STAT 4830 - SPRING 2026

Sung Cho, Maxwell Delorenzo, Gyubin Han

Optimizing Hyperparameter Tuning Methods for *Evolutionary Strategies*

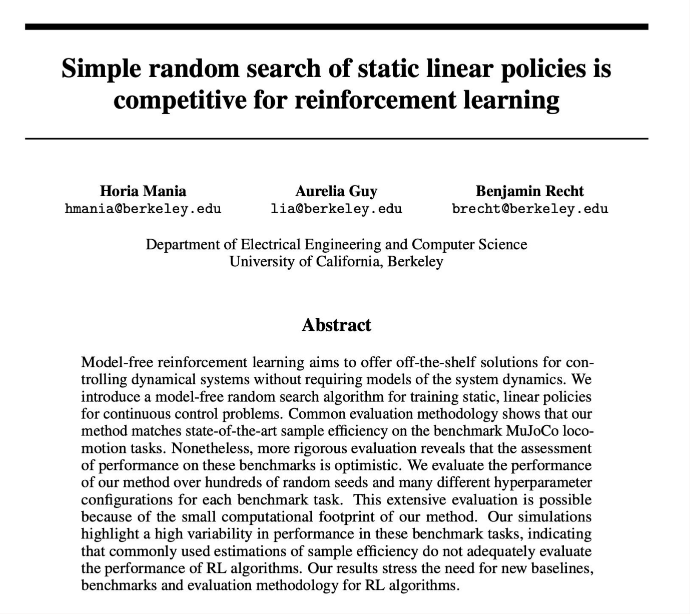
TABLE OF CONTENTS

1. Problem Statement
2. Technical Approach
3. Results
4. Limitations
5. Next Steps

01 Problem Statement

Problem Statement

- **Goal:** reproduce Mania et. al (2018) and test whether simple derivative-free augmented random search (ARS) can outperform Vanilla ES and REINFORCE under the same linear policy class and episode budget.
- **Tasks Implemented**: LQR and Pendulum (budget 3200 episodes per run)
- Success criteria: LQR eval_return >= -50, Pendulum threshold >= -200 (note rewards are negative and we're maximizing)
- **Why it matters**: isolates which ARS components actually drive performance (reward normalization, state normalization, top-b selection), which is useful for black-box optimization settings beyond RL.



Simple random search of static linear policies is competitive for reinforcement learning

Horia Mania hmania@berkeley.edu

Aurelia Guy lia@berkeley.edu

Benjamin Recht brecht@berkeley.edu

Department of Electrical Engineering and Computer Science
University of California, Berkeley

Abstract

Model-free reinforcement learning aims to offer off-the-shelf solutions for controlling dynamical systems without requiring models of the system dynamics. We introduce a model-free random search algorithm for training static, linear policies for continuous control problems. Common evaluation methodology shows that our method matches state-of-the-art sample efficiency on the benchmark MuJoCo locomotion tasks. Nonetheless, more rigorous evaluation reveals that the assessment of performance on these benchmarks is optimistic. We evaluate the performance of our method over hundreds of random seeds and many different hyperparameter configurations for each benchmark task. This extensive evaluation is possible because of the small computational footprint of our method. Our simulations highlight a high variability in performance in these benchmark tasks, indicating that commonly used estimations of sample efficiency do not adequately evaluate the performance of RL algorithms. Our results stress the need for new baselines, benchmarks and evaluation methodology for RL algorithms.

