Latent Trajectory: A New Framework for Deep Actor-Critic Reinforcement Learning

Lecture 25

Joint work: F. Shih and F. Liang (2025) published in TMLR

Motivation

- Reinforcement Learning (RL): sequential decision making via interaction with an environment.
- Central object: value function (or Q-function) under a policy.
- **Uncertainty quantification (UQ)** for value functions is important for:
  - safety
  - robustness
- In actor–critic RL, actor and critic are *interdependent*, complicating theory and UQ.

Joint work: F. Shih and F. Liang (2025) published in

Challenges in Deep RL UQ

- Deep neural networks: highly non-linear, over-parameterized
- RL: data are generated online by the current policy;
- Actor–critic:
  - Critic guides actor updates.
  - Actor formulates policy, which in turn determines the distribution of data for the critic.
- Many Bayesian / ensemble / bootstrap approaches:
  - often produce *mis-calibrated* intervals,
  - coverage can be far below nominal level when deep networks are involved.

Joint work: F. Shih and F. Liang (2025) published in

Existing Approaches for UQ in RL

- Bayesian / approximate Bayesian:
  - Randomized prior functions,
  - Bayesian dropout,
  - Gaussian-process–based TD.
- Frequentist / ensemble methods:
  - Bootstrapped DQN,
  - Deep ensembles, distributional RL.
- Issues in deep actor–critic setting:
  - Theoretical guarantees for coverage are largely absent.

Joint work: F. Shih and F. Liang (2025) published in

Our Contributions

- Introduce a **Latent Trajectory Framework (LTF)** for deep actor–critic RL:
    - Treat transition trajectories as latent variables.
    - Account for the interdependence between actor and critic updates.
- Propose an **adaptive stochastic gradient MCMC** (SGMCMC) algorithm:
    - SGD-style update for actor,
    - SGMCMC sampling for critic (conditional on actor).
- Prove convergence:
    - Actor parameters converge in probability to a solution of a mean-field equation.
    - Critic parameters converge in distribution to a target conditional law.
- Empirically:
    - Well-calibrated UQ for value function,
    - Improved policy search in benchmark environments.

Joint work: F. Shih and F. Liang (2025) published in

Background: Actor–Critic RL

- MDP with states s_t, actions a_t, rewards r_t, discount factor γ.
- Policy $\pi_\theta(a \mid s)$ parameterized by θ (actor).
- Critic network $V_\psi(s)$ approximates $V_{\pi_\theta}(s)$.
- Advantage Actor–Critic uses

$$A_\psi(s_t, a_t) = R_t - V_\psi(s_t),$$

where $R_t = \sum_{\tau=t}^{n} \gamma^{\tau-t} r_\tau$ is a truncated return up to horizon n.

Advantage Actor–Critic Policy Gradient

- Policy gradient:

$$g_{\psi}^{ac}(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{n} A_{\psi}(s_t, a_t)\, \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right].$$

- In practice:
  - Approximate expectation by sample trajectories.
  - Update θ by SGD:

$$\theta \leftarrow \theta + \upsilon\, \hat{g}_{\psi}^{ac}(\theta).$$

  - Update ψ to fit returns or TD targets.
- The difficulty:
  - θ-updates depend on ψ,
  - ψ-updates depend on trajectories generated under θ.

Joint work: F. Shih and F. Liang (2025) published in

Interdependence Between Actor and Critic

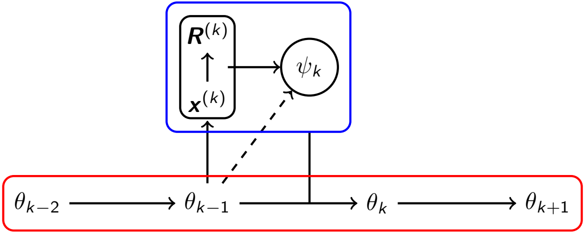
- For deep, multi-layer networks and general state spaces:
  - Theory is scarce.
  - UQ for values is even more challenging.
- Our key idea:

Treat trajectories as latent

Simulate *critic parameters* via MCMC, and learn *actor* parameters via optimization (SGD).

