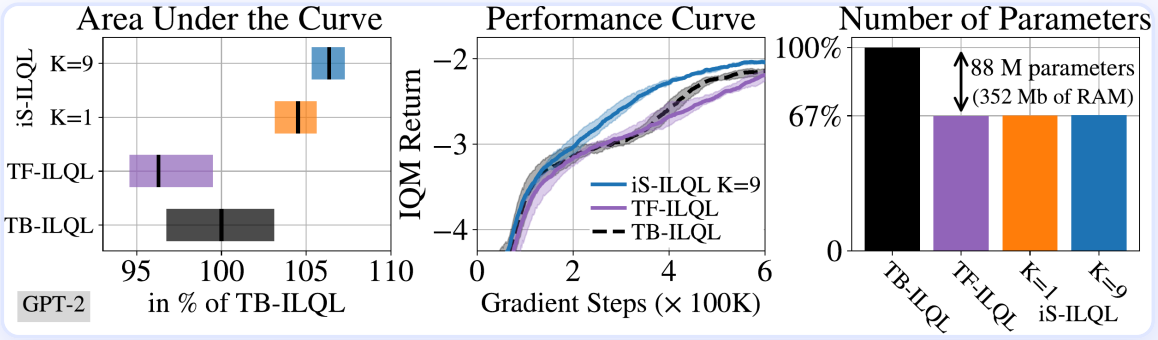


Figure 2. Performance on the Wordle offline RL task (GPT-2 small backbone). iS-ILQL K=9 improves over the target-based approach by more than 5% in normalized AUC while saving 33% of RAM (88 million parameters). Sharing features also enables computing the TD error in a single forward pass, reducing training time. From Vincent et al., ICLR 2026.

Method	Normalized AUC	Parameters
TF-ILQL	≈ TB-ILQL	−88M vs TB
TB-ILQL	100%	264M
iS-ILQL K=9	**> 105%**	**264M − 88M = 176M**

iS-ILQL K=9 **outperforms the target-based approach by more than 5%** and saves 88 million parameters (33% RAM reduction). Because both the online and target embeddings share a single forward pass, iS-ILQL also trains faster than TB-ILQL.

Streaming RL — Atari (no replay buffer)

Applied to Stream Q(λ) [Elsayed et al., 2024] on 7 Atari games without replay buffer or batch updates:

- iS-Stream Q(λ) K=3 improves over the target-free baseline by **more than 10%** in AUC, matching or outperforming the target-based reference on **6 out of 7 games**.

Conclusion

- **Simple modification, broad impact:** Sharing all parameters except the final linear head reduces memory to near target-free levels while restoring target-based stability across five distinct RL settings.
- **Iterated Bellman updates amplify the gain:** Learning K consecutive Bellman updates in parallel with the shared backbone significantly narrows, and in some settings eliminates, the performance gap with target-based methods.
- **Scalable to large architectures:** The 49% total parameter reduction on SimbaV2 and 33% RAM saving on GPT-2 confirm practical value for memory-constrained hardware.
- **Analysis-backed:** Gradient alignment, target churn, and srank measurements all confirm that iS-QL's learning dynamics are systematically closer to target-based than target-free, explaining the empirical gains.
- **Orthogonal to existing regularization:** iS-QL combines additively with LayerNorm, BatchNorm, MellowMax, and other target-free stabilizers; the gains are complementary.

References

1. **Vincent, T., Tripathi, Y., Faust, T., Akgül, A., Oren, Y., Kandemir, M., Peters, J., & D'Eramo, C. (2026).** Bridging the Performance-gap between Target-free and Target-based Reinforcement Learning. *Fourteenth International Conference on Learning Representations (ICLR 2026)*.
2. **Mnih, V., et al. (2015).** Human-level control through deep reinforcement learning. *Nature, 518*.

3. **Vincent, T., et al. (2025).** Iterated Q-Network: Beyond One-Step Bellman Updates in Deep Reinforcement Learning. *arXiv:2403.02107*.
4. **Gallici, M., et al. (2025).** Simplifying Deep Temporal Difference Learning. *ICLR 2025*.
5. **Bhatt, A., et al. (2024).** CrossQ: Batch Normalization in Deep Reinforcement Learning for Greater Sample Efficiency and Simplicity. *ICLR 2024*.
6. **Snell, C., et al. (2023).** Offline RL for Natural Language Generation with Implicit Language Q Learning. *ICLR 2023*.
7. **Elsayed, M., et al. (2024).** Streaming Deep Reinforcement Learning Finally Works. *arXiv:2410.10939*.
8. **Lee, H., et al. (2025).** Hyperspherical Normalization for Scalable Deep Reinforcement Learning. *arXiv:2502.15280*.

PAC4SAC: PAC-Bayesian Soft Actor-Critic Learning

AABI (2024) | *2nd Author*

Summary: 2-3x sample efficiency improvement on high-dimensional tasks (Ant), best cumulative regret across all four PyBullet environments. First actor-critic using a PAC-Bayesian generalization bound as the critic training objective. AABI 2024.

Links: - Paper - arXiv - Code - Scholar - View on Site

Introduction

No prior actor-critic algorithm had used a PAC-Bayesian generalization bound as its critic training objective — despite the theory offering exactly the worst-case guarantees that critic training needs. Standard actor-critic methods minimize plain Bellman error, which provides no guarantee on how well the critic generalizes to unseen states, allowing estimation errors to accumulate and destabilize training.

PAC4SAC (PAC-Bayes for Soft Actor-Critic) closes this gap by training a single randomized critic against a formal generalization bound, yielding three complementary properties in one loss: Bellman consistency, conservative value updates that eliminate overestimation without a second critic, and a principled exploration bonus that emerges from first principles rather than being added manually. A companion technique, **critic-guided multiple shooting**, leverages the randomized critic to search for better actions at interaction time, amplifying sample efficiency further.

In this work (2nd author), I co-designed the algorithm, designed and ran all experiments, and contributed substantially to the writing.

Problem Statement

- **Value overestimation bias:** When the same network computes both the prediction and its training target, the Q-learning update systematically overestimates values; even for zero-mean noise, the maximum of noisy estimates is larger than the true maximum (Thrun & Schwartz, 1993). Using two critics to counteract this introduces an underestimation bias instead (Fujimoto et al., TD3, 2018).
- **Catastrophic interference:** Updating the critic to fix poorly-estimated states inevitably degrades already-accurate estimates for other states. Polyak averaging (slow target updates) mitigates but does not eliminate this effect.
- **Sample inefficiency:** The combined effect of estimation errors and interference forces modern actor-critic methods (SAC, DDPG) to interact with the environment for many more steps than necessary before finding a good policy.
- **Gap:** No prior work uses a PAC-Bayesian bound as the critic training objective in an actor-critic algorithm, despite the theory offering exactly the kind of worst-case generalization guarantees that critic training needs.

Methodology

PAC4SAC makes two key contributions, illustrated in Figure 1.

