DO LLM AGENTS HAVE REGRET? A CASE STUDY IN ONLINE LEARNING AND GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have been increasingly employed for (interactive) decision-making, via the development of LLM-based autonomous agents. Despite their emerging successes, the performance of LLM agents in decision-making has not been fully investigated through rigorous metrics, especially in the multi-agent setting when they interact with each other, a typical scenario in real-world LLM-agent applications. To better understand the limits of LLM agents in these interactive environments, we propose to study their interactions in benchmark decision-making settings of *online learning* and *games*, through the performance metric of *regret*. We first empirically study the *no-regret* behaviors of LLMs in canonical (non-stationary) online learning problems, as well as the emergence of equilibria when LLM agents interact through playing repeated games. We then provide theoretical insights into the no-regret behaviors of LLM agents, under certain assumptions on *supervised* pre-training and *rationality* model of human decision-makers who generate the data. Notably, we also identify (simple) cases where advanced LLMs such as GPT-4 fail to be no-regret. To promote the no-regret behaviors, we propose a novel *unsupervised* training loss of *regret-loss*, which, in contrast to the supervised pre-training loss, does not require the labels of (optimal) actions. We then establish the statistical guarantee of generalization bound for regret-loss minimization, followed by the optimization guarantee that minimizing such a loss may automatically lead to known no-regret learning algorithms. Our further experiments demonstrate the effectiveness of our regret-loss, especially in addressing the above "regrettable" cases.

1 INTRODUCTION

Live Life with No Excuses. Travel with No Regret.

Oscar Wilde

Large language models (LLMs) have recently exhibited remarkable reasoning capabilities (Bubeck et al., 2023; Achiam et al., 2023; Wei et al., 2022b; Yao et al., 2023). As a consequence, a burgeoning body of work has been investigating the employment of LLMs as central controllers for (interactive) decision-making, through the construction of *LLM-based autonomous agents* (Hao et al., 2023; Shen et al., 2023; Yao et al., 2022; Shinn et al., 2023; Wang et al., 2023c; Significant Gravitas). Specifically, the LLM agent interacts with the (physical) world in a *dynamical/sequential* way: it uses LLMs as an oracle for reasoning, then acts in the environment based on the reasoning and the feedback it perceives over time. LLM agent has achieved impressive successes in embodied AI (Ahn et al., 2022; Huang et al., 2022a; Wang et al., 2023a), natural science (Wu et al., 2023; Swan et al., 2023), and social science (Park et al., 2022; 2023) applications.

Besides being *dynamic*, another increasingly captivating feature of LLM-based decision-making is the involvement of *strategic* interactions, oftentimes among multiple LLM agents. For example, it has been continually reported that the reasoning capability of LLMs can be improved by interacting with each other through negotiation and/or debate games (Fu et al., 2023; Du et al., 2023); LLM agents have now been widely used to *simulate* the strategic behaviors for social and economic studies, to understand the emerging behaviors in interactive social systems (Aher et al., 2023; Park et al., 2023). Moreover, LLMs have also exhibited remarkable potential in solving various games (Bakhtin et al., 2022; Mukobi et al., 2023), and in fact, a rapidly expanding literature has employed *repeated games* as a fundamental benchmark to understand the strategic behaviors of LLMs (Brookins & DeBacker, 2023; Akata et al., 2023; Fan et al., 2023). These exciting empirical successes call for a rigorous examination and understanding through a theoretical lens of decision-making.

Regret, on the other hand, has been a core metric in (online) decision-making. It measures how "sorry" the decision-maker is, in retrospect, not to have followed the best prediction in hindsight (Shalev-Shwartz, 2012). It provides not only a sensible way to *evaluate* the intelligence level of online decision-makers, but also a quantitative way to measure their *robustness* against arbitrary (and possibly adversarial) environments. More importantly, it inherently offers a connection to modeling and analyzing strategic behaviors: long-run interactions of no-regret learners lead to certain equilibria when they repeatedly play games (Cesa-Bianchi & Lugosi, 2006). In fact, *no-regret* learning has been posited as an important model of agents' "rational behavior" in playing games (Blum et al., 2008; Roughgarden, 2015; Roughgarden et al., 2017). Thus, it is natural to ask:

Can we examine and better understand the decision-making of LLMs through the lens of regret*?*

Acknowledging that LLM(-agents) are extremely complicated to analyze, to gain some insights into the question, we focus on benchmark decision-making settings: online learning with convex (linear) loss functions, and playing repeated games. We summarize our contributions as follows.

Contributions. First, we carefully examine the performance of several representative pre-trained LLMs in benchmark online decision-making settings as mentioned above, in terms of *regret*. We observe that oftentimes, LLM agents exhibit no-regret behaviors in these (non-stationary) online learning settings, where the loss functions change over time either arbitrarily (and even adversarially) or by following some pattern with bounded variation, and in playing both representative and randomly generated repeated games, where equilibria will emerge as the long-term behavior of the interactions. Second, we provide some theoretical insights into the observed no-regret behaviors, under certain assumptions on the *rationality* model of human decision-makers who generate the data, and the *supervised pre-training* procedure, a common practice in training large models for decision-making where the *(optimal) actions* are used as labels to predict. In particular, we make connections of pre-trained LLMs to the known no-regret algorithm of *follow-the-perturbed-leader* (FTPL) under certain assumptions. Third, we also identify (simple) cases where advanced LLMs as GPT-4 fail to be no-regret. We thus propose a novel *unsupervised* training loss, *regret-loss*, which, in contrast to the supervised pre-training loss, does not require the labels of (optimal) actions. We then establish both statistical and optimization guarantees for regret-loss minimization, showing that minimizing such a loss may automatically lead to known no-regret learning algorithms. Our further experiments demonstrate the effectiveness of regret-loss, especially in addressing the above "regrettable" cases.

1.1 Related Work

LLM(-agent) for decision-making. The impressive capability of LLMs for *reasoning* (Wei et al., 2022b; Srivastava et al., 2023; Yao et al., 2023) has inspired a growing line of research on *LLM for (interactive) decision-making* (Hao et al., 2023; Valmeekam et al., 2023; Ahn et al., 2022; Yao et al., 2022; Shinn et al., 2023; Driess et al., 2023). However, the performance has not been rigorously investigated via the regret metric in these works. Very recently, Liu et al. (2023c) proposed a principled architecture for LLM-agent, with provable *regret* guarantees in stationary and stochastic environments. 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, while Liu et al. (2023c) focused on *developing* a new architecture that uses LLM as an oracle *to achieve* sublinear (Bayesian) regret.

LLMs in multi-agent and social environments. Fu et al. (2023); Du et al. (2023); Liang et al. (2023) showed that LLMs can autonomously improve reasoning capabilities via negotiation and/or debate. The interaction has also been increasingly studied under the *game-theoretic* framework (Bakhtin et al., 2022; Brookins & DeBacker, 2023; Akata et al., 2023; Lorè & Heydari, 2023; Brookins & DeBacker, 2023). Nonetheless, these empirical studies have not been formally investigated through the lens of regret, nor online learning and equilibrium-computation, which are all fundamental in analyzing strategic multi-agent interactions. LLMs have also been used to *simulate* the (emerging) behaviors of humans, for social science and economics studies (Horton, 2023; Li et al., 2023b; Chen et al., 2023a;b; Park et al., 2022; 2023), which have motivated our work, a more quantitative understanding of the emerging behavior of LLMs as *computational human models*.

Online learning and games. Online learning has been extensively studied in the literature, see Shalev-Shwartz (2012); Hazan (2016) for comprehensive introductions of its (and regret notions') importance, and Cesa-Bianchi & Lugosi (2006) for a connection to game theory. Following the conventions in this literature, the online settings we focus on, which handle a potentially adversarial environment (which itself may consist of strategic agents), shall not be confused with the stationary and stochastic settings explored in other recent works on *Transformers for decision-making* (Lee et al., 2023; Lin et al., 2023) (see Appendix A for a detailed comparison).

2 PRELIMINARIES

Notation. For a finite set $\mathcal{S}$, we use $\Delta(\mathcal{S})$ to denote the simplex over $\mathcal{S}$. We denote $\mathbb{R}^+ = \{x \in \mathbb{R} \,|\, x \geq 0\}$. 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 the d-dimensional all-zero and all-one vector, respectively, and $\boldsymbol{O}_{d\times d}$ and $I_{d\times d}$ as the $d \times d$-dimensional zero matrix and identity matrix, respectively. For a positive integer d, we define $[d] = \{1, 2, \ldots, d\}$. For $p \in \mathbb{R}^d, R > 0$ and $C \subseteq \mathbb{R}^d$ being a convex set, define $B(p, R, \|\cdot\|) := \{x \in \mathbb{R}^d \mid \|x - p\| \leq R\}$ and $\texttt{Proj}_{C,\|\cdot\|}(p) = \arg\min_{x\in C} \|x - p\|$. For any $x \in \mathbb{R}^d$, define $\texttt{Softmax}(x) = \left(\frac{e^{x_i}}{\sum_{i\in[d]} e^{x_i}}\right)_{i\in[d]}$. For a vector $v \in \mathbb{R}^n$, we use $\|v\|_p$ to denote its ℓ_p-norm, with $\|v\|$ denoting the ℓ_2-norm by default. We define $\mathbb{1}(\mathcal{E}) = 1$ if some event $\mathcal{E}$ is true, and $\mathbb{1}(\mathcal{E}) = 0$ otherwise. For a random variable X, we use $\text{supp}(X)$ to denote its support.

2.1 ONLINE LEARNING & GAMES

Online learning. We first consider the online learning setting where an agent interacts with the environment for T rounds, by iteratively making decisions based on the feedback she receives. Specifically, at each time step t, the agent chooses her decision policy $\pi_t \in \Pi$ for some bounded domain Π, and after her commitment to π_t, a bounded loss function $f_t : \Pi \to [-B, B]$ for some constant $B > 0$ is revealed to her, which may be chosen adversarially. The agent thus incurs a loss of $f_t(\pi_t)$, and will update her decision to π_{t+1} using the feedback. We focus on the most basic setting where the agent chooses actions from a finite set $\mathcal{A}$ every round, which is also referred to as the *Experts Problem* (Littlestone & Warmuth, 1994; Hazan, 2016), without loss of much generality (c.f. Appendix B.4 for a discussion). In this case, Π becomes the simplex over $\mathcal{A}$, i.e., $\Pi = \Delta(\mathcal{A})$, and $f_t(\pi_t) = \langle \ell_t, \pi_t \rangle$ for some loss vector $\ell_t \in \mathbb{R}^d$ that may change over time, where $d := |\mathcal{A}|$.

At time step $t \in [T]$, the agent may receive either the full vector ℓ_t, or only the realized loss $\ell_t(a_t)$ for some $a_t \sim \pi_t(\cdot)$, as feedback, which will be referred to as online learning with *full-information feedback*, and that with *bandit feedback*, respectively. The latter is also referred to as the *adversarial/non-stochastic bandit* problem in the multi-armed bandit (MAB) literature. Note that hereafter, we will by default refer to this setting that does *not* make any assumptions on the loss sequence $(\ell_t)_{t\in[T]}$ simply as *online learning*. Moreover, if the loss functions change over time (usually with certain bounded variation), we will refer to it as *non-stationary online learning* for short, whose bandit-feedback version is also referred to as the *non-stationary bandit* problem.

Repeated games. The online learning setting above has an intimate connection to game theory. Consider a normal-form game $\mathcal{G} = \langle N, \{\mathcal{A}_n\}_{n\in[N]}, \{r_n\}_{n\in[N]} \rangle$, where N is the number of players, $\mathcal{A}_n$ and $r_n : \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \to \mathbb{R}$ are the action set and the payoff function of player n, respectively. The N players repeatedly play the game for T rounds, each player n maintains a strategy $\pi_{n,t} \in \Delta(\mathcal{A}_n)$ at time t, and takes action $a_{n,t} \sim \pi_{n,t}(\cdot)$. The *joint* action $a_t = (a_{1,t}, \cdots, a_{N,t})$ determines the payoff of each player at time t, $\{r_n(a_t)\}_{n\in[N]}$. From a single-player's (e.g., player n's) perspective, she encounters an online learning problem with (expected) loss function $\ell_t := -\mathbb{E}_{a_{-n,t}\sim\pi_{-n,t}(\cdot)}[r_n(\cdot, a_{-n,t})]$ at time t, where $-n$ denotes the index for players other than player n. We will refer to it as the *game setting* for short, and use the terms of "agent" and "player" interchangeably. The key difference between online learning and repeated games is in their interaction dynamics: online learning involves an agent facing a potentially adversarial, changing environment (or sequence of loss functions), while in repeated games, agents interact by playing the same game repeatedly, which might be less adversarial when they follow specific learning algorithms.

2.2 PERFORMANCE METRIC: REGRET

We now introduce *regret*, the core performance metric used in online learning and games. For a given algorithm $\mathcal{A}$, let $\pi_{\mathcal{A},t}$ denote the decision policy of the agent at time t generated by $\mathcal{A}$. Then, the regret, which is the difference between the accumulated (expected) loss incurred by implementing $\mathcal{A}$ and that incurred by the best-in-hindsight fixed decision, can be defined as

$$\text{Regret}_{\mathcal{A}}\left((f_t)_{t\in[T]}\right) := \sum_{t=1}^{T} f_t(\pi_{\mathcal{A},t}) - \inf_{\pi\in\Pi} \sum_{t=1}^{T} f_t(\pi).$$

In the Experts Problem, the definition can be instantiated as $\text{Regret}_{\mathcal{A}}((\ell_t)_{t\in[T]}) := \sum_{t=1}^{T}\langle \ell_t, \pi_{\mathcal{A},t}\rangle - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle$. With bandit-feedback, the regret guarantee may also take further expectation for $\text{Regret}_{\mathcal{A}}$, over the randomness of the $(\pi_{\mathcal{A},t})_{t\in[T]}$ generated. An algorithm is referred to as being *no-regret*, if $\text{Regret}_{\mathcal{A}}((f_t)_{t\in[T]}) \sim o(T)$, i.e., the regret grows sublinearly in T. Widely-known no-regret algorithms include follow-the-regularized-leader (FTRL) (Shalev-Shwartz & Singer, 2007), FTPL (Kalai & Vempala, 2005) (See Appendix B.3.1).

Dynamic regret		GPT-4	FTRL	FTPL
Full information	Gradual variation	12.61 ± 7.01	36.58 ± 24.51	35.19 ± 22.51
	Abrupt variation	30.0 ± 19.91	36.52 ± 27.68	36.24 ± 28.22
Bandit	Gradual variation	21.39 ± 10.86	37.64 ± 21.97	36.37 ± 20.7
	Abrupt variation	35.94 ± 28.93	36.52 ± 27.68	38.82 ± 26.17

Table 1: Dynamic regret of GPT-4 in non-stationary environment with either full information or bandit feedback. The experiments are validated by our framework (low p value and low $\widehat{\beta}_0$).

