Deterministic Policy Gradients in the Era of Soft RL

Mixture of Actors and Adaptive Ensembles

Guni Sharon Sheelabhadra Dey Komo Zhang

Department of Computer Science and Engineering, Texas A&M University

June 17, 2026

Problem Definition: Markov Decision Process (MDP)

An MDP is defined by:

- $\mathcal{S}$: State Space (the state of the agent).
- $\mathcal{A}$: Action Space (what the agent can do).
- $\mathcal{P}$: Transition Dynamics
  $(\mathcal{P} : S \times A \times S \to [0, 1])$.
- $\mathcal{R}$: Reward Function
  $(\mathcal{R} : S \times A \times S \to \mathbb{R})$.
- γ: Discount Factor $(\gamma \in [0, 1))$.

Problem Definition: Markov Decision Process (MDP)

An MDP is defined by:

- $\mathcal{S}$: State Space (the state of the agent).
- $\mathcal{A}$: Action Space (what the agent can do).
- $\mathcal{P}$: Transition Dynamics ($\mathcal{P} : S \times A \times S \to [0, 1]$).
- $\mathcal{R}$: Reward Function ($\mathcal{R} : S \times A \times S \to \mathbb{R}$).
- γ: Discount Factor ($\gamma \in [0, 1)$).

Deterministic Policy μ: A mapping from states to actions $\mu : \mathcal{S} \to \mathcal{A}$.

Problem Definition: Markov Decision Process (MDP)

An MDP is defined by:

- $\mathcal{S}$: State Space (the state of the agent).
- $\mathcal{A}$: Action Space (what the agent can do).
- $\mathcal{P}$: Transition Dynamics ($\mathcal{P} : S \times A \times S \to [0, 1]$).
- $\mathcal{R}$: Reward Function ($\mathcal{R} : S \times A \times S \to \mathbb{R}$).
- γ: Discount Factor ($\gamma \in [0, 1)$).

Deterministic Policy μ: A mapping from states to actions $\mu : \mathcal{S} \to \mathcal{A}$.

Stochastic Policy π: A mapping from states to a probability simplex over the action space $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.

Problem Definition: Markov Decision Process (MDP)

An MDP is defined by:

- $\mathcal{S}$: State Space (the state of the agent).
- $\mathcal{A}$: Action Space (what the agent can do).
- $\mathcal{P}$: Transition Dynamics ($\mathcal{P} : S \times A \times S \to [0, 1]$).
- $\mathcal{R}$: Reward Function ($\mathcal{R} : S \times A \times S \to \mathbb{R}$).
- γ: Discount Factor ($\gamma \in [0, 1)$).

Deterministic Policy μ: A mapping from states to actions $\mu : \mathcal{S} \to \mathcal{A}$.

Stochastic Policy π: A mapping from states to a probability simplex over the action space $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.

The RL Objective

The goal is to maximize the expected accumulated sum of discounted rewards $J(\pi)$:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \right]$$

Optimal Policy:

$$\pi^* = \underset{\pi}{\text{argmax}}\ J(\pi)$$

Background: Deterministic Policy Gradient (DPG)

1. The Deterministic Policy Objective

$$J(\mu_\theta) = \int_{\mathcal{S}} \rho^{\mu}(s) Q^{\mu}(s, \mu_\theta(s)) ds$$

Background: Deterministic Policy Gradient (DPG)

1. The Deterministic Policy Objective

$$J(\mu_\theta) = \int_{\mathcal{S}} \rho^{\mu}(s) Q^{\mu}(s, \mu_\theta(s)) ds$$

2. The DPG Theorem

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim\rho^{\mu}} \left[\nabla_\theta \mu_\theta(s) \; \nabla_a Q^{\mu}(s, a)|_{a=\mu_\theta(s)} \right]$$

Background: Deterministic Policy Gradient (DPG)

1. The Deterministic Policy Objective

$$J(\mu_\theta) = \int_{\mathcal{S}} \rho^{\mu}(s) Q^{\mu}(s, \mu_\theta(s)) ds$$

2. The DPG Theorem

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim\rho^{\mu}} \left[\nabla_\theta \mu_\theta(s) \, \nabla_a Q^{\mu}(s, a)|_{a=\mu_\theta(s)} \right]$$

3. Key Characteristics

Low Variance: does not require sampling over the action space $\mathcal{A}$, leading to low variance estimates.

Background: Deterministic Policy Gradient (DPG)

1. The Deterministic Policy Objective

$$J(\mu_\theta) = \int_{\mathcal{S}} \rho^{\mu}(s) Q^{\mu}(s, \mu_\theta(s)) ds$$

2. The DPG Theorem

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim\rho^{\mu}} \left[\nabla_\theta \mu_\theta(s) \, \nabla_a Q^{\mu}(s, a)|_{a=\mu_\theta(s)} \right]$$

3. Key Characteristics

Low Variance: does not require sampling over the action space $\mathcal{A}$, leading to low variance estimates.

Off-Policy Compatibility: DPG allows the expectation to be approximated using states sampled from an off-policy replay buffer.

Background: Deterministic Policy Gradient (DPG)

1. The Deterministic Policy Objective

$$J(\mu_\theta) = \int_{\mathcal{S}} \rho^\mu(s) Q^\mu(s, \mu_\theta(s)) ds$$

2. The DPG Theorem

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(s)\ \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right]$$

3. Key Characteristics

Low Variance: does not require sampling over the action space $\mathcal{A}$, leading to low variance estimates.

Off-Policy Compatibility: DPG allows the expectation to be approximated using states sampled from an off-policy replay buffer.

