Gradient Informed Proximal Policy Optimization

Sanghyun Son Laura Zheng Ryan Sullivan Yi-Ling Qiao Ming Lin

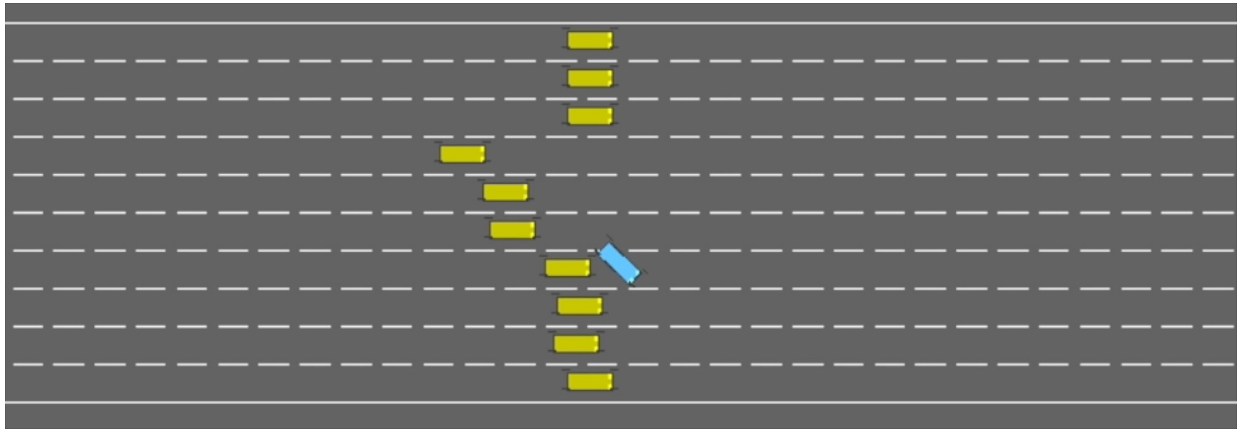
University of Maryland, College Park

UNIVERSITY OF MARYLAND

We study ...

"Application of analytical gradients in PPO framework"

Gradient about world dynamics



e.g., How will the other vehicles react to my vehicle's action, according to traffic model?

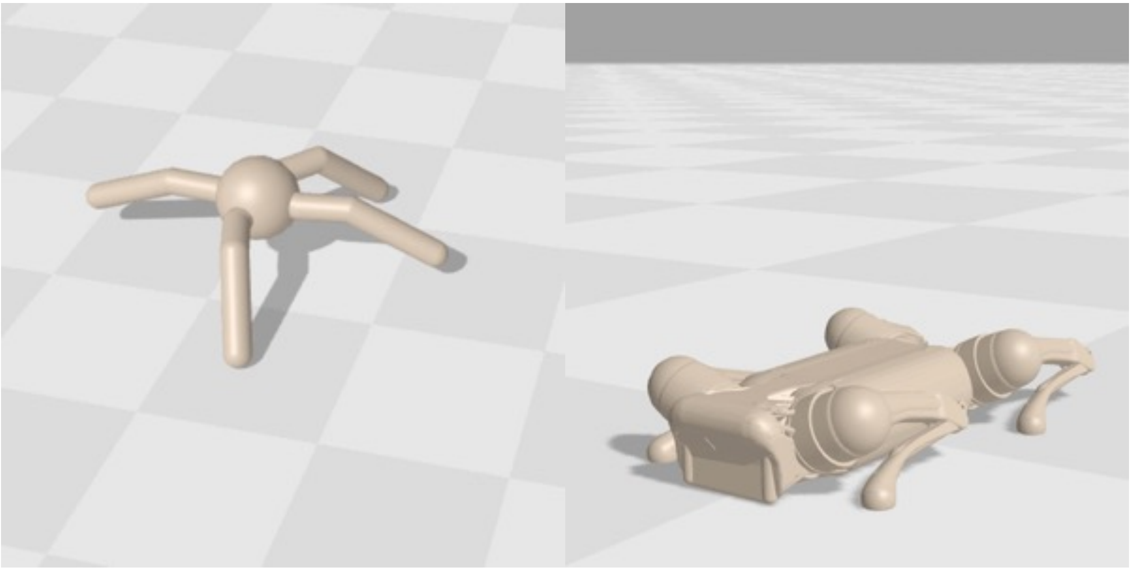
We study …

"Application of analytical gradients in PPO framework"

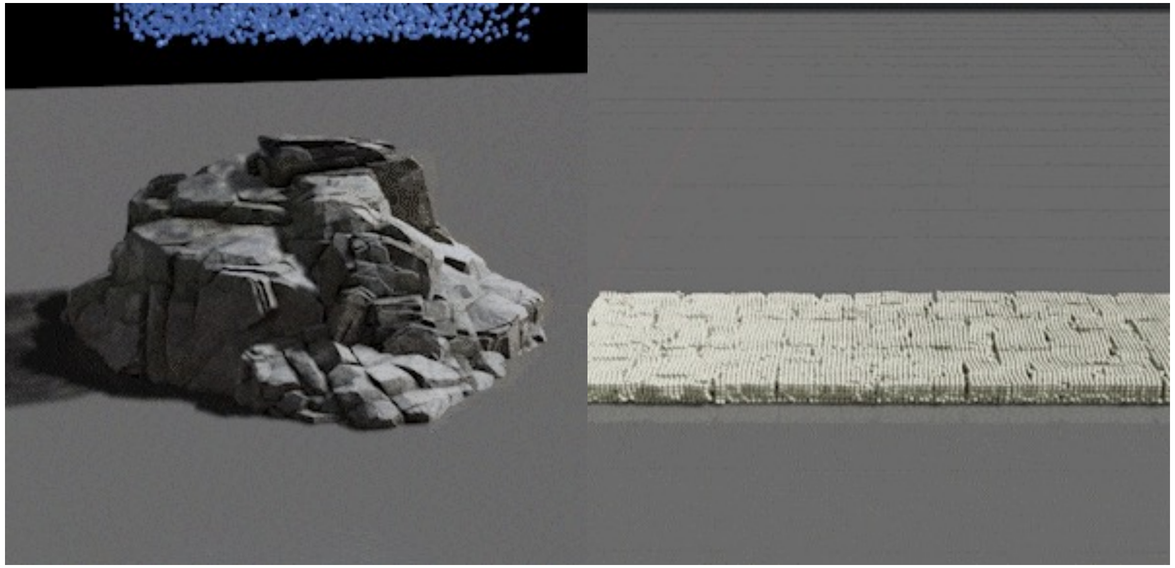
Has been one of the most widely used
model-free on-policy RL algorithms

First attempt to use analytical gradients in this scenario!