In non-stationary online learning, one also uses the metric of *dynamic regret* (Zinkevich, 2003), where the *comparator* in the definition also changes over time, as the best decision policy at each time t: $\text{D-Regret}_{\mathcal{A}}((f_t)_{t\in[T]}) := \sum_{t=1}^{T} f_t(\pi_{\mathcal{A},t}) - \sum_{t=1}^{T} \inf_{\pi\in\Pi} f_t(\pi)$, which is a stronger notion than $\text{Regret}_{\mathcal{A}}((f_t)_{t\in[T]})$ in that $\text{Regret}_{\mathcal{A}}((f_t)_{t\in[T]}) \le \text{D-Regret}_{\mathcal{A}}((f_t)_{t\in[T]})$.

3 Do Pre-Trained LLMs Have Regret? Experimental Validation

In this section, we explore the no-regret behaviors of representative LLMs (i.e., GPT-4 Turbo, GPT-4, and GPT-3.5), in the context of online learning and games. All experiments with LLMs are conducted using the public OpenAI Python API (Openai, 2023). We provided intuition why pre-trained LLM might have no-regret behavior in Appendix C.2.

Interaction protocol. To enable the sequential interactions with the LLM, we first describe the setup and objective of our experimental study. At each round, we incorporate the entire history of loss vectors of past interactions into our prompts, as concatenated texts, and ask the LLM agent to determine a policy that guides the decision-making for the next round. Note that since we hope to *evaluate* the intelligence level of pre-trained LLM through online learning, we only provide simple prompts that it should utilize the history information, without providing explicit rules of *how* to make use of the history information, nor asking it to *minimize regret* (in any sense). Detailed descriptions of the prompts are deferred to Appendix C.1.

3.1 Framework for No-Regret Behavior Validation

Before delving into the results, we propose two frameworks to rigorously validate no-regret behavior in algorithms over a *finite* T, which might be of independent interest. Details are in Appendix C.3.

Trend-checking framework. This framework is built upon non-parametric hypothesis testing.

H_0 : $(\text{Regret}(t)/t)_{t\in[T]}$ does not exhibit a decreasing trend;
H_1 : $(\text{Regret}(t)/t)_{t\in[T]}$ shows a decreasing trend.

Ideally, one should check if $\text{Regret}(t)/t$ approaches zero as t goes to infinity. With finite T values, testing these hypotheses provides a method to quantify this—whether we reject H_0 offers a way to measure it. To this end, one needs to count the number of $R(t)/t - R(t+1)/(t+1) > 0$, for which we have Proposition 1 to give some understanding of the probability it happens with various counts. We will report the p-value of H_0 as the output of this framework.

Regression-based framework. Alternatively, one can use the data $\{(t, \log \text{Regret}(t))\}_{t\in[T]}$ to fit a linear function $\log \text{Regret}(t) = \beta_0 \log t + \beta_1$, where the estimate of β_0, i.e., $\widehat{\beta}_0$, satisfying $\widehat{\beta}_0 < 1$ may be used to indicate the no-regret behavior.

3.2 Results: Online Learning

We now present the experimental results on the no-regret behavior of pre-trained LLMs in online learning in 1) arbitrarily changing environments, 2) non-stationary environments, and 3) bandit-feedback settings. We defer a detailed explanation to Appendix C.4. For **arbitrary changing environments**, the average regret (over multiple randomly generated instances) performance is presented in Figure 1, where we compare GPT-4 with well-known no-regret algorithms, FTRL with entropy regularization and FTPL with gaussian perturbations (with tuned parameters). It is seen that these pre-trained LLMs can indeed achieve no-regret, and often have smaller regrets than baselines. For **non-stationary environments**, the average dynamic regret results are presented in Table 1. It can be seen that GPT-4 achieves sublinear dynamic regret and outperforms Restart FTRL/FTPL. For **bandit-feedback settings**, we compare the performance with the counterparts of FTRL in the bandit-feedback setting, e.g., EXP3 (Auer et al., 2002) and the bandit-version of FTPL (Abernethy et al., 2015) in both Figure 10 and Table 1, where GPT-4 consistently achieves lower regret.

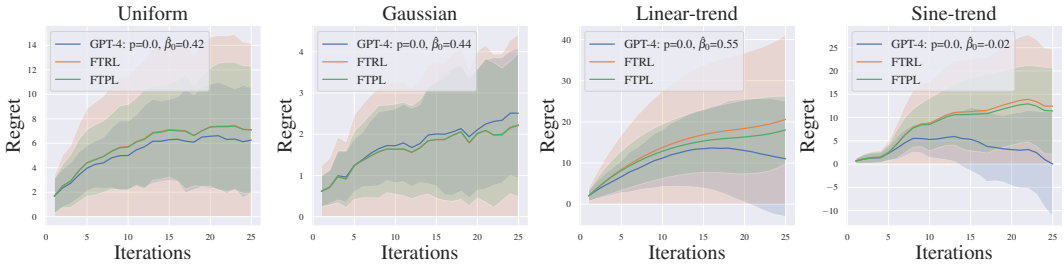


Figure 1: Regret of GPT-4 for online learning with full information feedback in 4 different settings. It performs comparably or better than well-known no-regret algorithms, FTRL, FTPL.

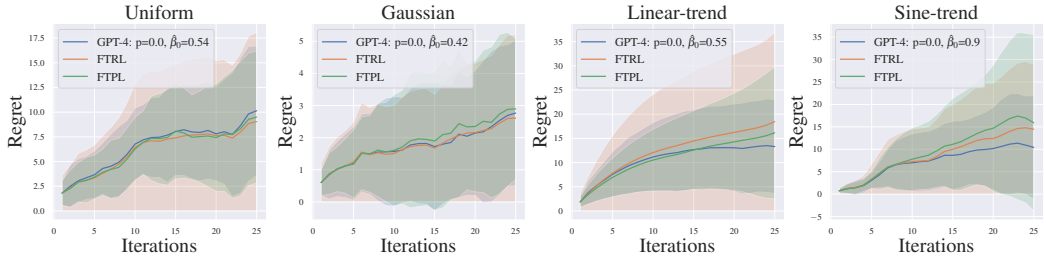


Figure 2: Regret of GPT-4 for online learning with bandit feedback in 4 different settings. It performs comparably or better than well-known no-regret algorithms, FTRL, FTPL.

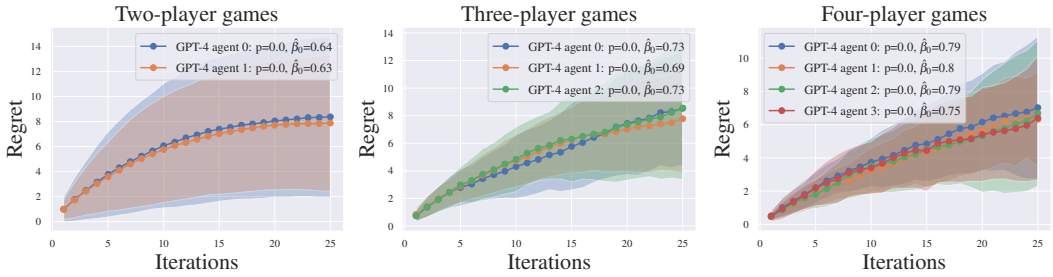


Figure 3: Regret of GPT-4 agents for repeated games of 3 different game sizes, where sublinear regret is validated by our both statistical framework.

3.3 Results: Multi-Player Repeated Games

We now consider the setting when multiple LLM agents make online strategic decisions in a *shared* environment. In this setting, at each round, the loss vectors each agent receives are determined by both her payoff matrix and the strategies of all other players. Note that the payoff matrix is not directly revealed to the LLM agent, but she has to make decisions in a completely online fashion (See Figure 7 for the prompt). This is a typical scenario in learning in games (Fudenberg & Levine, 1998). We introduce the detailed games in Appendix C.5. The results (Figure 3, 9 and 6) show that: 1) GPT-4 agents indeed have no-regret behavior when interacting in repeated games; 2) GPT-4 agents' regrets are comparable with those obtained by the FTRL algorithm, according to the frameworks in Section 3.1 and the graphic trends.

3.4 Pre-Trained LLM Agents May Still Have Regret

It seems tempting to conclude that pre-trained LLMs can achieve no-regret in both online learning and playing repeated games. However, is this capability *universal*? We show that the no-regret property might break for LLM agents if the loss vectors are generated in a more adversarial fashion. We provided two scenarios for regrettable behavior of GPT-4: (1) less-predictable loss sequences and 2) Adaptive loss sequences) in Figure 4. Detailed explanations for the counterexamples are in Appendix C.6. This observation has thus inspired us to design new ways to better promote the no-regret property of LLM agents, as to be detailed in Section 5. Before delving into the design of such a *stronger* LLM agent, we first provide some theoretical insights into why pre-trained LLMs already exhibit good no-regret behaviors oftentimes.

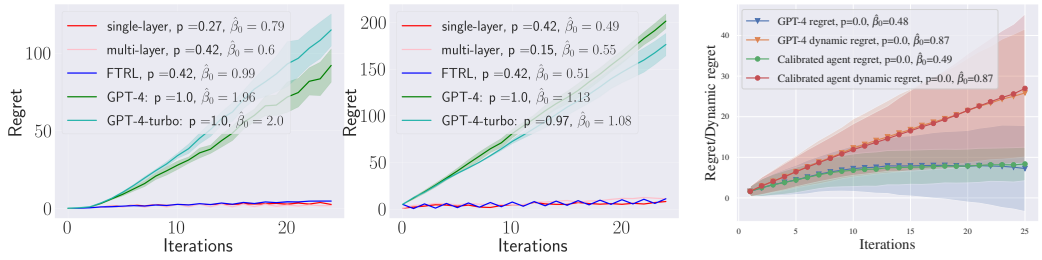


Figure 4: (left, mid) Failure of GPT-4 and GPT-4-turbo on two scenarios for regrettable behavior of GPT (left = less-predictable trend, right = adaptive loss), while transformers with regret-loss provide no-regret behaviors. (right) Comparison of GPT-4 with a calibrated agent on the test set, where the calibrated quantal response can perfectly capture the behavior of the GPT-4 agent.

4 Why are Pre-Trained LLMs (No-)Regret? Theoretical Insights

We now provide some plausible explanations about the no-regret behavior of pre-trained LLMs, as observed in Sections 3.2 and 3.3. Note that our explanations have to be *hypothetical* by nature, since to the best of our knowledge, the details of pre-training these popular LLMs (e.g., GPT-3.5 and GPT-4), regarding data distribution, training algorithm, etc., have not been revealed. We instead make the explanations based on some common assumptions and arguments in the recent literature on understanding LLMs/Transformers. As a preliminary result, we provided Observation 1 which indicates that pre-trained LLMs have similar regret as humans (who generate data). Detailed explanations are deferred in Appendix D. We discuss next under what (natural) behavioral models of humans (who generate the pre-training data), the no-regret behavior of pre-trained LLM agents emerge.

In Appendix D.2, we newly defined quantal response for the multiple-losses (Definition 2), which is defined as the standard quantal response against some scaled summation of the losses. We also provided implications of our generalized quantal response from behavioral economics. Moreover, our generalized quantal response is equivalent to the FTPL algorithm (Kalai & Vempala, 2005) with proper perturbation (Observation 2).

Case study: pre-training under canonical data distribution assumptions. Although some empirical validation, e.g., Ding et al. (2022), has confirmed that our generalized quantal response can model *human* behaviors in sequential decision-making, it remains unclear *how* to concretely achieve such behavior via pre-training. We here provide a case study of pre-training to gain some insights.

The training of LLMs often involves the method of *next token prediction*. When applying LLMs to sequential decision-making, the model receives the context of the decision-making sequence and then generates a series of *actions*. This process can be conceptualized as *predicting the optimal action* in the form of the next token prediction. For instance, (Yao et al., 2022; Shinn et al., 2023; Liu et al., 2023a;c) demonstrated how decision-making can be framed in this way. Meanwhile, large models such as Transformers are often (pre-)trained for sequential decision-making problems under a *stationary* underlying loss vector (Lin et al., 2023; Lee et al., 2023), which limits their ability to generalize the no-regret behavior to *arbitrary* loss sequences in our online learning setup. Thus, it is natural to ask: *Is it possible to have a generalized quantal response emerging as a consequence of this (optimal) action prediction, under* stationary *pre-training data distributions over environments?*

We model the pre-training data distribution as follows: there exists a random variable z, representing a static underlying loss of the individual sequential decision-making problem. The pre-training dataset, however, only contains t partial observations $(\ell_t)_{t\in[T]}$ of z due to the noises in data generation. The presence of noises could be attributed to z, a variable *privately* observed by the data-generator (human), representing the intentions of a human being/data-generator. LLM will be pre-trained with partial and noisy information about z. We assume that the optimal action a w.r.t the static underlying loss vector z is available in the pre-training dataset as the label. Note that this is consistent with the supervised pre-training process in the recent studies of Transformers for decision-making (Lee et al., 2023; Lin et al., 2023). Formally, we consider the following:

Assumption 1 (Factorization of pre-training distribution)**.** *We assume the joint distribution of* $(z, (\ell_i)_{i\in[T]}, a)$ *satisfies* $\mathbb{P}_{pre}(z, (\ell_i)_{i\in[T]}, a) = \mathbb{P}_{pre}(z)\mathbb{P}_{pre}((\ell_i)_{i\in[T]} \mid z)\mathbb{1}[a \in \arg\min_i z_i]$.