Limited Policy Representation: Cannot capture multiple modes of optimality $\Rightarrow \pi^*$ might be unlearnable + inefficient exploration.

Background: Stochastic Policy Gradient (SPG)

1. The Standard SPG Theorem (REINFORCE/A2C)

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[Q^\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)\right]$$

Background: Stochastic Policy Gradient (SPG)

1. The Standard SPG Theorem (REINFORCE/A2C)

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[Q^\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)\right]$$

2. Key Characteristics

High Variance: The expectation involves sampling over both states and actions, $\mathbb{E}_{s,a}$, leading to high variance, often necessitating strong variance reduction techniques (e.g., advantage estimation).

Background: Stochastic Policy Gradient (SPG)

1. The Standard SPG Theorem (REINFORCE/A2C)

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[Q^\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)\right]$$

2. Key Characteristics

High Variance: The expectation involves sampling over both states and actions, $\mathbb{E}_{s,a}$, leading to high variance, often necessitating strong variance reduction techniques (e.g., advantage estimation).

On-Policy Requirement: The expectation must be computed over trajectories (ρ^π distribution) induced by the current policy π_θ, making it inherently on-policy (or requiring complex and high variance off-policy corrections like importance sampling).

Background: Stochastic Policy Gradient (SPG)

1. The Standard SPG Theorem (REINFORCE/A2C)

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[Q^\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)\right]$$

2. Key Characteristics

High Variance: The expectation involves sampling over both states and actions, $\mathbb{E}_{s,a}$, leading to high variance, often necessitating strong variance reduction techniques (e.g., advantage estimation).

On-Policy Requirement: The expectation must be computed over trajectories (ρ^π distribution) induced by the current policy π_θ, making it inherently on-policy (or requiring complex and high variance off-policy corrections like importance sampling).

Flexible Policy Representation: Underlying policy is stochastic. Can capture multiple modes of optimality. Provides intrinsic exploration.

DPG for training a soft actor

The Challenge: How to combine the low variance of DPG with the multimodal flexibility of general stochastic policies?

DPG for training a soft actor

The Challenge: How to combine the low variance of DPG with the multimodal flexibility of general stochastic policies?

DDPG [Lillicrap et al., 2015]: Sampling a noise value from an Ornstein-Uhlenbeck process (Uhlenbeck & Ornstein, 1930) around $\mu(s)$. Alternatively, considers a Gaussian with $\mu(s)$ and a fixed (low) Σ.

* This is Not a general stochastic policy!

DPG for training a soft actor

The Challenge: How to combine the low variance of DPG with the multimodal flexibility of general stochastic policies?

DDPG [Lillicrap et al., 2015]: Sampling a noise value from an Ornstein-Uhlenbeck process (Uhlenbeck & Ornstein, 1930) around $\mu(s)$. Alternatively, considers a Gaussian with $\mu(s)$ and a fixed (low) Σ.

* This is Not a general stochastic policy!

Our solution [Dey & Sharon, 2024]: Consider a Gaussian Mixture Model (GMM) with component means $\{\mu_i(s)\}_{i=1}^K$ and shared fixed covariance Σ. Train each $\mu_i(s)$ using DPG.

DPG for training a soft actor

The Challenge: How to combine the low variance of DPG with the multimodal flexibility of general stochastic policies?

DDPG [Lillicrap et al., 2015]: Sampling a noise value from an Ornstein-Uhlenbeck process (Uhlenbeck & Ornstein, 1930) around $\mu(s)$. Alternatively, considers a Gaussian with $\mu(s)$ and a fixed (low) Σ.

* This is Not a general stochastic policy!

Our solution [Dey & Sharon, 2024]: Consider a Gaussian Mixture Model (GMM) with component means $\{\mu_i(s)\}_{i=1}^K$ and shared fixed covariance Σ. Train each $\mu_i(s)$ using DPG.

A general stochastic policy? GMMs are universal approximators of densities, i.e., a GMM with sufficient components can approximate any other density function to arbitrary precision [Alspach & Sorenson, 1972].

GMM Policy & The Challenge of Mode Collapse

1. **The GMM Policy Structure:** A mixture of K Gaussian components.

$$\pi_\theta(a|s) = \sum_{k=1}^{K} \omega_k(s)\mathcal{N}(a|\mu_k(s), \Sigma_k)$$

GMM Policy & The Challenge of Mode Collapse

1. The GMM Policy Structure: A mixture of K Gaussian components.

$$\pi_\theta(a|s) = \sum_{k=1}^{K} \omega_k(s)\mathcal{N}(a|\mu_k(s), \Sigma_k)$$

2. Naive Deterministic Update: Applying the DPG theorem to each component independently:

$$\Delta\mu_k(s) \propto \nabla_a Q(s,a)\big|_{a=\mu_k(s)}$$

GMM Policy & The Challenge of Mode Collapse

1. **The GMM Policy Structure:** A mixture of K Gaussian components.

$$\pi_\theta(a|s) = \sum_{k=1}^{K} \omega_k(s)\mathcal{N}(a|\mu_k(s), \Sigma_k)$$

2. **Naive Deterministic Update:** Applying the DPG theorem to each component independently:

$$\Delta\mu_k(s) \propto \nabla_a Q(s,a)\big|_{a=\mu_k(s)}$$

3. **Mode Collapse Issue.**

   Problem: All GMM components μ_k optimize against the *same* Q-function landscape.

   Result: They might all converge to the same extremum.

   Consequence: The GMM degenerates into a single mode $(\mu_1 \approx \mu_2 \approx \cdots \approx \mu_K)$.