Background



Brax simulator, Google



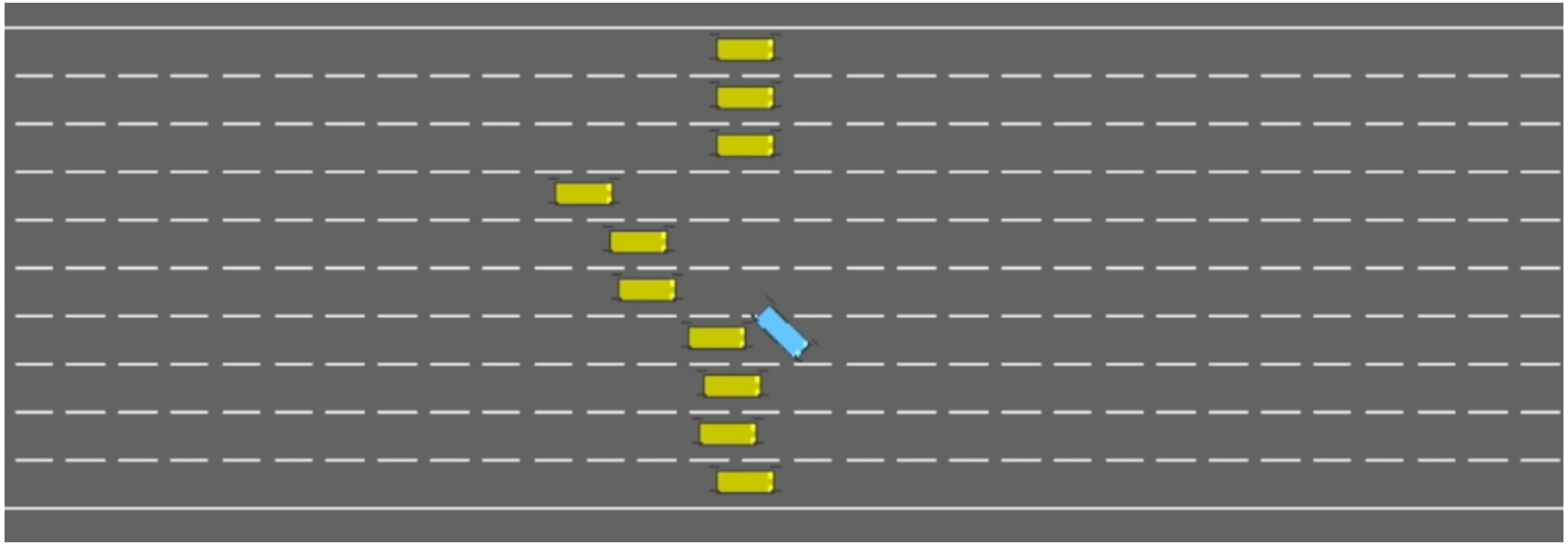
Warp simulator, NVIDIA

(Fully) Differentiable physics simulations provide gradients used for training

However, **complete differentiability is often hard to achieve**!

Freeman, C. Daniel, et al. "Brax--A Differentiable Physics Engine for Large Scale Rigid Body Simulation." *arXiv preprint arXiv:2106.13281* (2021).
https://github.com/NVIDIA/warp

Background



e.g., In traffic environment, lane change is a discrete behavior

Gradients become **biased** when a vehicle changes its lane

How can we leverage this **biased** gradient in PPO framework?

Preliminaries: Problem definitions

- Our problem: Markov Decision Process (MDP)

State transition (dynamics) Reward discount

$$(S, A, P, r, \rho_0, \gamma),$$

state action reward Initial state distribution

- Goal: Train a parameterized stochastic policy to maximize its expected sum of discounted rewards

$$\pi_\theta : S \times A \to \mathbb{R}^+, \quad \eta(\pi_\theta) = \mathbb{E}_{s_0, a_0, \ldots \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right].$$

Stochastic policy, θ = parameters Expected sum of discounted rewards

Preliminaries: Analytical gradients

- Assumption:
  - State and action spaces (S, A) are continuous, differentiable spaces
  - Dynamics and reward models (P, r) are differentiable models

$$\frac{\partial s_{t+1+k}}{\partial a_t}, \frac{\partial r_{t+k}}{\partial a_t},$$

Then, environment provides above basic analytical gradients!
With these, we can compute analytical gradients for advantage functions…

$$\frac{\partial A}{\partial a}$$

We use Generalized Advantage Estimator (GAE).
Please see our paper for derivation!

Schulman, John, et al. "High-dimensional continuous control using generalized advantage estimation." *arXiv preprint arXiv:1506.02438* (2015).

Preliminaries: Policy Updates (RP)

- Reparameterization (RP) Gradient based approach
  - Sample an action from our stochastic policy by sampling a random variable ϵ from another independent distribution q.

$$g_\theta(s, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

For a given ϵ (sampled from q), g maps it to an action that we use.

g is a bijective function!

$$\left|\det\left(\frac{\partial g_\theta(s,\epsilon)}{\partial \epsilon}\right)\right| > 0, \forall \epsilon \in \mathbb{R}^n$$

Because of bijectivity, above relationship holds.

$$q = \mathcal{N}(0, I)$$

$$g_\theta(s,\epsilon) = \mu_\theta(s) + \sigma_\theta(s) \cdot \epsilon \quad (||\sigma_\theta(s)||_2 > 0),$$

In our work, we use above q and g.

Preliminaries: Policy Updates (RP)

- Reparameterization (RP) Gradient based approach
  - Sample an action from our stochastic policy by sampling a random variable ϵ from another independent distribution q.

$$\pi_\theta \triangleq g_\theta$$

A Function g is "equivalent" to a stochastic policy if…

$$\int_{T_a} \pi_\theta(s,a)da = \int_{T_\epsilon} q(\epsilon)d\epsilon,$$

$$T_a = g_\theta(s, T_\epsilon).$$

Please see our paper (Definition 3.1) for details!

Preliminaries: Policy Updates (RP)

- Reparameterization (RP) Gradient based approach

$$\frac{\partial \eta(\pi_\theta)}{\partial \theta} = \mathbb{E}_{s_0,\epsilon_0,\ldots\sim q}\left[\sum_{t=0}^{\infty}\sum_{k=t}^{\infty}\gamma^k \frac{\partial g_\theta(s_t,\epsilon_t)}{\partial \theta}\frac{\partial r(s_k,a_k)}{\partial a_t}\right].$$

*As shown here, we use deterministic function **g** to compute RP gradient of the current policy, which can be used for gradient ascent.*

Preliminaries: Policy Updates (PPO)

- Proximal Policy Optimization (PPO) based approach

$$\eta(\pi_\theta) = \eta(\pi_{\bar{\theta}}) + \int_s \rho_{\pi_\theta}(s) \int_a \pi_\theta(s,a) A_{\pi_{\bar{\theta}}}(s,a),$$

We can evaluate a policy (π_θ) with our current policy ($\pi_{\bar{\theta}}$).
However, since ρ_{π_θ} is not available…

$$L_{\pi_{\bar{\theta}}}(\pi_\theta) = \eta(\pi_{\bar{\theta}}) + \int_s \rho_{\pi_{\bar{\theta}}}(s) \int_a \pi_\theta(s,a) A_{\pi_{\bar{\theta}}}(s,a).$$

Use this surrogate loss function and maximize it!
It holds as far as π_θ is "not very different" from $\pi_{\bar{\theta}}$.