Furthermore, we consider the standard pre-training objective of Maximum likelihood estimation:

$$\min_{\theta} \quad \mathbb{E}_{\mathbb{P}_{\text{pre}}(z,(\ell_i)_{i\in[T]},a)} \sum_{t=1}^{T} [\log \text{LLM}_\theta(a \,|\, (\ell_i)_{i\in[t]})], \tag{4.1}$$

where LLM_θ denotes the LLM (usually a Transformer) parameterized by θ. We now analyze the performance of a trained LLM_θ in the following theorem:

Theorem 4.1. (Emergence of no-regret behavior)**.** *Suppose Assumption 1 holds with $\mathbb{P}_{pre}(z) = \mathcal{N}(0, \sigma^2 I)$, $\mathbb{P}_{pre}((\ell_i)_{i\in[T]} \,|\, z) = \prod_{i\in[T]} \mathbb{P}_{pre}(\ell_i \,|\, z)$ with $\mathbb{P}_{pre}(\ell_i \,|\, z) = \mathcal{N}(z, \sigma^2 I)$ for some $\sigma > 0$, and $\text{LLM}_{\theta^\star}$ that is sufficiently expressive minimizes Equation (4.1). Then, we have $\text{LLM}_{\theta^\star}(a \,|\, (\ell_i)_{i\in[t]}) = \mathbb{P}^{\eta_t}_{quantal}(a \,|\, (\ell_i)_{i\in[t]})$ with $P_{noise} = \mathcal{N}(0, I)$ and $\eta_t = \Theta(\sqrt{t})$ for any $t \in [T]$. Correspondingly, there exist algorithms that can utilize $\text{LLM}_{\theta^\star}$ to achieve no (dynamic) regret for (nonstationary) online learning with full-information/adversarial bandit.*

We presented the statement and proof of non-asymptotic bounds for the (dynamic) regret in various online learning problems using LLM in Appendix D.3. Furthermore, we demonstrated that the prior distribution of z could be replaced with a general distribution in Proposition 2. We also point out in Remark 3 that the pre-training distributions can be further relaxed. It is important to observe that even when pre-training is conducted solely with *stationary* loss vector generation, it can still lead to the *emergence of no-regret behavior* in online learning with potentially adversarial losses. Key in the proof is our newly established connection of pre-trained LLM models to the online learning algorithm of FTPL. Note that the data assumption here mostly follows that used in the recent literature, for the theoretical case study, and can be possibly generalized as follows. We provide comparison to Lee et al. (2023); Lin et al. (2023) in Appendix D.4.

Calibrating the degree of bounded rationality of actual LLMs. Here we here propose to calibrate the parameter $\{\eta_t\}_{t\in[T]}$, the degree of bounded rationality through behaviors of actual LLM agents with $\mathbb{P}_{\text{noise}}$ to be standard normal distribution. Then we run the generalized quantal response model with the calibrated $\{\eta_t^\star\}_{t\in[T]}$ on the N episodes of $\{(\ell_t^i)_{t\in[T]}\}_{j\in[N]}$ and compare it with the behavior of the real LLM agents. In Figure 4, we show the averaged regret for LLM agent and the calibrated generalized quantal response over N episodes. It can be seen that calibrated generalized quantal response can *very well capture* the behavior of the LLM agent, justifying the superiority of our proposed generalized quantal response model. We refer the details of calibration to Appendix D.5.

Finally, we acknowledge that for popular pre-trained LLM models like GPT-4, the canonical assumptions above may not hold. Moreover, the supervision labels, i.e., the optimal action given z, may not be available in practice in pre-training. Hence, it is completely possible to observe regret-table behaviors (c.f. Section 3.4). Motivated by these caveats, we next propose a new training loss that is *unsupervised*, and can promote no-regret behavior provably.

5 Guaranteed No-Regret by an Unsupervised Training Loss

In light of the observations in Section 3, we ask the question: *Is there a way to further enhance the no-regret property of LLM agents, hopefully* ***without*** *(optimal) action labels?* To address this question, we propose to train LLMs with a new unsupervised loss that naturally provides no-regret behaviors. This approach is akin to "instruction tuning" (Wei et al., 2021), which has been shown to have enhanced LLM ability when learning from context, both theoretical (Ahn et al., 2023; Mahankali et al., 2023; Zhang et al., 2023b) and empirical (Lu et al., 2023) support.

5.1 A New Unsupervised Training Loss: Regret-Loss

Intuitively, our new training loss is designed to enforce the trained LLMs to minimize the regret under an arbitrary sequence of loss vectors. Specifically, letting $\theta \in \Theta$ parameterize LLM_θ, we define the training loss as

$$\mathcal{L}(\theta) := \max_{\ell_1,\ldots,\ell_T} \quad \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}) \tag{5.1}$$

where $\|\ell_t\|_\infty \le B$ for $t \in [T]$. As shown in Kirschner et al. (2023), directly minimizing the max regret is computationally intractable. Hence, one may parameterize the LLM and resort to differentiable programming to solve it approximately. However, Equation (5.1) is not necessarily

differentiable with respect to parameter θ, if it does not satisfy the condition of Danskin's Theorem (Danskin, 1966), or even if it is differentiable (i.e., the maximizer of $(\ell_t)_{t\in[T]}$ is unique), computation of derivative is intractable since we need to calculate $\arg\max_{(\ell_i)_{i\in[T]}} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$ since we have $\inf$ in the definition of regret. Therefore, we provide a general framework so that we can approximate Equation (5.1) by the following surrogate:

$$\mathcal{L}(\theta, k, N) := \mathbb{E}\left[\frac{\sum_{j\in[N]} h(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}), k)}{\sum_{j\in[N]} f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}), k)}\right], \tag{5.2}$$

where $k \in \mathbb{R}$, N is a positive integer, $h : \mathbb{R} \to \mathbb{R}^+$ is a continuous function, and $f : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^+$ is a continuous function such that $\lim_{k\to\infty} \frac{f(R_1,k)}{f(R_2,k)} = \infty \cdot \mathbb{1}(R_1 > R_2) + \mathbb{1}(R_1 = R_2)$ where we use the convention of $\infty \cdot 0 = 0$. Examples of such an f include $f(x, k) = x^k$ and $\exp(kx)$.

We will sample N trajectories of loss sequences $(\ell_t^{(j)})_{t\in[T],j\in[N]}$ from some continuous probability distribution supported on $[-B, B]^{T\times N}$, and the expectation in Equation (5.2) is taken with respect to this distribution. Note that we do not require any statistical assumptions on $(\ell_t^{(j)})_{t\in[T],j\in[N]}$, in contrast to those in Section 4 when justifying the no-regret property of pre-trained LLMs. In Appendix E.2, we prove that $\lim_{N,k\to\infty} \mathcal{L}(\theta, k, N) = h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$, and also the uniform convergence of $\mathcal{L}(\theta, k, N)$ (i.e., $\lim_{N,k\to\infty} \sup_{\theta\in\Theta} |h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta, k, N)| = 0$ where Θ is some compact set of the LLM parameter). Hence, one can expect that minimizing the loss function Equation (5.2) with large enough k and N may promote the trained LLM to have a small regret value. We will refer to Equation (5.2) as the *regret-loss*. Similarly, we can also define *dynamic-regret-loss*, and the results to be presented next will also hold in this case (Remark 4 in Appendix E.3).

5.2 Guarantees via Regret-Loss Minimization

We first establish a *statistical* guarantee under general parameterizations of LLM_θ that is Lipschitz with respect to θ, including the Transformer-based models such as GPT-4 and most existing LLMs (see Proposition 3 for an example with formal statement). This guarantee focuses on their generalization ability when trained to minimize the empirical regret loss (c.f. Equation (E.2) in Appendix E.3), denoted as $\widehat{\mathcal{L}}(\theta, k, N, N_T)$. This involves replacing the expectation $\mathbb{E}$ with the empirical mean with N_T sampling in Equation (5.2). We will denote $\widehat{\theta}_{k,N,N_T} \in \arg\min_{\theta\in\Theta} \widehat{\mathcal{L}}(\theta, k, N, N_T)$.

Theorem 5.1. (Generalization gap). *For any $0 < \epsilon < 1/2$, with probability at least $1 - \epsilon$, we have*

$$\mathcal{L}\left(\widehat{\theta}_{k,N,N_T}, k, N\right) - \inf_{\theta\in\Theta} \mathcal{L}(\theta, k, N) \le \mathcal{O}\left(\frac{1 + \log(1/\epsilon)}{\sqrt{N_T}}\right), \tag{5.3}$$

for any N and sufficiently large k, where the empirical loss $\mathcal{L}$ is computed with N_T samples.

Through a careful use of Berge's Maximum Theorem (Berge, 1877), we prove that the right-hand side of Equation (5.3) does *not* depend on k and N, which allows us to take the limit of $\lim_{N\to\infty} \lim_{k\to\infty}$ without affecting the generalization bound. Thanks to the uniform convergence of $\mathcal{L}(\theta, k, N)$ (c.f. Appendix E.2), we further obtain the following corollary:

Corollary 1. (Regret). *Suppose h is a non-decreasing function and $\log f$ is a supermodular twice-continuously-differentiable function (i.e., $\frac{\partial^2 \log f}{\partial x \partial k} \ge 0$). For any $0 < \epsilon < 1/2$, with probability at least $1 - \epsilon$, we have*

$$h\Big(\lim_{N\to\infty} \lim_{k\to\infty} \max_{\|\ell_t\|_\infty \le B} \textit{Regret}_{\textit{LLM}_{\widehat{\theta}_{k,N,N_T}}}((\ell_t)_{t\in[T]})\Big) \le h\Big(\inf_{\theta\in\Theta} \max_{\|\ell_t\|_\infty \le B} \textit{Regret}_{\textit{LLM}_\theta}((\ell_t)_{t\in[T]})\Big) + \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{N_T}}\right). \tag{5.4}$$

Proofs of Theorem 5.1 and Corollary 1 are deferred to Appendix E.3. Therefore, if additionally, the LLM parameterization (i.e., Transformers) can realize a no-regret algorithm (for example, the single-layer self-attention model can construct FTRL, as to be shown in Section 5.3), then Corollary 1 means that the with a large enough number of samples N_T, the learned $\text{LLM}_{\widehat{\theta}_{k,N,N_T}}$ becomes no-regret, i.e., $\text{Regret}_{\text{LLM}_{\widehat{\theta}_{k,N,N_T}}}((\ell_t)_{t\in[T]}) = o(T)$, since the first term on the right-hand-side of Equation (5.4) would directly be $o(T)$ under the choice of $h(x) = \max\{0, x\}$. For other choices of h, one can use the inverse of h^{-1} (which always exists by our requirement of h) to ensure $\text{Regret}_{\text{LLM}_{\widehat{\theta}_{k,N,N_T}}}((\ell_t)_{t\in[T]})$ is of order $o(T)$.

Despite the power of previous results, one cannot use an *infinitely large* N and k in practical training. Hence, in the next subsection, we provide results when N is finite, for specific parameterizations of the LLMs using Transformers.

5.3 Minimizing Regret-Loss Can Automatically Produce Known Online Learning Algorithms

We now study the setting of minimizing Equation (5.2) when LLM_θ is specifically parameterized by Transformers. As an initial step, we focus on single-layer (linear) self-attention models, as in most recent theoretical studies of Transformers (Ahn et al., 2023; Zhang et al., 2023b; Mahankali et al., 2023), and the more practical setting with a finite $N = 1$. In this section, we drop superscript (N) in Equation (5.2). We sample ℓ_t for $t \in [T]$ by realizing some random variable Z. Here, Z is symmetric about zero (i.e., $Z \overset{d}{=} -Z$), and $\text{Var}(Z) = \Sigma$ is positive definite. Recent works of Ahn et al. (2023); Zhang et al. (2023b); Mahankali et al. (2023) have demonstrated that when a Transformer is trained by a certain loss, an optimal solution within the single-layer linear self-attention model class can emulate the *gradient descent* algorithm for linear regression. We aim to have a similar result for our regret-loss, justifying its usefulness in online learning. Firstly, we consider the following structure of single-layer self-attention model g (see a formal introduction in Appendix B.2):

$$g(Z_t; V, K, Q, v_c, k_c, q_c) := (V\ell_{1:t} + v_c\mathbf{1}_t^\intercal)\texttt{Softmax}\left((K\ell_{1:t} + k_c\mathbf{1}_t^\intercal)^\intercal \cdot (Qc + q_c)\right), \tag{5.5}$$

where $Z_t = (\ell_1, \ldots, \ell_t, c)$ and $V, K, Q \in \mathbb{R}^{d\times d}$ correspond to the value, key, and query matrices, respectively, $v_c, k_c, q_c \in \mathbb{R}^d$ correspond to the bias terms of the value, key, and query matrices, and $c \neq \mathbf{0}_d$ is a constant vector. We then have the following result.

Theorem 5.2. *The configuration in Equation (5.5) and $\Pi = B(0, R_\Pi, \|\cdot\|)$ for some $R_\Pi > 0$, (V, K, Q, v_c, k_c, q_c) such that $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ and $V = -R_\Pi \frac{T}{\sum_{t=1}^{T-1} 1/t}\Sigma^{-1}\mathbb{E}\left[\|\sum_{t=1}^T \ell_t\|\ell_1\ell_2^\intercal\right]\Sigma^{-1}$ is a first-order stationary point of Equation (5.2) with $N = 1$, $h(x) = x^2$. Moreover, if Σ is a diagonal matrix, then plugging this configuration to Equation (5.5) then $\texttt{Proj}_{\Pi,\|\cdot\|}$ would perform FTRL with an L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.*

We also consider the single-layer *linear* self-attention as follows, for which we can strengthen the results above from a stationary-point to an optimal-solution argument:

$$g(Z_t; V, K, Q, v_c, k_c, q_c) = \sum_{i=1}^{t}(V\ell_i + v_c)((K\ell_i + k_c))^\intercal \cdot (Qc + q_c)). \tag{5.6}$$

Theorem 5.3. *The configuration of a single-layer linear self-attention model in Equation (5.6) (V, K, Q, v_c, k_c, q_c) such that $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ and $\Pi = B(0, R_\Pi, \|\cdot\|)$ for some $R_\Pi > 0$, $V = -2R_\Pi\Sigma^{-1}\mathbb{E}\left(\|\sum_{t=1}^T \ell_t\|\ell_1\ell_2^\intercal\right)\Sigma^{-1}$ is **a global optimal solution** of Equation (5.2) with $N = 1$, $h(x) = x^2$. Moreover, every global optimal configuration of Equation (5.2) within the parameterization class of Equation (5.6) has the same output function g. If Σ is a diagonal matrix, plugging any global optimal configuration to Equation (5.6) then $\texttt{Proj}_{\Pi,\|\cdot\|}$ would perform FTRL with an-L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.*

Theorem 5.3 shows the capacity of self-attention Transformer model structures to realize online learning algorithms, thanks to the regret-loss we proposed. In particular, this can be achieved automatically by optimizing the new loss, *without* hard-coding the parameters of the Transformer.

The above results are for the case of FTRL with an L_2-regularizer, and it is possible to consider FTRL with an *entropy regularizer*, leading to the well-known Hedge algorithm (Freund & Schapire, 1997) that is more compatible with the simplex constraint on π. We defer the discussion of this case to Appendix E.7, together with the empirical validations through the training of our regret-loss. Through these theorems, we can also guarantee in the game setting that we can **efficiently find coarse correlated equilibria** since each player exhibits no-regret behavior in the matrix game.

5.4 Experimental Results for Minimizing Regret-Loss

We now provide **experimental results** for minimizing our regret-loss: 1) randomly-generated loss sequences (Figure 13); 2) loss sequences with a predictable trend (Figure 14); 3) repeated games (Figure 15; and 4) regrettable behavior examples for current LLMs (Figure 4). Details of the training setup can be found in Appendix E.8. We provide detailed experimental settings in Appendix E.9. We also provided an ablation study for the training Equation (5.2) loss in Appendix E.10.