MaxEnt GMM Gradient

1. The Proposed Gradient: The gradient is defined for each Gaussian $i \in [0 \dots N-1]$ based on two (conflicting) objectives:

1.1. Maximize Return (DDPG): Maximize the expected return for a policy following Gaussian i:

$$\max_{\theta[i]} [Q(s; \theta[i])]$$

MaxEnt GMM Gradient

1. The Proposed Gradient: The gradient is defined for each Gaussian $i \in [0 \dots N-1]$ based on two (conflicting) objectives:

1.1. Maximize Return (DDPG): Maximize the expected return for a policy following Gaussian i:

$$\max_{\theta[i]} [Q(s; \theta[i])]$$

1.2. Maximize Entropy (MaxEnt): Maximize the cross-entropy between Gaussian i and the GMM excluding Gaussian i:

$$\max_{\theta[i]} \left[- \int \mathcal{N}(x; \mu(s; \theta[i]), \Sigma) \log \text{GMM}_{N \setminus i}(x; \mu(s; \theta)) dx \right]$$

MaxEnt GMM Gradient

1. The Proposed Gradient: The gradient is defined for each Gaussian $i \in [0 \dots N-1]$ based on two (conflicting) objectives:

1.1. Maximize Return (DDPG): Maximize the expected return for a policy following Gaussian i:

$$\max_{\theta[i]} [Q(s; \theta[i])]$$

1.2. Maximize Entropy (MaxEnt): Maximize the cross-entropy between Gaussian i and the GMM excluding Gaussian i:

$$\max_{\theta[i]} \left[- \int \mathcal{N}(x; \mu(s; \theta[i]), \Sigma) \log \text{GMM}_{N \setminus i}(x; \mu(s; \theta)) dx \right]$$

2. Resulting Update: Gaussian i is updated to increase the approximated Q-value **AND** increase the Cross Entropy w.r.t. the rest of the mixture (acting as a repulsive force).

Cross-Entropy vs KL

Computation Challenge: No known closed form expression for computing the Cross-Entropy over GMM components.

Cross-Entropy vs KL

Computation Challenge: No known closed form expression for computing the Cross-Entropy over GMM components.

Proposition

Cross-entropy and KL-divergence result in the same gradients with respect to a single Gaussian, i, and the GMM excluding i, denoted $GMM_{N \setminus i}$.

Proof.

The *Shannon entropy* of a single Gaussian, $f = \mathcal{N}(\mu, \Sigma)$, is $H(f) = 0.5 \ln \det(2\pi e \Sigma)$ [Huber, 2008] (note that π refers to the mathematical constant and not a policy here).

Since $H(f)$ is not a function of μ, we get $\nabla_\mu H(f) = 0$. As a result, for any other density g (e.g., a GMM), we have $\nabla_\mu H(f, g) = \nabla_\mu D_{KL}(f \parallel g)$.

Note that $\nabla_\Sigma H(f, g)$ is not needed for Gamid because it assumes a constant Σ. □

Approximating KL Divergence

Challenge: Compute $D_{KL}(f\|g)$, f is a single Gaussian and g is a GMM.

- **Problem:** No Known closed-form expression.

Approximating KL Divergence

Challenge: Compute $D_{KL}(f\|g)$, f is a single Gaussian and g is a GMM.

- **Problem:** No Known closed-form expression.

1. Variational Approximation [Hershey & Olsen, 2007]

$$D_{var}(f\|g) = -\log \sum_b p_b \exp(-D_{KL}(f\|g_b))$$

where g_b are the individual Gaussian components of g.
A closed form D_{KL} between two Gaussians is well defined!

Approximating KL Divergence

Challenge: Compute $D_{KL}(f\|g)$, f is a single Gaussian and g is a GMM.

- **Problem:** No Known closed-form expression.

1. Variational Approximation [Hershey & Olsen, 2007]

$$D_{var}(f\|g) = -\log \sum_b p_b \exp(-D_{KL}(f\|g_b))$$

where g_b are the individual Gaussian components of g.
A closed form D_{KL} between two Gaussians is well defined!

2. Practical Implementation For shared covariance $\Sigma = \mathbf{I}$ and uniform weights $p_b = 1/N$, the pairwise KL simplifies to Euclidean distance:

