Supplementary Materials for
"Do LLM Agents Have Regret? A Case Study in Online Learning and Games"

CONTENTS

A Related Work

LLM(-agent) for decision-making. The impressive capability of LLMs for *reasoning* (Bubeck et al., 2023; Achiam et al., 2023; Wei et al., 2022b;a; Srivastava et al., 2023; Yao et al., 2023a) has inspired a growing line of research on *LLM for (interactive) decision-making*, i.e., an LLM-based autonomous agent interacts with the environment by taking actions repeatedly/sequentially, based on the feedback it perceives. Some promises have been shown from a *planning* perspective (Hao et al., 2023; Valmeekam et al., 2023; Huang et al., 2022b; Shen et al., 2023). In particular, for embodied AI applications, e.g., robotics, LLMs have achieved impressive performance when used as the controller for decision-making (Ahn et al., 2022; Yao et al., 2023b; Shinn et al., 2023; Wang et al., 2023d; Driess et al., 2023; Significant Gravitas, 2023). However, the performance of decision-making has not been rigorously characterized via the regret metric in these works. Very recently, Liu et al. (2023e) has proposed a principled architecture for LLM-agent, with provable regret guarantees in stationary and stochastic decision-making environments, under the Bayesian adaptive Markov decision processes framework. In contrast, our work focuses on online learning and game-theoretic settings, in potentially adversarial and non-stationary environments. Moreover, (first part of) our work focuses on *evaluating* the intelligence level of LLM per se in decision-making (in terms of the regret metric), while Liu et al. (2023e) focused on *developing* a new architecture that uses LLM as an oracle for reasoning, together with memory and specific planning/acting subroutines, *to achieve* sublinear (Bayesian) regret, in stationary and stochastic environments.

LLMs in multi-agent environments. The interaction of multiple LLM agents has garnered significant attention lately. For example, Fu et al. (2023) showed that LLMs can autonomously improve each other in a negotiation game by playing and criticizing each other. Similarly, (Du et al., 2023; Liang et al., 2023; Xiong et al., 2023; Chan et al., 2024; Li et al., 2023c) showed that multi-LLM *debate* can improve the reasoning and evaluation capabilities of the LLMs. Qian et al. (2023); Schick et al. (2023); Wu et al. (2023) demonstrated the potential of multi-LLM interactions and collaboration in software development, writing, and problem-solving, respectively. Zhang et al. (2024) exhibited a similar potential in embodied cooperative environments. More formally, multi-LLM interactions have also been investigated under a *game-theoretic* framework, to characterize the *strategic* decision-making of LLM agents. Bakhtin et al. (2022); Mukobi et al. (2023) and Xu et al. (2023b;a) have demonstrated the promise of LLMs in playing Diplomacy and WereWolf games, respectively, which are both language-based games with a mixture of competitive and cooperative agents. Note that these works utilized LLM to solve a specific rather than a general game. Related to our work, Brookins & DeBacker (2023); Akata et al. (2023); Lorè & Heydari (2023); Brookins & DeBacker (2023); Fan et al. (2023) have also used (repeated) matrix games as a benchmark to evaluate the reasoning capability and rationality of LLM agents. In contrast to our work, these empirical studies have not formally investigated LLM agents using the metric of *regret*, nor through the lenses of *online learning* and *equilibrium-computation*, which are all fundamental in modeling and analyzing strategic multi-agent interactions. Moreover, our work also provides theoretical results to explain and further enhance the no-regret property of LLM agents.