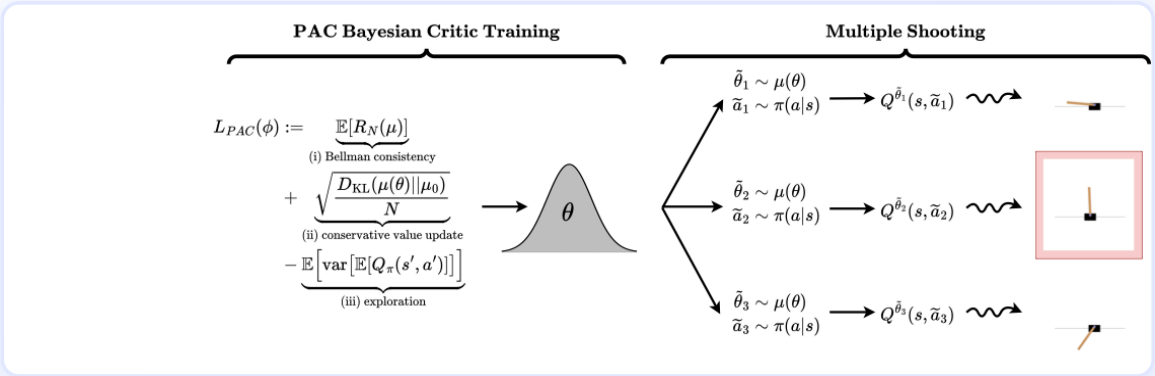


Figure 1. The PAC4SAC framework. Left: The PAC-Bayesian critic loss has three interpretable terms: Bellman consistency (accurate value estimation), conservative value update (KL regularization prevents overestimation), and an exploration bonus (variance of next-state value, derived from first principles). Training this loss yields a critic with Gaussian-distributed output layer weights, i.e., a distribution over Q-values. Right: Multiple shooting uses this randomized critic at action-selection time: S candidate actions are sampled from the stochastic actor, each evaluated by a fresh critic sample, and the action with the highest sampled Q-value is executed.

1: PAC-Bayesian Critic Training

PAC4SAC treats the critic as a **Bayesian posterior** over Q-functions: the final linear layer has normally distributed weights, making every forward pass a random sample from a distribution over value estimates.

The training objective minimizes a PAC-Bayesian generalization bound on Bellman error, adapted from McAllester (1999) and the RL-specific bound of Fard et al. (2012), producing three complementary terms:

$$\mathcal{L}_{\mathrm{PAC}}(\phi) := \underbrace{\mathbb{E}\left[R_N(\mu)\right]}_{\text{Bellman consistency}} + \underbrace{\sqrt{\frac{D_{\mathrm{KL}}(\mu \parallel \mu_0)}{N}}}_{\text{conservative value update}} - \xi \underbrace{\mathbb{E}\left[\mathrm{var}\left[\mathbb{E}\left[Q_\pi(s', a')\right]\right]\right]}_{\text{exploration}}$$

- **Bellman consistency** encourages accurate policy evaluation, as in standard SAC.
- **Conservative value update** (KL divergence between the learned posterior and a standard-normal prior) penalizes overconfident value estimates, replacing the need for a second critic to combat overestimation.
- **Exploration** maximizes the expected variance of the next-state value, promoting the agent to visit uncertain states. Crucially, this term emerges **from first principles** out of the PAC-Bayes bound, unlike the manually added entropy term in SAC.

The KL term and the variance term are both analytically tractable given a Gaussian penultimate layer, so no Monte Carlo sampling is needed during training.

2: Critic-Guided Optimal Action Search (Multiple Shooting)

At each environment interaction, the agent draws S candidate actions $a_1, \ldots, a_S$ from the stochastic actor and evaluates each with an independent sample from the critic distribution $Q^{\tilde{\theta}_i}(s, a_i)$. The action with the highest sampled Q-value is executed:

$$a_* = \arg \max_{i=1,\ldots,S} Q^{\tilde{\theta}_i}(s, a_i)$$

This **multiple shooting** strategy turns the critic's randomness into a directed search: the stochastic actor explores broadly, and the stochastic critic selects the most promising candidate without over-exploiting a potentially biased deterministic estimate. A convergence proof (Theorem 2 in the paper) shows that policy improvement holds for any sample count $S > 0$.

Implementation: PyTorch 1.13.1 · PyBullet Gymperium · Adam optimizer (lr = 0.001) · replay buffer of 25,000 · batch size 32 · $\xi = 0.01$ · $S = 500$ shooting samples.

Results

Evaluated on four continuous control tasks from the **PyBullet Gymperium** suite (Coumans & Bai, 2019; Benelot, 2018) with increasing difficulty: Cartpole Swingup ($d_s = 5, d_a = 1$), Half Cheetah ($d_s = 17, d_a = 6$), Ant ($d_s = 111, d_a = 11$), and Humanoid ($d_s = 376, d_a = 17$). Baselines: SAC (Haarnoja et al., 2018), DDPG (Lillicrap et al., 2015), OAC (Ciosek et al., 2019). All results averaged over 5 seeds.

Two metrics: (1) **Cumulative Regret**: total reward gap relative to a task-completion threshold, measuring how efficiently the agent solves the task; (2) **Episodes Until Task Solved**: how quickly the agent first crosses the threshold consistently, measuring sample efficiency.

Cumulative Regret ($\times 10^3$) — lower is better

Method	Cartpole Swingup	Half Cheetah	Ant	Humanoid
DDPG	7.2 ± 1.0	317.8 ± 24.0	210.8 ± 21.2	906.2 ± 7.8
SAC	6.5 ± 0.3	166.8 ± 10.4	165.5 ± 17.0	539.0 ± 20.0
OAC	22.7 ± 1.4	213.0 ± 15.5	443.8 ± 128.3	1223.7 ± 44.5
PAC4SAC (Ours)	**5.7 ± 0.3**	**132.8 ± 10.8**	**113.3 ± 10.9**	**528.8 ± 36.5**

Mean ± std over 5 seeds. Bold = lowest mean.

Episodes Until Task Solved — lower is better

Method	Cartpole (max 40)	Half Cheetah (max 250)	Ant (max 500)	Humanoid (max 500)
DDPG	34.2 ± 3.8	250.0 ± 0.0	298.6 ± 56.4	500.0 ± 0.0
SAC	24.4 ± 5.7	250.0 ± 0.0	302.0 ± 44.8	482.2 ± 15.9
OAC	40.0 ± 0.0	250.0 ± 0.0	315.0 ± 82.6	500.0 ± 0.0
PAC4SAC (Ours)	**22.0 ± 5.1**	**223.6 ± 14.6**	**146.4 ± 12.3**	**473.8 ± 18.5**

PAC4SAC achieves the best result on every task in both metrics. The improvement is most dramatic on **Ant**, where PAC4SAC solves the task in 146 episodes versus 298–443 for the baselines, a 2–3× sample efficiency gain.

Ablation: All Three Loss Terms Matter

Bellman	Conservative	Exploration	Cartpole Regret ($\times 10^3$)	Half Cheetah Regret ($\times 10^3$)
✓	×	×	6.5 ± 1.5	182.2 ± 14.4
✓	✓	×	5.9 ± 0.3	141.6 ± 21.5
✓	×	✓	7.1 ± 1.0	153.0 ± 17.3
✓	✓	✓	**5.7 ± 0.3**	**132.8 ± 10.8**

All three terms are required to minimize cumulative regret. The conservative update alone reduces regret substantially; combining it with the exploration bonus reaches the lowest regret on both tasks. The ablation on multiple shooting confirms the trend: more actor samples S monotonically reduce cumulative regret and improve sample efficiency.