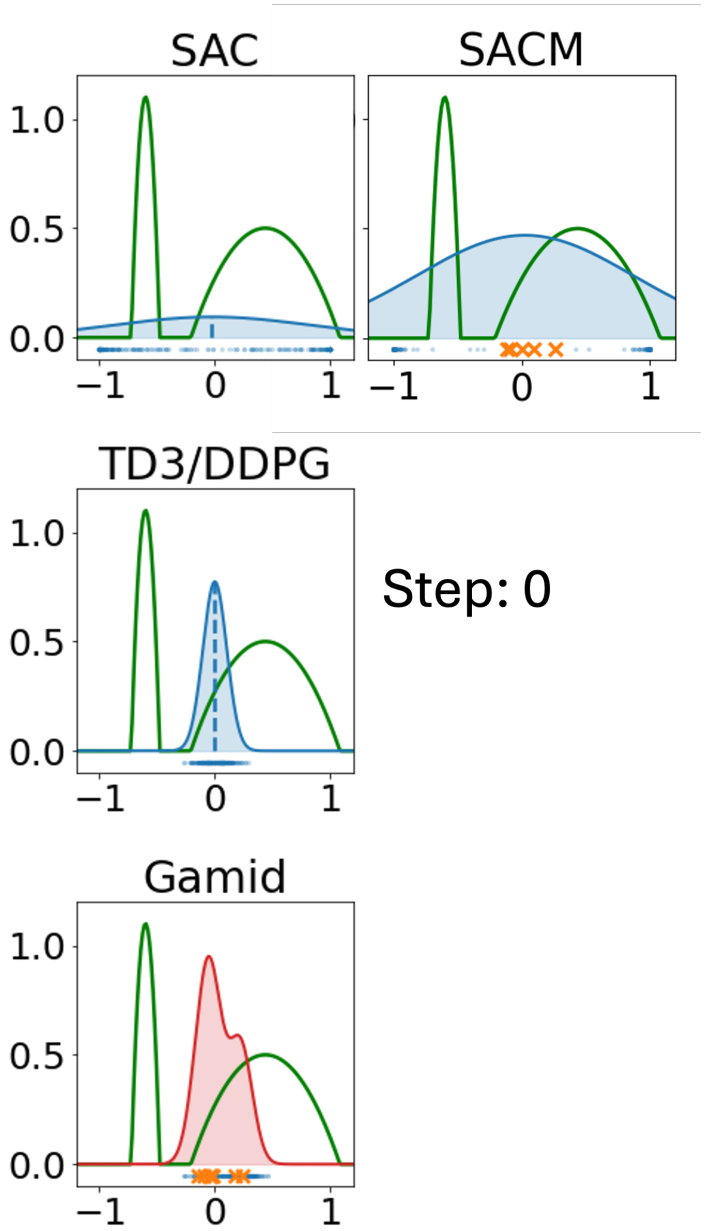
$$D_{KL}(f\|g_b) \propto ||\mu_f - \mu_b||^2$$

Substituting this back:

$$D_{var} \propto -\log \sum_{j \neq i} \exp\left(-||\mu_i - \mu_j||^2\right)$$

Acts as a repulsive potential field, pushing μ_i away from all other means μ_j.

1-D continuous bandit demo



Axes:

x-axis: Action space

y-axis: Probability density

Observation:

Unimodal DDPG/TD3 and SAC as well as GMM-based SACM **Converge on suboptimal mode.**

Benchmark Environments

Evaluation Domains

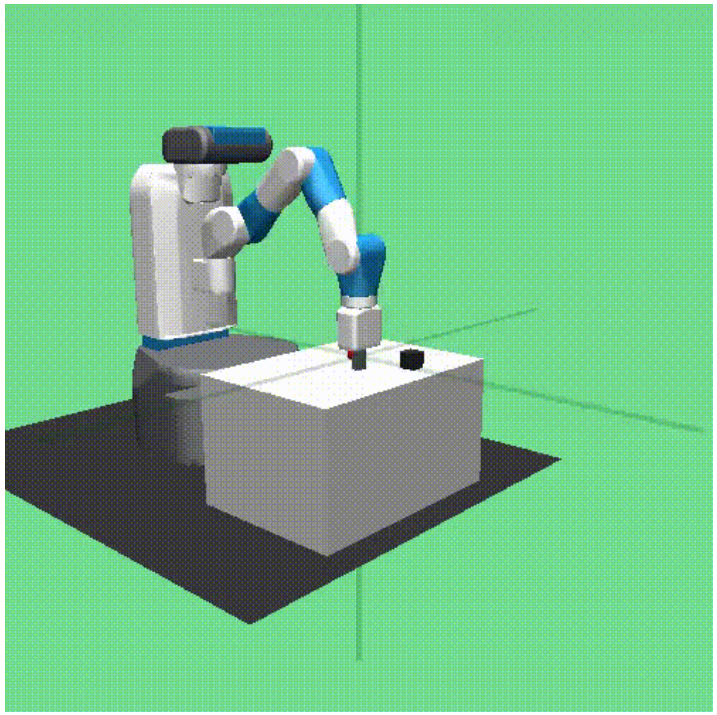
1. MuJoCo (Dense Rewards)

Continuous Locomotion Control

Tasks: Ant, HalfCheetah, Walker2d, Humanoid, etc.