02
Technical Approach

Technical Approach: RL Algorithms

The RL Objective: $\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$

Policy Gradient

$$a \sim \pi_\theta(a), \quad a \in \{0,1\}^K$$

- Define a Bernoulli policy over parameters

$$\theta \leftarrow \theta + \alpha(R - b)\nabla_\theta \log \pi_\theta(a)$$

- Update parameters using REINFORCE

Evolutionary Strategies

$$\epsilon_i \sim \mathcal{N}(0, I), \quad R_i = R(\theta + \sigma\epsilon_i)$$

- Define policy parameters and apply Gaussian perturbations

$$\nabla_\theta J \approx \frac{1}{N}\sum_i (R_i - b)\epsilon_i.$$

- Calculate a gradient estimate

Technical Approach: ARS Algorithm

Linear Policy $\pi_\theta(s) = \theta s$

$$\text{For } k = 1, \dots, N: \quad \delta_k \sim \mathcal{N}(0, I), \quad r_k^+ = R(\pi_{\theta+\sigma\delta_k}), \quad r_k^- = R(\pi_{\theta-\sigma\delta_k})$$

Select top-b directions by $\max(r_k^+, r_k^-)$

$$\theta \leftarrow \theta + \frac{\alpha}{b\,\sigma_R} \sum_{k \in \text{top-}b} (r_k^+ - r_k^-)\delta_k$$

Technical Approach: Evaluation Protocol

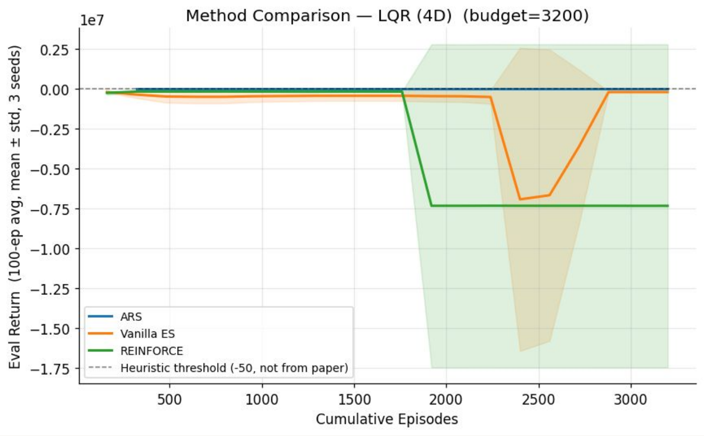
- **Main sweep:** 3 seeds
- **Two-phase protocol:** grid search (2*2*2 configs) then multi-seed eval
  - Grid phase used 3 fixed seeds, evaluating hyperparameter grid and select best config by minimum mean episodes-to-threshold.
  - Eval phase ran the selected config on many unseen seeds and report final performance distribution
- Eval uses frozen running norm (no norm updates during, preventing contamination of test performance by eval trajectories
- **Fairness controls**: same episode budget, evaluation cadence, and number of eval rollouts per checkpoint across methods