REFERENCES

Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pp. 807–823. PMLR, 2014.

Jacob D Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pp. 337–371. PMLR, 2023.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*, 2023.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Mohammad Ahsanullah, Valery B Nevzorov, and Mohammad Shakil. *An introduction to order statistics*, volume 8. Springer, 2013.

Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *arXiv preprint arXiv:2305.16867*, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Anonymous. Large language models as rational players in competitive economics games. *ICLR 2024 submission*, 2023a. URL https://openreview.net/forum?id=NMPLBbjYFq.

Anonymous. Large language models as gaming agents. *ICLR 2024 submission*, 2023b. URL https://openreview.net/forum?id=iS3fQooCaa.

Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, 2023.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Claude Berge. *Topological spaces: Including a treatment of multi-valued functions, vector spaces and convexity*. Oliver & Boyd, 1877.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27, 2014.

Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. Regret minimization and the price of total anarchy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 373–382, 2008.

Philip Brookins and Jason Matthew DeBacker. Playing games with GPT: What can we learn about a large language model from canonical strategic games? *Available at SSRN 4493398*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Colin F Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2011.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolo Cesa-Bianchi, Philip M Long, and Manfred K Warmuth. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Transactions on Neural Networks*, 7(3):604–619, 1996.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2023a.

Yiting Chen, Tracy Xiao Liu, You Shan, and Songfa Zhong. The emergence of economic rationality of gpt. *arXiv preprint arXiv:2305.12763*, 2023b.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966.

Jingying Ding, Yifan Feng, and Ying Rong. Myopic quantal response policy: Thompson sampling meets behavioral economics. *arXiv preprint arXiv:2207.01028*, 2022.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Christoph Engel, Max RP Grossmann, and Axel Ockenfels. Integrating machine behavior into human subject experiments: A user-friendly toolkit and illustrations. *Available at SSRN*, 2023.

Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. Can large language models serve as rational players in game theory? a systematic analysis. *arXiv preprint arXiv:2312.05488*, 2023.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*, 2023.

Drew Fudenberg and David K Levine. *The theory of learning in games*, volume 2. MIT Press, 1998.

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *arXiv preprint arXiv:2301.13196*, 2023.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pp. 9118–9147. PMLR, 2022a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022b.

Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Mikołaj J Kasprzak, Ryan Giordano, and Tamara Broderick. How good is your gaussian approximation of the posterior? finite-sample computable error bounds for a variety of useful divergences. *arXiv preprint arXiv:2209.14992*, 2022.

Johannes Kirschner, Alireza Bakhtiari, Kushagra Chandak, Volodymyr Tkachuk, and Csaba Szepesvari. Regret minimization via saddle point optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Jonathan N Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *arXiv preprint arXiv:2306.14892*, 2023.

Chao Li, Xing Su, Chao Fan, Haoying Han, Cong Xue, and Chunmo Zheng. Quantifying the impact of large language models on collective opinion dynamics. *arXiv preprint arXiv:2308.03313*, 2023a.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*, 2023b.

Siyu Li, Jin Yang, and Kui Zhao. Are you in a masquerade? exploring the behavior and impact of large language model driven social bots in online social networks. *arXiv preprint arXiv:2307.10337*, 2023c.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. 2023d.

Zifan Li and Ambuj Tewari. Beyond the hazard rate: More perturbation algorithms for adversarial multi-armed bandits. *J. Mach. Learn. Res.*, 18:183–1, 2017.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *arXiv preprint arXiv:2310.08566*, 2023.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. Chain of hindsight aligns language models with feedback. *arXiv preprint arXiv:2302.02676*, 3, 2023a.

Yueyang Liu, Benjamin Van Roy, and Kuang Xu. Nonstationary bandit learning via predictive sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 6215–6244. PMLR, 2023b.

Zhihan Liu, Hao Hu, Shenao Zhang, Hongyi Guo, Shuqi Ke, Boyi Liu, and Zhaoran Wang. Reason for future, act for now: A principled architecture for autonomous llm agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023c.

Nunzio Lorè and Babak Heydari. Strategic behavior of large language models: Game structure vs. contextual framing. *arXiv preprint arXiv:2309.05898*, 2023.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*, 2023.

Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

Daniel L McFadden. Quantal choice analaysis: A survey. *Annals of Economic and Social Measurement, Volume 5, number 4*, pp. 363–390, 1976.

Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*, 2023.

Openai. Gpt-4 technical report. 2023.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–18, 2022.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

David Robinson and David Goforth. *The topology of the 2x2 games: a new periodic table*, volume 3. Psychology Press, 2005.

Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5): 1–42, 2015.

Tim Roughgarden, Vasilis Syrgkanis, and Eva Tardos. The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, 59:59–101, 2017.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*, 2022.

Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. Hebrew University, 2007.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69:115–142, 2007.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving AI tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Significant Gravitas. AutoGPT. URL https://github.com/Significant-Gravitas/AutoGPT.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Melanie Swan, Takashi Kido, Eric Roland, and Renato P dos Santos. Math agents: Computational infrastructure, mathematical embedding, and genomics. *arXiv preprint arXiv:2307.02502*, 2023.

Chen Feng Tsai, Xiaochen Zhou, Sierra S Liu, Jing Li, Mo Yu, and Hongyuan Mei. Can large language models play text games well? current state-of-the-art and open questions. *arXiv preprint arXiv:2304.02868*, 2023.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 2023b.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023c.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022b.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. Diving into the inter-consistency of large language models: An insightful analysis through debate. *arXiv preprint arXiv:2305.11595*, 2023.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023a.

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*, 2023b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

H Peyton Young. *Strategic learning and its limits*. OUP Oxford, 2004.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023a.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023b.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023c.

Qinlin Zhao, Jindong Wang, Yixuan Zhang, Yiqiao Jin, Kaijie Zhu, Hao Chen, and Xing Xie. Competeai: Understanding the competition behaviors in large language model-based agents. *arXiv preprint arXiv:2310.17512*, 2023.

Julian Zimmert and Yevgeny Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *The Journal of Machine Learning Research*, 22(1):1310–1358, 2021.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pp. 928–936, 2003.

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

CONTENTS

A Detailed 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., 2023) 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., 2022; Shinn et al., 2023; Wang et al., 2023c; Driess et al., 2023; Significant Gravitas). However, the performance of decision-

making has not been rigorously characterized via the regret metric in these works. Very recently, Liu et al. (2023c) 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. (2023c) 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. (2023) showed that multi-LLM *debate* can improve the reasoning and evaluation capabilities of the LLMs. Qian et al. (2023); Schick et al. (2022); Wu et al. (2023) demonstrated the potential of multi-LLM interactions and collaboration in software development, writing, and problem-solving, respectively. Zhang et al. (2023a) 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, with more recent observations in Anonymous (2023a;b). 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. (2023); 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. (2023a;b). Li et al. (2023c;a) studied the opinion/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 well-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. (2021) 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. (2022); Von Oswald et al. (2023); Dai et al. (2022); Giannou et al. (2023) showed

(among other results) that ICL comes from the fact 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. (2023b); 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. The result of a similar type (but for no-regret learning) will also be established in our work. Li et al. (2023d) established generalization bounds of ICL from a multi-task learning perspective. Zhang et al. (2023c) 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. (2022); Lee et al. (2023); Lin et al. (2023) investigated the in-context reinforcement learning property of Transformers. In particular, they showed that Transformers after supervised pretraining, where the supervision signals come from either good RL algorithms or optimal actions, can approximate online reinforcement learning algorithms for 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 an *unsupervised* strategic training loss to enforce the no-regret behavior. The ICL property has also played a critical role in the framework in Liu et al. (2023c) mentioned above.

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 (i.e., of order $o(T)$), 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., 2012), 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., 2023). Centering around the regret metric, our work has also explored the non-stationary bandit (Besbes et al., 2014) settings, as well as the repeated game setting where the environment itself consists of strategic agents (Cesa-Bianchi & Lugosi, 2006).

B Deferred Notation and Definition

B.1 Notation

For a finite set $\mathcal{S}$, we use $\Delta(\mathcal{S})$ to denote the simplex over $\mathcal{S}$. 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 $\boldsymbol{O}_{d\times d}$ as a $d \times d$ dimensional zero matrix. We define $[d] = \{1, 2, \ldots, d\}$. 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 $A \in \mathbb{R}^{m\times n}$, define $\|A\|_{\text{op}} := \max_{\|x\|_2\le 1} \|Ax\|_2$, $\|A\|_{2,\infty} := \sup \|A_i\|_2$, $\|A\|_{2,2} := \|(\|A_i\|_2)\|_2$ (which is also known as Frobenius norm), and $A_{-1} := A_n$ which indicates the last column vector. We define $\mathbb{R}^+ := \{x \mid x \ge 0\}$. For a set Π, $\text{diam}(\Pi) := \sup_{\pi_1,\pi_2\in\Pi} \|\pi_1 - \pi_2\|_2$. We define $\mathbb{1}(\mathcal{E}) = 1$ if $\mathcal{E}$ is true, and $\mathbb{1}(\mathcal{E}) = 0$ otherwise. For random variable sequence $(X_n)_{n\in\mathbb{N}}$ and random variable X, Y, we denote F_X as the cumulative distribution function of a random variable X, $X_n \xrightarrow{p} X$ if $\forall\epsilon > 0, \lim_{n\to\infty} P(|X_n - X| > \epsilon) = 0$,

$X_n \xrightarrow{d} X$ 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 $P(\lim_{n\to\infty} X_n = X) = 1$, and $\text{esssup}(X) = \inf\{M \in \mathbb{R} : P(X > M) = 0\}$. For a random variable X, we use $\text{supp}(X)$ to denote its support.

B.2 Detailed Definition for Appendix

(Linear) Self-attention. One key component in Transformers (Vaswani et al., 2017), the backbone of modern language models, is the *(self-)attention* mechanism. For simplicity, we here focus on introducing the *single-layer* self-attention architecture. The mechanism takes a sequence of vectors $Z = [z_1, \ldots, z_t] \in \mathbb{R}^{d\times t}$ as input, and outputs some sequence of $[\widehat{z}_1, \ldots, \widehat{z}_t] \in \mathbb{R}^{d\times t}$. For each $i \in [t]$ where $i > 1$, the output is generated by $\widehat{z}_i = (Vz_{1:i-1})\sigma((Kz_{1:i-1})^\intercal(Qz_i))$, where $z_{1:i-1}$ denotes the 1 to $i-1$ columns of Z, σ is either the $\texttt{Softmax}$ or $\texttt{ReLU}$ activation function, and for the initial output, $\widehat{z}_1 = \mathbf{0}_d$. Here, $V, Q, K \in \mathbb{R}^{d\times d}$ are referred to as the *Value*, *Query*, and *Key* matrices, respectively. Following the theoretical framework in Von Oswald et al. (2023); Mahankali et al. (2023), we exclude the attention score for a token z_i in relation to itself. For theoretical analysis, we also consider the *linear* self-attention model, where $\widehat{z}_i = (Vz_{1:i-1})((Kz_{1:i-1})^\intercal(Qz_i))$. We write this (linear) self-attention layer's output as $\texttt{(L) SA}_{(V,Q,K)}(Z)$. We define an M-head self-attention layer with $\theta = \{(V_m, Q_m, K_m)\}_{m\in[M]}$ as $\texttt{M- (L) SA}_\theta(Z) := \sum_{m=1}^M \texttt{(L) SA}_{(V_m,Q_m,K_m)}(Z)$. We define $\|\cdot\|_{\texttt{M- (L) SA}}$ as $\|\theta\|_{\texttt{M- (L) SA}} := \max_{m\in[M]}\{\|Q_m\|_{\texttt{op}}, \|K_m\|_{\texttt{op}}\} + \sum_{m=1}^M \|V_m\|_{\texttt{op}}$.

Transformers. For a multilayer perceptron (MLP) layer, it takes $Z = [z_1, \ldots, z_t] \in \mathbb{R}^{d\times t}$ as input, with parameter $\theta = (W_1, W_2) \in \mathbb{R}^{d'\times d} \times \mathbb{R}^{d\times d'}$ such that for each $i \in [t]$, the output is $\widehat{z}_i := W_2\sigma(W_1 z_i)$ where σ is either $\texttt{Softmax}$ or $\texttt{ReLU}$. We write the output of an MLP layer with parameter θ as $\texttt{MLP}_\theta(Z)$. Defining $\|\cdot\|_{\texttt{MLP}}$ as $\|\theta\|_{\texttt{MLP}} := \|W_1\|_{\texttt{op}} + \|W_2\|_{\texttt{op}}$ and $\texttt{ResNet}(f, Z) := Z + f(Z)$, we can define an L-layer Transformer with parameter $\theta = (\theta^{(lm)}, \theta^{(la)})_{l\in[L]}$ as $\texttt{TF}_\theta(Z) := Z^{(L)}$, where the output $Z^{(L)}$ is defined iteratively from $Z^{(0)} = \texttt{clip}_R(Z) := \min(-R, \max(R, Z))$ and

$$Z^{(l)} = \texttt{clip}_R\left(\texttt{ResNet}\left(\texttt{MLP}_{\theta^{(la)}}, \texttt{ResNet}\left(\texttt{M- (L) SA}_{\theta^{(lm)}}, Z^{(l-1)}\right)\right)\right).$$

We define a class of Transformers with certain parameters as $\Theta_{d,L,M,d',B_{\texttt{TF}}} := \{\theta = (\theta^{(lm)}, \theta^{(la)})_{l\in[L],m\in[M]} : \|\theta\|_{\texttt{TF}} \le B_{\texttt{TF}}\}$ where M is the number of head of self-attention. where $\|\theta\|_{\texttt{TF}} := \max_{l\in[L]}\left\{\|\theta^{(la)}\|_{\texttt{M- (L) SA}} + \|\theta^{(lm)}\|_{\texttt{MLP}}\right\}$, and $B_{\texttt{TF}} > 0$ is some constant. We assume R to be sufficiently large so that $\texttt{clip}$ does not take effect on any of our approximation results.

For general one: train without time-embedding.

input: $Z_t = [z_1, \cdots, z_t] \in \mathbb{R}^{d\times t}$, $z_i \in \mathbb{R}^d$. Here, $Z_{1:i} = [z_1, \cdots, z_i]$ for $i = 1, 2, \cdots, t$.

output: $O_l = O_{l-1} + \text{attn}(O_{l-1}) \in \mathbb{R}^{d\times t}$

where $\text{attn}(O_{l-1,i}) = V_{l-1}Z_{1:i}\sigma((K_{l-1}O_{1:i})^\intercal q_{l-1})$

Here $O_{l-1,i}$ is ith column of $O_{l-1} \in \mathbb{R}^{d\times t}$ and $V_{l-1}, K_{l-1} \in \mathbb{R}^{d\times d}$, $q_{l-1} \in \mathbb{R}^d$ are the trainable variable.