Conclusion

- **First PAC-Bayesian actor-critic:** PAC4SAC is the first algorithm to use a PAC-Bayesian generalization bound as a critic training objective, combining Bellman consistency, conservative value updates, and a principled exploration bonus in a single loss.
- **Single critic suffices:** The KL regularization term replaces the standard practice of training two critics to counter overestimation bias, simplifying the architecture without sacrificing (in fact, improving) performance.
- **Multiple shooting pays off:** Critic-guided random action search at interaction time consistently reduces regret and improves sample efficiency, with gains increasing monotonically with the number of samples.
- **Consistent improvement across difficulty:** PAC4SAC outperforms SAC, DDPG, and OAC on all four tasks in both cumulative regret and sample efficiency, with the largest gains on the high-dimensional Ant and Half Cheetah environments.
- **Theory-grounded exploration:** The variance-based exploration bonus arises from the PAC-Bayes derivation itself rather than being added manually, providing a principled alternative to entropy regularization.

References

1. **Tasdighi, B., Akgül, A., Haußmann, M., Brink, K. K., & Kandemir, M. (2024).** PAC-Bayesian Soft Actor-Critic Learning. *Sixth Symposium on Advances in Approximate Bayesian Inference (AABI 2024)*.

2. **Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018).** Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ICML 2018*.
3. **Fujimoto, S., van Hoof, H., & Meger, D. (2018).** Addressing Function Approximation Error in Actor-Critic Methods (TD3). *ICML 2018*.
4. **Lillicrap, T. P., et al. (2015).** Continuous Control with Deep Reinforcement Learning (DDPG). *ICLR 2016*.
5. **Ciosek, K., Vuong, Q., Loftin, R., & Hofmann, K. (2019).** Better Exploration with Optimistic Actor Critic (OAC). *NeurIPS 2019*.
6. **Fard, M. M., Pineau, J., & Szepesvári, C. (2012).** PAC-Bayesian Policy Evaluation for Reinforcement Learning. *arXiv:1206.1839*.
7. **McAllester, D. A. (1999).** PAC-Bayesian Model Averaging. *COLT 1999*.

ObjectRL: An Object-Oriented Reinforcement Learning Codebase

arXiv (2025) | *2nd Author*

Summary: Extending SAC to a new algorithm takes roughly 5 lines: just override the two methods that change. Full OOP codebase where encapsulation, inheritance, and polymorphism map directly to RL algorithm components. arXiv 2025.

Links: - Paper - arXiv - Code - Scholar - View on Site

Introduction

Deep reinforcement learning (RL) research depends on codebases that are fast to prototype in, easy to debug, and straightforward to extend. Existing libraries, while powerful, tend to prioritize scalability or off-the-shelf usability over research flexibility: tightly-coupled architectures, deep functional abstractions, and complex configuration systems make it hard to isolate and modify individual algorithmic components. **ObjectRL** is the first deep RL codebase built from the ground up on **Object-Oriented Programming (OOP)** principles, organizing every element of a modern RL algorithm into a clear class hierarchy that mirrors the way researchers think about the problem: agents, actors, critics, replay buffers, and update rules as distinct, composable entities.

Problem Statement

- **Tightly coupled architectures** in existing codebases (RLlib [Liang et al., ICML 2018], Pearl [Zhu et al., JMLR 2024], skrl [Serrano-Muñoz et al., JMLR 2023]) target production robustness and large-

scale deployment, but their complex APIs hinder rapid prototyping and low-level access.

- **Monolithic single-file designs** (CleanRL [Huang et al., JMLR 2022]) improve readability but sacrifice modularity; everything lives in one file, making component-level reuse impossible.
- **Unstructured APIs** (Stable-Baselines3 [[Raffin et al., JMLR 2021]]) provide stable implementations but navigation and customization are non-trivial.
- **Deep abstraction layers** (Tianshou [Weng et al., JMLR 2022], TorchRL [Bou et al., 2023]) can obscure the connection between code and algorithm, making intuitive prototyping difficult.
- **Partial OOP use** (MushroomRL [D'Eramo et al., JMLR 2021]) leaves algorithmic exploration less straightforward than a fully OOP-structured alternative.
- **Gap:** No existing codebase fully applies OOP principles (encapsulation, inheritance, composition, and polymorphism) at the level of individual algorithmic components, limiting how easily researchers can swap, extend, or debug a single part without touching the rest.

Methodology

ObjectRL is organized around three core OOP principles applied directly to RL algorithm design:

- **Encapsulation:** Each RL component (agent, critic, actor, replay buffer, logger) is a self-contained class. Attributes describe RL concepts; methods reflect their interactions. Modifications to one class do not propagate unexpectedly.
- **Inheritance and Composition:** A base `Agent` class defines common utilities (replay buffer, logger, training loop). `ActorCritic` inherits from it and defines the actor-critic structure. Algorithm-specific classes (e.g., `SoftActorCritic`, `TD3`) then inherit from `ActorCritic`, specializing only what changes. Actors and critics are composed into agents.
- **Polymorphism:** Shared interfaces (e.g., `update()`, `loss()`, `get_bellman_target()`) allow different actor/critic types to be swapped without modifying the surrounding training loop.

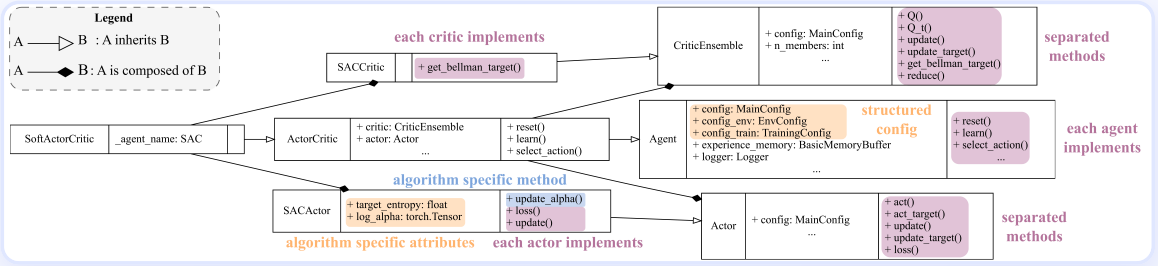


Figure 1. Class diagram of Soft Actor-Critic (SAC) in ObjectRL. Inheritance (arrows) and composition (diamonds) map directly to how RL researchers conceptualize the algorithm: SoftActorCritic is composed of a SACActor and a SACCritic (itself a CriticEnsemble); both inherit from shared base classes. Key attributes and methods are color-coded; algorithm-specific elements (orange) are clearly separated from shared infrastructure (grey).

Module structure: `agents` — base agent classes; `models` — actors, critics, their compositions, and algorithm implementations; `config` — hyperparameters via Python dataclasses; `experiments` — training and evaluation loops; `loggers` — result tracking; `nets` — network architectures; `replay buffers` — experience storage; `utils` — helper functions.