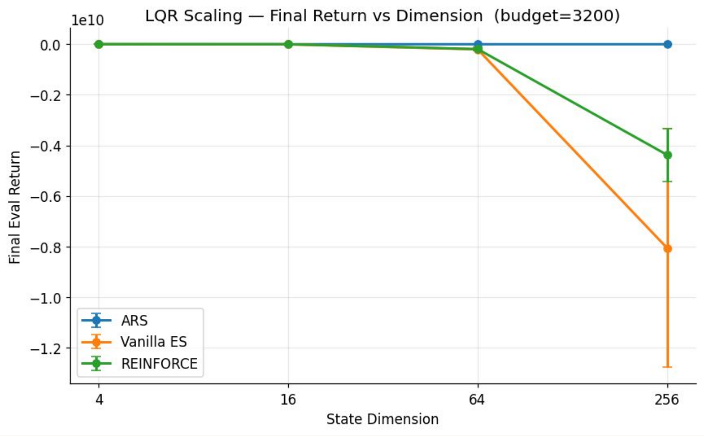
03

Results

Results

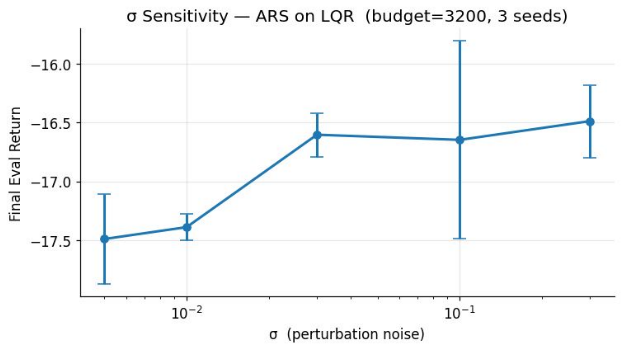
- **Core Performance and Methods Comparison**
  - ARS outperforms baselines and produces a rapid convergence for the LQR environment
  - Additional benefits stem from normalization and insensitivity to initialization
- **Ablation and Dimension Scaling**
  - In higher dimensions, ARS doesn’t collapses while Vanilla ES and REINFORCE collapses exponentially

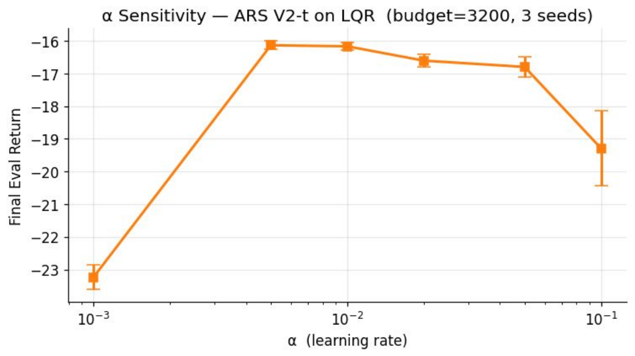


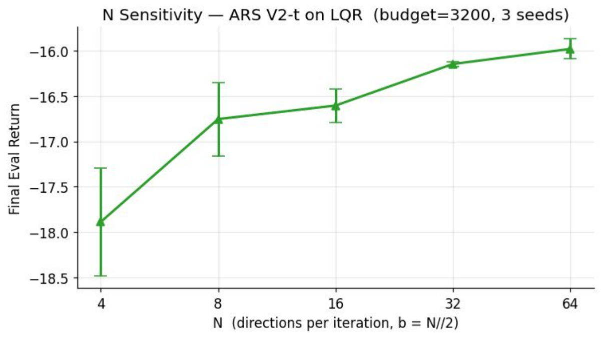


Results

- **Hyperparameter Sensitivity**
  - ARS remained stable across a full decade [0.03, 0.3] of noise standard deviation σ. Meaningful degradations only occur when perturbations are too small
  - There is a tight bound on the optimal range [0.005, 0.01] for the learning rate α
  - Performance improve monotonically as population N increases, but experiences strong diminishing returns


σ Sensitivity — ARS on LQR (budget=3200, 3 seeds)


α Sensitivity — ARS V2-t on LQR (budget=3200, 3 seeds)


N Sensitivity — ARS V2-t on LQR (budget=3200, 3 seeds)

04

Limitations

Limitations

- External validity: no MuJoCo locomotion results in current artifacts, so headline paper claim is not yet directly reproduced.
- Multi-seed reliability: protocol directory contains 20-seed eval, not 100-seed
- Narrow method-space coverage: no comparison against stronger modern baselines (e.g. PPO/TRPO)
- Reproduction vs novelty gap: the work is currently strongest as a careful reproduction/ablation study

05

Next Steps

Next Steps

- **Fully reimplement the paper**
    - Expand Pendulum search (N up to 64, budget 10000+) and rerun two-phase protocol
    - Execute MuJoCo fully to test transfer from toy tasks
- **LLM Extension**
    - Thesis: ARS-style updates are competitive for gradient-free prompt/policy tuning under sparse scalar rewards in LLM setting
- **Adaptive ARS**
    - Thesis: dynamic schedule with adaptive N, sigma, and b outperforms fixed hyperparameters under fixed episode budget

THANK YOU

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**