2. Fetch (Sparse Rewards)

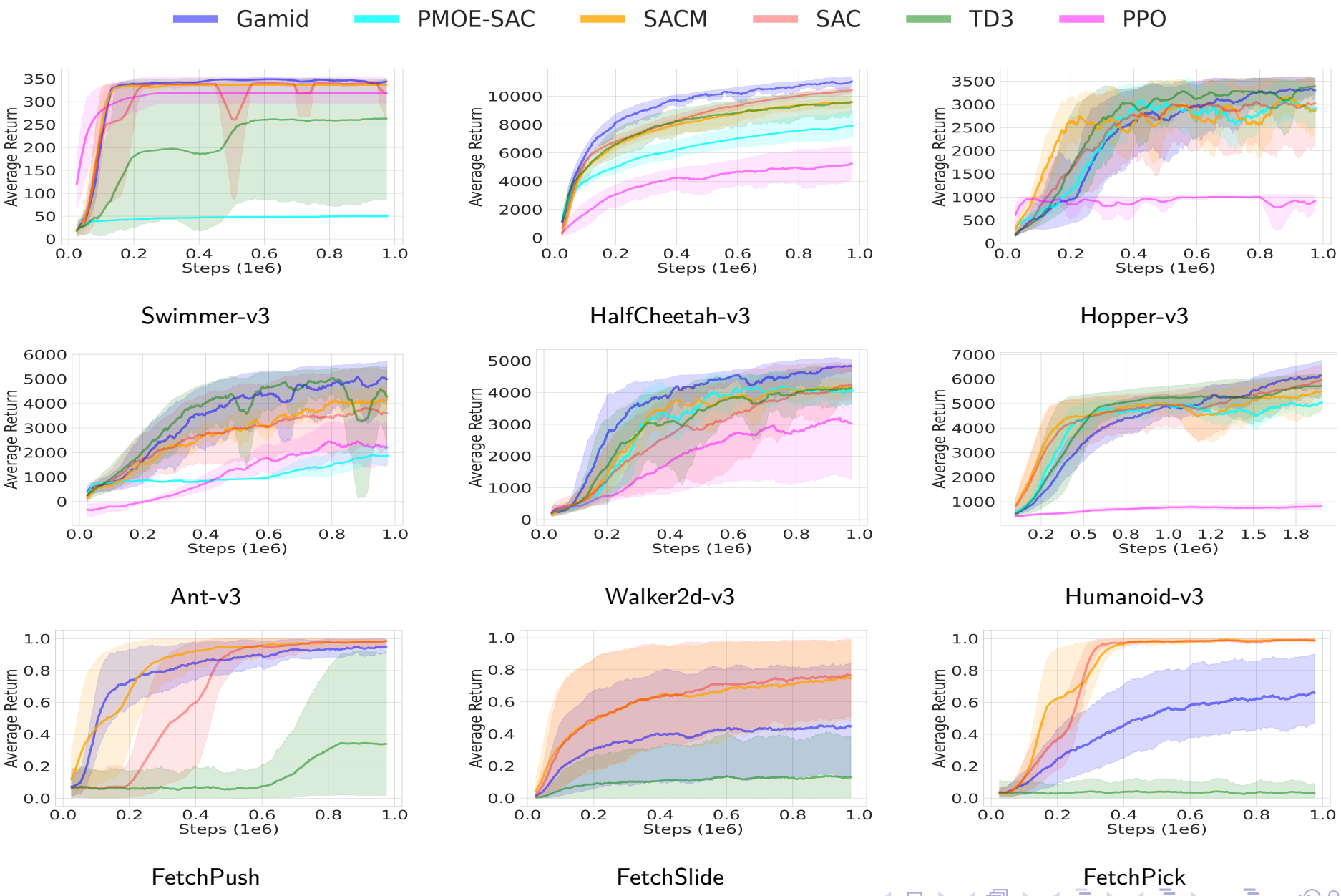
Goal-Conditioned Manipulation



Tasks: Push, Slide, PickAndPlace

These environments test the agent's ability to learn complex dynamics and handle varying reward densities.

Empirical Results: Gamid $\geq$ TD3



Practical Algorithm

Gamid **tuning burden**: MaxEnt temp? GMM component sampling? Number of GMM components?

Notation: **Actor** := a single Gaussian, **Actor Ensemble** := GMM.

1. Implicit Exploration and diversification. Introduce inherent random processes which implicitly encourage exploration and actor diversity.

- ➡ **Dual-Randomized Actor Selection**:
  - ➥ **Exploration:** One random actor interacts with the environment.
  - ➥ **Learning:** Another random actor receives gradient updates.

Practical Algorithm

Gamid **tuning burden**: MaxEnt temp? GMM component sampling? Number of GMM components?

Notation: **Actor** := a single Gaussian, **Actor Ensemble** := GMM.

1. Implicit Exploration and diversification. Introduce inherent random processes which implicitly encourage exploration and actor diversity.

- ➡ **Dual-Randomized Actor Selection**:
  - ➥ **Exploration:** One random actor interacts with the environment.
  - ➥ **Learning:** Another random actor receives gradient updates.

2. Adaptive Ensemble Size (N). Dynamically adapts the ensemble size:

- ➡ **Dual-Pruning Mechanism:** Prune actors based on both:
  - ➥ **Performance-Estimates:** Removing underperforming actors.
  - ➥ **Action-Space Distances:** Removing redundant (similar) actors.

Validating Exploration: Implicit vs. Explicit

1. **Measuring Exploration**

   New State Exploration Rate. Using RSNorm [Lee et al., 2025].

2. **Setup**

   Gamid: Uses explicit MaxEnt regularization (requires tuning τ).

   Dual-Randomized (AEAP): Uses implicit randomization (no parameters).

Conclusion: Dual-randomization achieves similar exploration coverage without the tuning burden.

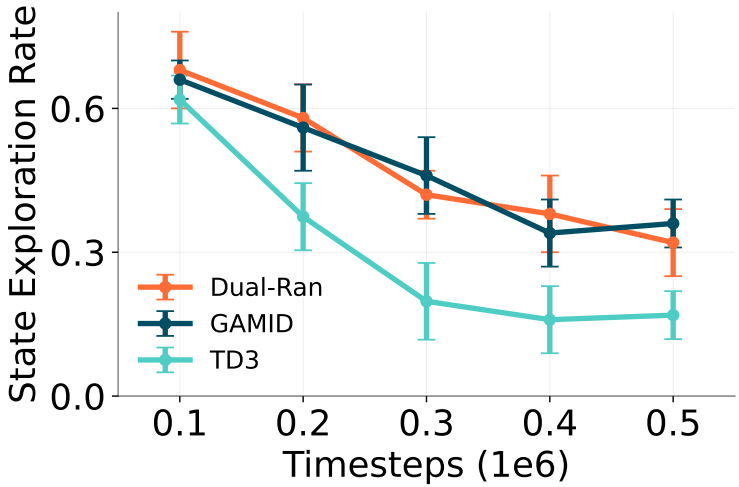


Figure: Dual-randomized exhibits similar new state exploration rates as a tuned Gamid on Walker2d.