Preliminaries: Policy Updates (PPO)

- Proximal Policy Optimization (PPO) based approach

$$1 - \epsilon_{clip} < \frac{\pi_\theta(s_i, a_i)}{\pi_{\bar{\theta}}(s_i, a_i)} < 1 + \epsilon_{clip}.$$

PPO enforces π_θ to stay near $\pi_{\bar{\theta}}$ using above condition.

Approach: α-policy

- When we are given the gradient of advantage function with respect to an action, $\frac{\partial A}{\partial a}$ we can define an α-policy (π_α) of current policy ($\pi_{\bar{\theta}}$) as follows:

$$\pi_\alpha(s, \tilde{a}) = \begin{cases} \frac{\pi_{\bar{\theta}}(s,a)}{\boxed{|\det(I+\alpha\nabla_a^2 A_{\pi_{\bar{\theta}}}(s,a))|}} & \text{if} \quad \exists a \ s.t.\ \tilde{a} = f(a) = a + \alpha \cdot \nabla_a A_{\pi_{\bar{\theta}}}(s,a) \\ 0 & \text{else} \end{cases}$$

1. We define a better action $\tilde{a}$ than a using gradient

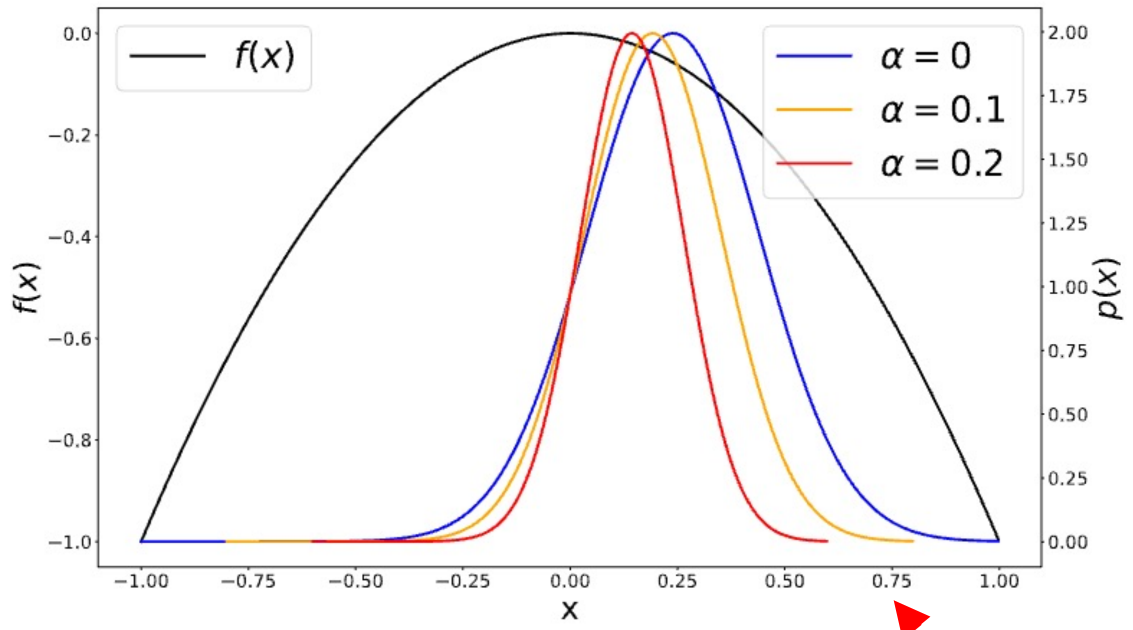
2. Then, intuitively, π_α can be thought of as a policy that selects "slightly better" action than $\pi_{\bar{\theta}}$ with the "same" probability!

(Denominator is used to make π_α a valid policy, which sums to 1)

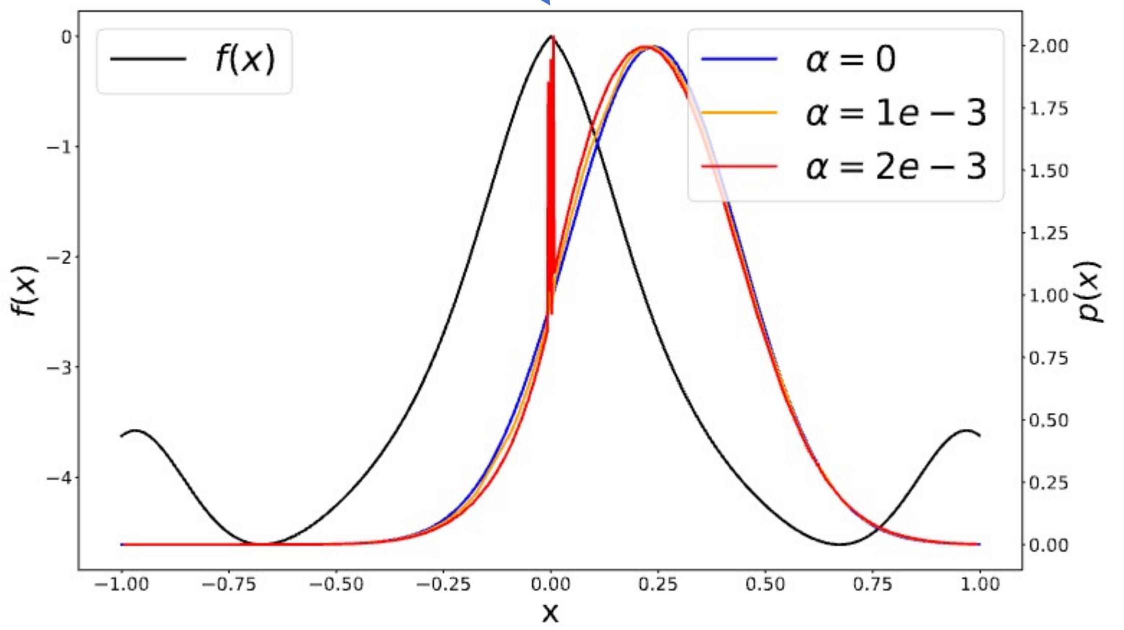
Please see our paper (Definition 4.1, Lemma 4.2) for details!

α-policy

However, when α is large ($\mathbf{2e^{-3}}$), π_α is not well-defined!