LLMs & Human/Social behavior. LLMs have also been used to *simulate* the behavior of human beings, for social science and economics studies (Engel et al., 2023). The extent of LLMs simulating human behavior has been claimed as a way to evaluate the level of its intelligence in a controlled environment (Aher et al., 2023; Tsai et al., 2023). For example, Li et al. (2023b); Hong et al. (2024); Zhao et al. (2023) showed that by specifying different "roles" to LLM agents, certain collaborative/competitive behaviors can emerge. Argyle et al. (2023) showed that LLMs can emulate response distributions from diverse human subgroups, illustrating their adaptability. Horton (2023) argued that an LLM, as a computational model of humans, can be used as *homo economicus* when given endowments, information, preferences, etc., to gain new economic insights by simulating its interaction with other LLMs. Park et al. (2022; 2023) proposed scalable simulators that can generate realistic social behaviors emerging in populated and interactive social systems, and the emerging behaviors of LLM agents in society have also been consistently observed in Chen et al. (2024; 2023). Li et al. (2023d;a) studied the behavioral dynamics of LLM agents on social networks. These empirical results have inspired our work, which can be viewed as an initial attempt towards quantitatively understanding the *emerging behavior* of LLMs as computational human models, given the known

justification of *equilibrium* being a long-run emerging behavior of *learning dynamics* (Fudenberg & Levine, 1998) and strategic interactions (Young, 2004; Camerer, 2011).

Transformers & In-context-learning. LLMs nowadays are predominantly built upon the architecture of Transformers (Vaswani et al., 2017). Transformers have exhibited a remarkable capacity of *in-context-learning* (ICL), which can construct new predictors from sequences of labeled examples as input, without further parameter updates. This has enabled the *few-shot learning* capability of Transformers (Brown et al., 2020; Garg et al., 2022; Min et al., 2022). The empirical successes have inspired burgeoning theoretical studies on ICL. Xie et al. (2022) used a Bayesian inference framework to explain how ICL works, which has also been adopted in Wang et al. (2023b); Jiang (2023). Akyürek et al. (2023); Von Oswald et al. (2023); Dai et al. (2023); Giannou et al. (2023) showed (among other results) that ICL comes from that Transformers can implement the gradient descent (GD) algorithm. Bai et al. (2023) further established that Transformers can implement a broad class of machine learning algorithms in context. Moreover, Ahn et al. (2023); Zhang et al. (2023a); Mahankali et al. (2023) proved that a *minimizer* of the certain training loss among single-layer Transformers is equivalent to a single step of GD for linear regression. Li et al. (2023e) established generalization bounds of ICL from a multi-task learning perspective. Zhang et al. (2023b) argued that ICL implicitly implements Bayesian model averaging, and can be approximated by the attention mechanism. They also established a result on some *regret* metric. However, the regret notion is not defined for (online) decision-making, and is fundamentally different from ours that is standard in online learning and games. Also, we provide extensive experiments to validate the no-regret behavior by our definition. More recently, the ICL property has also been generalized to decision-making settings. Laskin et al. (2023); Lee et al. (2023); Lin et al. (2024) investigated the in-context reinforcement learning (RL) property of Transformers under supervised pre-training, for solving stochastic bandits and Markov decision processes. In contrast, our work focuses on online learning settings with an arbitrary and *potentially adversarial* nature, as well as *game-theoretic* settings. We also provide a new *unsupervised* loss to promote the no-regret behavior in our settings.

Online learning and games. Online learning has been extensively studied to model the decision-making of an agent who interacts with the environment sequentially, with a potentially arbitrary sequence of loss functions (Shalev-Shwartz, 2012; Hazan, 2016), and has a deep connection to game theory (Cesa-Bianchi & Lugosi, 2006). In particular, regret, the difference between the incurred accumulated loss and the best-in-hindsight accumulated loss, has been the core performance metric, and a good online learning algorithm should have regret at most sublinear in time T, which is referred to as being *no-regret*. Many well-known algorithms can achieve no-regret against *arbitrary* loss sequences, e.g., multiplicative weight updates (MWU)/Hedge (Freund & Schapire, 1997; Arora et al., 2012b), EXP3 (Auer et al., 2002), and more generally follow-the-regularized-leader (FTRL) (Shalev-Shwartz & Singer, 2007) and follow-the-perturbed-leader (FTPL) (Kalai & Vempala, 2005). In the bandit literature (Lattimore & Szepesvári, 2020; Bubeck et al., 2012), such a setting without any statistical assumptions on the losses is also referred to as the *adversarial/non-stochastic* setting. Following the conventions in this literature, the online settings we focus on shall not be confused with the stationary and *stochastic*(-bandit)/(-reinforcement learning) settings that have been explored in several other recent works on *Transformers for decision-making* (Lee et al., 2023; Lin et al., 2024). Centering around the regret metric, our work has also explored the non-stationary bandit setting (Besbes et al., 2014), as well as the repeated game setting where the environment itself consists of strategic agents (Cesa-Bianchi & Lugosi, 2006).