Pruning Strategy: Dual Criteria

Objective: Exploration diversity and computational efficiency.

1. Dual-Pruning Mechanism

☞ **Performance-Based:** Remove actor i if its mean Q-value falls below ratio ξ of the best performer:

$$\mathbb{E}_s \left[Q\left(s, \mu_i(s)\right) \right] < \xi \cdot \mathbb{E}_s \left[\max_j Q\left(s, \mu_j(s)\right) \right]$$

Pruning Strategy: Dual Criteria

Objective: Exploration diversity and computational efficiency.

1. Dual-Pruning Mechanism

- **Performance-Based:** Remove actor i if its mean Q-value falls below ratio ξ of the best performer:

$$\mathbb{E}_s\left[Q\left(s, \mu_i(s)\right)\right] < \xi \cdot \mathbb{E}_s\left[\max_j Q\left(s, \mu_j(s)\right)\right]$$

- **Redundancy-Based:** For any actors i and j, if their max pairwise action distance $< \zeta$, remove the actor with lower Q-value.

$$\max_s \left\|\mu_i(s) - \mu_j(s)\right\|_2 < \zeta \implies \text{prune}\left(\arg\min_{a \in \{i,j\}} \left(\mathbb{E}_s\left[Q\left(s, \mu_a(s)\right)\right]\right)\right)$$

Pruning Strategy: Dual Criteria

Objective: Exploration diversity and computational efficiency.

1. Dual-Pruning Mechanism

- **Performance-Based:** Remove actor i if its mean Q-value falls below ratio ξ of the best performer:

$$\mathbb{E}_s\left[Q\left(s, \mu_i(s)\right)\right] < \xi \cdot \mathbb{E}_s\left[\max_j Q\left(s, \mu_j(s)\right)\right]$$

- **Redundancy-Based:** For any actors i and j, if their max pairwise action distance $< \zeta$, remove the actor with lower Q-value.

$$\max_s \|\mu_i(s) - \mu_j(s)\|_2 < \zeta \implies \text{prune}\left(\arg\min_{a \in \{i,j\}} \left(\mathbb{E}_s\left[Q\left(s, \mu_a(s)\right)\right]\right)\right)$$

2. Hyperparameters: Initial Ensemble Size N, Performance Threshold ξ, Distance Threshold ζ, and Pruning Frequency κ.

AEAP Hyperparameters & Configurations

1. Standard Hyperparameters (Robust across domains)

- **Performance Threshold** $\xi = 0.85$**:** Eliminates actors performing below 85%. Empirically robust against overestimation bias.
- **Pruning Frequency** $\kappa = 10,000$ **steps:** Ensures adequate training before making pruning decisions.
- **Distance Threshold** ζ**:** Scales with action space dimension:

$$\zeta = c \cdot \sqrt{\dim(\mathcal{A})} \cdot a_{\max}$$

AEAP Hyperparameters & Configurations

1. Standard Hyperparameters (Robust across domains)

- **Performance Threshold** $\xi = 0.85$**:** Eliminates actors performing below 85%. Empirically robust against overestimation bias.
- **Pruning Frequency** $\kappa = 10,000$ **steps:** Ensures adequate training before making pruning decisions.
- **Distance Threshold** ζ**:** Scales with action space dimension:

$$\zeta = c \cdot \sqrt{\dim(\mathcal{A})} \cdot a_{\max}$$

2. Recommended Configurations:

Conservative Setting ($N = 3, c = 1.0$)

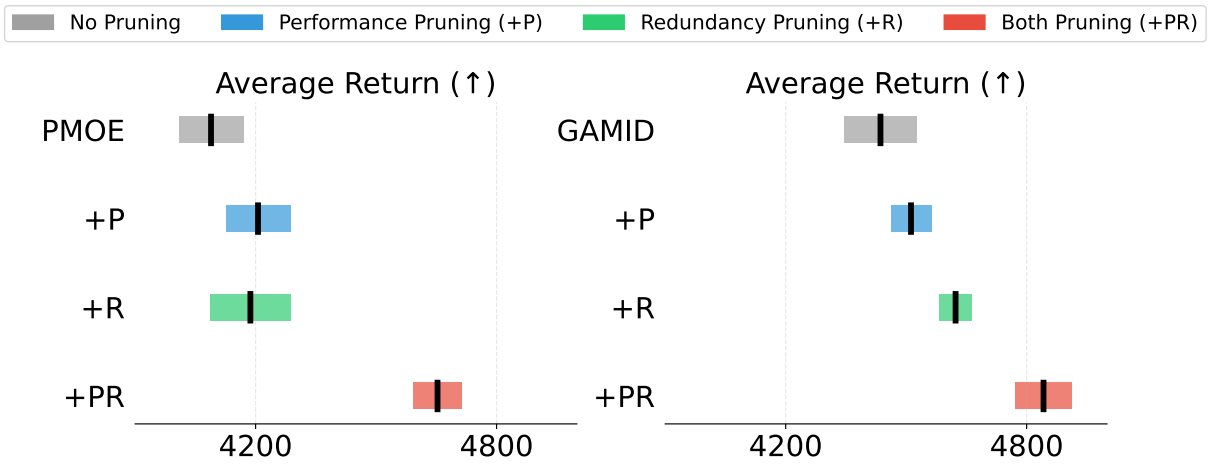
Small initial ensemble with a conservative threshold.

