Fast Adaption by Policy Deviation Integral Meta-reinforcement Learning with Applications to High-speed Trains Operation

Extended Abstract

Haotong Zhang
Chongqing Jiaotong University
Chongqing, China
tongzh@mails.cqjtu.edu.cn

Wanyuan Wang
Southeast University
Nanjing, China
wywang@seu.edu.cn

ABSTRACT

Current deep reinforcement learning (DRL)-based to optimize trajectories for real-world high-speed rail (HSR) face two issues: 1) a driver-centric Markov decision process (MDP) with sparse rewards and 2) single-trajectory optimization (i.e., single-task), poorly suited for real-world HSR scenarios that demand rapid adaptation to changing conditions (i.e., multi-task). To address these issues, we propose two innovations. First, a trajectory loop optimization (RTLO)-centric MDP that directly computes rewards from trajectory states, providing dense rewards. Second, a policy deviation integral meta-reinforcement learning (PDIMRL) method that enhances multi-task learning by leveraging HSR inter-task similarities, while the initial policy of the new task is linearly adjusted by the policy deviation integral between tasks' sub-optimal policies. Experiments demonstrate that 1) compared to existing driver-centric MDP, RTLO is 16.8× faster for single task training, and 2) based on RTLO-centric MDP, PDIMRL requires 2.3× fewer meta-training iterations than benchmark meta-RL methods.

KEYWORDS

Policy deviation integral; Meta-reinforcement learning; Rail trajectory loop optimization; High-speed rail

ACM Reference Format:
Haotong Zhang and Wanyuan Wang. 2025. Fast Adaption by Policy Deviation Integral Meta-reinforcement Learning with Applications to High-speed Trains Operation: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

The global drive for carbon neutrality has boosted interest in high-speed rail (HSR) as a sustainable transport option. However, driver-dependent operations cause energy fluctuations over 10%, undermining efficiency[3, 7]. Consequently, rail trajectory optimization (RTO) has been proposed with the aim of reducing overall operating costs of HSR by minimizes energy use under fixed timetables constraint while balancing efficiency, punctuality, and comfort[2].

Deep reinforcement learning (DRL) has applied to RTO[6, 10, 12, 13], but driver-focused models often produce sparse rewards that slow training[4], and single-task methods struggle with changing conditions[11]. We address these challenges by formulating a rail trajectory loop optimization (RTLO)-centric Markov Decision Process (MDP) in which trajectories serve as states, power outputs as actions, and transitions yield dense rewards. To enable multi-task adaptation, our policy deviation integral meta-reinforcement learning (PDIMRL) method leverages similarity between HSR tasks and adjusts the policy linearly quantifying the discrepancy between the current and optimal policies of previous iterations, thereby facilitating faster adaptation with less data than meta-reinforcement learning (Meta-RL).Experiments on real-world HSR operations show that our method cuts decision delays from hours to minutes while maintaining robust policy transfer under dynamic conditions.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 PROBLEM STATEMENT AND SOLUTIONS

2.1 RTLO Provides Intensive Rewards to Agent

Traditional RTO models driver behavior as an MDP with fixed spatial discretization, but sparse rewards and long decision sequences in HSR hinder effective exploration. RTLO addresses this by adjusting suboptimal trajectories with gradient segment, providing immediate optimization signals. Rather than controlling every step, the agent selects key power transition points aligned with gradient shifts, thereby simplifying the action space and enabling real-time evaluation of energy cost and punctual performance.

PMP Baseline: An analytical solution u^* based on Pontryagin's Maximum Principle (PMP)[1] serves as a baseline. Although it neglects gravitational effects on gradients, it efficiently computes a near-optimal by partitioning the trajectory into four phases: 1) Full traction until a transition point η_1, then 2) cruising, constant speed with balanced forces, ending at η_2, keep 3) coasting, inertia-driven motion until speed constraints require intervention, ending at η_3, and 4) full braking to reach the final stop f.

The PMP baseline effectively guides the train to near-optimal operation on each gradient segment, allowing gradient-based methods to converge quickly, significantly lower than naive DRL managing thousands of fine-grained steps. The agent refines traction strategies to exploit gravitational potential energy, dramatically narrowing the agent's search space.

Action Space: Rather than applying a continuous at every meter, the agent selects transition points and power levels for each gradient segment $[g_i, g_{i+1}]$ in a compact set:

$$A_t = \{p_t, p_f, p_b\}, \quad p_t, p_f, p_b \in [0, 1] \tag{1}$$

where p_t indicates the relative position within a gradient segment. p_f and p_b denote the power levels (as fractions of maximum traction) at the front and back of the segment, respectively. The control

scope of the agent is $[\eta_1, \eta_2]$. To maintain a continuous trajectory by aligning actions with natural gradient transitions, RTLO avoids unnecessary exploration over a wide range of irrelevant states.

State Space: RTLO uses a state space that fuses local and upcoming track information. The state includes travel time and energy consumption of previous optimization, current and next gradient speed, minimum power needed to maintain speed, and upcoming gradient set of track gradients and segment lengths.

