How to Fine-tune the Model: Unified Model Shift and Model Bias Policy Optimization

Hai Zhang Hang Yu Junqiao Zhao* Di Zhang Chang Huang Hongtu Zhou Xiao Zhang Chen Ye

Department of Computer Science, Tongji University, Shanghai, China MOE Key Lab of Embedded System and Service Computing, Tongji University, Shanghai, China

Introduction

Core Problem: How to adaptively adjust the impacts of model shift to get a better performance improvement guarantee ?

- MBPO[1]-style: Return Discrepancy Scheme

$$V^{\pi|M} \geq V^{\pi}_{\hat{M}} - C(\epsilon_m, \epsilon_\pi) \tag{1}$$

1. Does not consider the impacts of model shift.

- CMLO[2]-style: Performance Difference Bound Scheme

$$\begin{aligned} &V^{\pi_2|M_2} - V^{\pi_1|M_1} \\ &\geq \kappa(\mathbb{E}_{s,a\sim d^{\pi_1}} D_{TV}(P||P_{M_1}) - \mathbb{E}_{s,a\sim d^{\pi_2}} D_{TV}(P||P_{M_2})) - \frac{\gamma}{1-\gamma} L(2\sigma_{M_1,M_2}) - \epsilon_{opt} \\ &s.t. D_{TV}(P_{M_2}||P_{M_1}) \leq \sigma_{M_1,M_2}, \forall (s,a) \in \mathcal{S} \times \mathcal{A} \end{aligned} \tag{2}$$

1. The lower threshold impairs the subsequent optimization process.
2. The bigger threshold collapses the performance improvement guarantee.
3. Fixed threshold lacks flexibility.

An illustrative experiment

To valid our statement, we devise an experiment that sets three different thresholds for CMLO on Walker2d environment in MuJoCo.