Implemented algorithms: DQN [Mnih et al., Nature 2015], DDPG [Lillicrap et al., ICLR 2016], PPO [Schulman et al., 2017], TD3 [Fujimoto et al., ICML 2018], SAC [Haarnoja et al., ICML 2018], OAC [Ciosek et al., NeurIPS 2019], REDQ [Chen et al., ICLR 2021], DRND [Yang et al., ICML 2024], PBAC [Tasdighi et al., 2024].

Prototyping example: SAC → DRND in two steps. DRND augments SAC with an exploration bonus derived from ensemble disagreement over Q-values. In ObjectRL: 1. Define `DRNDBonus` — a class encapsulating the uncertainty estimation and its hyperparameters. 2. Create `DRNDActor` and `DRNDCritic` — subclasses of `SACActor` and `SACCritic` that override just `loss()` and `get_bellman_target()` to add the bonus term (highlighted below):

```
# DRNDActor.loss() - adds bonus to the standard entropy-regularized objective
loss, act_dict = super().loss(state, critics)
bonus = bonus_ensemble.bonus(state, action).mean()
return loss + bonus, act_dict

# DRNDCritic.get_bellman_target() - injects bonus into the Bellman backup
bonus = bonus_ensemble.bonus(next_state, next_action)
q_target = target_reduced - alpha * log_prob - self.lambda_critic * bonus
return q_target
```

The existing training loop requires no changes; polymorphism handles the dispatch automatically. Full documentation and additional examples at objectrl.readthedocs.io.

Results

All implemented algorithms are evaluated on **five standard MuJoCo continuous control environments** (Ant, HalfCheetah, Hopper, Humanoid, and Walker2D) using the Gymnasium interface [Towers et al., 2024], with 5 seeds per algorithm.

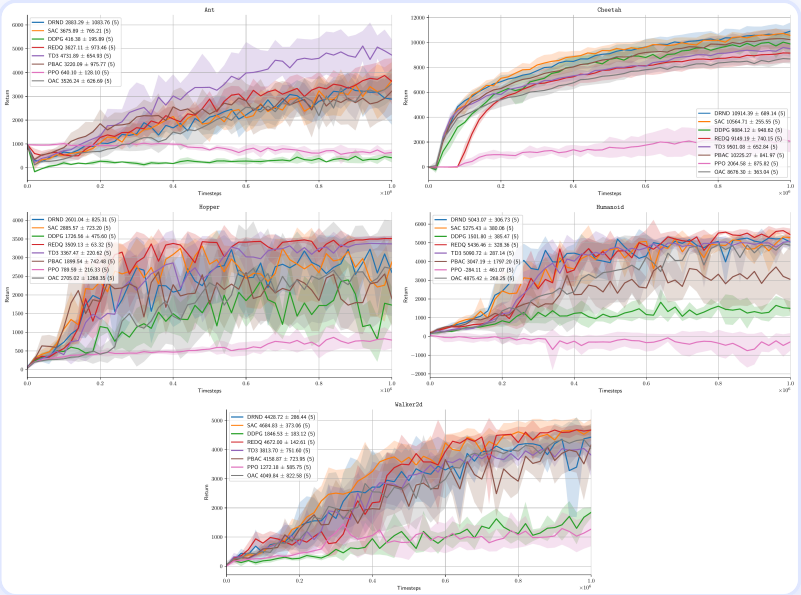


Figure 2. Learning curves for all nine implemented algorithms across five MuJoCo environments (mean ± std, 5 seeds). Top row: Ant and HalfCheetah; middle row: Hopper and Humanoid; bottom: Walker2D. Exploration-augmented methods (OAC, DRND, PBAC) lead on challenging tasks like Ant and Humanoid. All implementations reproduce expected relative performance orderings from the published literature, confirming correctness of the codebase.

The primary purpose of the benchmark is correctness verification: that each algorithm's implementation within ObjectRL's OOP framework reproduces the relative performance ordering and scale reported in the original papers. As one example, on Ant (mean ± std, 5 seeds): TD3 leads at 4,732 ± 655, followed by SAC (3,676 ± 765), OAC (3,526 ± 527), REDQ (3,607 ± 973), and PBAC (3,220 ± 976), while PPO and DDPG score lower as expected for this high-DoF task. All algorithms integrate within the same training infrastructure with no ad-hoc modifications to the loop.

Conclusion

- **First fully OOP deep RL codebase:** ObjectRL applies encapsulation, inheritance, composition, and polymorphism at the level of individual algorithmic components, not just the training infrastructure.
- **Rapid prototyping with minimal changes:** Extending SAC to DRND requires overriding two methods (~5 lines total); the rest of the training loop, logging, and evaluation machinery is inherited unchanged.
- **Readable, debuggable structure:** The class hierarchy mirrors the conceptual building blocks researchers use, making it straightforward to locate, understand, and modify any part of an algorithm.
- **Nine algorithms, five environments:** Clean, verified implementations of DQN, DDPG, PPO, TD3, SAC, OAC, REDQ, DRND, and PBAC, all benchmarked on standard MuJoCo tasks.
- **Open source:** Code at github.com/adinlab/objectrl · Documentation at objectrl.readthedocs.io

References

1. **Baykal, G., Akgül, A., Haußmann, M., Tasdighi, B., Werge, N., Wu, Y.S., & Kandemir, M. (2025).** ObjectRL: An Object-Oriented Reinforcement Learning Codebase. *arXiv:2507.03487*. objectrl.readthedocs.io
2. **Mnih, V., et al. (2015).** Human-level control through deep reinforcement learning. *Nature*.
3. **Lillicrap, T.P., et al. (2016).** Continuous control with deep reinforcement learning. *ICLR 2016*.
4. **Haarnoja, T., et al. (2018).** Soft Actor-Critic. *ICML 2018*.
5. **Fujimoto, S., et al. (2018).** Addressing Function Approximation Error in Actor-Critic Methods. *ICML 2018*.
6. **Schulman, J., et al. (2017).** Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.
7. **Chen, X., et al. (2021).** Randomized Ensembled Double Q-Learning. *ICLR 2021*.
8. **Ciosek, K., et al. (2019).** Better Exploration with Optimistic Actor Critic. *NeurIPS 2019*.
9. **Yang, K., et al. (2024).** Exploration and Anti-Exploration with Distributional Random Network Distillation. *ICML 2024*.
10. **Tasdighi, B., et al. (2024).** Deep Exploration with PAC-Bayes. *arXiv:2402.03055*.
11. **Raffin, A., et al. (2021).** Stable-Baselines3: Reliable Reinforcement Learning Implementations. *JMLR*.
12. **Huang, S., et al. (2022).** CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *JMLR*.
13. **Todorov, E., et al. (2012).** MuJoCo: A physics engine for model-based control. *IROS 2012*.

BFL: Aggregating Variational Bayesian Networks in Federated Learning

NeurIPS 2022 Workshop (2022) | *3rd Author*

Summary: Low-spread aggregation rules match or beat deterministic FedAvg while providing significantly better calibration across clients. First systematic study of aggregation strategies for Variational Bayesian Neural Networks in federated learning. NeurIPS 2022 Workshop.