B.3 In-Context Learning

In-context learning is an emergent behavior of LLMs (Brown et al., 2020), which means that these models can adapt and learn from a limited number of examples provided within their immediate input context. In in-context learning, the prompt is usually constituted by a length of T in-context (independent) examples $(x_t, y_t)_{t\in[T]}$ and $(T+1)$th input x_{T+1}, so the $\text{LLM}((z_t)_{t\in[T]}, x_{T+1}))$ provides the inference of y_{T+1}.

B.3.1 Definition of FTRL algorithm and FTPL algorithm

Follow-the-regularized-leader (FTRL). The Follow-the-Regularized-Leader (FTRL) algorithm (Shalev-Shwartz, 2007) is an iterative method that updates policy based on the observed data and a regularization term. The idea is to choose the next policy that minimizes the sum of the past losses and a regularization term.

Mathematically, given a sequence of loss vectors $\ell_1, \ell_2, \ldots, \ell_t$, the FTRL algorithm updates the policy π at each time step t as follows:

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} \left(\sum_{i=1}^{t} \langle \ell_i, \pi \rangle + R(\pi) \right)$$

where $R(\pi)$ is a regularization term. The regularization term $R(\pi)$ is introduced to prevent overfitting and can be any function that penalizes the complexity of the model. A function $R(\pi)$ is said to be λ-strongly convex with respect to a norm $\|\cdot\|$ if for all $\pi, \pi' \in \Pi$:

$$R(\pi) \geq R(\pi') + \langle \nabla R(\pi'), \pi - \pi' \rangle + \frac{\lambda}{2} \|\pi - \pi'\|_2^2.$$

A key property that ensures the convergence and stability of the FTRL algorithm is the strong convexity of the regularization term $R(\pi)$. Strong convexity of $R(\pi)$ ensures that the optimization problem in FTRL has a unique solution. The FTRL algorithm's flexibility allows it to encompass a wide range of online learning algorithms, from gradient-based methods like online gradient descent to decision-making algorithms like Hedge.

Connection to online gradient descent (OGD). The Online Gradient Descent (OGD) (Cesa-Bianchi et al., 1996) algorithm is a special case of the FTRL algorithm when the regularization term is the L_2 norm, i.e., $R(\pi) = \frac{1}{2}\|\pi\|_2^2$ and $\Pi = \mathbb{R}^d$. In OGD, at each time step t, the policy π is updated using the gradient of the loss function:

$$\pi_{t+1} = \pi_t - \ell_t,$$

which is exactly the OGD algorithm. Therefore, the connection between FTRL and OGD can be seen by observing that the update rule for FTRL with L_2 regularization can be derived from the OGD update rule.

Connection to the Hedge algorithm. The Hedge algorithm (sometimes called as Multiplicative Weight Update algorithm) (Arora et al., 2012) is an online learning algorithm designed for problems where the learner has to choose from a set of actions (denoted as $\mathcal{A}$) at each time step and suffers a loss based on the chosen action. The FTRL framework can be used to derive the Hedge algorithm by considering an entropy regularization term. Specifically the regularization term is the negative entropy $R(\pi) = \sum_{j \in [d]} \pi_j \log \pi_j$, (here, d is the dimension of policy π) then the FTRL update rule yields the Hedge algorithm as

$$\pi_{(t+1)j} = \pi_{tj} \frac{\exp(-\ell_{tj}\pi_{tj})}{\sum_{i \in [d]} \exp(-\ell_{ti}\pi_{ti})}$$

for $j \in [d]$ where $d := |\mathcal{A}|$.

Follow-the-perturbed-leader (FTPL). Given a sequence of loss vectors $\ell_1, \ell_2, \ldots, \ell_{t-1}$, the Follow-the-Perturbed-Leader algorithm (Kalai & Vempala, 2005) updates the policy π at each time step t by incorporating a perturbation vector ϵ_t. This perturbation is sampled from a predefined distribution. The policy π_t for the next time step is chosen by solving the following optimization problem:

$$\pi_t = \mathbb{E}\left[\arg\min_{\pi \in \Pi} \langle \epsilon_t, \pi \rangle + \sum_{i=1}^{t-1} \langle \ell_i, \pi \rangle \right] \tag{B.1}$$

Here ϵ_t introduces randomness to the decision-making.

Relationship between FTRL and FTPL. FTPL with Exponential distribution Perturbations and FTRL with Entropy Regularization (i.e., Hedge) are equivalent. In addition, FTPL with Gaussian distribution Perturbations and FTRL with L_2 Regularization (i.e., OGD) are equivalent. However, this equivalence is typically not exact due to the randomization in FTPL. It's more of a theoretical observation that under certain mathematical conditions, the algorithms' update rules can be aligned. Usually, the two algorithms will have different behaviors, especially since FTPL inherently includes randomness while FTRL does not.

B.4 Why Focusing on Linear Loss Function $f_t(\pi) := \langle \ell_t, \pi \rangle$?

We note that focusing on the linear loss function $f_t(\pi) := \langle \ell_t, \pi \rangle$ does not lose generality. Specifically, for the general convex loss function $(f_t)_{t\in[T]}$, we have $f_t(\pi_{\mathcal{A},t}) - f_t(\pi) \leq \langle \nabla f_t(\pi_{\mathcal{A},t}), \pi_{\mathcal{A},t} - \pi \rangle$ for any $\pi \in \Pi$, which indicates

$$\begin{aligned}&\text{Regret}_{\mathcal{A}}((f_t)_{t\in[T]})\\ &\leq \sum_{t=1}^{T} \mathbb{E}[\langle \nabla f_t(\pi_{\mathcal{A},t}), \pi_{\mathcal{A},t} \rangle] - \inf_{\pi\in\Pi} \sum_{t=1}^{T} \mathbb{E}[\langle \nabla f_t(\pi_{\mathcal{A},t}), \pi \rangle].\end{aligned}$$

Therefore, one can regard the loss vector $(\ell_t)_{t\in[T]}$ as $\ell_t := \nabla f_t(\pi_{\mathcal{A},t})$ for $t \in [T]$, and control the actual regret by studying the linear loss function (Hazan, 2016). The same argument regarding the general convex f_t can be applied to the dynamic-regret value. In sum, an algorithm designed for online *linear* optimization can be adapted for online *convex* optimization, with the understanding that the instance received at round t corresponds to the gradient of the convex function evaluated at that round.

B.5 Six Representative Two-player General-sum Games

In game theory, there are six representative two-player general-sum games (Robinson & Goforth, 2005). Firstly, consider **the win-win game** represented by matrices $A = \begin{pmatrix} 1 & 4 \\ 1 & 2 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 4 \\ 1 & 2 \end{pmatrix}$ for players A and B, respectively. This setup fosters a cooperative dynamic, as both players receive identical payoffs, encouraging strategies that benefit both parties equally.

Contrastingly, **the Prisoner's Dilemma**, is depicted by $A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$. This game illustrates the conflict between individual and collective rationality, where players are tempted to pursue individual gain at the collective's expense, often resulting in suboptimal outcomes for both.

In an **unfair game**, represented by $A = \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 4 & 3 \\ 1 & 2 \end{pmatrix}$, the asymmetry in the payoff structure places one player at a disadvantage, regardless of the chosen strategy. This imbalance often reflects real-world scenarios where power or information asymmetry affects decision-making.

Cyclic games, with matrices $A = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & 4 \\ 2 & 1 \end{pmatrix}$, present a scenario where no stable equilibrium exists. The best strategy for each player changes in response to the other's actions, leading to a continuous cycle of strategy adaptation without a clear resolution.

Biased games, denoted by $A = \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}$, inherently favor one player, often reflecting situations where external factors or inherent advantages influence outcomes, leading to consistently unequal payoffs.

Finally, **the second-best game**, with matrices $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 4 \\ 3 & 2 \end{pmatrix}$, encapsulates scenarios where players settle for less-than-optimal outcomes due to constraints like risk aversion or limited options. This often results in players choosing safer, albeit less rewarding, strategies.

Each of these games exemplifies distinct aspects of strategic decision-making and interaction. From cooperative to competitive and balanced to biased scenarios, these matrices provide a rich landscape for exploring the nuances of game theory and human behavior.

C Deferred Explanations in Section 3

C.1 Ablation Study on Prompts

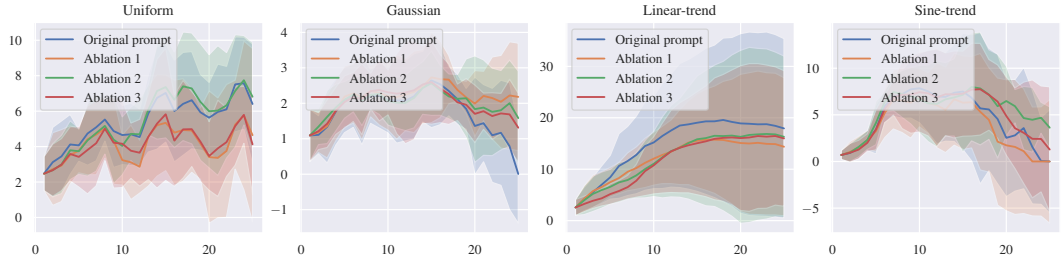


Figure 5: Ablation study on our prompt design.

To systematically understand the effects of our prompt on the final performance of the LLM agent, we create three different variants of our prompt and report the regret from different prompts in Figure 5. Specifically, for ablation1, we remove examples to illustrate the game rules. For ablation2, we remove the number of iterations. For ablation3, we remove the hints. We can see in Figure 5 that the performance of the LLM agent is consistent under different variants of prompts.

C.2 Intuition of Why Pre-trained Language Models May Exhibit No-regret Behaviors

Transformer-based LLMs have demonstrated impressive *in-context-learning* and few-shot learning capabilities (Brown et al., 2020; Garg et al., 2022; Min et al., 2022). One theoretical explanation is that, trained Transformers can implement the *gradient descent algorithm* on the testing loss (Akyürek et al., 2022; Von Oswald et al., 2023; Dai et al., 2022; Ahn et al., 2023; Zhang et al., 2023b; Mahankali et al., 2023), which is inherently *adaptive* to the loss function used at test time. On the other hand, it is known in online learning that the simple algorithm of *online gradient descent* (Zinkevich, 2003) can achieve no-regret. Hence, it seems reasonable to envision the no-regret online learning behavior of such meta-learners. However, it is not trivial due to the fundamental difference between multi-task/meta-learning and online learning settings, as well as that between *stationary* and *non-stationary/adversarial* environments in decision-making. Next, we provide both experimental and theoretical studies to validate the intuition above.

C.3 Statistical Frameworks for Validating No-Regret Behavior

We now introduce two statistical frameworks for validating the no-regret behaviors, which might be of independent interest.

Trend-checking framework We propose the following hypothesis test:

H_0 : The sequence $(\text{Regret}(t)/t)_{t=1}^{\infty}$ either diverges or converges to a positive constant.
H_1 : The sequence $(\text{Regret}(t)/t)_{t=1}^{\infty}$ converges to 0.

The notion of convergence is related to $T \to \infty$ by definition, making it challenging to verify directly. As an alternative, we propose a more tractable hypothesis test, albeit a weaker one, that still captures the essence of our objective:

H_0 : The sequence $(\text{Regret}(t)/t)_{t\in[T]}$ does not exhibit a decreasing trend.
H_1 : The sequence $(\text{Regret}(t)/t)_{t\in[T]}$ shows a decreasing trend.

For our analysis, we will employ non-parametric testing. Given the sequence $(\text{Regret}(1)/1, \ldots, \text{Regret}(T)/T)$, we aim to identify its decreasing behavior. Specifically, we will count the number where $R(t)/t - R(t+1)/(t+1) > 0$.

Proposition 1. [p-value of the null hypothesis] *Define the event* $\mathcal{E}(s,T) := \{$*The number of* $R(t)/t - R(t+1)/(t+1) > 0$ *for* $t = 1, \ldots, T$ *is at least* $s \geq \frac{T-1}{2}\}$. *Under the assumption that the null hypothesis* H_0 *holds, the probability of this event happening is bounded as* $\mathbb{P}_{H_0}(\mathcal{E}(s,T)) \leq \frac{1}{2^{T-1}} \sum_{t=s}^{T-1} \binom{T-1}{t}$.

Proof. Under the null hypothesis H_0, the probability p that $R(t)/t - R(t+1)/(t+1) > 0$ is less than $\frac{1}{2}$. Therefore, if we consider the event $\mathcal{E}(s,T)$, we have

$$\mathbb{P}_{H_0}(\mathcal{E}(s,T)) = \sum_{k=s}^{T-1} p^s (1-p)^{T-1-s} \binom{T-1}{k} \leq \frac{1}{2^{T-1}} \sum_{k=s}^{T-1} \binom{T-1}{k}$$

since $s \geq \frac{T-1}{2}$. □

For our experiments, where the primary focus is on $T = 25$, it's noteworthy that: $\mathbb{P}_{H_0}(\mathcal{E}(17,25)) < 0.032, \mathbb{P}_{H_0}(\mathcal{E}(19,25)) < 0.0035, \mathbb{P}_{H_0}(\mathcal{E}(21,25)) < 0.00014$, i.e., one can easily reject H_0 with high probability. We will report the p-value of H_0 as the output of this framework.

Regression-based framework. In complement to the statistical framework above, we propose an alternative approach by fitting the data. In particular, one can use the data $\{(t, \log \text{Regret}(t))\}_{t\in[T]}$ to fit a linear function $\log \text{Regret}(t) = \beta_0 \log t + \beta_1$, where the estimate of β_0, i.e., $\widehat{\beta}_0$, satisfying $\widehat{\beta}_0 < 1$ may be used to indicate the no-regret behavior. While being simple, it cannot be directly used when $\text{Regret}(t) < 0$, so we set $\log \text{Regret}(t)$ as -10. We will report $\widehat{\beta}_0$ as the output of this framework.

C.4 Detailed Environment Settings for the Experiments in Section 3.2

Online learning in arbitrarily changing environment. We first consider the setting with arbitrarily changing environments, with the following instantiations: 1) *Randomly-generated loss sequences*. At every timestep, we generate a random loss vector $\ell_t \sim \text{Unif}([0,10]^d)$ or $\ell_t \sim \mathcal{N}(5 \cdot \mathbf{1}_d, I)$ with clipping to $[0,10]$ to ensure the boundedness, such that the loss vectors of different timesteps can be arbitrarily distinct; 2) *Loss sequences with predictable trend.* Although real-world environments can change arbitrarily, they could often exhibit certain patterns. Therefore, we consider two representative trends, a *linear* trend and a *periodic* (sinusoid) trend. For the linear trend, we sample $a, b \sim \text{Unif}([0,10]^d)$ and let $\ell_t = (b-a)\frac{t}{T} + a$ for each $t \in [T]$. For the periodic trend, we sample $a, b \sim \text{Unif}[0,10]^d$ and let $\ell_t = 5(1 + \sin(at + b))$ for each $t \in [T]$. In the experiments, we choose $d = 2$. The average regret (over multiple randomly generated instances) performance is presented in Figure 1, where we compare GPT-4 with well-known no-regret algorithms, FTRL with entropy regularization and FTPL with gaussian perturbations (with tuned parameters). It is seen that these pre-trained LLMs can indeed achieve no-regret, and often have smaller regret than baselines.