Joint work: F. Shih and F. Liang (2025) published in

Latent Markov Sampling Diagram



Trajectories act as latent variables: $\psi_k \perp \theta_{k-1} \mid \boldsymbol{x}^{(k)}$.

Joint work: F. Shih and F. Liang (2025) published in

Pseudo-Population and Conditional Law

- For a fixed policy π_θ, consider a large (but finite) *pseudo-population* of transitions of size $\mathcal{N}$.
- Critic parameter ψ has a conditional distribution

$$\pi_{\mathcal{N}}(\psi \mid \theta),$$

  induced by fitting V_ψ to this pseudo-population.
- Intuition:
  - Avoid degenerate behavior as number of tuples $\to \infty$.
  - Keep a non-trivial distribution over critic parameters, even asymptotically.
- Actor update should account for the variability in ψ: we work with a *mean-field* policy gradient.

Joint work: F. Shih and F. Liang (2025) published in

Mean-Field Equation for the Actor

- Define the mean-field gradient

$$g(\theta) = \int g_{\psi}^{ac}(\theta)\, \pi_{\mathcal{N}}(\psi \mid \theta)\, d\psi.$$

- Actor parameter is defined by solving

$$g(\theta^*) = 0.$$

- Training actor–critic networks becomes a stochastic approximation problem driven by samples from $\pi_{\mathcal{N}}(\psi \mid \theta)$.

Joint work: F. Shih and F. Liang (2025) published in

Adaptive SGMCMC Overview

High-level iteration (at step k)

1. Sample a trajectory batch $(\boldsymbol{x}^{(k)}, \boldsymbol{R}^{(k)})$ using current policy $\pi_{\theta_{k-1}}$.
2. ψ-**sampling:** approximately sample

$$\psi_k \sim \pi_{\mathcal{N}}(\psi \mid \theta_{k-1})$$

   using SGMCMC (e.g. SGLD or SGHMC).
3. θ-**update:**

$$\theta_k = \theta_{k-1} + \upsilon_k \, \hat{g}^{ac}_{\psi_k}(\theta_{k-1}),$$

   where $\hat{g}^{ac}$ is an unbiased estimator of g^{ac}_{ψ}.

- Stepsizes ϵ_k (for ψ) and υ_k (for θ) decay over time with appropriate rates.
- Under mild conditions, we obtain convergence of both θ_k and ψ_k.

Joint work: F. Shih and F. Liang (2025) published in

Reward Model and Gradient for ψ

- For convenience, assume a Gaussian reward model:

$$R_t \mid x_t, \psi \sim \mathcal{N}\big(V_\psi(s_t), \sigma^2\big).$$

- Then

$$\nabla_\psi \log \pi_{\mathcal{N}}(\psi \mid \boldsymbol{x}^{(k)}, \boldsymbol{R}^{(k)}) = \sum_{t=1}^{n} \nabla_\psi \log \pi(R_t^{(k)} \mid x_t^{(k)}, \psi) + \frac{n}{\mathcal{N}} \nabla_\psi \log \pi(\psi).$$

- This is the "critic gradient" used in SGMCMC:

$$\psi_k = \psi_{k-1} + \frac{\epsilon_k}{2} \nabla_\psi \tilde{L}(\theta_{k-1}, \psi_{k-1}) + \sqrt{\tfrac{n}{\mathcal{N}} \epsilon_k}\, e_k,$$

where e_k is Gaussian noise.