Aggressive Setting ($N = 7, c = 1.5$)

Large initial ensemble + higher threshold. Rapid pruning while leveraging initial exploration. *Particularly beneficial for sparse-reward environments.*

Pruning Component Ablation

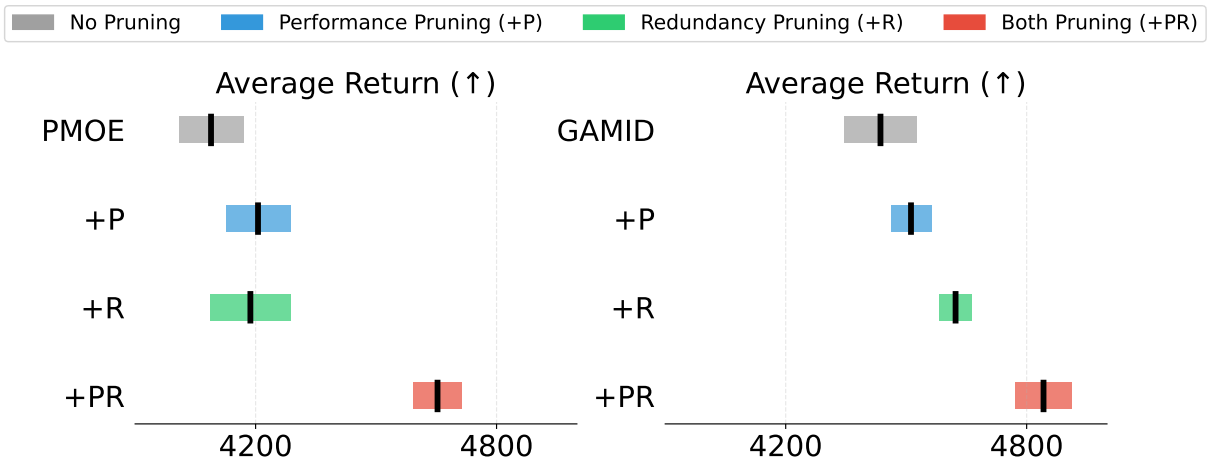
- **Baseline algorithms:** (1) Probabilistic Mixture-of-Experts SAC (PMOE) (Ren et al., 2021), (2) GAMID (Dey & Sharon, 2024).
- **Domain:** Ant-V5, 1M training steps (see paper for more results).
- **Variants:** (1) no pruning, (2) Performance Pruning (+P), (3) Redundancy Pruning (+R), (4) both (+PR).



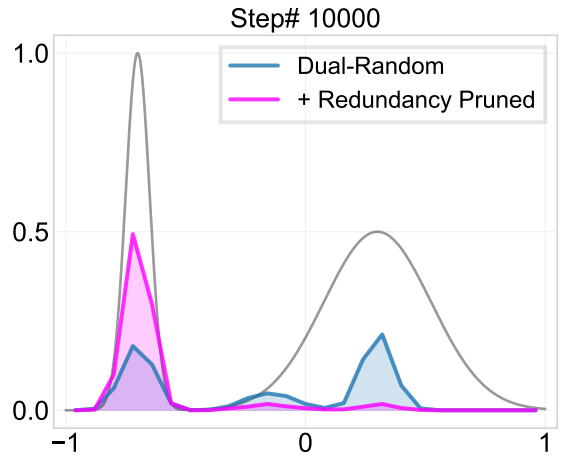
(a)

Pruning Component Ablation

- **Baseline algorithms:** (1) Probabilistic Mixture-of-Experts SAC (PMOE) (Ren et al., 2021), (2) GAMID (Dey & Sharon, 2024).
- **Domain:** Ant-V5, 1M training steps (see paper for more results).
- **Variants:** (1) no pruning, (2) Performance Pruning (+P), (3) Redundancy Pruning (+R), (4) both (+PR).



(a)