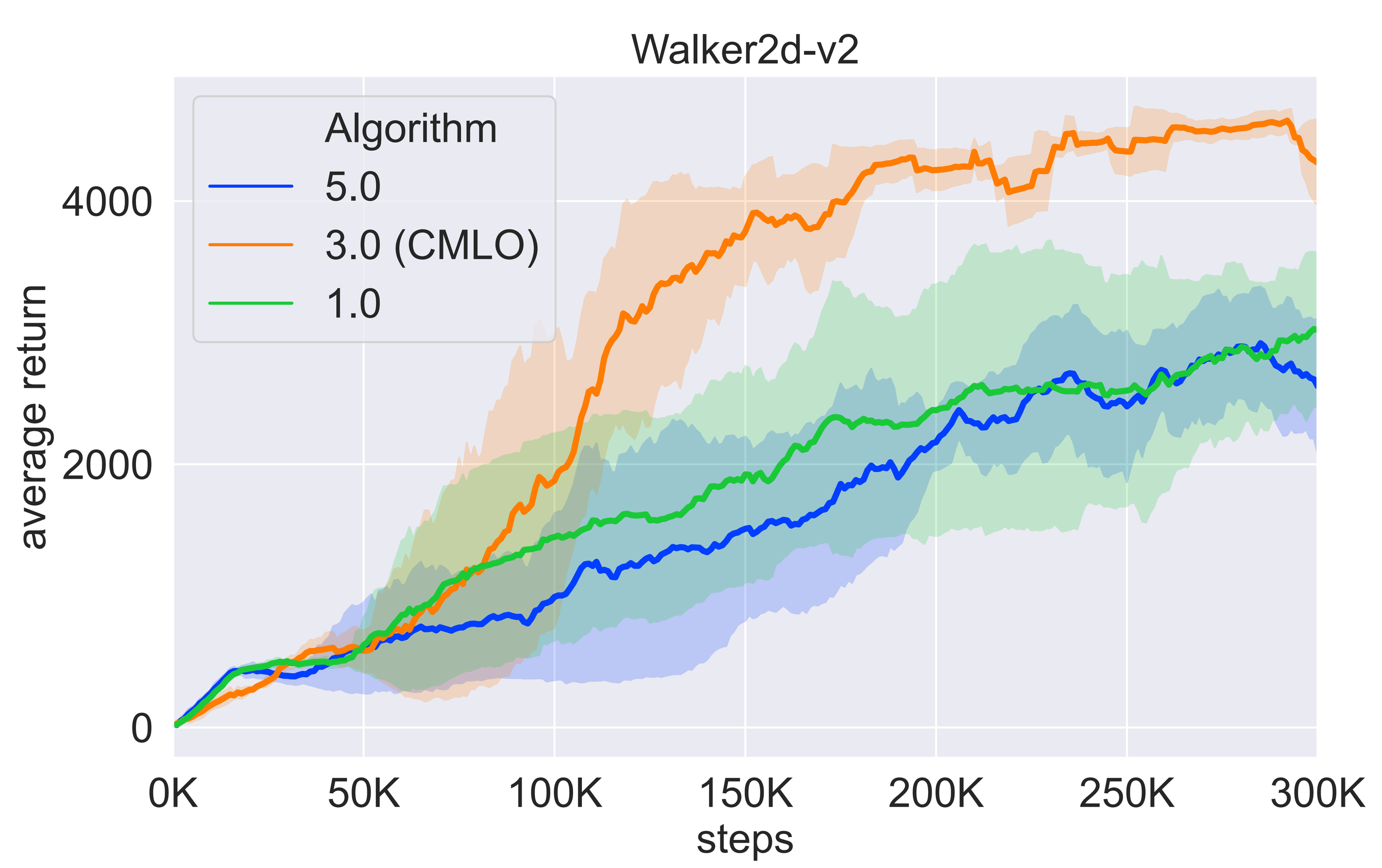


Figure 1. CMLO performance curves for different threshold settings over different random seeds, where 3.0 is the threshold recommended in the paper.

As Figure 1 shows, the performance corresponding to the other two thresholds (1.0 and 5.0) is severely affected. Therefore, setting a fixed threshold to constrain is inappropriate, and a smarter way like USB-PO should be applied to adaptively adjust the impacts of model shift.

[1] Michael Janner et al. "When to trust your model: Model-based policy optimization". In: Advances in neural information processing systems 32 (2019).

[2] Tianying Ji et al. "When to Update Your Model: Constrained Model-based Reinforcement Learning". In: Advances in Neural Information Processing Systems. Ed. by Alice H. Oh et al.2022.

Unified Model Shift and Model Bias Policy Optimization (USB-PO)

Theoretical Proof

- Unified Model Shift and Model Bias Bound

$$\begin{aligned} &V^{\pi_2|M_2} - V^{\pi_1|M_1} \\ &\geq \kappa(\gamma(\mathbb{E}_{(s,a)\sim d^{\pi_1}_{M_1}}[D_{TV}(p_{M_1}||p_{M^*}) - D_{TV}(p_{M_1}||p_{M_2}) - D_{TV}(p_{M_2}||p_{M^*})] + \Delta) - \epsilon_\pi) \end{aligned} \tag{3}$$

- $|\Delta|$ Upper Bound

$$|\Delta| \leq \frac{2\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\pi_1}_{M_1}}[D_{TV}(p_{M_1}||p_{M_2}) \max_{s,a} D_{TV}(p_{M_2}||p_{M^*})] + \frac{2\epsilon_\pi}{1-\gamma}\max_{s,a} D_{TV}(p_{M_2}||p_{M^*}) \tag{4}$$

$$D_{TV}(p_{M_1}||p_{M_2}) + D_{TV}(p_{M_2}||p_{M^*})$$

Practical Implementation

- Integral Probability Metrics

$$\sup_{f\in\mathcal{F}} |\mathbb{E}_{s'\sim p_M}[f(s')] - \mathbb{E}_{s'\sim p_{M'}}[f(s')]| = \frac{R_{max}}{1-\gamma} D_{TV}(p_M||p_{M'}) = L_v W_1(p_M, p_{M'}) \tag{5}$$