(a) De Jong's Function



(b) Ackley's Function

π_α selects slightly better action than original policy, which ends up in slightly better policy

1. *Reward function (black)*
2. *Probability distribution of original policy (blue)*
3. *Probability distribution of α-policies for different αs (yellow, red)*

α-policy

$$\pi_\alpha(s,\tilde{a}) = \begin{cases} \frac{\pi_{\bar{\theta}}(s,a)}{|\det(I+\alpha\nabla_a^2 A_{\pi_{\bar{\theta}}}(s,a))|} & \textit{if} \quad \exists a \ s.t. \ \tilde{a} = f(a) = a + \alpha \cdot \nabla_a A_{\pi_{\bar{\theta}}}(s,a) \\ 0 & \textit{else} \end{cases}$$

$$\textit{where constant } |\alpha| < \frac{1}{\max_{(s,a)}|\lambda_1(s,a)|}.$$

Here $\lambda_1(s,a)$ *represents the minimum eigenvalue of* $\nabla_a^2 A_{\pi_{\bar{\theta}}}(s,a)$.

Condition to make π_α a valid policy

** α should be sufficiently small!*

α-policy (PPO's viewpoint)

Proposition 4.3 *If* $|\alpha| \ll 1$,

$$\boxed{L_{\pi_{\bar{\theta}}}(\pi_\alpha)} - \eta(\pi_{\bar{\theta}}) = O(|\alpha|) \textit{ when } \alpha > 0,$$
$$\eta(\pi_{\bar{\theta}}) - L_{\pi_{\bar{\theta}}}(\pi_\alpha) = O(|\alpha|) \textit{ when } \alpha < 0,$$

where $L_{\pi_{\bar{\theta}}}$ *denotes estimated expected return defined in Equation* $\boxed{4}$.

In fact, if $\boldsymbol{\alpha > 0}$ is sufficiently small,
$\boldsymbol{\pi_\alpha}$ is better than $\boldsymbol{\pi_{\bar{\theta}}}$ in the PPO's framework!

$$\boxed{L_{\pi_{\bar{\theta}}}(\pi_\theta) = \eta(\pi_{\bar{\theta}}) + \int_s \rho_{\pi_{\bar{\theta}}}(s) \int_a \pi_\theta(s,a) A_{\pi_{\bar{\theta}}}(s,a).}$$

α-policy (PPO's viewpoint)

*Therefore, if we update our policy to **α**-policy,*
it aligns with PPO's objective.

*However, how can we gain **α**-policy?*
It has second-order derivative in its definition…

$$\pi_\alpha(s,\tilde{a}) = \begin{cases} \frac{\pi_{\bar{\theta}}(s,a)}{|\det(I+\alpha\nabla_a^2 A_{\pi_{\bar{\theta}}}(s,a))|} & \textit{if} \quad \exists a \; s.t. \; \tilde{a} = f(a) = a + \alpha \cdot \nabla_a A_{\pi_{\bar{\theta}}}(s,a) \\ 0 & \textit{else} \end{cases}$$

Computing α-policy

Assumption: $\pi_{\bar{\theta}} \triangleq g_{\bar{\theta}}$

Definition: $g_\alpha(s, \epsilon) = \boxed{a + \alpha \cdot \nabla_a A_{\pi_{\bar{\theta}}}(s, a),}$ where $a = g_{\bar{\theta}}(s, \epsilon)$.

Note that $\boldsymbol{g_\alpha}$ shares the same spirit as $\boldsymbol{\pi_\alpha}$ -
it selects "slightly better" action than the original mapping $\boldsymbol{g_{\bar{\theta}}}$ for the same $\boldsymbol{\varepsilon}$.

Computing α-policy

Proposition 4.5 *If* $\pi_{\bar{\theta}} \triangleq g_{\bar{\theta}}$*, for* α *that satisfies the constraint in Definition* $\boxed{4.1}$*,* $\pi_\alpha \triangleq g_\alpha$*.*

In fact, not only π_α *and* g_α *share the same spirit, they are "equivalent"!*

That is, we can gain π_α *by approximating* g_α*, which is possible by minimizing following loss:*

$$L(\theta) = \mathbb{E}_{s_0,\epsilon_0,\ldots\sim q}\left[||g_\theta(s_t,\epsilon_t) - g_\alpha(s_t,\epsilon_t)||^2\right].$$

α-policy (RP's viewpoint)

If we use following advantage function for defining g_α,

$$\hat{A}_{\pi_\theta}(s_t, a_t) = \frac{1}{2}\mathbb{E}_{s_t, a_t, \ldots \sim \pi_\theta}\left[\sum_{k=t}^{\infty} \gamma^k r(s_k, a_k)\right],$$

The RP gradient corresponds to $\frac{\partial L}{\partial \theta}$.

$$\frac{\partial \eta(\pi_\theta)}{\partial \theta} = \mathbb{E}_{s_0, \epsilon_0, \ldots \sim q}\left[\sum_{t=0}^{\infty}\sum_{k=t}^{\infty} \gamma^k \frac{\partial g_\theta(s_t, \epsilon_t)}{\partial \theta}\frac{\partial r(s_k, a_k)}{\partial a_t}\right].$$

$$L(\theta) = \mathbb{E}_{s_0, \epsilon_0, \ldots \sim q}\left[||g_\theta(s_t, \epsilon_t) - g_\alpha(s_t, \epsilon_t)||^2\right].$$

Please see our paper (Lemma 4.6) for details!

α-policy

To sum up, $\boldsymbol{\alpha}$-policy is a policy that aligns with both RP and PPO method, where $\boldsymbol{\alpha}$ stands for the influence of analytical gradients.

We can approximate $\boldsymbol{\alpha}$-policy by minimizing regression loss function $\boldsymbol{L}$.

Algorithm

1. Update current policy to α-policy

2. Adjust α for next iteration

3. Update again using PPO-based approach

PPO can be regarded a ***safeguard*** *that promises certain amount of policy update even when the analytical gradients are undesirable and therefore* $\boldsymbol{\alpha} = 0$.

Therefore, our algorithm is tightly bounded to PPO!

Algorithm

1. Update current policy to α-policy

*We can do it by minimizing regression loss **L** shown before.*

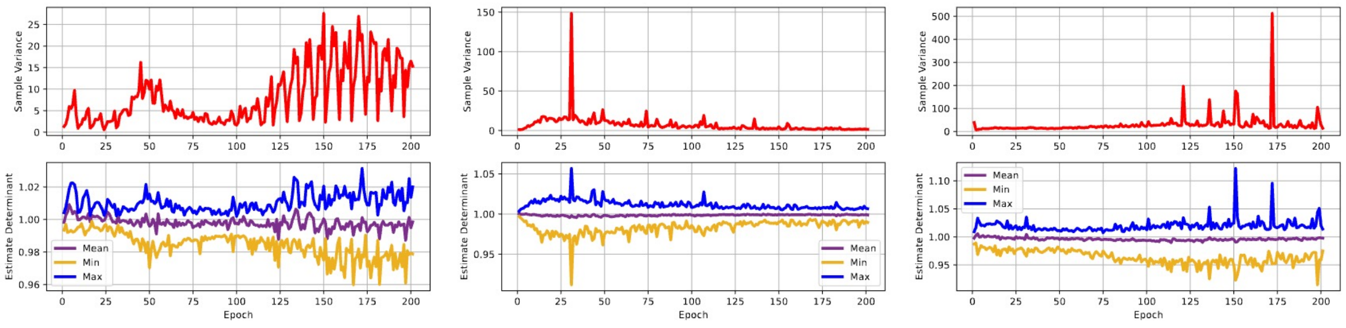
$$L(\theta) = \mathbb{E}_{s_0,\epsilon_0,\ldots\sim q}\left[||g_\theta(s_t,\epsilon_t) - g_\alpha(s_t,\epsilon_t)||^2\right].$$

Algorithm

2. Adjust α for next iteration

✓ Bias : Analytical gradients can be biased, not only explicitly, but also implicitly. Use PPO's formulation to detect biasedness.