Online learning (in non-stationary environment). We then experiment on the setting when the losses are still changing over time but their variations across time are bounded, more concretely, sublinear in T. Correspondingly, we consider the stronger metric of *dynamic regret* here to measure the performance. Note that without constraining the variation of the loss vectors, dynamic regret can be linear w.r.t T in the worst case. Hence, we generate the loss vectors in two different ways: 1) *Gradual variation.* We firstly sample $\ell_1 \sim \text{Unif}[0,10]^d$. Then for each $t \geq 2$, we uniformly and randomly generate ℓ_{t+1} under the constraint $\|\ell_{t+1} - \ell_t\|_\infty \leq \frac{1}{\sqrt{t}}$, such that the variations over time are guaranteed to satisfy $\sum_{t=1}^{T-1} \|\ell_{t+1} - \ell_t\|_\infty = \mathcal{O}(\sqrt{T})$; 2) *Abrupt variation.* We randomly generate $\ell_1 \sim \text{Unif}[0,10]^d$ and m time indices $\{t_i\}_{i\in[m]}$ from $\{1, 2, \cdots, T\}$. At each time step t_i for $i \in [m]$, the sign of the loss vector ℓ_{t_i} is flipped, i.e., we let $\ell_{t_i} \leftarrow 10 - \ell_{t_i}$. For the specific choice of $T = 25$ in our experiments, we choose $m = 3$. For both cases, the average dynamic regret results are presented in Table 1. It can be seen that GPT-4 achieves sublinear dynamic regret and outperforms Restart FTRL/FTPL.

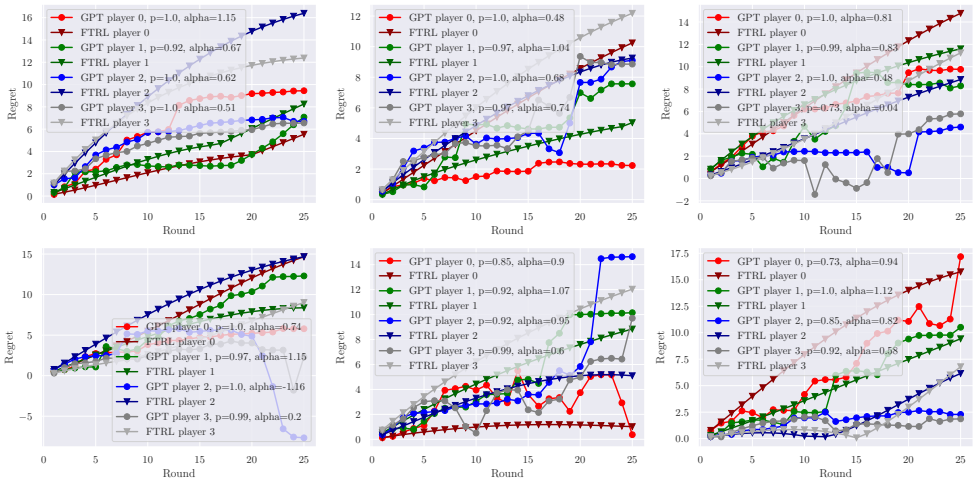


Figure 6: Regret of GPT-4 agents and the FTRL algorithm in 6 randomly generated four-player general-sum games. GPT-4 agents have comparable (even better) no-regret properties when compared with the FTRL algorithm.

Extension to bandit-feedback settings. Although pre-trained LLMs have achieved good performance in online learning with full-information feedback, it is unclear whether they can still maintain no-regret with only bandit feedback. For such problems, we modify the prompt and protocol of interactions slightly, where we still ask the LLM agent to provide a policy π_t at time step t, but manually sample one $a_t \sim \pi_t$ and then inform the agent of the sampled action a_t, together with the loss corresponding to that action, i.e., $\ell_{tj} \leftarrow \frac{\ell_{tj}}{\pi_{tj}} \mathbb{1}(a_t = j)$ for all $j \in [d]$ instead of providing ℓ_{ta_t}. Note such an operation of *re-weighting* the loss by the inverse of the probability is standard in online learning when adapting full-information-feedback no-regret algorithms to the bandit-feedback ones. Later, we will also show the provable benefits of such operations (c.f. Section 4). We compare the performance with the counterparts of FTRL in the bandit-feedback setting, e.g., EXP3 (Auer et al., 2002) and the bandit-version of FTPL (Abernethy et al., 2015) in both Figure 10 and Table 1, where GPT-4 consistently achieves lower regret.

C.5 Detailed Environment Settings for the Experiments in Section 3.3

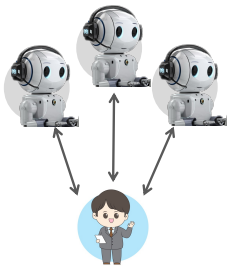
Representative games. We first test on all 6 representative two-player general-sum games (*win-win, prisoner's dilemma, unfair, cyclic, biased,* and *second best*) studied in Robinson & Goforth (2005) (see a detailed introduction of these games in Appendix B.5). For each type of the game, we conduct 20 repeated experiments.

Randomly generated games. To further validate the no-regret behavior of LLM agents, we also test on 50 randomly generated three-player general-sum games, and 50 randomly generated four-player general-sum games, where each entry of the payoff matrix is sampled randomly from $\mathrm{Unif}[0, 10]$. These are larger and more challenging settings than the two-player and structured cases above.

We summarize experimental results in Figure 3, which are similar to the above: for all types of games, GPT-4 agents achieve sublinear regret, which is comparable with that obtained by FTRL for most games (See Figure 9 and Figure 6 for more results).

C.6 Detailed Environment Settings for the Experiments in Section 3.4

To begin with, we consider a well-known example that *follow-the-leader* (FTL) algorithm Shalev-Shwartz (2012) suffers from linear regret Hazan (2016), where $\ell_{11} = 5, \ell_{12} = 0$ and $\ell_{t(2-t\%2)} = 10, \ell_{t(1+t\%2)} = 0$ for $t \geq 2$ where $\%$ is the modulo operation. Interestingly, GPT-4 agent can easily



Human Moderator's Prompt

You are playing a matrix game problem for T rounds. There are A number of actions. At each round, you need to choose a policy; it specifies your probability of choosing each action. This policy should be A-dimensional, and the sum of its components should equal 1. After that, you will be shown the reward vector for choosing each action. Remember that this reward vector is decided by the external system and can be potentially different for different rounds. It is not decided by what policies you have chosen. The reward vector is also A-dimensional. You can adjust your policy based on the reward vectors for all previous rounds. You're required to provide your policy in numeric format. Your response's last line should be formatted as 'Policy: [your A-dimensional policy]'. Let's think step by step. Explicitly examining history is important. Please explain how you chose the policy by guessing what reward you might receive for each action according to the history.

Figure 7: Demonstration of the prompts used for multi-player repeated games. A human moderator does not provide the game's payoff matrices to the LLM agents. Instead, at each round, the human moderator provides each player's own payoff vector history.

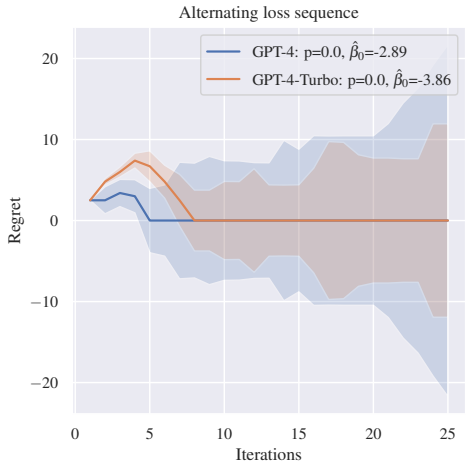


Figure 8: Regret of both GPT-4 and GPT-4-Turbo under the seminal counter-example for FTL.

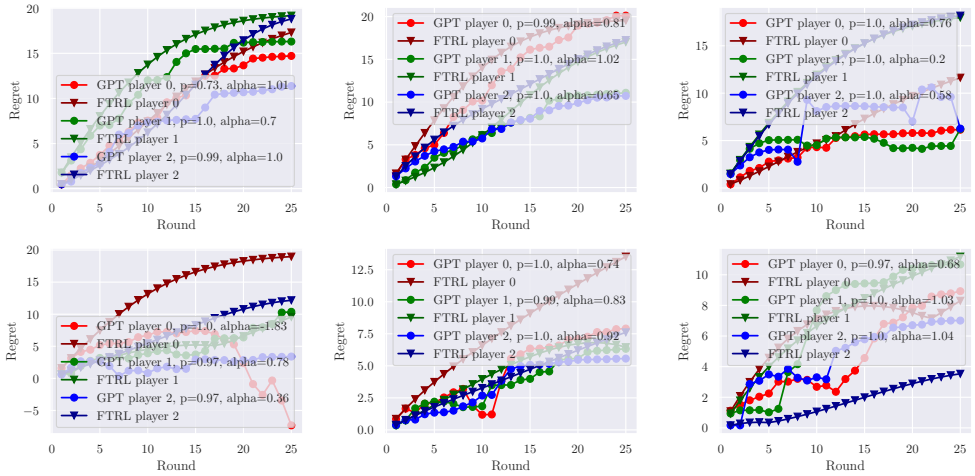


Figure 9: Regret of GPT-4 agents and the FTRL algorithm in 6 randomly generated three-player general-sum games. GPT-4 agents have comparable (even better) no-regret properties when compared with the FTRL algorithm.

identify the pattern for the loss sequence that the optimal action alternates, thus accurately predicting the loss it will receive and achieving near zero regret in Figure 8.

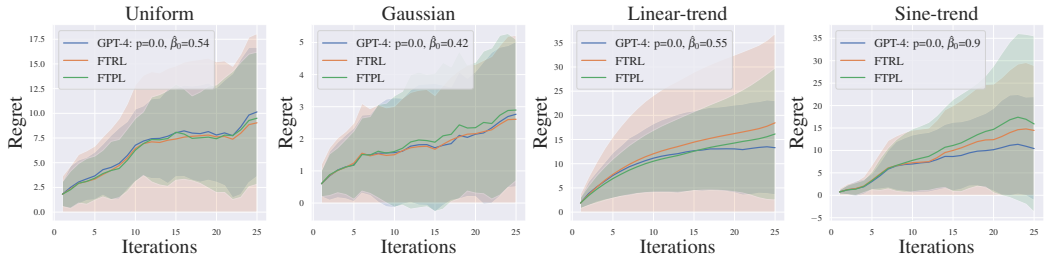


Figure 10: We report regret of GPT-4 for online learning with bandit feedback in 4 different settings. It performs comparably or better than the bandit version of well-known no-regret algorithms, FTRL, FTPL.

Less predictable loss sequence. Inspired by FTL observation, we design a new loss sequence that is *similar but less predictable*. Specifically, we construct the following (simple) loss sequence with 2 actions such that $\ell_{t(1+t\%2)} = \min(25/t, 10), \ell_{t(2-t\%2)} \sim \text{Unif}[9, 10]$ for $t \in [25]$.

Adaptive loss sequence. We also develop a simpler but more *adaptive* loss sequence that takes the full power of the adversary in the online learning setup. After the GPT-4 agent provides π_t, we choose ℓ_t such that $\ell_{t(\arg\max_i \pi_{ti})} = 10$ and $\ell_{t(3-\arg\max_i \pi_{ti})} = 0$. We report the regret averaged for the later two settings over 20 repeated experiments using GPT-4 and more advanced GPT-4 Turbo in Figure 4, where linear regret is confirmed by both trend-checking and regression-based frameworks.

D Deferred Explanation in Section 4

D.1 Pre-Trained LLMs Have Similar Regret as Humans (Who Generate Data)

Recently, a growing literature has evidenced that the intelligence level of LLM agents are determined by, and in fact mimic, those of human beings who generate the data for pre-training the models (Park et al., 2022; Argyle et al., 2023; Horton, 2023). The key rationale was that, LLMs (with Transformer parameterization) can approximate the *pre-training data distribution* very well (Xie et al., 2021; Zhang et al., 2023c; Lee et al., 2023). In such a context, one can expect that LLM agents can achieve similar regret as human decision-makers who generate the pre-training data, as we formally state below.

Observation 1. *An LLM agent is said to be pre-trained with an* ϵ-decision error *if, for any arbitrary* t *and loss sequences* $(\ell_i)_{i\in[t]}$*, the following condition holds:*

$$\sup_{\pi\in\Pi} \left|\mathbb{P}_{data}(\pi \mid (\ell_i)_{i\in[t]}) - \mathbb{P}_{LLM}(\pi \mid (\ell_i)_{i\in[t]})\right| \le \epsilon,$$

where $\mathbb{P}_{data}$ *and* $\mathbb{P}_{LLM}$ *are the pre-training data distribution and the pre-trained LLM model, respectively. Then the regret of an LLM agent with* ϵ*-decision error is bounded as:*

$$\begin{aligned} &\textit{(D-)Regret}_{LLM}\left((\ell_t)_{t\in[T]}\right) \\ &\in \left[\textit{(D-)Regret}_{data}\left((\ell_t)_{t\in[T]}\right) \pm \epsilon\|\ell_t\| \sup_{\pi\in\Pi}\|\pi\|\right], \end{aligned}$$

where $[a \pm b] := [a - b, a + b]$.

Observation 1 shows that the pre-trained LLM-agent's regret can be controlled by that of the pre-training dataset and the decision error ϵ. A small ϵ can be achievable if LLM is constructed with the Transformer architecture (Zhang et al., 2023c; Lin et al., 2023).