Importance Weighting and Latent Gradient

- Latent gradient uses importance weight

$$w_k = \frac{\pi(\boldsymbol{R}^{(k)} \mid \boldsymbol{x}^{(k)}, \psi_k)}{\pi(\boldsymbol{R}^{(k)} \mid \boldsymbol{x}^{(k)})}.$$

- Numerator from Gaussian model; denominator can be estimated by:
  - Monte Carlo over auxiliary ψ,
  - or kernel conditional density estimator (Nadaraya–Watson).
- Effective critic gradient:

$$\nabla_\psi \tilde{L}(\theta_{k-1}, \psi_{k-1}) = w_k \left\{ \sum_{t=1}^{n} \nabla_\psi \log \pi(R_t^{(k)} \mid x_t^{(k)}, \psi_{k-1}) + \frac{n}{\mathcal{N}} \nabla_\psi \log \pi(\psi_{k-1}) \right\}.$$

Algorithm: LT-A2C (High-Level)

Algorithm Latent Trajectory A2C (LT-A2C) – overview

1: Initialize actor π_{θ_0}, critic V_{ψ_0}.
2: **for** $k = 1$ to $\mathcal{K}$ **do**
3: Generate trajectories $\boldsymbol{x}^{(k)}, \boldsymbol{R}^{(k)}$ with policy $\pi_{\theta_{k-1}}$.
4: Draw auxiliary critic samples $\tilde{\psi}_j$ via SGMCMC to approximate $\pi(\psi \mid \boldsymbol{x}^{(k)})$.
5: Compute importance weight $\hat{w}_k$ using auxiliary samples.
6: Update critic via SGMCMC:

$$\psi_k = \psi_{k-1} + \text{SGLD step using } \hat{w}_k.$$

7: Update actor:

$$\theta_k = \theta_{k-1} + \upsilon_k \sum_{t=1}^{n} A_{\psi_k}(s_t^{(k)}, a_t^{(k)}) \nabla_\theta \log \pi_{\theta_{k-1}}(a_t^{(k)} \mid s_t^{(k)}).$$

8: **end for**

Joint work: F. Shih and F. Liang (2025) published in

Convergence Conditions (Sketch)

- Step sizes:

$$\epsilon_k = \frac{C_\epsilon}{c_\epsilon + k^\alpha}, \quad \upsilon_k = \frac{C_\upsilon}{c_\upsilon + k^\beta},$$

  with $0 < \beta \leq \alpha \leq \min\{1, 2\beta\}$.
- Regularity:
  - Smoothness of $L(\theta, \psi)$ in θ and ψ.
  - Dissipativity in ψ (e.g. Gaussian prior on ψ).
- Gradient noise:
  - Unbiased stochastic gradients,
  - Controlled second moments (depend on $\|\theta - \theta^*\|^2$ and $\|\psi\|^2$).

Joint work: F. Shih and F. Liang (2025) published in

Theorem 1: Convergence of Actor

Theorem (informal)

Under the regularity assumptions and step-size conditions, there exists a solution θ^* to the mean-field equation $g(\theta) = 0$ such that

$$\mathbb{E}\|\theta_k - \theta^*\|^2 \le \xi \, \upsilon_k, \quad k \ge k_0,$$

where $\xi > 0$ is a constant.

- θ_k converges (in mean square) to a root of $g(\theta)$.
- This root corresponds to a stationary point of the marginal actor objective that accounts for uncertainty in ψ.
- Standard in stochastic approximation / adaptive MCMC theory.

Joint work: F. Shih and F. Liang (2025) published in

Theorem 2: Convergence of Critic Distribution

Theorem (informal)

Let $\pi^* = \pi(\psi \mid \theta^*)$ and let $T_k = \sum_{i=1}^k \epsilon_i$. Let μ_{T_k} be the law of ψ_k. Then

$$\mathcal{W}_2(\mu_{T_k}, \pi^*) \;\leq\; (\hat{C}_0 \delta_{\tilde{L}}^{1/4} + \hat{C}_1 \upsilon_1^{1/4})\, T_k + \hat{C}_2 e^{-T_k/c_{LS}},$$

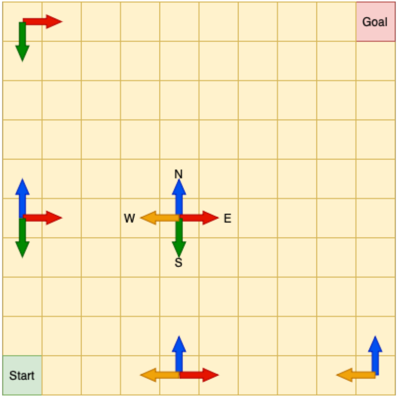
where $\mathcal{W}_2$ is the 2-Wasserstein distance and c_{LS} is a logarithmic Sobolev constant of π^*.