Links: - Paper - arXiv - Code - Scholar - View on Site

Introduction

Federated Learning (FL) enables multiple data centers or devices to collaboratively train a shared model **without sharing raw data**, making it essential for privacy-sensitive applications such as healthcare and finance. While most FL methods use deterministic neural networks, real-world safety-critical deployments require not only accurate predictions but also calibrated **uncertainty estimates**. Variational Bayesian Neural Networks (VBNNs) provide this capability by treating network weights as probability distributions rather than fixed values. However, the standard FL aggregation strategy (FedAvg) was designed for point-estimate weights and cannot be directly applied to distributional weights. **BFL** (Bayesian Federated Learning) is the first systematic empirical survey of statistical aggregation rules for VBNNs in the federated setting, identifying when and why different strategies succeed or fail.

In this work (3rd author), I supervised two undergraduate students on implementation, structured the survey, and led the writing and editing of the manuscript.

Problem Statement

- **FedAvg and its deterministic variants** aggregate model parameters by (weighted) averaging scalars. When model weights are Gaussian distributions, a naive extension is ambiguous: how should two Gaussian distributions be merged into one global distribution?
- **Different aggregation rules yield fundamentally different distributions.** As illustrated in Figure 1, combining two client distributions with different rules can produce global distributions with very different means and variances, making the choice of aggregation rule a design decision with significant empirical consequences.
- **No prior work** had systematically investigated the effect of the aggregation rule choice on VBNNs trained with Variational Inference in a federated setting; existing probabilistic FL methods (FedPA, pFedBayes, FedSparse) address different concerns.
- **Gap:** It is unknown which statistical properties of the aggregated distribution (its variance, in particular) matter most for accuracy, calibration, and uncertainty quantification.

Methodology

Five aggregation rules are derived and evaluated, all operating on the Gaussian weight distributions (parameterized by mean μ and variance σ^2) of each client's VBNN:

Rule	Aggregated Variance	Key Property
EAA — Empirical Arithmetic Aggregation	Weighted average of σ^2	Naive extension of FedAvg; high spread
GAA — Gaussian Arithmetic Aggregation	Weighted average of $\sigma^2 \times \beta_k$	Uses Gaussian sum rule; $\sigma^2_{GAA} < \sigma^2_{EAA}$ always
AALV — Arithmetic Aggregation with Log Variance	Geometric mean of σ^2	Equivalent to FedAvg gradient averaging for log σ^2
PPA — Population Pooling Based Aggregation	Empirical variance of pooled samples	Sampling-based; computationally heavier
CF — Conflation Aggregation	Precision-weighted combination	Product of distributions; tends toward low spread

$$\mu_{CF} = \frac{\sum_k \beta_k \mu_k / \sigma_k^2}{\sum_k \beta_k / \sigma_k^2}, \qquad \sigma^2_{CF} = \frac{\beta_{\max}}{\sum_k \beta_k / \sigma_k^2}$$

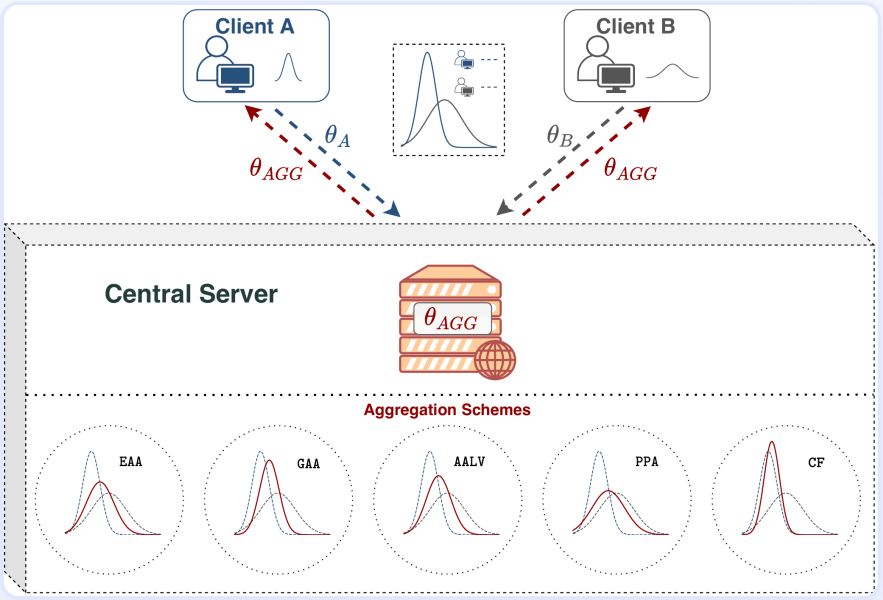


Figure 1. Illustration of the five aggregation rules applied to two client weight distributions (depicted as Gaussians). After local training, clients send their weight distributions to the server; the server applies the aggregation rule and returns the resulting global distribution. Different rules produce substantially different distributions, varying in both mean and variance, which motivates a principled comparison.

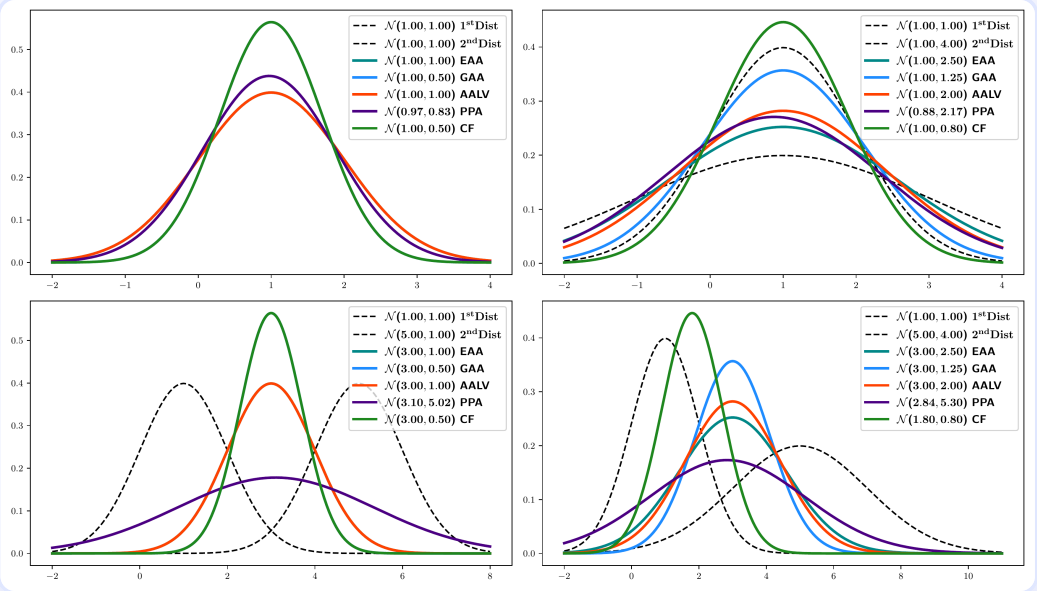


Figure 2. Four scenarios comparing the five aggregation rules on pairs of Gaussian client distributions with varying means and variances (shown as dashed black curves). EAA (orange) and PPA (dark blue) consistently produce wider aggregated distributions; GAA (purple), AALV (cyan), and CF (green) maintain tighter, lower-variance results. This spread difference, which seems small in toy examples, compounds dramatically when aggregating across hundreds of clients.