Proof. For given $(\ell_t)_{t\in[T]}$,

$$\sum_{t=1}^{T} \mathbb{P}_{\text{LLM}}(\pi_t \mid (\ell_i)_{i\in[t]})\langle \ell_t, \pi_t\rangle \le \sum_{t=1}^{T} (\mathbb{P}_{\text{data}}(\pi_t \mid (\ell_i)_{i\in[t]}) + \epsilon)\langle \ell_t, \pi_t\rangle$$

holds, so

$$\begin{aligned}
\text{Regret}_{\text{LLM}}((\ell_t)_{t\in[T]}) &= \sum_{t=1}^{T} \mathbb{P}_{\text{LLM}}(\pi_t \mid (\ell_i)_{i\in[t]})\langle \ell_t, \pi_t\rangle - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle \\
&\le \sum_{t=1}^{T} (\mathbb{P}_{\text{data}}(\pi_t \mid (\ell_i)_{i\in[t]}) + \epsilon)\langle \ell_t, \pi_t\rangle - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle \\
&= \sum_{t=1}^{T} \mathbb{P}_{\text{data}}(\pi_t \mid (\ell_i)_{i\in[t]})\langle \ell_t, \pi_t\rangle - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle + \sum_{t=1}^{T}\epsilon\langle \ell_t, \pi_t\rangle \\
&\le \text{Regret}_{\text{data}}((\ell_t)_{t\in[T]}) + \epsilon\|\ell\|_p\|\pi\|_q T
\end{aligned}$$

where $\frac{1}{p}+\frac{1}{q}=1$ and $p, q \ge 1$. Similarly, we can establish the lower bound for $\text{Regret}_{\text{LLM}}((\ell_t)_{t\in[T]})$. To prove the dynamic regret cases, we can simply change the term $\inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle$ in the regret case to $\sum_{t=1}^{T}\inf_{\pi\in\Pi}\langle \ell_t, \pi\rangle$. □

D.2 Deferred definition of generalized quantal response in Section 4

Generalized quantal response gives rise to follow-the-perturbed-leader. A seminal model for human behaviors is the *quantal response* model, which assumes that humans are often not *perfect* decision-makers, and their bounded rationality can be modeled through unseen latent variables that influence the decision-making process (McFadden, 1976; McKelvey & Palfrey, 1995). Formally, the quantal response is defined as follows:

Definition 1 (Quantal response)**.** *Given a loss vector $l \in \mathbb{R}^d$, a noise distribution $\epsilon \sim \mathbb{P}_{noise}$, and $\eta > 0$, the quantal response is defined as*

$$\mathbb{P}^{\eta}_{quantal}(a \mid \ell) = \mathbb{P}_{noise}\left(a \in \arg\min_{i\in[d]} \ (\ell + \eta\epsilon)[i]\right).$$

In essence, this implies that humans are rational but with respect to the latent variable $\ell + \eta\epsilon$ instead of ℓ. This addition of noise to the actual loss vector characterizes the bounded rationality of humans in decision-making.

Traditional quantal response formulations primarily focused on scenarios with a single loss vector. In online learning, given the *history* information, the decision-maker (either human or LLM agent) at each time t is faced with *multiple* loss vectors. Hence, we propose the following generalization to model human behavior in online decision-making.

Definition 2 (Quantal response against multiple losses)**.** *Given a set of loss vectors $(\ell_i)_{i\in[t]}$, a noise distribution $\mathbb{P}_{noise}$, and $\eta_t > 0$, the generalized quantal response is defined as* $\mathbb{P}^{\eta_t}_{quantal}\left(a \mid (\ell_i)_{i\in[t]}\right) := \mathbb{P}^{\eta_t}_{quantal}\left(a \,\middle|\, \sum_{i=1}^{t}\ell_i\right)$.

In simpler terms, the generalized quantal response is defined as the standard quantal response against some scaled summation of the losses. Note that such a *dynamic* version of quantal response also has implications from behavior economics, and has been recently used to model human behaviors in sequential decision-making (Ding et al., 2022) (in stochastic and stationary environments). Indeed, there is a direct relationship between our Definition 2 and a well-known no-regret learning algorithm in online learning, *follow-the-perturbed-leader* (Kalai & Vempala, 2005), whose formal definition can be found in Appendix B.3.1.

Observation 2. *Suppose at each time step t, the decision-maker (i.e., human or LLM agent) response follows Definition 2, then the decision-making process is equivalent to using the FTPL algorithm with proper perturbation.*

Before we move to the proof, we will define the random variable which has distribution $\mathbb{P}_{\text{noise}}$ as Z_{noise}

Proof.

$$\mathbb{P}^{\eta_t}_{\text{quantal}}(a \mid (\ell_i)_{i\in[t]}) := \mathbb{P}_{\text{noise}}\left(a \in \arg\min_{j\in[d]}(\sum_{i=1}^{t}\ell_i + \eta_t\epsilon)_j\right)$$

which is exactly the case that ϵ_t in Equation (B.1) satisfies $\epsilon_t \overset{d}{=} \eta_t\epsilon$. □

D.3 Deferred Proof of Theorem 4.1

Theorem 4.1. (Emergence of no-regret behavior). *Suppose Assumption 1 holds with $\mathbb{P}_{pre}(z) = \mathcal{N}(0, \sigma^2 I)$, $\mathbb{P}_{pre}((\ell_i)_{i\in[T]} \mid z) = \prod_{i\in[T]}\mathbb{P}_{pre}(\ell_i \mid z)$ with $\mathbb{P}_{pre}(\ell_i \mid z) = \mathcal{N}(z, \sigma^2 I)$ for some $\sigma > 0$, and $\text{LLM}_{\theta^\star}$ that is sufficiently expressive minimizes Equation (4.1). Then, we have $\text{LLM}_{\theta^\star}(a \mid (\ell_i)_{i\in[t]}) = \mathbb{P}^{\eta_t}_{quantal}(a \mid (\ell_i)_{i\in[t]})$ with $P_{noise} = \mathcal{N}(0, I)$ and $\eta_t = \Theta(\sqrt{t})$ for any $t \in [T]$. Correspondingly, there exist algorithms that can utilize $\text{LLM}_{\theta^\star}$ to achieve no (dynamic) regret for (nonstationary) online learning with full-information/adversarial bandit.*

To be specific,

(1) For online learning with full-information feedback, $\text{Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]}) \le \mathcal{O}(\sqrt{T\log d})$;

(2) For non-stationary online learning with full-information feedback, $\text{D-Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]}) \le \mathcal{O}((d\log d\, V_T)^{1/3}T^{2/3})$;

(3) For adversarial bandits, $\mathbb{E}[\text{Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]})] \le \mathcal{O}((\log d)^{1/2}T^{1/2+2\log\log T/\log T})$;

(4) For non-stationary bandits, $\mathbb{E}[\text{D-Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]})] \le \mathcal{O}((d\log d\, V_T)^{1/3}T^{2/3+2\log\log T/\log T})$,

where we define $V_T := \sum_{t=1}^{T-1}\|\ell_{t+1} - \ell_t\|_\infty$.

Proof. Due to the same reason as (Lee et al., 2023; Lin et al., 2023, Theorem 1), the minimizer of $\mathbb{E}_{\mathbb{P}_{\text{pre}}(z,(\ell_i)_{i\in[T]},a)}\sum_{t=1}^{T}[\log \text{LLM}_\theta(a \mid (\ell_i)_{i\in[t]})]$ is given by the posterior distribution, so $\text{LLM}_{\theta^\star}(a \mid (\ell_i)_{i\in[t]}) = \mathbb{P}_{\text{pre}}(a \mid (\ell_i)_{i\in[t]})$ as long as the LLM is sufficiently expressive. Therefore, we will calculate $\mathbb{P}_{\text{pre}}(z \mid (\ell_i)_{i\in[t]})$ for each $t \in [T]$. Since $z \sim \mathcal{N}(0, \sigma_1 I)$, and $\ell_i \mid z \sim \mathcal{N}(z, \sigma I)$, we have

$$z \mid (\ell_i)_{i\in[t]} \sim \mathcal{N}\left(\frac{1}{t+1}\sum_{i\in[t]}\ell_i, \frac{\sigma^2}{t+1}I\right)$$

by the posterior distribution of the normal distribution. Therefore, the corresponding noise level η_t in the procedure of FTPL is $\eta_t = \sqrt{t}$.

(1) Combining the above result with Lemma 1, we can derive the regret bound.

(2) Combining the above result with Lemma 1 and Lemma 3, we can prove a regret guarantee for online learning in a non-stationary environment with full information feedback.

(3) Combining the above result with Lemma 2, we can prove regret guarantee for adversarial bandits.

(4) Combining this result with Lemma 2 and Lemma 3, we can prove regret guarantee for adversarial bandits.

Now, we present Lemma 1 - Lemma 3.

Lemma 1 (FTPL Regret guarantee with full-information feedback.). *Suppose the noise distribution satisfies that $\mathbb{P}_{noise} = \mathcal{N}(\mathbf{0}_d, I)$ and $\eta_t = \Theta(\sqrt{t})$, then for online learning with full information feedback,*

$$\textit{Regret}((\ell_i)_{i\in[T]}) \le \mathcal{O}(\sqrt{T\log d}).$$

Proof. By Theorem 8 of Abernethy et al. (2014), we have

$$\text{Regret}((\ell_i)_{i\in[T]}) \le \sqrt{2\log d}\left(\eta_T + \sum_{t=1}^{T}\frac{1}{\eta_t}\|\ell_t\|_\infty^2\right).$$

Therefore, plugging $\eta_t = \Theta(\sqrt{t})$ and $\|\ell_t\|_\infty^2 \le 1$ provides

$$\text{Regret}((\ell_i)_{i\in[T]}) \le \sqrt{2\log d}\left(\sqrt{T} + \sum_{t=1}^{T}\frac{1}{\sqrt{t}}\right) \le \mathcal{O}(\sqrt{T\log d}).$$

□

Lemma 2 (Regret guarantee of FTPL with bandit feedback.)**.** *Suppose the noise distribution satisfies that $\mathbb{P}_{noise} = \mathcal{N}(\mathbf{0}_d, I)$ and $\eta_t = \Theta(\sqrt{t})$, then for online learning with full information feedback,*

$$\mathbb{E}[\textit{Regret}((\ell_i)_{i\in[T]})] \le \mathcal{O}(\sqrt{T\log d}).$$

Proof. The proof of the bandit problem is more complex. We first define the following notations. We denote $G_t = \sum_{t'=1}^{t}\ell_t$, $\widehat{G}_t = \sum_{t'=1}^{t}\widehat{\ell}_t$, $\Phi(G) = \max_\pi\langle\pi, G\rangle$, $\Phi_t(G) = \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0}_d,I)}\Phi(G+\eta_t\epsilon)$, and D_{Φ_t} to be the Bregman divergence with respect to Φ_t. By Li & Tewari (2017), $\pi_t = \nabla\Phi_t(\widehat{G}_t)$. Due to the convexity of Φ,

$$\Phi(G_T) = \Phi(\mathbb{E}[\widehat{G}_T]) \le \mathbb{E}\Phi(\widehat{G}_T).$$

Therefore,

$$\mathbb{E}[\text{Regret}((\ell_i)_{i\in[T]})] = \Phi(G_T) - \mathbb{E}[\sum_{t=1}^{T}\langle\pi_t,\widehat{\ell}_t\rangle] \le \mathbb{E}[\Phi(\widehat{G}_T) - \sum_{t=1}^{T}\langle\pi_t,\widehat{\ell}_t\rangle].$$

By recalling the definition of the Bregman divergence, we have

$$\begin{aligned}-\sum_{t=1}^{T}\langle\pi_t,\widehat{\ell}_t\rangle &= -\sum_{t=1}^{T}\langle\nabla\Phi_t(\widehat{G}_t),\widehat{\ell}_t\rangle = -\sum_{t=1}^{T}\langle\nabla\Phi_t(\widehat{G}_t),\widehat{G}_t-\widehat{G}_{t-1}\rangle\\ &= \sum_{t=1}^{T}D_{\Phi_t}(\widehat{G}_t,\widehat{G}_{t-1}) + \Phi_t(\widehat{G}_{t-1}) - \Phi_t(\widehat{G}_t).\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathbb{E}[\text{Regret}((\ell_i)_{i\in[T]})]\\ &\le \underbrace{\mathbb{E}[\sum_{t=1}^{T}D_{\Phi_t}(\widehat{G}_t,\widehat{G}_{t-1})]}_{(i)} + \underbrace{\mathbb{E}[\sum_{t=1}^{T}\Phi_t(\widehat{G}_{t-1}) - \Phi_{t-1}(\widehat{G}_{t-1})]}_{(ii)} + \underbrace{\mathbb{E}[\Phi(\widehat{G}_T) - \Phi_T(\widehat{G}_T)]}_{(iii)}.\end{aligned}$$

$(iii) \le 0$ due to the convexity of Φ. For (ii), we use Lemma 10 of Abernethy et al. (2014): we have

$$\mathbb{E}[\sum_{t=1}^{T}\Phi_t(\widehat{G}_{t-1}) - \Phi_{t-1}(\widehat{G}_{t-1})] \le \eta_T\mathbb{E}_\epsilon[\Phi(\epsilon)] \le \mathcal{O}(\sqrt{2T\log d}).$$

For (i), by Theorem 8 of Li & Tewari (2017), for any $\alpha\in(0,1)$, the following holds:

$$\mathbb{E}[\sum_{t=1}^{T}D_{\Phi_t}(\widehat{G}_t,\widehat{G}_{t-1})] \le \sum_{t=1}^{T}\eta_t^{\alpha-1}\frac{4}{\alpha(1-\alpha)} \le \frac{4}{\alpha(1-\alpha)}\mathcal{O}(T^{\frac{1+\alpha}{2}}).$$

By tuning $\alpha = \frac{2}{\log T}$, we proved that $\mathbb{E}[\text{Regret}((\ell_i)_{i\in[T]})] \le \mathcal{O}((\log d)^{\frac{1}{2}}T^{\frac{1}{2}+\frac{2\log\log T}{\log T}})$. □

Lemma 3. *Denote the variation of loss vectors as $V_T = \sum_{t=1}^{T-1} \|\ell_{t+1} - \ell_t\|_\infty$. Suppose there exists an algorithm for online learning with full information feedback with regret guarantee that Regret $\leq f(T, d)$ for some function f, where T denotes the horizon and d denotes the policy dimension. Then, there exists an algorithm that can achieve*

$$\textit{D-Regret}((\ell_i)_{i\in[T]}) \leq \min_{\Delta_T \in [T]} \left(\frac{T}{\Delta_T} + 1\right) f(\Delta_T, d) + 2\Delta_T V_T.$$

Similarly, suppose there exists an algorithm for adversarial bandit problem with regret guarantee that $\mathbb{E}$Regret $\leq g(T, d)$ for some function g; then there exists an algorithm that can achieve

$$\mathbb{E}[\textit{D-Regret}((\ell_i)_{i\in[T]})] \leq \min_{\Delta_T \in [T]} \left(\frac{T}{\Delta_T} + 1\right) g(\Delta_T, d) + 2\Delta_T V_T.$$

Proof. This is a direct result of the proof of Theorem 2 of Besbes et al. (2014). □

□

Note that in the first part of Theorem 4.1, we establish the fact that pre-trained LLM agent (under mild pre-training distribution assumptions) mimics FTPL with Gaussian perturbations and time-varying learning rate $\eta_t = \mathcal{O}(\sqrt{t})$ for $t \in [T]$. However, existing literature for FTPL usually does not address such a kind of learning rate η_t, especially for bandit problems, which makes the known regret guarantee not directly applicable. Nevertheless, it's still possible to extend the analysis to the time-varying learning rate case. Moreover, for a similar reason to the above, a non-stationary setting cannot be directly derived from the literature.