✓ Variance : Analytical gradients can exhibit exploding gradients. Detect this case using our Lemma 4.4.



(a) Cartpole (b) Ant (c) Hopper

We can estimate the sample variance of analytical gradients (upper row) using statistics we get after we update our policy (lower row) to α-policy in step 1 much more efficiently!

Please see our paper (Section 4.3.1) for details!

Algorithm

2. Adjust α for next iteration

✓ Out-of-range-ratio : Since PPO requires the updated policy to stay near current policy, we adjust α so that α-policy is not far away from current policy.

$$\text{out-of-range-ratio} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(\left|\frac{\pi_\theta(s_i, a_i)}{\pi_{\bar{\theta}}(s_i, a_i)} - 1\right| > \epsilon_{clip}\right),$$

Adjust $\boldsymbol{\alpha}$ to maintain this value under certain threshold!

This is the main reason why our method is tightly bound to PPO.

Please see our paper (Section 4.3.1) for details!

Algorithm

3. Update again using PPO-based approach

$$\pi_h(s,a) = \frac{1}{2}(\pi_{\bar{\theta}}(s,a) + \pi_\alpha(s,a)).$$

Use this function for importance sampling function to preserve updates from step 1.

Please see Algorithm 1 for the entire algorithm!

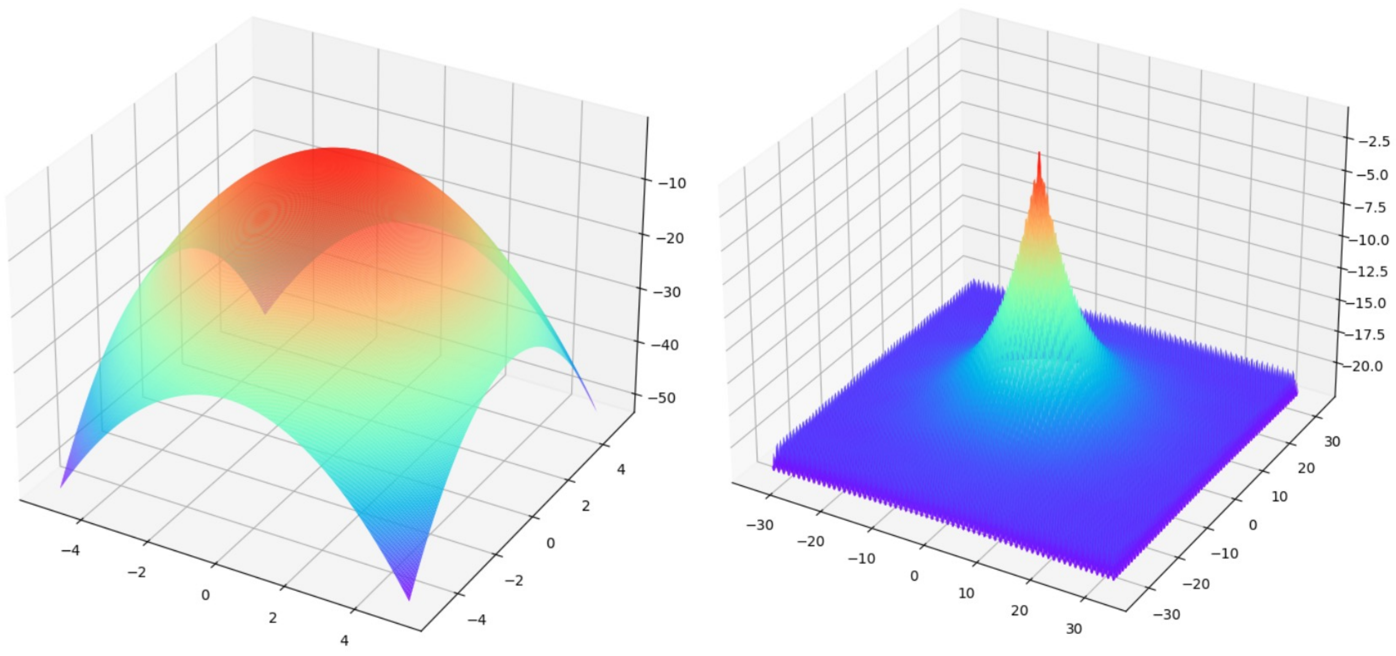
Experimental Results

Algorithms used for comparisons

- LR: Policy gradient method based on LR gradient.
- RP: Policy gradient method based on RP gradient. For physics and traffic problems, we adopted a truncated time window of [Xu et al., 2022] to reduce variance.
- PPO: Proximal Policy Optimization [Schulman et al., 2017].
- LR+RP: Policy gradient method based on interpolation between LR and RP gradient using sample variance [Parmas et al., 2018].
- PE: Policy enhancement scheme of [Qiao et al., 2021], for physics environments only.
- GI-PPO: Our method based on Section 4.3.

Please see our paper (Appendix 7.4) for details!

Experimental Results: Function Optimization



(a) De Jong's Function (b) Ackley's Function

Figure 8: Landscape of target functions in 2 dimensions.

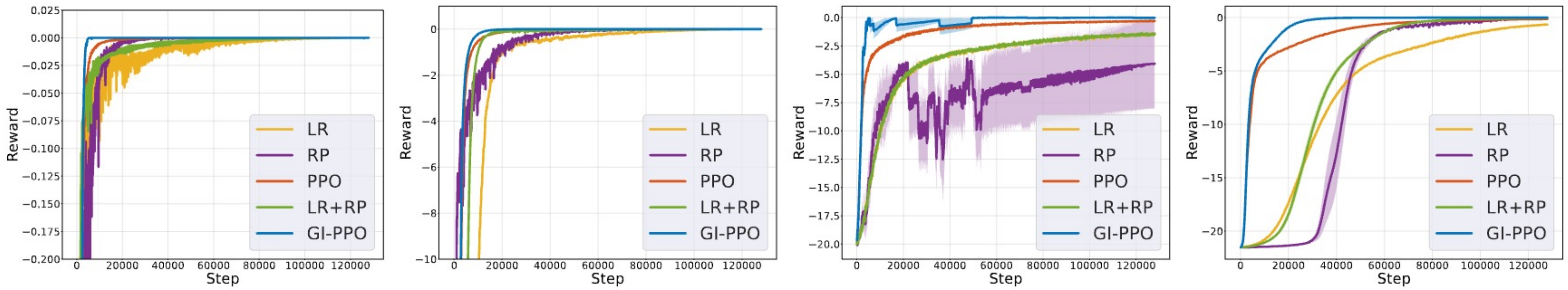
Smooth landscape, Smaller variance

Noisy landscape, Higher variance

Experimental Results: Function Optimization

Table 1: Average maximum reward (↑) for function optimization problems.