A.1 Comparison with Concurrent Work Krishnamurthy et al. (2024)

After submitting the first version of our manuscript, we were aware of a concurrent work Krishnamurthy et al. (2024), which considered using LLMs to solve multi-arm *stochastic* bandit problems entirely in-context, with a focus on the *exploration* behaviors of LLMs. Specifically, Krishnamurthy et al. (2024) claimed that LLMs may not show robust exploratory behaviors under a variety of prompt configurations, although there does exist some successful prompt configuration that enabled satisfactory exploratory behaviors. We here provide a detailed comparison between Krishnamurthy et al. (2024) and the first experimental part of our paper, i.e., Section 3 and related appendices.

- **(Focused settings).** We mainly considered the *full-information* online learning setting with potentially *adversarial* loss vectors, as well as the multi-agent *repeated-game* setting. In

contrast, Krishnamurthy et al. (2024) focused on the *stochastic* setting with *bandit* feedback, where the loss vectors at different rounds are drawn i.i.d. from a *fixed* distribution. Therefore, both the *metrics* and most *results* are not directly comparable. For example, i) some failure cases in Krishnamurthy et al. (2024) for stochastic bandits did not appear in our setting (as will be detailed next); ii) for some adversarial loss instances (e.g., those from Feder et al. (1992), see the introduction in Section 3.4), the *summarized history input* that was claimed essential in Krishnamurthy et al. (2024) is not very effective in our settings, while a *raw-history input* as in our experiments can be more effective (see Section 3.4 and Figure C.6); iii) as studied in Krishnamurthy et al. (2024), *uniform-like* behaviors constitute one of the main failures in stochastic bandits. However, uniform-like policies do not necessarily correspond to failure cases in our setting, especially when the loss vectors are highly adversarial (c.f. examples in Section 3.4). In particular, such a metric may be irrelevant/inapplicable to validating the no-regret behaviors in our full-information non-stochastic/adversarial settings. These results/facts demonstrated the fundamentally different features in addressing the distinct settings in both works.

- **(Configuration/Prompt design choices).** Despite the negative results under many prompt configurations, Krishnamurthy et al. (2024) still found one successful prompt configuration that can lead to robust exploratory behaviors in stochastic bandits, which in fact shares many similarities with our default prompt configurations. For example, Krishnamurthy et al. (2024) found that asking the LLMs to output a *distribution* over the action space (instead of one *single action*) can address the *suffix failure* for stochastic bandits, which was indeed the default prompt we used in our settings. Moreover, as a standard technique, our default prompt asked the model to have the Chain-of-Thought (CoT) reasoning, while Krishnamurthy et al. (2024)'s successful prompt also emphasized the importance of CoT. Krishnamurthy et al. (2024) also showed the importance of *summarizing* the history, i.e., summarizing the mean reward associated with each arm, while we found that when we feed the LLMs with (raw) *full-information feedback in the vector form*, the LLMs may automatically choose to summarize the history and make decisions based on the summarized statistics (c.f. the output examples in Appendix C.10).

- **(Horizons v.s. No-regret behaviors).** In light of the findings from Krishnamurthy et al. (2024) that LLMs may fail when the problem horizon is long, we conduct experiments on problems with comparable horizons as in Krishnamurthy et al. (2024). Our results show that for the full-information non-stochastic setting we focused on, LLMs can still be no-regret with longer horizons (Figure 3.2 and Table 1), under the loss sequences we studied.