- Wasserstein Distance Inequality

$$W_1(p_M, p_{M'}) \leq W_2(p_M, p_{M'}), \forall M, M' \in \mathcal{M} \tag{6}$$

$$W_2(p_{M_1}, p_{M_2}) + W_2(p_{M_2}, p_{M^*})$$

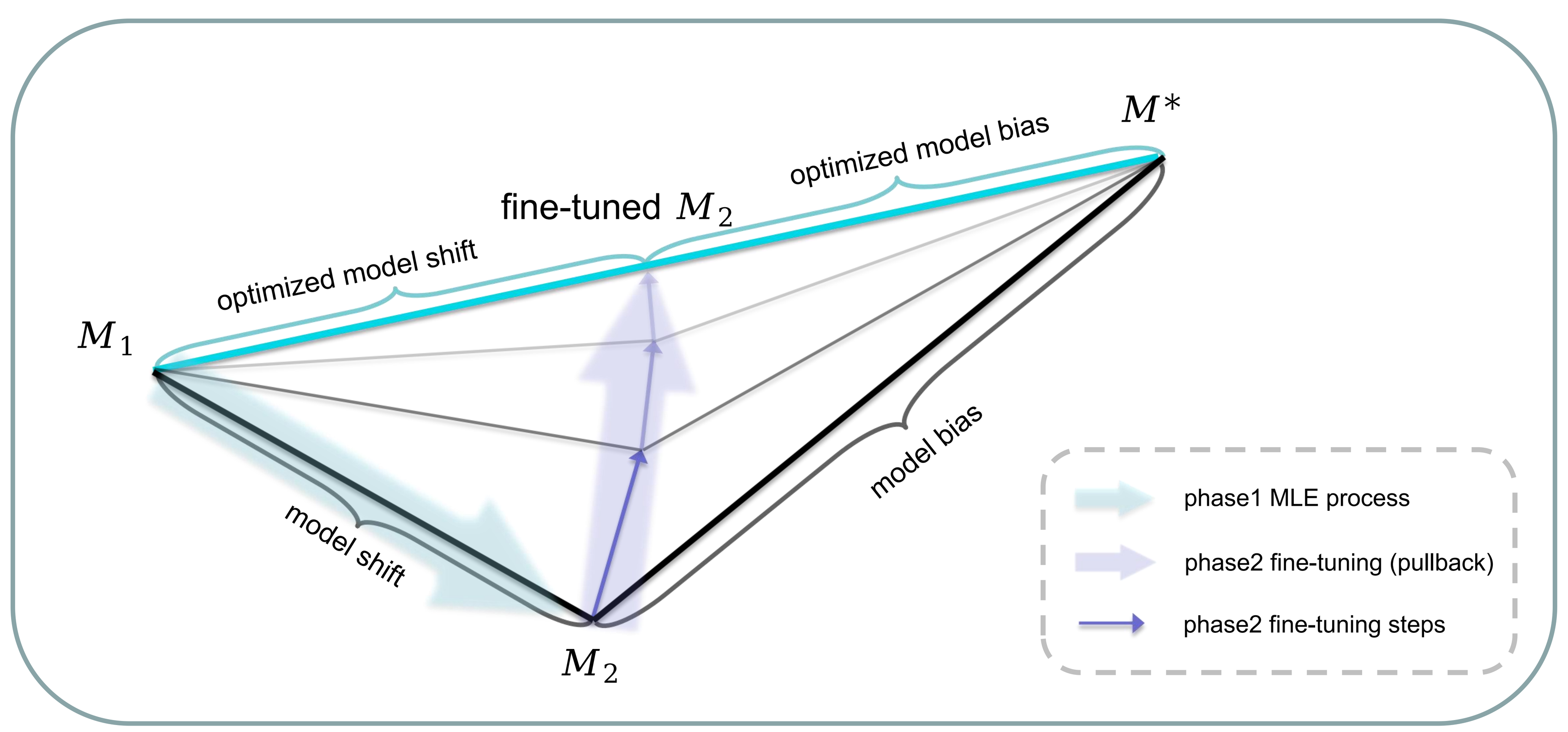
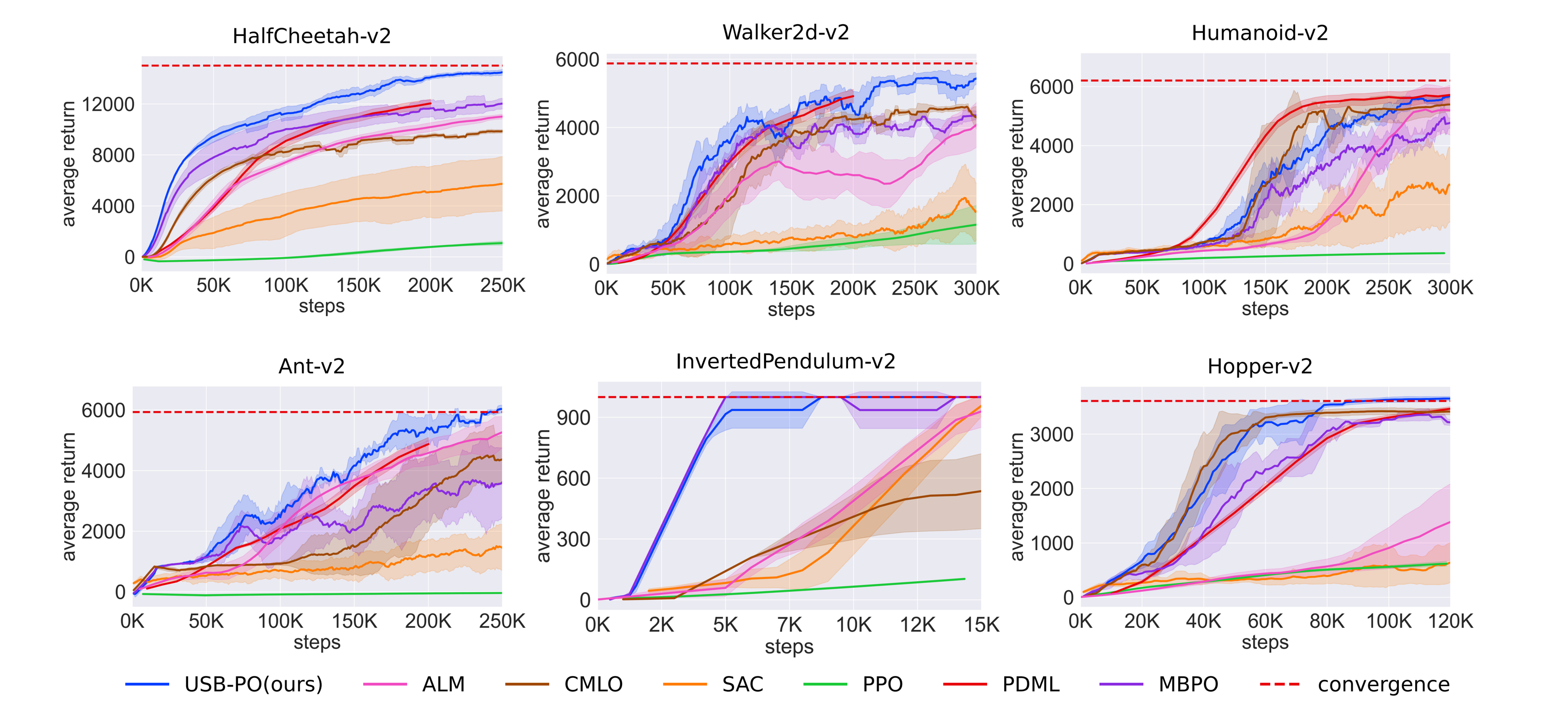