Problem	LR	RP	PPO	LR+RP	GI-PPO
Dejong (1)	$-1.24*10^{-6}$	$-1.42*10^{-8}$	$-5.21*10^{-5}$	$-6.36*10^{-8}$	$\mathbf{-3.84*10^{-10}}$
Dejong (64)	-0.0007	$\mathbf{-9.28*10^{-7}}$	-0.0011	$-3.05*10^{-6}$	$-1.04*10^{-6}$
Ackley (1)	-1.2772	-0.4821	-0.2489	-1.2255	**-0.0005**
Ackley (64)	-0.6378	-0.0089	-0.1376	-0.0326	**-0.0036**

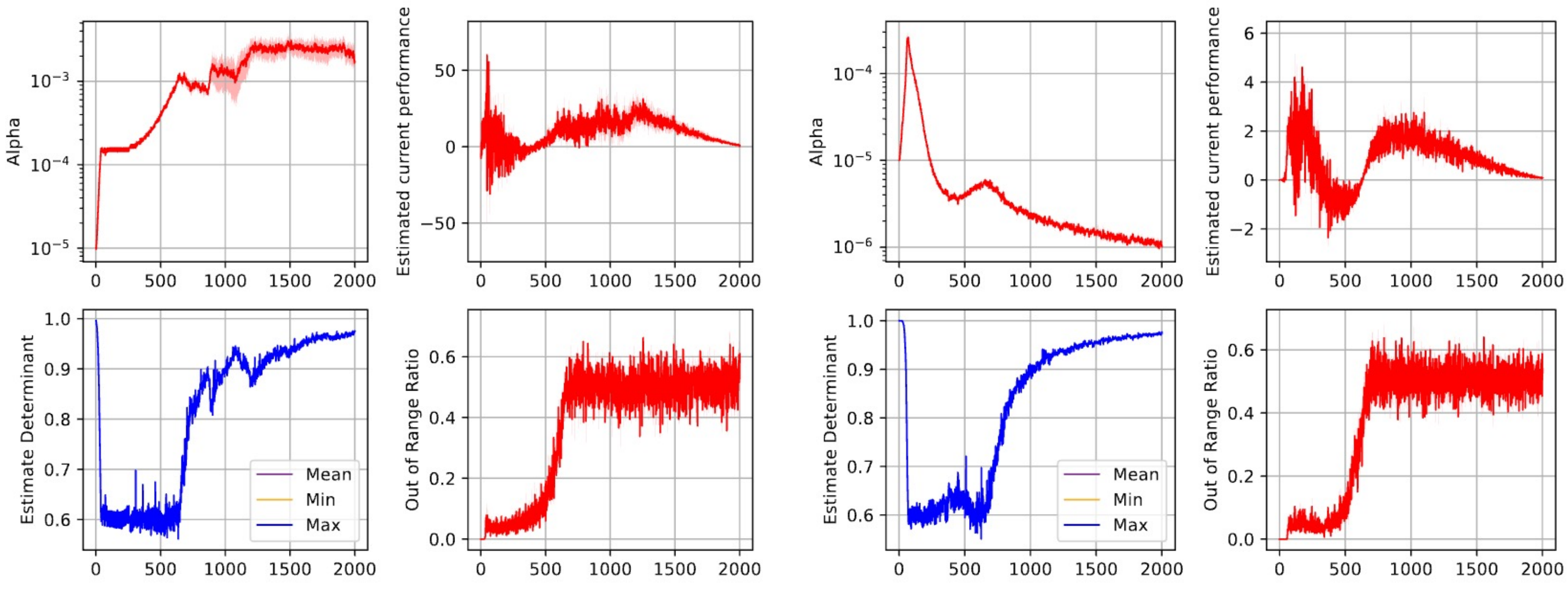


(a) De Jong (1) (b) De Jong (64) (c) Ackley (1) (d) Ackley (64)

Figure 2: Optimization curves for Dejong's and Ackley's function of dimension 1 and 64.

Faster convergence to better optimal values!

Experimental Results: Function Optimization

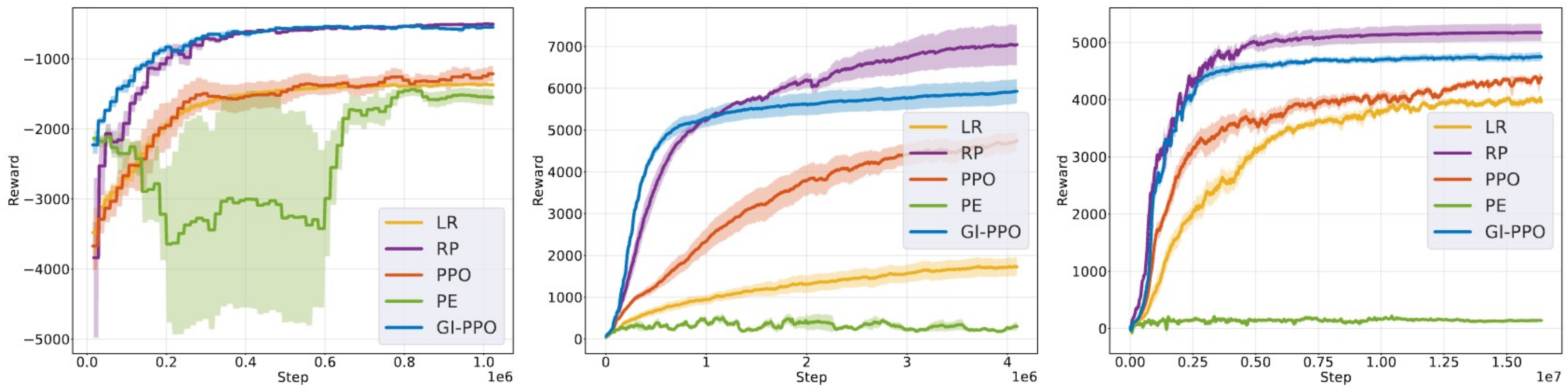


(a) De Jong's Function (64)

(b) Ackley's Function (64)