Reward: Dense feedback is provided immediately after each sub-trajectory rather than waiting until the end of an episode. The reward balances multiple factors—energy consumption r_e, punctuality r_p, and passenger comfort r_j (via jerk):

$$R_t = \begin{cases} -r_j - (r_e r_p)^{\frac{1}{2}}, & E < 0, P < 0, \\ -r_j + C_1 (r_e r_p)^{\frac{1}{2}}, & E \geq 0, P \geq 0, \\ -r_j + s(E) r_e + s(P) r_p, & otherwise, \end{cases} \tag{2}$$

where r_e is calculated by $C_2(s(E) - 1 + E\eta_1)$, r_p is calculated by $C_2(1 - P\eta_2)$, and r_j is equal to $\eta_3 J$. $s(\cdot)$ is a switch function (0 if input $\geq$, otherwise 1). Here, η_1, η_2, and η_3 are positive weights for energy saving E relative to PMP baseline, punctuality P, and jerk J, respectively, while C_1 and C_2 scale the incentives. In order to prevent daily rewards from diluting settlement rewards, the reward for non-settlement rounds should be diluted.

2.2 Policy Deviation Integral Guided Meta-RL

By leveraging the linear relationship among suboptimal policies across different tasks in HSR, we proposes a PDIMRL, a novel first-order gradient-based meta-RL algorithm, migrates the new task's policy parameters closer to previously discovered solutions, and adapting by initializing the neural network model's parameters. The algorithm integrates meta-learning via Reptile[8] and stable policy updates via PPO[9], with a novel policy deviation integre (PDI) to migrate policy parameters between tasks.

Initialization: The algorithm begins by initializing meta parameters θ_0 that define a shared policy and task-specific variables, denoted local gradient $\xi(s)$ and speed constraints $v_{lim}(s)$, while $S_p(T)$ defines the task distribution for scheduling. A suboptimal baseline $u^*(T)$ pre-computed via PMP provides an initial for PDI.

Inner Loop: Task-specific DRL, for each task T_i, task-specific parameters $\hat{\phi}_i$ are derived from θ_i and refined using several rounds of PPO updates. Trajectories sampled from the policy $\pi(s|a, \hat{\phi}_i)$ are evaluated against the baseline, and the PPO loss[9] as:

$$L(\hat{\phi}_i) = \mathbb{E}_t \left[\min \left(r_t(\hat{\phi}_i) \hat{A}_t, \text{clip}(r_t(\hat{\phi}_i), 1 - \epsilon_c, 1 + \epsilon_c) \hat{A}_t \right) \right] \tag{3}$$

It is minimized via gradient descent to ensure stable updates. In cases where the current trajectory outperforms the reference, $u^*(T_i)$ is updated, enhancing the fidelity of PDI update.

Outer loop: Meta-learning, during each T_i, the inner loop performs v_{tasks} successive updates to refine task-specific parameters. Upon completion, the outer loop is invoked and employed Reptile to recalibrate the overarching model parameters, facilitating cross-task generalization, as follows:

$$\theta_0 \leftarrow \theta_0 + \beta_r (\hat{\phi}_i - \theta_0), \tag{4}$$

where $\hat{\phi}_i$ is the task-specific parameter in T_i and β_i^r is the learning rate for the Reptile update, decayed over time.

After switching to task T_{i+1}, initializing θ_0 to θ_{i+1}, compute the trajectory u using the policy $\pi(s_{t_0}|a, \hat{\phi}_i)$, where the parameters $\hat{\phi}_i$ are adapted from the previous task T_i, the initial state is denoted as s_{t_0}, and using $p(u, s)$ compute the traction power distribution:

$$\mathcal{E}_p^{t(s)} = p(\pi(s_{t_0}|a, \hat{\phi}_i), s) \tag{5}$$

where $p(u, s)$ is the traction power on trajectory u at point s.

The PDI is computed to quantify the discrepancy σ between the current and optimal policies from prior iterations, computed as:

$$\sigma = \int_{\eta_1}^{\eta_2} \frac{\mathcal{E}_p^{t(s)}}{p(u_{i+1}^*, s)} \, ds \tag{6}$$

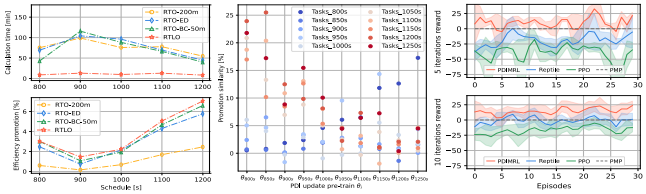
where η_1 and η_2 are the agent's control scope for task T_i.

Meta-parameters are adjusted as

$$\theta_{i+1} \leftarrow \theta_{i+1} + \beta_p \theta_{i+1} (\sigma - 1), \tag{7}$$

where the deviation σ controls how much the current policy is adjusted toward the previous one, and the β_p denotes the learning rate, decayed over time. This migration rapidly shifts the policy parameters closer to feasible, high-performing regions.

3 EXPERIMENTAL RESULTS



(a) Calculated cost (b) Boosted similarity (c) Different meta tests

Figure 1: DRTO and PDIMRL Case Study.