- **(Results in bandit setting & Failure cases).** As an extension and sanity check of our full-information-setting results, we have also experimented with the (adversarial) bandit setting. This extension setting is more comparable to that in Krishnamurthy et al. (2024). However, different from the focus therein, we did not ask the LLMs to *directly explore* in context. Instead, we manually input a *re-weighting* estimate of the full-information loss vector, a standard technique in online learning (Auer et al., 2002; Hazan, 2016; Lattimore & Szepesvári, 2020), to balance exploration and exploitation. We viewed this approach as a natural way to exploit the no-regret behaviors of LLMs in the full-information setting. In fact, with such a re-weighting, we show in Table 2 that the failure cases in Krishnamurthy et al. (2024) for the bandit setting may not appear, in the exact hard instance proposed therein, even under a relatively *long horizon* of $T = 100$. Complementing Krishnamurthy et al. (2024), our bandit-setting results may suggest that such *human-intervened input* may enhance LLMs' decision-making capabilities. This is perhaps also in line with the observation in Krishnamurthy et al. (2024) that some additional "human intervention" (i.e., the *summarized* history input therein) may be critical in the (stochastic) bandit setting. Specifically, in Table 2, we validate that although LLMs may fail in bandit-feedback settings *without interventions*, such a simple re-weighting technique may be useful to handle exploration tasks by leveraging LLMs' performance in the full-information setting.

Hard MAB instance of Krishnamurthy et al. (2024)	TS	UCB	Successful case of Krishnamurthy et al. (2024)	Ours (GPT-4)	Naive (GPT-4)	Ours (GPT-4o)	Naive (GPT-4o)
Median reward (higher is better)	0.47	0.55	0.47	0.46	0.46	0.475	0.455
SuffFailFreq($T/2$) (lower is better)	0.01	0.02	0.00	0.00	0.00	0.00	0.2
$n*$MinFrac (lower is better)	0.28	0.18	0.33	0.27	0.38	0.1	0.09

Table 2: Comparing Thompson Sampling (TS), Upper Confidence Bound (UCB), and the successful prompt configuration of Krishnamurthy et al. (2024) (from Figure 4 therein) with our approaches (named *Ours* in the table), on the *hard* MAB instance therein. We also conducted ablation studies by *removing* our re-weighting technique (named *Naive* in the table). Note that both *Ours* and *Naive* use *distributional output*, as it is the default prompt configuration we used throughout our paper. Specifically, as introduced in Krishnamurthy et al. (2024), for this hard instance, rewards associated with each arm follow a Bernoulli distribution, the horizon is $T = 100$, the number of actions is $n = 5$, and the reward gap is 0.2. For GPT-4, the model adopted by Krishnamurthy et al. (2024), we have observed similar results with their case using the *distributional* output, where although the median reward is comparable with the successful cases, *Naive* suffers from the uniform-like failure as indicated by a high $n*$MinFrac value. For GPT-4o, the model not studied by Krishnamurthy et al. (2024), we have a slightly different observation that *Naive* (with distributional output as in our default configurations) seems to still suffer from suffix failure, indicated by a slightly high SuffFailFreq($T/2$), while Krishnamurthy et al. (2024) reported that distributional output *can avoid* such a failure for GPT-4. In contrast to *Naive*, our re-weighting technique enabled the LLMs to avoid *both* the suffix and the uniform-like failures in this (stochastic) bandit-feedback case, *without* external history summarization, and achieve comparable rewards.