Change of α: Note that higher α is maintained in De Jong's function

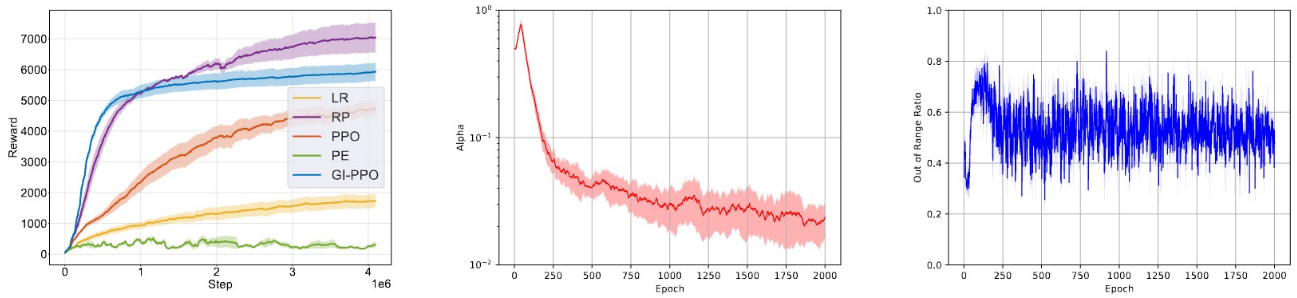
Experimental Results: Physics Simulation



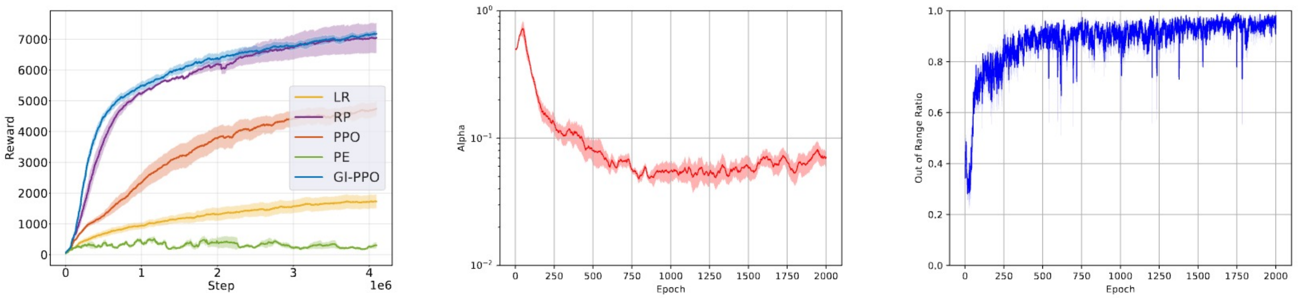
(a) Cartpole (b) Ant (c) Hopper

Our method achieved far better results than the baseline PPO, but could not do better than RP in Ant and Hopper.

Experimental Results: Physics Simulation



(a) Statistics of training in **Ant** environment when $\delta_{oorr} = 0.5$.

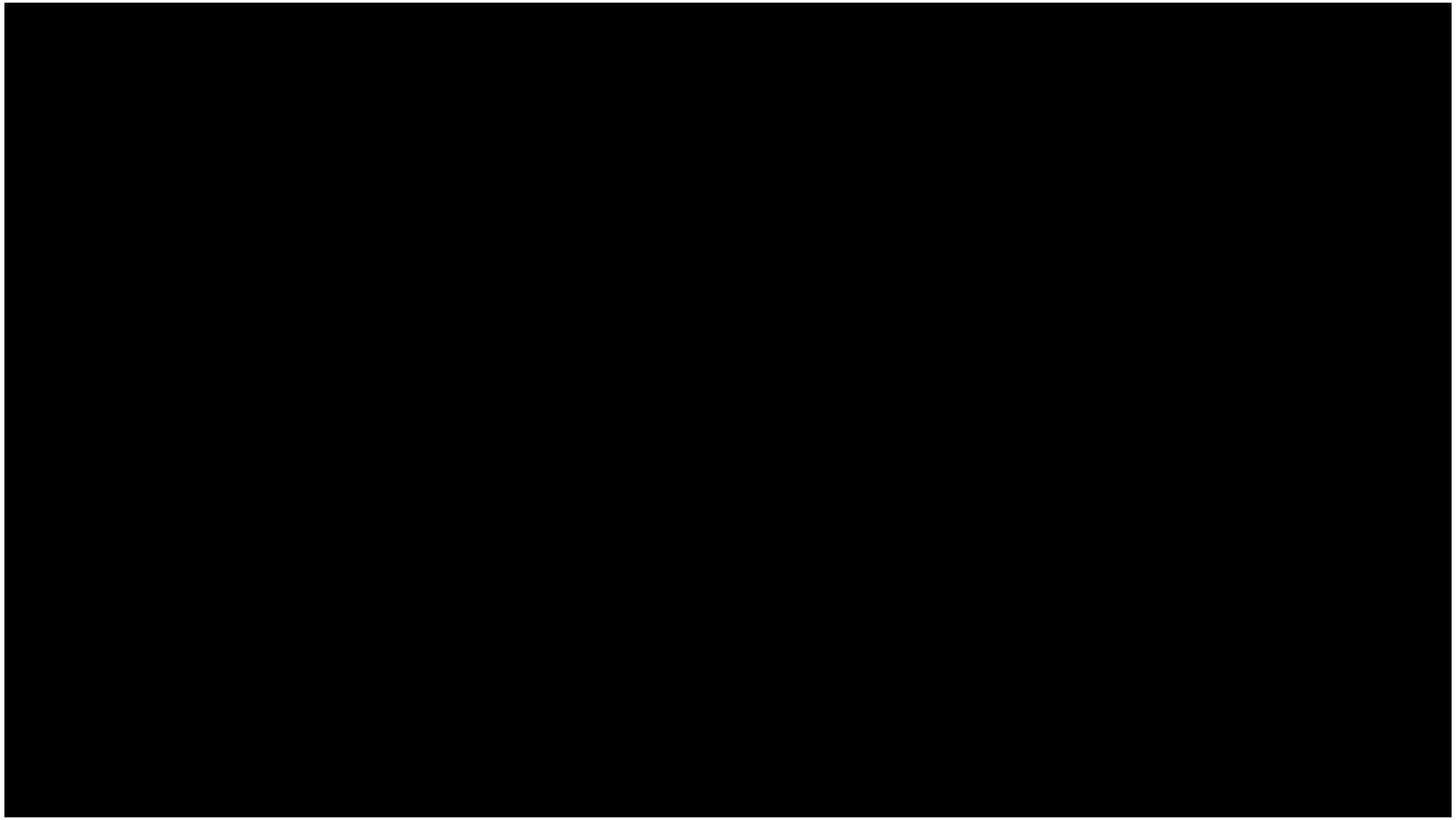


(b) Statistics of training in **Ant** environment when $\delta_{oorr} = 1.0$.

This is because our method is tightly bound to PPO.
If we do not bound it to PPO, our method performs better.