- Critic samples ψ_k converge in distribution to $\pi(\psi \mid \theta^*)$.
- Enables **proper UQ** for V_ψ and Q-values.
- This is not available from plain SGD-based training.

Joint work: F. Shih and F. Liang (2025) published in

Experiments: Indoor Escape Environment

- 10×10 grid, start at bottom left, goal at top right.
- Actions: N, E, S, W.
- Reward: $r_t \sim \mathcal{N}(-1, 0.01)$ each step.
- Episode length capped (e.g. 200 steps).



Indoor Escape environment

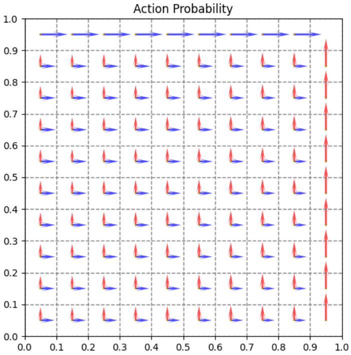
Policy Diversity: KL Divergence

- Define optimal policy $\pi^*(\cdot \mid s)$:
  - uniform over optimal actions,
  - zero on sub-optimal actions.
- For a learned policy $\pi_\theta(\cdot \mid s)$, measure

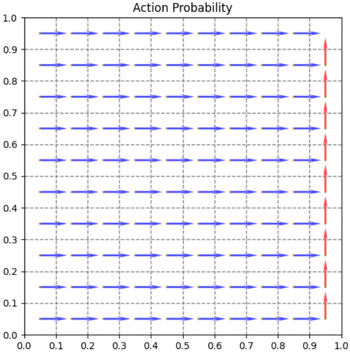
$$D_{KL}(\pi^* \,\|\, \pi_\theta) = \sum_a \pi^*(a \mid s) \log \frac{\pi^*(a \mid s)}{\pi_\theta(a \mid s)}.$$

- Small D_{KL}: policy spreads mass correctly over optimal actions (good diversity).
- Large D_{KL}: policy collapses onto a single action, poor coverage.

Policy Distributions Across States



LTF (LT-A2C): nearly uniform over optimal actions



A2C: collapsed policy, large D_{KL}

Joint work: F. Shih and F. Liang (2025) published in

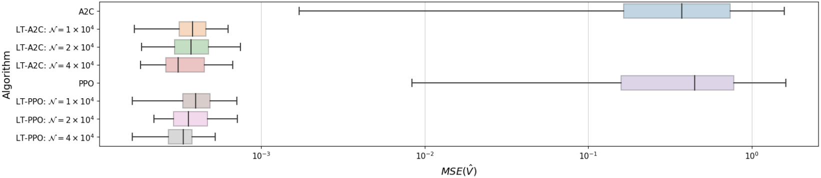
Indoor Escape: Quantitative Metrics

Algorithm	$\mathcal{N}$	$D_{KL}(\pi^* \| \pi_{\theta^*})$	MSE($\hat{V}$)	Coverage
A2C	–	4.647	0.535	0.489
LT-A2C	10k	0.010	0.00038	0.947
LT-A2C	20k	0.014	0.00039	0.947
LT-A2C	40k	0.014	0.00033	0.947
PPO	–	4.773	0.561	0.487
LT-PPO	10k	0.011	0.00041	0.947
LT-PPO	20k	0.009	0.00038	0.947
LT-PPO	40k	0.011	0.00032	0.947

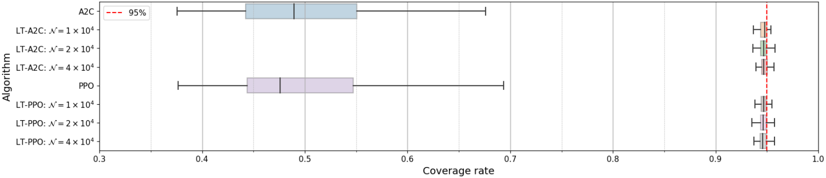
Averages over 100 runs. LT methods dramatically reduce KL and MSE, and achieve $\approx$ 95% coverage.