B Deferred Background

B.1 Notation

We use $\mathbb{N}$ and $\mathbb{N}^+$ to denote the sets of non-negative and positive integers, respectively. For a finite set $\mathcal{S}$, we use $\Delta(\mathcal{S})$ to denote the simplex over $\mathcal{S}$. For $d \in \mathbb{N}^+$, we define $[d] := \{1, 2, \ldots, d\}$. For two vectors $x, y \in \mathbb{R}^d$, we use $\langle x, y\rangle$ to denote the inner product of x and y. We define $\mathbf{0}_d$ and $\mathbf{1}_d$ as a d-dimensional zero or one vector, and $\boldsymbol{O}_{d\times d}$ and $I_{d\times d}$ as a $d\times d$-dimensional zero matrix and identity matrix, respectively. We omit d when it is clear from the context. We define e_i as a unit vector (with proper dimension) whose i-th coordinate equal to 1. For $p \in \mathbb{R}^d, R > 0$ and $C \subseteq \mathbb{R}^d$ is a convex set, define $B(p, R, \|\cdot\|) := \{x \in \mathbb{R}^d \mid \|x-p\| \le R\}$, $\texttt{Proj}_{C,\|\cdot\|}(p) = \arg\min_{x\in C} \|x-p\|$ (which is well defined as C is a convex set), and $\texttt{clip}_R(x) := [\texttt{Proj}_{B(0,R,\|\cdot\|_2),\|\cdot\|_2}(x_i)]_{i\in[d]}$. Define $\texttt{Softmax}(x) := \left(\frac{e^{x_i}}{\sum_{i\in[d]} e^{x_i}}\right)_{i\in[d]}$ and $\texttt{ReLU}(x) = \max(0, x)$ for $x \in \mathbb{R}^d$. For $A \in \mathbb{R}^{m\times n}$ with A_i denoting its i-th column, we define $\|A\|_{\text{op}} := \max_{\|x\|_2\le 1} \|Ax\|_2$, $\|A\|_{2,\infty} := \sup_{i\in[n]} \|A_i\|_2$, $\|A\|_F$ as the Frobenius norm, and $A_{-1} := A_n$ to denote the last column vector of A. We define $\mathbb{R}^+ := \{x \mid x \ge 0\}$. For a set Π, define $\text{diam}(\Pi, \|\cdot\|) := \sup_{\pi_1,\pi_2\in\Pi} \|\pi_1 - \pi_2\|$. We define $\mathbb{1}(\mathcal{E}) := 1$ if $\mathcal{E}$ is true, and $\mathbb{1}(\mathcal{E}) := 0$ otherwise. For a random variable sequence $(X_n)_{n\in\mathbb{N}}$ and random variables X, Y, we denote F_X as the cumulative distribution function of a random variable X, $X_n \xrightarrow{p} X$ if $\forall\epsilon > 0, \lim_{n\to\infty} \mathbb{P}(|X_n - X| > \epsilon) = 0$, $X_n \xrightarrow{d} X$ if $\lim_{n\to\infty} F_{X_n}(x) = F_X(x)$ for all x where $F_X(x)$ is continuous, $X \overset{d}{=} Y$ if $F_X(x) = F_Y(x)$ for all x, $X_n \xrightarrow{a.s.} X$ if $\mathbb{P}(\lim_{n\to\infty} X_n = X) = 1$, and $\text{esssup}(X) := \inf\{M \in \mathbb{R} : \mathbb{P}(X > M) = 0\}$. For a random variable X, we use $\text{supp}(X)$ to denote its support. For functions $f, g : \mathbb{R} \to \mathbb{R}$, we define $g(x) = \mathcal{O}(f(x))$ if there exist $x_0, M < \infty$ such that $|g(x)| \le M|f(x)|$ for all $x > x_0$. We use f' to denote the derivative of f. Let $F : \Omega \to \mathbb{R}$ be a continuously-differentiable, strictly convex function defined on a convex set Ω. The Bregman divergence associated with F for points p, q is defined as $D_F(p, q) := F(p) - F(q) - \langle\nabla F(q), p - q\rangle$. For a sequence $(\ell_t)_{t\in[T]}$ for some $T \in \mathbb{N}^+$, we define $\ell_{a:b} := (\ell_a, \cdots, \ell_b)$ for $1 \le a \le b \le T$. If $a > b$, we define $\ell_{a:b} = \emptyset$.