Two federation frameworks are considered: **FVBA** (equal client weights, $\beta_k = 1/K$) and **FVBWA** (data-size-weighted, $\beta_k = |D_k|/|D|$). Deterministic baselines FED (uniform weights) and FEDAVG (size-weighted) serve as comparisons.

Architecture: Convolutional network (two 5×5 conv layers + three linear layers); variational Bayesian layers replace deterministic linear layers, parameterizing each weight as $(\mu, \log \sigma^2)$.

Datasets: Three image classification benchmarks, each with 10 labels: Fashion-MNIST (FMNIST, 60K train), CIFAR-10 (50K train), and SVHN (73K train).

Experiments: Two client settings (10 active clients vs. 100 total / 10 active) × two data partitions (IID and non-IID via Dirichlet distribution), 5 seeds. Evaluation uses accuracy (Acc ↑), Expected Calibration Error (ECE ↓), and Negative Log-Likelihood (NLL ↓).

Results

10-client experiment — all aggregation rules viable. With a coarser data split (fewer clients), high-spread methods remain competitive. VBNNs with lower spread (GAA, AALV, CF) match or exceed deterministic baselines on accuracy while providing substantially better calibration:

Setting	FED (Acc/ECE)	FVBA + GAA (Acc/ECE)	FVBA + CF (Acc/ECE)
FMNIST — IID	89.62 / 7.99	89.82 / 6.38	89.90 / 6.35
Cifar-10 — IID	70.09 / 3.35	**71.29** / **3.11**	**71.17** / **2.81**
SVHN — non-IID	86.65 / 9.88	87.52 / 6.24	87.60 / 6.28

Lower ECE is better. Bold indicates best-performing models within standard error.

100-client experiment — spread becomes critical. When data is split across 100 clients (each client has far fewer samples), high-spread aggregations (EAA, PPA) collapse entirely, while low-spread methods (GAA, AALV, CF) continue to outperform deterministic baselines:

Setting	FED	FVBA + EAA	FVBA + GAA	FVBA + AALV
Cifar-10 — IID (Acc)	64.40	45.10	**66.97**	**67.45**
SVHN — IID (Acc)	87.20	79.51	**90.16**	**90.10**
Cifar-10 — non-IID (Acc)	**61.23**	26.87	**60.85**	**60.60**
SVHN — non-IID (Acc)	85.00	67.47	**87.34**	**87.46**

EAA collapses under 100-client non-IID (26.87% on Cifar-10 vs. 60.85% for GAA). GAA/AALV/CF match or surpass deterministic baselines while offering better calibration.

Key finding — degree of spread dominates. Across all settings, the empirical standard deviation of the final aggregated model is the most consistent predictor of performance: high-spread methods (EAA, PPA) produce overconfident, poorly calibrated models under 100 clients, while low-spread methods (GAA, AALV, CF) learn stably. No single low-spread rule consistently beats the others in all scenarios.

Calibration advantage of VBNNs. When accuracy is competitive, VBNNs universally provide lower ECE and NLL than deterministic counterparts, confirming that probabilistic FL models are better suited for uncertainty-sensitive downstream use.

Conclusion

- **First systematic study:** BFL is the first work to empirically investigate the effect of statistical aggregation rules on VBNNs in federated learning, filling a gap left by prior probabilistic FL methods that focus on convergence guarantees or personalization.
- **Spread is the dominant factor:** Aggregation methods that keep the variance of the global distribution low (GAA, AALV, CF) are consistently more stable, especially as the number of clients grows and local datasets shrink.

- **High-spread methods collapse at scale:** EAA and PPA maintain performance with 10 clients but degrade sharply under 100 clients, particularly in non-IID settings where local distributions are highly heterogeneous.
- **Better calibration without sacrificing accuracy:** Low-spread VBNN aggregations match or outperform deterministic FedAvg on accuracy while providing significantly lower calibration error (ECE) and negative log-likelihood.
- **Reproducible pipeline:** A parallelized multi-process simulation framework (BFL-P) is released, substantially reducing wall-clock training time and enabling full reproducibility.

References

1. **Ozer, A., Buldu, K.B., Akgül, A., & Unal, G. (2022).** How to Combine Variational Bayesian Networks in Federated Learning. *NeurIPS 2022*. github.com/ituvisionlab/BFL-P
2. **McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B.A.Y. (2017).** Communication-Efficient Learning of Deep Networks from Decentralized Data. *AISTATS 2017*.
3. **Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020).** Federated Optimization in Heterogeneous Networks. *MLSys 2020*.
4. **Zhang, X., Li, Y., Li, W., Guo, K., & Shao, Y. (2022).** Personalized Federated Learning via Variational Bayesian Inference. *ICML 2022*.
5. **Al-Shedivat, M., Gillenwater, J., Xing, E., & Rostamizadeh, A. (2021).** Federated Learning via Posterior Averaging. *ICLR 2021*.
6. **Hill, T.P. (2008).** Conflations of Probability Distributions. *Transactions of the American Mathematical Society*.
7. **Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015).** Weight Uncertainty in Neural Networks. *ICML 2015*.

Thesis

Probabilistic Methods for Sample-Efficient Reinforcement Learning

Ph.D. Thesis (2026) | *First Author*

Summary: Doctoral thesis presenting six peer-reviewed algorithms at NeurIPS, ICML, ICLR, TMLR, and UAI, unified by one claim: probabilistic uncertainty representations make reinforcement learning agents faster, more adaptive, and more data-efficient.

Links: - Paper - Scholar - View on Site

Introduction

Reinforcement learning (RL) is a learning paradigm in which an agent maximizes cumulative reward through interaction with an environment, learning via trial and error without labeled supervision. This framework underpins applications in robotics, autonomous systems, recommendation engines, and game playing.

Yet every RL interaction carries a cost: hardware wear, safety risk, time, or data budget. **Sample efficiency** (learning effective policies from as few interactions as possible) is a foundational challenge that cuts across all RL settings. This thesis addresses it through a single research question:

> *How can probabilistic modeling be used to improve sample efficiency in reinforcement learning across diverse learning settings?*

The answer is six published methods, spanning three learning paradigms, that collectively show how representing and acting on uncertainty leads to faster, more robust learning.

Problem Statement

RL manifests in three settings, each with distinct obstacles to sample efficiency:

- **Online RL**: the agent interacts with a stationary environment. Standard approaches suffer from overestimation bias in value estimates, error propagation through bootstrapping, and lack of principled exploration strategies.
- **Non-stationary RL**: environment dynamics change over time (e.g., a robot whose joints gradually degrade). Agents must detect the shift, preserve plasticity (the ability to keep learning), and re-explore efficiently without the option of a full reset.
- **Offline RL**: no environment interaction is possible; learning proceeds entirely from a fixed, pre-collected dataset. The core danger is distributional shift: value estimates for actions underrepresented in the data become inflated, with no corrective feedback from the environment.

Across all three settings, the shared gap is the same: standard methods do not represent uncertainty explicitly, missing the principal mechanism for learning faster and more reliably.

Methodology

Six methods are organized under three research objectives.