Joint work: F. Shih and F. Liang (2025) published in

Value Estimation and Coverage



MSE of $\hat{V}(s)$ across states



Coverage rate of 95% prediction intervals for $V^*(s)$

Joint work: F. Shih and F. Liang (2025) published in

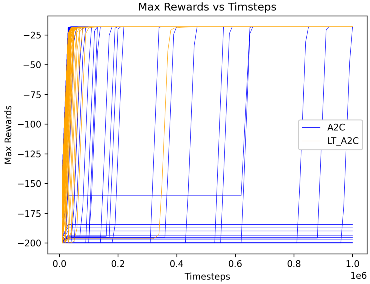
Existing UQ Methods for Q in Escape

Algorithm	MSE($\hat{Q}$)	Coverage	CI width
BootDQN	0.0998	0.39	0.188
QR-DQN	0.0046	0.82	0.278
RPN (0.1)	0.0334	0.80	0.679
RPN (1.0)	0.0372	0.82	0.693
RPN (5.0)	0.0366	0.79	0.782

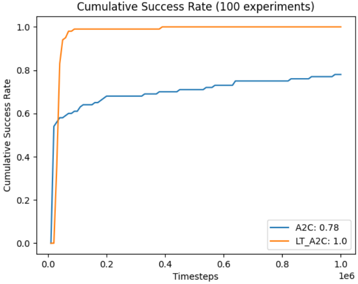
- None achieves nominal 95% coverage.
- Calibration of UQ with deep networks is nontrivial.
- Our LTF gives calibrated coverage for V; conceptually extendable to Q.

Joint work: F. Shih and F. Liang (2025) published in

Optimal Policy Search and Success Rate



Best-so-far reward vs. time steps



Success rate of reaching optimal policy

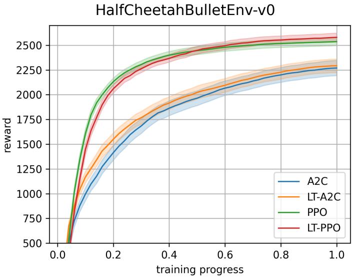
LT-A2C finds the optimal policy in essentially all runs; A2C fails in a substantial fraction.

Joint work: F. Shih and F. Liang (2025) published in

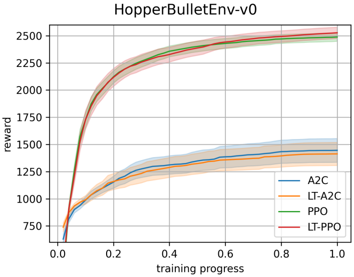
PyBullet Continuous Control Tasks

- PyBullet environments:
  - HalfCheetahBulletEnv-v0
  - HopperBulletEnv-v0
  - ReacherBulletEnv-v0
  - Walker2DBulletEnv-v0
- Use RL Baselines3 Zoo for implementations.
- Compare:
  - A2C vs LT-A2C,
  - PPO vs LT-PPO.
- Evaluate training curves and best evaluation rewards over time.

PyBullet: Performance Curves (1)



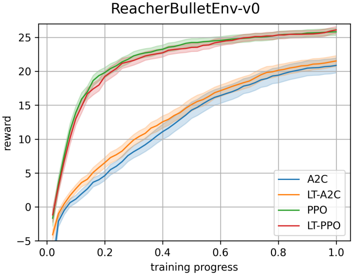
HalfCheetahBulletEnv-v0



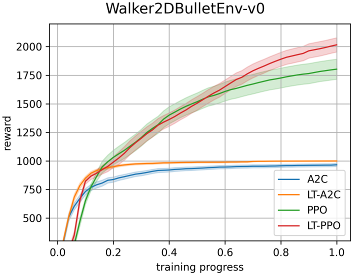
HopperBulletEnv-v0

Joint work: F. Shih and F. Liang (2025) published in

PyBullet: Performance Curves (2)