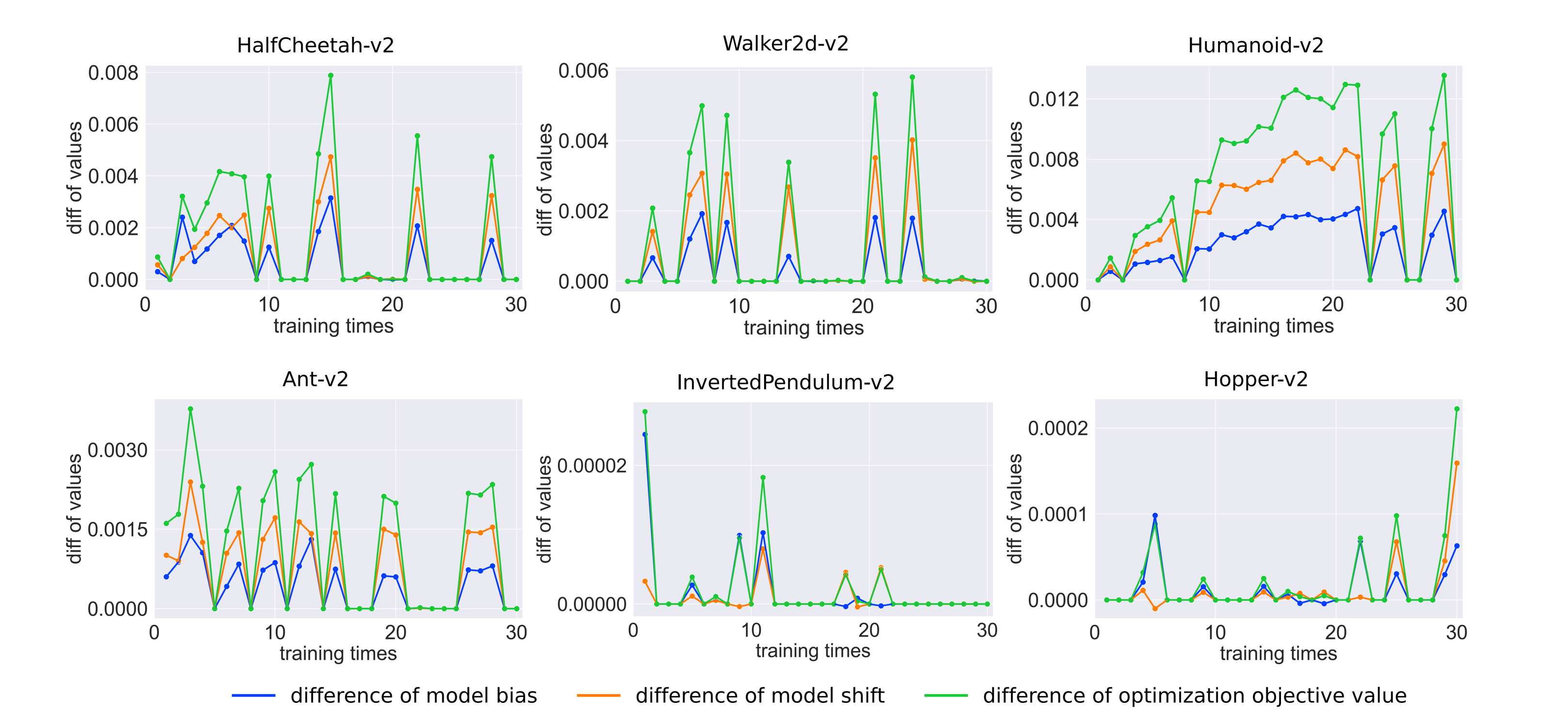
Figure 2. A schematic diagram to describe USB-PO. USB-PO adopts a two-phase model learning process. The model backed up before MLE update (phase 1) is denoted as M_1. M denotes the real environment and M_2 denotes the model after phase 1. M_2 will be further fine-tuned by Eq.(3) (phase 2) to get a performance improvement guarantee.

Experiments

- Higher sample efficiency and asymptotic performance.



- Automatically fine-tune the model updates.
- Model overfitting avoidance.



- Prevent diminishing sample efficiency and performance improvement.