We conducted experiments using a CRH380A train—with traction parameters calibrated on a flat track[5]—on a 46-km modified segment of the Beijing–Shanghai railway [14]. We evaluated RTLO's generalization and adaptability on five task durations (800–1,200 seconds) using PPO, alongside baselines RTO-BC-50m [14], RTO-ED [15], and RTO-50m [16]. Compared to RTO-BC-50m, RTLO achieves a 16.8× average speedup across five single-task training runs. Figure 1b shows that a similarity analysis (averaged over four seeds) reveals significant behavior alignment improvements post-PDI, particularly when actual and planned execution times deviate. Figure 1c presents the meta-testing reward profile of PDIMRL on task-875s after 5 and 10 meta-training iterations, with error bands indicating stability. Leveraging the RTLO-centric MDP, PDIMRL outperforms traditional PMPs and requires 2.3× fewer meta-training iterations than the Reptile baseline.

4 CONCLUSION

We introduce an RTLO-centric MDP with dense, trajectory-based rewards to overcome the sparse-reward issue in driver-centric models. Our PDIMRL algorithm leverages a linear relationship among suboptimal HSR policies to mitigate the "lazy agent" problem, enabling near-real-time trajectory optimization, dynamic task adaptation, and efficient HSR operations.

REFERENCES

[1] Amie R Albrecht, Phil G Howlett, Peter J Pudney, and Xuan Vu. 2013. Energy-efficient train control: From local convexity to global optimization and uniqueness. *Automatica* 49, 10 (2013), 3072–3078.

[2] Yuan Cao, Zixuan Zhang, Fanglin Cheng, and Shuai Su. 2022. Trajectory optimization for high-speed trains via a mixed integer linear programming approach. *IEEE Transactions on Intelligent Transportation Systems* 23, 10 (2022), 17666–17676.

[3] Anne de Bortoli and Adélaïde Féraille. 2024. Banning short-haul flights and investing in high-speed railways for a sustainable future? *Transportation Research Part D: Transport and Environment* 128 (2024), 103987.

[4] Hairong Dong, Lingbin Ning, Min Zhou, Haifeng Song, and Weiqi Bai. 2024. Deep Reinforcement Learning for Integration of Train Trajectory Optimization and Timetable Rescheduling Under Disturbances. *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[5] Hebi Li. 2021. *Research on technology of high-speed railway train group operation simulation system.* Ph.D. Dissertation. China Academy of Railway Sciences.

[6] Jia Liu, Yunduan Cui, Jianghua Duan, Zhengmin Jiang, Zhongming Pan, Kun Xu, and Huiyun Li. 2024. Reinforcement learning-based high-speed path following control for autonomous vehicles. *IEEE Transactions on Vehicular Technology* (2024).

[7] Hongjie Ma, Hui Xie, Denggao Huang, and Shuo Xiong. 2015. Effects of driving style on the fuel consumption of city buses under different road conditions and vehicle masses. *Transportation Research Part D: Transport and Environment* 41 (2015), 205–216.

[8] A Nichol. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[10] Hui Wang, Zhigang Liu, Guiyang Hu, Xufan Wang, and Zhiwei Han. 2024. Offline Meta-Reinforcement Learning for Active Pantograph Control in High-Speed Railways. *IEEE Transactions on Industrial Informatics* (2024).

[11] Yihui Wang, Songwei Zhu, Shukai Li, Lixing Yang, and Bart De Schutter. 2022. Hierarchical model predictive control for on-line high-speed railway delay management and train control in a dynamic operations environment. *IEEE Transactions on Control Systems Technology* 30, 6 (2022), 2344–2359.

[12] Jianpeng Xu and Bo Ai. 2021. Experience-driven power allocation using multi-agent deep reinforcement learning for millimeter-wave high-speed railway systems. *IEEE Transactions on Intelligent Transportation Systems* 23, 6 (2021), 5490–5500.

[13] Haotong Zhang and Gang Xian. 2023. ASTPSI: Allocating Spare Time and Planning Speed Interval for Intelligent Train Control of Sparse Reward. In *International Conference on Neural Information Processing*. Springer, 65–77.

[14] Haotong Zhang, Kai Xu, Deqing Huang, Deqiang He, Shixun Wu, and Gang Xian. 2024. Hybrid Decision-Making for Intelligent High-Speed Train Operation: A Boundary Constraint and Pre-Evaluation Reinforcement Learning Approach. *IEEE Transactions on Intelligent Transportation Systems* 25, 11 (2024), 17979–17992.

[15] Liqing Zhang, Mingliang Zhou, Zhenning Li, et al. 2021. An intelligent train operation method based on event-driven deep reinforcement learning. *IEEE Transactions on Industrial Informatics* 18, 10 (2021), 6973–6980.

[16] Zicong Zhao, Jing Xun, Xuguang Wen, and Jianqiu Chen. 2022. Safe reinforcement learning for single train trajectory optimization via shield SARSA. *IEEE Transactions on Intelligent Transportation Systems* 24, 1 (2022), 412–428.