ReacherBulletEnv-v0

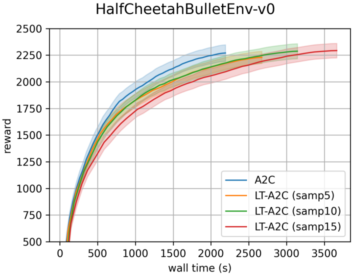


Walker2DBulletEnv-v0

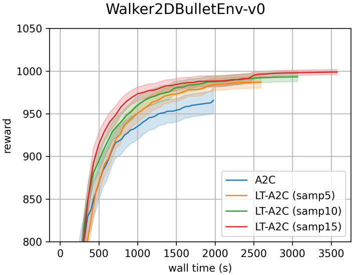
LT-A2C / LT-PPO perform comparably to baselines on some tasks and substantially better on Walker-2D.

Joint work: F. Shih and F. Liang (2025) published in

Wall-Clock Time and Inner Sampling Steps



HalfCheetah: best reward vs wall time



Walker2D: best reward vs wall time

Using 5–10 critic-sampling steps is enough in practice; improvements plateau beyond that.

Joint work: F. Shih and F. Liang (2025) published in

When Does LTF Help Most?

- Rugged actor objective landscapes:
- LTF adds *structured randomness* through critic sampling:
  - Additional exploration in parameter space,
  - Helps escape bad local optima.

Flexibility of the Framework

- Critic-sampling step:
  - SGLD, SGHMC, or other SGMCMC variants.
- Actor step:
  - A2C, PPO, SAC, etc.
- Critic distributions can encode more advanced RL tricks:
  - Ensembles (e.g., REDQ),
  - Optimistic critics (e.g., OAC),
  - SUNRISE-style uncertainty scaling.
- LTF provides a *general wrapper*:
  - improves and stabilizes training,
  - while providing UQ for the value function.

Joint work: F. Shih and F. Liang (2025) published in

Computational Considerations

- Per-iteration cost of SGLD / SGHMC is similar to SGD:
  - one gradient evaluation + Gaussian noise.
- Extra cost comes from:
  - inner critic-sampling loop (5–10 steps).
- In practice:
  - Training time increases moderately,
  - Gains in policy quality and calibrated UQ can justify the cost.
- Scales to large networks:
  - no change in big-O complexity,
  - implementation similar to modern deep RL pipelines.

Summary

- Proposed a **Latent Trajectory Framework** for deep actor–critic RL.
- Trained via **adaptive SGMCMC**:
  - Actor: stochastic approximation to a mean-field gradient.
  - Critic: MCMC sampling from $\pi_{\mathcal{N}}(\psi \mid \theta)$.
- Theoretical guarantees:
  - Actor convergence to stationary point,
  - Critic distribution converges in Wasserstein distance.
- Empirical results:
  - Calibrated uncertainty for value function in Indoor Escape.
  - Improved policy search, especially in harder tasks (e.g. Walker-2D).

Joint work: F. Shih and F. Liang (2025) published in

Practical Tips

- Step sizes:
  - Choose ϵ_k, υ_k decreasing, with ϵ_k not smaller than υ_k.
- Number of critic samples per iteration:
  - 5–10 SGMCMC steps typically enough.
- Pseudo-population size $\mathcal{N}$:
  - Larger $\mathcal{N} \Rightarrow$ more accurate UQ, slightly narrower CIs.
- Initialization:
  - Using standard deep RL initializations and hyperparameters works well as a starting point.

Thank You

Questions?

- Paper: Shi, F. and Liang, F. (2025). *Latent Trajectory: A New Framework for Deep Actor-Critic Reinforcement Learning with Uncertainty Quantification*, *Transactions on Machine Learning Research*.
- OpenReview: `https://openreview.net/forum?id=8B74xdaRHa`