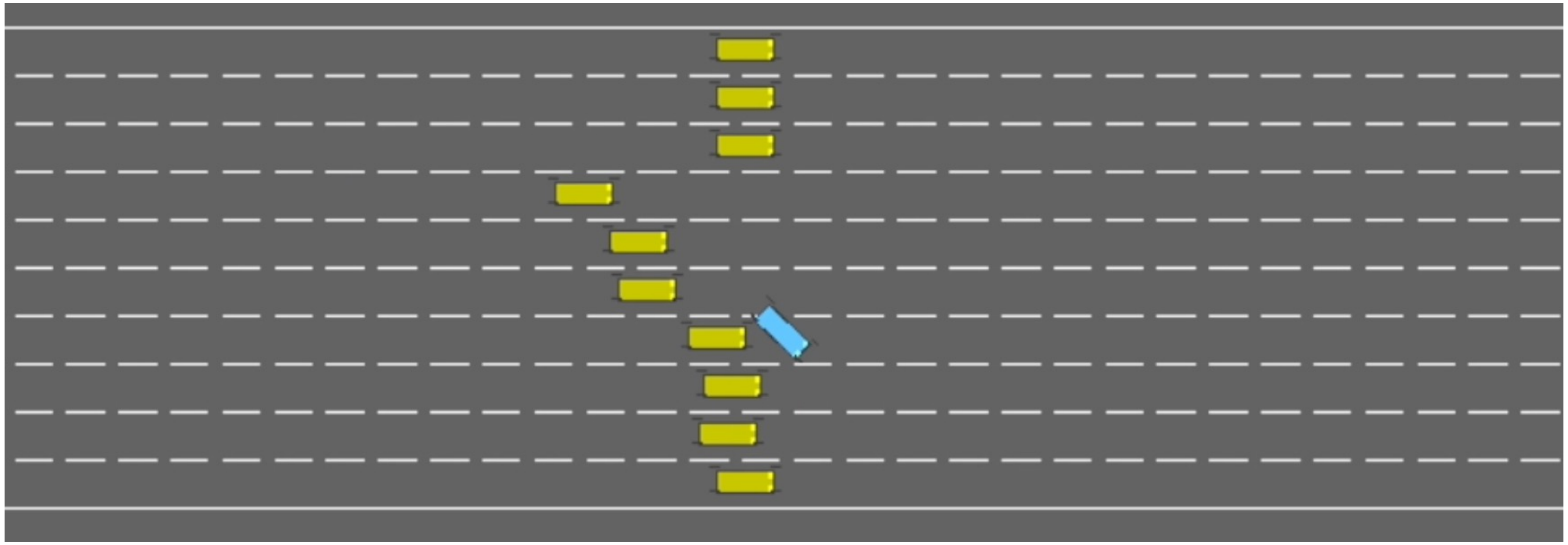
However, we cannot detect such cases yet, with our current approach...

Experimental Results: Traffic simulation



Our vehicle should intervene other vehicles to regulate speed.

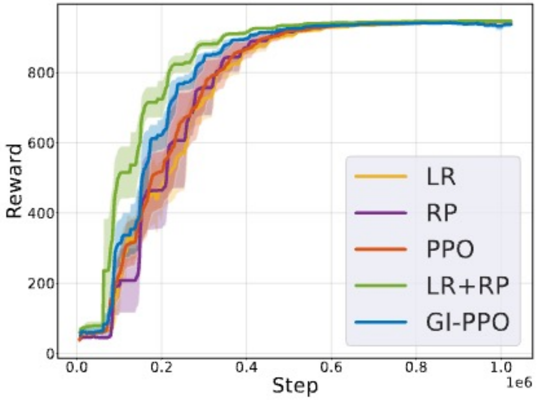
Experimental Results: Traffic simulation



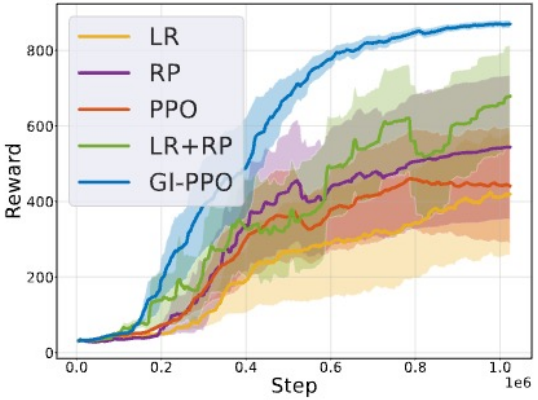
Represents environment with biased gradient

Even though our method uses the biased gradients, since it uses PPO as a safeguard, our method can still exploit useful information from the biased gradient!

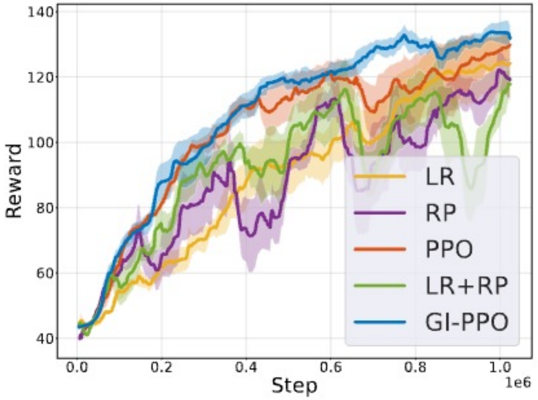
Experimental Results: Traffic simulation



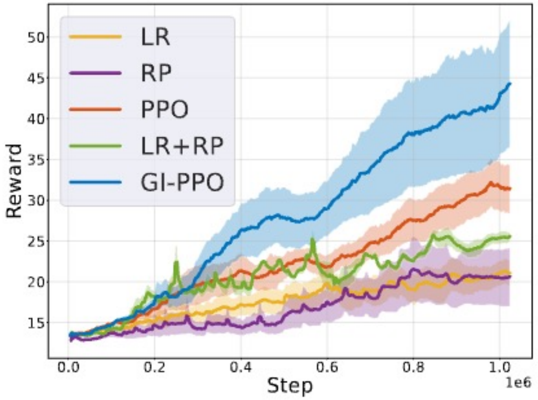
(a) Single Lane



(b) 2 Lanes



(c) 4 Lanes



(d) 10 Lanes

Achieved best results in most of the environments!

Experimental Results: Computational cost

Table 2: Average (wall clock) training time (sec) for traffic problems.

Problem	LR	RP	PPO	LR+RP	GI-PPO
Single Lane	120	282	181	1335	332
2 Lanes	142	330	218	1513	411
4 Lanes	304	646	366	2093	813
10 Lanes	244	496	294	2103	533

Since we should compute analytical gradients and do PPO updates, it is a little bit slower than RP, which also computes analytical gradients.

However, faster than LR+RP, which also combines analytical gradients with LR gradients, which corresponds to PPO in our case.

Conclusion

- We presented a novel approach to leverage analytical gradients in PPO framework.
- We defined α-policy, where α stands for the influence of the analytical gradients. We suggested criteria to adaptively change α during training, to find balance between analytical gradients and PPO.
- We achieved much better learning results than the baseline PPO in every environment, even in the challenging environments with biased gradients.

Limitations

- Our method is tightly bound to PPO. Therefore, even when the analytical gradients are much more useful, we cannot fully utilize them.

- Our approach to control α is naïve, not optimal – there are still a lot of rooms for developing another fine-grained algorithm.

Thank you.