When to Add, When to Multiply: Probability Boosting Strategies for LLM-Guided RL Agents

Tan Yung Liang Joshua
Singapore University of Technology and Design joshua3$_tan@mymail.sutd.edu.sg$

***Abstract*—Large language models (LLMs) have emerged as promising sources of guidance for reinforcement learning agents during periods of policy uncertainty. While prior work establishes *when* to query LLMs using entropy thresholding, *how* to integrate LLM guidance into the action distribution remains underexplored. We present the first systematic comparison of additive versus multiplicative probability boosting for LLM-guided navigation, evaluating 36 configurations on a grid-based maze task. Both methods achieve comparable peak performance (+7.3% over baseline), but with different characteristics: multiplicative boosting reaches this peak at threshold 0.8 with more frequent queries, while additive achieves the same at threshold 1.0 with fewer queries. Multiplicative exhibits stable performance across parameters, while additive shows a narrower sweet spot requiring careful tuning. Analysis of action distributions reveals that additive succeeds when the correct action has low base probability (mean 0.27 vs 0.46 for multiplicative), supporting the hypothesis that additive's constant offset rescues agents from states where multiplicative scaling provides insufficient boost. These findings provide practical guidance for LLM-RL integration and lay groundwork for dynamic environments where classical planners fail.**

I. Introduction

Reinforcement learning has achieved remarkable success in navigation tasks, yet learned policies often struggle in unfamiliar states where training data is sparse [**?**]. This challenge is well-documented: RL policies generalize poorly to unseen environments. When they make an error, such as running into a wall or doubling back, they repeatedly make the same mistake, leading to failure [**?**]. Large language models (LLMs) offer a compelling complement. Their broad world knowledge and reasoning capabilities can provide guidance precisely when learned policies are uncertain. Recent work on hybrid LLM-RL systems demonstrates that querying an LLM when the policy exhibits high entropy can improve task success rates. This "Ask When Uncertain" (AWU) approach leverages the strengths of both paradigms [**?**].

While the AWU framework establishes *when* to query an LLM, triggering requests when policy entropy exceeds a calibrated threshold, the question of *how* to integrate LLM suggestions into the action distribution remains underexplored. Prior work typically applies a fixed multiplicative boost to the probability of the LLM-recommended action, but this design choice lacks systematic justification. An alternative approach, additive boosting, offers a constant probability offset regardless of the action's base likelihood. The distinction is consequential: when the correct action has very low probability (e.g., 0.05), multiplicative scaling (2x) yields only 0.10, while additive boosting (+0.3) yields 0.35. This is a qualitatively different intervention. To our knowledge, no prior work has systematically compared these integration strategies or characterized their relative strengths.

In this work, we present the first systematic comparison of additive versus multiplicative probability boosting for LLM-guided navigation. We evaluate a PPO-based agent on 150 held-out maze environments, conducting a comprehensive ablation across 36 configurations (6 thresholds by 6 boost values). Our baseline agent achieves 80% success, consistent with known RL generalization limitations. We find that both boosting methods can achieve comparable peak performance (+7.3% over baseline), but with distinct characteristics. Multiplicative boosting reaches peak performance at a lower entropy threshold, requiring more frequent LLM queries, while exhibiting stable performance across a broad range of parameters. Additive boosting achieves the same peak at a higher threshold with fewer queries, but shows sensitivity to parameter selection, with a narrow optimal region.

In this work, we present the first systematic comparison of additive versus multiplicative probability boosting for LLM-guided navigation. We evaluate a PPO-based agent on 150 held-out maze environments, conducting a comprehensive ablation across 36 configurations (6 thresholds $\times$ 6 boost values). Our baseline agent achieves 80% success, consistent with known RL generalization limitations. We find that both boosting methods can achieve comparable peak performance (+7.3% over baseline), but with distinct characteristics. Multiplicative boosting reaches peak performance at a lower entropy threshold, requiring more frequent LLM queries, while exhibiting stable performance across a broad range of parameters. Additive boosting achieves the same peak at a higher threshold with fewer queries, but shows sensitivity to parameter selection, with a narrow optimal region.

To understand why these methods differ, we analyze action probability distributions at LLM query points in representative rescue cases, defined as mazes where only one boosting method succeeded over baseline. Our analysis suggests that additive boosting tends to succeed when the correct action has low base probability. For example, in a maze solved only by additive boosting, the mean probability of the LLM-suggested action was 0.27, compared to 0.46 in a maze solved only by multiplicative boosting. This is consistent with our hypothesis that additive boosting can rescue agents from states where they are "more stuck." When the policy assigns very low probability to the correct action, multiplicative scaling

provides insufficient lift, while a constant additive offset can meaningfully shift the distribution. However, this same property may make additive boosting more volatile: strong additive boosts could introduce noise when the agent is only mildly uncertain, potentially explaining its narrower optimal parameter range.

II. Methodology

III. Results

IV. Conclusion