Objective 1: Sample-Efficient Value Estimation

Inaccurate value functions slow policy improvement by propagating estimation errors through bootstrapping. Three complementary methods address this:

- **PAC4SAC**: augments Soft Actor-Critic with a PAC-Bayesian critic objective that bounds generalization error directly. Uncertainty estimates from the stochastic critic guide directed exploration via multiple shooting. *(AABI 2024)*
- **iS-QL**: bridges the target-free/target-based dichotomy by sharing all network parameters except the final output layer, and learning K+1 parallel Bellman iterations simultaneously. Achieves target-based stability at near target-free memory cost across online control, offline control, and language modeling. *(ICLR 2026)*
- **DAIF**: formulates quantile regression as Bayesian quantile regression under a Normal-Inverse-Gamma (NIG) generative model. Each (state, action, quantile) triple maps to a full distribution over the return, capturing both aleatoric and epistemic uncertainty. Exploration follows from minimizing Expected Free Energy, with no learned dynamics model required. *(ICML 2026)*

Objective 2: Sample-Efficient Adaptation to Non-Stationary Environments

When dynamics shift, agents must rapidly detect the change and re-adapt their policies. Two probabilistic methods enable this:

- **EPPO**: replaces the scalar value head in Proximal Policy Optimization with an evidential (NIG) critic, providing closed-form estimates of both aleatoric and epistemic uncertainty. Epistemic uncertainty identifies distribution shifts, preserving plasticity via hyperprior regularization; uncertainty propagates through the Generalized Advantage Estimator as a UCB bonus for directed exploration. *(TMLR 2025)*

- **WSB**: introduces Weighted Sequential Bayesian inference for non-stationary linear contextual bandits. A Bayesian posterior over drifting reward parameters is maintained through exponentially weighted updates. Novel concentration inequalities explicitly account for prior beliefs and their decay over time, yielding provably efficient algorithms (WSB-LinUCB, WSB-RandLinUCB, WSB-LinTS) that match or improve on frequentist baselines in regret. *(UAI 2026)*

Objective 3: Sample-Efficient Learning from Fixed Datasets

With no access to new interactions, every data point must be used as effectively as possible:

- **MOMBO**: identifies high Monte Carlo variance as the primary source of training instability in model-based offline RL. Replaces single-sample Bellman targets with deterministic moment matching: next-state distributions from an ensemble dynamics model are propagated analytically through the Q-network, yielding a closed-form pessimistic Bellman operator. Two theoretical contributions accompany the algorithm: a suboptimality bound for sampling-based approaches showing explicit dependence on the number of samples, and a tighter deterministic bound for the moment-matching counterpart. *(NeurIPS 2024)*

Results

Evaluated across MuJoCo, PyBullet, DeepMind Control Suite, EvoGym, Atari, and D4RL benchmarks, the six methods consistently improve sample efficiency over their respective baselines:

- **MOMBO** achieves state-of-the-art or competitive normalized reward and AULC on D4RL offline benchmarks across halfcheetah, hopper, and walker2d environments, and provides a provably tighter suboptimality guarantee than sampling-based offline RL methods.
- **EPPO** ranks first in average AULC among seven baselines on non-stationary MuJoCo locomotion (average rank 1.5), outperforming methods designed for plasticity alone, exploration alone, and standard PPO, showing that both properties are simultaneously necessary.
- **DAIF** achieves the best average task ranking across 19 continuous control tasks on three benchmark suites, with improvements of up to +62% AULC over the next-best distributional baseline on individual tasks, and competitive performance on pixel-observation tasks without pixel-specialist tuning.
- **PAC4SAC** achieves the lowest cumulative regret and fewest episodes to task completion across four PyBullet continuous control environments compared to DDPG, SAC, and OAC baselines.
- **WSB** yields provably lower cumulative regret than frequentist non-stationary bandit methods, validated empirically across multiple non-stationary configurations.
- **iS-QL** matches target-based stability while using near target-free memory, with gains across Atari, DMControl, and offline language modeling tasks.

Conclusion

This thesis demonstrates that probabilistic modeling is an effective and general mechanism for sample-efficient reinforcement learning. Across settings as different as online control, continual adaptation, and offline policy learning, probabilistic frameworks enable agents to:

- **Learn faster** by grounding exploration in principled uncertainty estimates rather than heuristics.
- **Adapt more reliably** under changing dynamics by detecting distributional shifts in value estimates and preserving plasticity.
- **Extract more from limited data** by replacing high-variance sampling with analytical uncertainty propagation.

Spanning PAC-Bayesian bounds, evidential learning, Bayesian inference, distributional representations, deterministic moment matching, and parameter sharing, the six contributions form a coherent program: treating uncertainty not as a nuisance to be minimized, but as the primary signal for efficient learning.

References

1. **Akgül, A. (2026).** Probabilistic Reinforcement Learning for Sample-Efficient Control. *PhD Thesis, University of Southern Denmark*.
2. **Akgül, A., Haußmann, M., & Kandemir, M. (2024).** Deterministic Uncertainty Propagation for Improved Model-Based Offline Reinforcement Learning. *NeurIPS 2024*.
3. **Akgül, A., Baykal, G., Haußmann, M., & Kandemir, M. (2025).** Overcoming Non-stationary Dynamics with Evidential Proximal Policy Optimization. *TMLR 2025*.
4. **Akgül, A., Baykal, G., Haußmann, M., Çelikok, M. M., & Kandemir, M. (2026).** Distributional Active Inference. *ICML 2026*.
5. **Tasdighi, B., Akgül, A., Haußmann, M., Brink, K. K., & Kandemir, M. (2024).** PAC-Bayesian Soft Actor-Critic Learning. *AABI 2024*.
6. **Werge, N., Wu, Y.S., Akgül, A., & Kandemir, M. (2026).** Weighted Sequential Bayesian Inference for Non-Stationary Linear Contextual Bandits. *Conference on Uncertainty in Artificial Intelligence (UAI) 2026*.
7. **Vincent, T., Tripathi, Y., Faust, T., Akgül, A., Oren, Y., Kandemir, M., Peters, J., & D'Eramo, C. (2026).** Bridging the Performance Gap between Target-free and Target-based Reinforcement Learning. *ICLR 2026*.

Memory-based Approaches to Problems in Probabilistic

Modeling

Master's Thesis (2022) | *First Author*

Summary: Master's thesis at Istanbul Technical University demonstrating that external memory solves two open problems in probabilistic ML: total calibration of neural networks (ETP, ICLR 2022) and continual learning of multi-modal dynamical systems (CDDP, L4DC 2024).

Links: - Paper - Scholar - View on Site

Introduction

External memory (an explicit, addressable store of information that a neural network can read from and write to) has proven effective in many machine learning settings, from question answering to meta-learning. This thesis asks whether external memory can serve as a general-purpose mechanism for solving open problems in **probabilistic modeling**: the branch of machine learning concerned with representing and reasoning under uncertainty.

The answer is yes. Two distinct, previously unsolved problems in probabilistic modeling are addressed: one in uncertainty quantification, one in continual learning. External memory turns out to be the key ingredient in both.