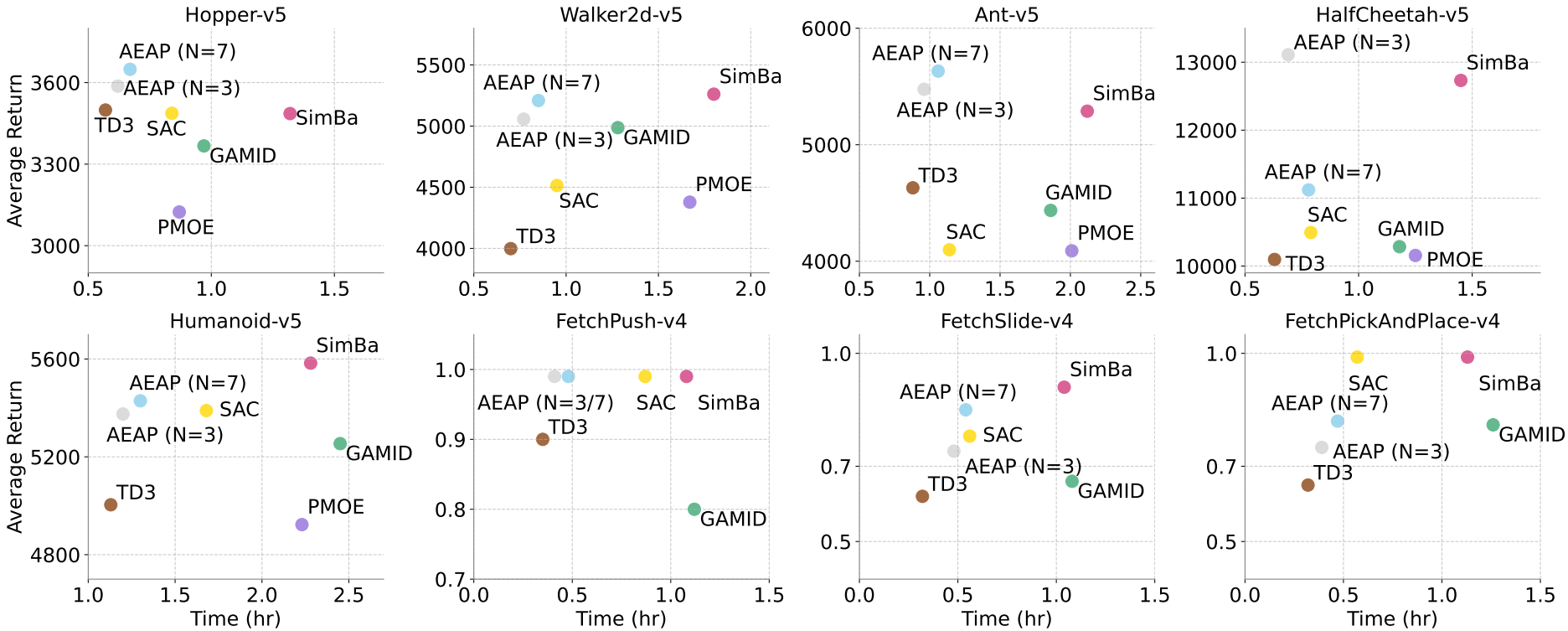


(b)

Figure (b): Probability densities for dual-randomized actor selection with and without pruning.

Performance-Efficiency Trade-off

Comparing: (1) AEAP-Conservative ($N = 3$), (2) AEAP-Aggressive ($N = 7$), (3) GAMID, (4) SAC, (5) PMOE-SAC, (6) SimBa, (7) TD3.
Domains: MuJoCo-v5 and Fetch-V4, 1M training steps.



AEAP is Pareto Optimal (computation time + return) in 7/8 tasks!

Summary

Part I: The Theoretical Foundation

- ☞ GMMs capture multimodal behavior AND allow DPG.
- ☞ DPG MaxEnt objective for GMMs via Variational KL approximation.
- ☞ **Lesson:** DPG *can* optimize multimodal GMMs effectively.

Summary

Part I: The Theoretical Foundation

- GMMs capture multimodal behavior AND allow DPG.
- DPG MaxEnt objective for GMMs via Variational KL approximation.
- **Lesson:** DPG *can* optimize multimodal GMMs effectively.

Part II: The Practical Solution (AEAP)

- **The Problem:** Explicit entropy regularization requires careful hyperparameter tuning (τ, ϵ).
- **The Solution:**
  - ✓ *Dual-Randomized Selection:* Implicit exploration without tuning.
  - ✓ *Adaptive Pruning:* Eliminates redundancy, improving efficiency.
- *Result:* AEAP offers a superior performance-efficiency trade-off compared to fixed ensembles and soft baselines.

Practical Takeaways

1. Reconsider Deterministic Gradients: For expressive policies like GMMs, deterministic gradients (with proper diversity mechanisms) can outperform stochastic gradients, especially in dense-reward settings.

Practical Takeaways

1. Reconsider Deterministic Gradients: For expressive policies like GMMs, deterministic gradients (with proper diversity mechanisms) can outperform stochastic gradients, especially in dense-reward settings.

2. Implicit vs. Explicit Exploration Regularization: Architectural choices (like randomizing actor selection) can provide robust exploration comparable to explicit MaxEnt terms, often with less tuning sensitivity.

Practical Takeaways

1. Reconsider Deterministic Gradients: For expressive policies like GMMs, deterministic gradients (with proper diversity mechanisms) can outperform stochastic gradients, especially in dense-reward settings.

2. Implicit vs. Explicit Exploration Regularization: Architectural choices (like randomizing actor selection) can provide robust exploration comparable to explicit MaxEnt terms, often with less tuning sensitivity.

3. Dynamic Ensembles are Efficient: You don't need to pay the computational cost of a full ensemble throughout training. Adaptive pruning allows you to "explore widely, then exploit cheaply".

Practical Takeaways

1. Reconsider Deterministic Gradients: For expressive policies like GMMs, deterministic gradients (with proper diversity mechanisms) can outperform stochastic gradients, especially in dense-reward settings.

2. Implicit vs. Explicit Exploration Regularization: Architectural choices (like randomizing actor selection) can provide robust exploration comparable to explicit MaxEnt terms, often with less tuning sensitivity.

3. Dynamic Ensembles are Efficient: You don't need to pay the computational cost of a full ensemble throughout training. Adaptive pruning allows you to "explore widely, then exploit cheaply".

4. AEAP Configuration Guide:

- ☞ **Stability needed?** Use Conservative ($N = 3, c = 1.0$).
- ☞ **Hard exploration/Sparse rewards?** Use Aggressive ($N = 7, c = 1.5$).

Collaborators: Sheelabhadra Dey, Komo Zhang.

- ✍ **Dey, S. & Sharon, G. (2024).** Comparing Deterministic and Soft Policy Gradients for Optimizing Gaussian Mixture Actors. *Transactions on Machine Learning Research*. `https://openreview.net/forum?id=qS9pPu8ODt`
- ✍ **Zhang, W. & Sharon, G. (2025).** AEAP: A Reinforcement Learning Actor Ensemble Algorithm with Adaptive Pruning. *Transactions on Machine Learning Research*. `https://openreview.net/forum?id=I5ymMVdmaR`