Problem Statement

Problem 1: Total Calibration of Neural Networks

Deploying a probabilistic neural network in a safety-critical domain (medical diagnostics, autonomous driving) requires it to satisfy three properties at the same time:

- **Model fit** — the model accurately captures the in-domain data distribution
- **Class overlap calibration** — predicted probabilities faithfully reflect genuine ambiguity at class boundaries
- **Out-of-distribution (OOD) detection** — the model reliably flags inputs that fall outside the training domain

Prior work treated these as separate problems: Bayesian Neural Networks (BNNs) handle OOD detection well but not class calibration; Evidential Deep Learning (EDL) handles class calibration but has no global signal for OOD detection. No single model, and no unified formal framework, existed for achieving all three simultaneously. This is the problem of **total calibration**.

Problem 2: Continual Learning of Multi-modal Dynamical Systems

Probabilistic State-Space Models (SSMs) are the gold standard for dynamics modeling, with applications in weather forecasting, robotics, and stochastic optimal control. A key open challenge is **continual learning (CL)**: a model must learn new behavioral modes of a dynamical system one task at a time, without forgetting the modes it has already learned. The standard CL fix of transferring model parameters

from one task to the next (as in Variational Continual Learning) fails for multi-modal dynamics because different modes can have fundamentally different transition structures that cannot coexist in the same parameter space. Prior work on CL had not studied this setting at all.

Methodology

Both contributions adopt external memory as their core mechanism, in different probabilistic modeling contexts.

ETP — Evidential Turing Processes (for Total Calibration)

The key insight is that total calibration requires *two* kinds of uncertainty to coexist: a **global** uncertainty that shrinks with more data (handling OOD detection) and a **local**, per-input uncertainty that captures class overlap. This motivates:

- **Complete Bayesian Models (CBM):** A new theoretical framework that combines a global parameter θ (from BNNs) and a local class-probability variable π (from EDL). A variance decomposition shows that a CBM is the *minimal* structure necessary to represent all three calibration components simultaneously.
- **Turing Process:** ETP instantiates CBM using a neural episodic memory: a set of learnable memory slots updated during training via an explicit write rule. At inference, input queries retrieve relevant uncertainty information from memory via attention, without needing a held-out context set. This memory acts as the mechanism that makes the global uncertainty signal accurate and data-driven.

CDDP — Continual Dynamic Dirichlet Process (for Continual Learning of Dynamics)

Instead of transferring model parameters between tasks (which overwrites old knowledge), CDDP stores a compact **mode descriptor** per dynamical mode in an external memory and retrieves it on demand:

- **Bayesian State-Space Model (BSSM)** provides the probabilistic dynamics backbone.
- **Neural episodic memory with a Dirichlet Process prior** on attention weights stores one descriptor per discovered mode. The DP prior encourages efficient slot usage and supports automatic mode discovery without explicit mode labels.
- **Cross-task transfer via retrieval:** When a new mode is encountered, similar past-mode descriptors are retrieved from memory and fed into the transition kernel as an additional input, reusing past knowledge without modifying earlier representations.

A competitive VCL-based baseline is curated from scratch, as no prior baseline existed for this new problem.

Results

ETP — Total Calibration:

Evaluated across five real-world benchmarks (IMDB text, Fashion MNIST, SVHN, CIFAR10, CIFAR100) with 10 random seeds. Four metrics are reported: test error (model fit), ECE (class overlap), NLL (unified model fit), and AUROC (OOD detection).

- ETP achieves the **best NLL on all five datasets**, a result no other method matches.
- ETP is the **only model** that consistently ranks among top performers on all three calibration axes; every baseline collapses on at least one metric (EDL on IMDB accuracy, ENP on IMDB accuracy, BNN on ECE across all datasets).
- The advantage holds under **19 corruption types at 5 severity levels**, confirming that external memory improves the *structure* of uncertainty, not just performance on clean data.

CDDP — Continual Multi-modal Dynamics:

Evaluated on synthetic and adapted real-world time-series datasets under sequential task arrivals. Measured by Normalized Mean Squared Error and NLL.

- CDDP **compares favorably** to the VCL parameter-transfer baseline across all settings, showing that memory-based knowledge transfer is a more effective strategy than parameter reuse for multi-modal dynamics.

Conclusion

The central finding of this thesis is that **external memory is highly beneficial for problems of probabilistic modeling**:

- For **uncertainty quantification**, memory provides the global signal needed to simultaneously fit in-domain data, calibrate class probabilities, and detect out-of-distribution inputs. No prior method could achieve this without memory.
- For **continual learning of dynamics**, memory enables cross-task knowledge transfer without parameter overwriting, solving catastrophic forgetting in a setting (multi-modal dynamical systems) where the standard parameter-transfer remedy fails.

Together, ETP and CDDP establish external memory as a principled, broadly applicable tool in the probabilistic modeling toolkit, effective wherever a model must accumulate and selectively reuse structured knowledge about its domain.

References

1. **Akgül, A. (2022).** Memory-based Approaches to Problems in Probabilistic Modeling. *Master's Thesis, Istanbul Technical University.*
2. **Kandemir, M., Akgül, A., Haußmann, M., & Ünal, G. (2022).** Evidential Turing Processes. *International Conference on Learning Representations (ICLR 2022).*

3. **Akgül, A., Ünal, G., & Kandemir, M. (2024).** Continual Learning of Multi-modal Dynamics with External Memory. *Learning for Dynamics and Control Conference (L4DC 2024)*.

Industry Experience

Signature Verification for Fraud Detection

Vakifbank (2020) | *ML Engineer*

Summary: Siamese CNN trained on handwritten signatures deployed at Vakifbank R&D for cheque fraud detection — 95% accuracy on internal data, 88% on the public CEDAR benchmark.

Overview

Handwritten signature verification is a key fraud prevention mechanism in banking: detecting whether a signature on a cheque or document matches the account holder's reference signature. Classical rule-based systems are brittle under natural signing variability; the challenge is distinguishing genuine intra-personal variation from skilled forgeries.

This project, completed as a part-time ML engineer in Vakifbank's R&D and Innovation department, delivered a deep learning–based signature verification system from data pipeline through deployment evaluation.

Approach

Model: A **Siamese Convolutional Neural Network** — two weight-sharing CNN branches that each encode a signature image into a fixed-length embedding, with a contrastive loss that pulls genuine-pair embeddings together and pushes forged-pair embeddings apart.

Why Siamese networks: Traditional classifiers require retraining when a new account holder is enrolled. A Siamese network learns a similarity metric that generalizes to previously unseen signers at test time, making it practical for a banking system where the customer base grows continuously.

Pipeline: - Preprocessing: binarization, noise removal, and size normalization of raw signature scans - Training: contrastive loss with margin, Adam optimizer - Inference: threshold on embedding distance to classify genuine vs. forged

Results

Dataset	Accuracy
Vakifbank internal test set	**95%**
CEDAR public benchmark	**88%**

The CEDAR benchmark (Cherry, Eaddy, and Dupont Advanced Research) is a standard public dataset for offline handwritten signature verification, providing an independent measure of generalization beyond the bank's own data distribution.

Skills Applied

Python · TensorFlow 2 · Keras · Convolutional Neural Networks · Siamese Networks · Contrastive Learning · Image Preprocessing · OpenCV