Remark 1. *In the context of an LLM minimizing Equation (4.1), the condition that a minimizer is equivalent to the FTPL algorithm does not strictly require that both the prior distribution and the conditional distribution of ℓ_i given z, t must be normal distributions. It is possible to consider the emergent algorithm as FTPL even if the posterior distribution of z aligns with $\sum_{i\in[t]} \ell_i + \epsilon_t$, where ϵ_t is independent of the sequence $(\ell_i)_{i\in[t]}$. The assumption that the prior distribution and the distribution of ℓ_i given z, t are normal is made not just to facilitate FTPL, but also to naturally encourage no-regret behavior in the algorithm.*

Remark 2. *Although Lee et al. (2023); Lin et al. (2023) have shown pre-trained LLM agents can solve stochastic bandit provably in light of the equivalence to posterior sampling, it cannot be used for adversarial bandit since posterior sampling can perform almost as badly as a worst-performing agent in some non-stationary environments (Liu et al., 2023b). In contrast, due to the equivalence to FTPL, our approach solves adversarial bandit problems with simple modifications.*

Remark 3 (Weaker data assumption)**.** *When interacting with LLMs, users will explicitly* prompt *the task, i.e., online learning in our context. Therefore, the LLM agent's policy is essentially $\textit{LLM}_\theta(a \mid (\ell_t)_{t\in[T]}, \texttt{OL=True})$, where the $\texttt{OL}$ (denoting "online learning") represents the extra prompts fed into LLMs (like a description of the problem setting) beyond only $(\ell_i)_{i\in[T]}$. This implies that Assumption 1 on $\mathbb{P}(z, (\ell_t)_{t\in[T]}, a)$ is essentially only required on $\mathbb{P}(z, (\ell_i)_{i\in[T]}, a, \texttt{OL=True})$ and the objective generalizes to $\mathbb{E}_{\mathbb{P}_{pre}(z,(\ell_i)_{i\in[T]},a,\texttt{OL})} \sum_{t=1}^{T} [\log \textit{LLM}_\theta(a \mid (\ell_i)_{i\in[t]}, \texttt{OL})]$, while we do not need any assumptions on $\mathbb{P}(z, (\ell_i)_{i\in[T]}, a, \texttt{OL=False})$, i.e., the training data that is not related to online learning problems.*

D.3.1 Extending Theorem 4.1 with a General Task Distribution

Proposition 2. *In Theorem 4.1, we can relax the assumption on $\mathbb{P}_{pre}(z)$ that we only require $\mathbb{P}_{pre}(z)$ to be i.i.d for each coordinate and $0 < \mathbb{P}_{pre}(z_j) < \infty$ for any $j \in [d]$, $z_j \in \mathbb{R}$, and the guarantee for (1) and (2) only increase by $\mathcal{O}(d^2 \log T)$.*

The key idea of the proof is that when t is large enough, the prior distribution does not affect the posterior distribution, which is also called the Bayesian Central Limit Theorem.

Proof. Since we extend Theorem 4.1 to settings with general task prior distribution only requiring the coordinates to be i.i.d, from now on, we consider j-th coordinate only. To begin with, fix $t \in [T]$,

we define the log-likelihood of the posterior as

$$L_t(z_j) = \log \prod_{i=1}^{t} \frac{1}{\sigma^d (2\pi)^{d/2}} e^{-\frac{1}{2\sigma^2}(\ell_{ij} - z_j)^2} = -n \log \sigma - \frac{n}{2} \log 2\pi - \sum_{i=1}^{n} \frac{1}{2\sigma^2} (\ell_{ij} - z_j)^2.$$

Then MLE estimator $\widehat{z}_{j,t}$ is defined as

$$\widehat{z}_{j,t} := \arg\max_{z_j \in \mathbb{R}} L_t(z_j) = \frac{1}{t} \sum_{i=1}^{t} \ell_{ij}.$$

We also define $\widehat{J}_t$ as:

$$\widehat{J}_t(z_j) := -\frac{\nabla^2 L_t(z_j)}{n} = \frac{1}{\sigma^2}.$$

For Assumption 1 of Kasprzak et al. (2022) to hold, any $\delta > 0$, $M_2 > 0$ suffices.

For Assumption 2 of Kasprzak et al. (2022) to hold, we can choose $\widehat{M}_1 = \max_{z_j \in [-\delta, 1+\delta]} \frac{1}{\mathbb{P}_{pre}(z_j)}$

For Assumption 7 of Kasprzak et al. (2022) to hold, we choose δ to be σ.

For Assumption 8 of Kasprzak et al. (2022) to hold, one can choose $M_2 = \frac{\sigma}{2}$.

For Assumption 9 of Kasprzak et al. (2022) to hold, we have

$$\kappa \leq - \sup_{(z_j - \widehat{z}_j)^2 \geq \delta} \frac{L_t(z_j) - L_t(\widehat{z}_{j,t})}{t} = -\frac{1}{2\sigma^2 t} \sup_{(z_j - \widehat{z}_{j,t})^2 \geq \delta} \sum_{i=1}^{t} (\ell_{ij} - \widehat{z}_{j,t})^2 - (\ell_{ij} - z_j)^2 = \frac{1}{4\sigma}.$$

For Assumption 10 of Kasprzak et al. (2022) to hold, we choose $M_1 = \sup_{z_j \in [-\delta, 1+\delta]} \| \frac{\nabla \mathbb{P}_{pre}(z_j)}{\mathbb{P}_{pre}(z_j)} \|_2$, $\widetilde{M}_1 = \sup_{z_j \in [-\delta, 1+\delta]} |\mathbb{P}_{pre}(z_j)|$.

By Theorem 6.1 of Kasprzak et al. (2022),

$$\begin{aligned}
&\int_{z_j} |\mathbb{P}(z_j/\sqrt{t} + \widehat{z}_j \mid (\ell_{ij})_{i\in[t]}) - Ce^{-\frac{1}{2\sigma^2} z^2}| dz_j \\
&\quad = \sqrt{t} \int_{z_j} |\mathbb{P}(z_j \mid (\ell_{ij})_{i\in[t]}) - \mathcal{N}(\widehat{z}_j, \sigma^2 t)| dz_j \leq D_1 t^{-1/2} + D_2 t^{1/2} e^{-t\kappa} + 2\widehat{\mathcal{D}}(t, \delta),
\end{aligned}$$

where

$$\begin{aligned}
D_1 &= \frac{\sqrt{\widetilde{M}_1 \widehat{M}_1}}{\sigma} \left(\frac{\sqrt{3}\sigma^2}{2\left(1 - \sqrt{\widehat{\mathcal{D}}(t,\delta)}\right)} M_2 + M_1 \right) \\
D_2 &= \frac{2\widehat{M}_1 \widehat{J}_t^p(\widehat{z}_j, \delta)}{(2\pi)^{1/2}(1 - \widehat{\mathcal{D}}^p(t,\delta))} \\
\widehat{\mathcal{D}}(t,\delta) &= e^{-\frac{1}{2}(\sqrt{t}-1)^2} \\
\widehat{J}_t^p(\widehat{z}_j, \delta) &= \frac{1}{\sigma^2} + \frac{\delta M_2}{3}.
\end{aligned}$$

Therefore, we conclude that TV distance for the joint random variable z is guaranteed that

$$\int_z |\mathbb{P}(z \mid (\ell_i)_{i\in[t]}) - \mathcal{N}(\widehat{z}, \sigma^2 t)| dz \leq \sum_{j=1}^{d} \int_{z_j} |\mathbb{P}(z_j \mid (\ell_{ij})_{i\in[t]}) - \mathcal{N}(\widehat{z}_j, \sigma^2 t)| dz_j \leq \mathcal{O}(d/t),$$

due to the independence of $(z_j)_{j\in[d]}$ conditioned on $(l_i)_{i\in[t]}$. Now we denote policy $\widehat{\pi}_t$ to be the policy obtained by smoothing using noise distribution $\mathbb{P}(z \mid (\ell_i)_{i\in[t]})$ and its corresponding regret as $\widehat{\text{Regret}}$. Similarly, we define π_t and Regret to be associated with $\mathcal{N}(\widehat{z}_j, \sigma^2 t)$. Therefore, we have

$$|\widehat{\text{Regret}} - \text{Regret}| \leq \sum_{t=1}^{T} d \|\pi_t - \widehat{\pi}_t\|_\infty \leq d \sum_{t=1}^{T} \int_z |\mathbb{P}(z \mid (\ell_i)_{i\in[t]}) - \mathcal{N}(\widehat{z}, \sigma^2 t)| dz = \mathcal{O}(d^2 \log T).$$

In other words, using $\mathbb{P}(z \,|\, (\ell_i)_{i\in[t]})$ as the smoothing distribution only increase regret by $\mathcal{O}(d^2 \log T)$. Similarly, it is easy to see that

$$|\widehat{\text{D-Regret}} - \text{D-Regret}| \leq \mathcal{O}(d^2 \log T).$$

□

D.4 Comparison to Lee et al. (2023); Lin et al. (2023)

Intriguingly, similar assumptions and objectives have also been considered in the very recent work of Lee et al. (2023); Lin et al. (2023) for studying in-context reinforcement learning (RL) property of Transformers under supervised pre-training. Lee et al. (2023) established its equivalence to *posterior sampling* (Osband et al., 2013), an important RL algorithm with provable regret guarantees when the environments are *stationary*, and Lin et al. (2023) generalized the study of the settings with algorithm distillation as in Laskin et al. (2022). However, their results cannot imply the no-regret guarantee in our online learning setting, due to the known facts that posterior sampling can perform poorly under potentially *adversarial* or *non-stationary* environments (Zimmert & Seldin, 2021; Liu et al., 2023b). In contrast, we here establish the equivalence of the pre-trained LLM to the FTPL algorithm (under different pre-training distribution specifications), with the ability to handle arbitrary loss sequences, even though the LLMs are only trained on a fixed distribution of *stationary* online learning problems.

D.5 Details on Calibration

Given N episodes of the LLM agent's behavior $\{(\ell_t^j, \pi_t^j)_{t\in[T]}\}_{j\in[N]}$, we propose to calibrate $\{\eta_t\}_{t\in[T]}$ by solving the following problem for each $t \in [T]$

$$\eta_t^\star \in \arg\min_{\eta^t} \sum_{j\in[N]} \left\| \pi_t^j - \mathbb{P}^{\eta_t}_{quantal}\Big(\cdot \mid \sum_{t'=1}^{t-1} \ell_{t'} \Big) \right\|_1.$$

We solve this single-variable optimization problem by grid search over $[0, 10]$.

E Deferred Explanation in Section 5

E.1 Basic Lemmas

Lemma 4 (Double sequences's iterated limit)**.** *Suppose that* $\lim_{m,n\to\infty} a_{mn} = L$*. Then the following are equivalent:*

- *For each* m, $\lim_{n\to\infty} a_{mn}$ *exist*
- $\lim_{m\to\infty} \lim_{n\to\infty} a_{mn} = L$.

Lemma 5 (Hoeffeding's inequality)**.** *Let* $X_1, X_2, \ldots, X_n$ *be independent random variables bounded by the intervals* $[a_i, b_i]$ *respectively. Define* $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ *and let* $\mu = \mathbb{E}[\bar{X}]$ *be the expected value of* $\bar{X}$*. Then, for any* $t > 0$*,*

$$P(|\bar{X} - \mu| \geq t) \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

Lemma 6 (Uniform convergence allow order changing of limit and infimum)**.** *If* $(f_n(x) : X \to \mathbb{R})_{n\in\mathbb{N}}$ *is a sequence of continuous functions that uniformly converges to a function* $f(x)$*, then* $\lim_{n\to\infty} \inf_{x\in X} f_n(x) = \inf_{x\in X} f(x)$ *holds.*

E.2 Mathematically Rigorous Argument for Section 5.1

In this section, we prove the mathematical details regarding $\mathcal{L}(\theta, k, N)$.

Claim 1 (Iterated limit of $\mathcal{L}(\theta, k, N)$ is the same with double limit of $\mathcal{L}(\theta, k, N)$)**.**

$$\lim_{N\to\infty}\lim_{k\to\infty} \mathcal{L}(\theta, k, N) = \lim_{N,k\to\infty} \mathcal{L}(\theta, k, N) = \lim_{k\to\infty}\lim_{N\to\infty} \mathcal{L}(\theta, k, N) = h(\max_{\ell_1,\ldots,\ell_T} \textit{Regret}_{\textit{LLM}_\theta}((\ell_t)_{t\in[T]}))$$

Proof. **Step 1.** $\lim_{N\to\infty}\lim_{k\to\infty}\mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T}\textbf{Regret}_{\textbf{LLM}_\theta}((\ell_t)_{t\in[T]}))$

Firstly, as h is a non-negative function, we have

$$\begin{aligned}\lim_{k\to\infty}\mathcal{L}(\theta,k,N) &= \mathbb{E}\lim_{k\to\infty}\left[\frac{\sum_{j\in[N]}h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]}f(R_{\text{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k)}\right]\\ &= \mathbb{E}_{(\ell_t^{(j)})_{t\in[T],j\in[N]}}\left[h(\max_{j\in[N]}\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))\right].\end{aligned}$$

By (Ahsanullah et al., 2013), we have $h(\max_{j\in[N]}\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]})) \xrightarrow{p} h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$ when $N\to\infty$, so $\lim_{N\to\infty}\lim_{k\to\infty}\mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$ holds.

Step 2. $\lim_{N,k\to\infty}\mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T}\textbf{Regret}_{\textbf{LLM}_\theta}((\ell_t)_{t\in[T]}))$

Now, we will calculate $\lim_{N,k\to\infty}\mathcal{L}(\theta,k,N)$.

Lemma 7. *For any* $0<\epsilon<1$,

$$\lim_{N,k\to\infty}\frac{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)<1-\epsilon)}{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)>1-\epsilon/2)} = 0$$

and

$$\lim_{N,k\to\infty}\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)<1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)>1-\epsilon/2)} = 0$$

hold with probability 1 where X_i*'s are i.i.d. random variable and* $\operatorname{esssup}(h(X_i)) = 1$

Proof of Lemma 7. Since f,h is nonnegative, and f and h is an non-decreasing function, we have

$$\frac{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)<1-\epsilon)}{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)>1-\epsilon/2)} \le \frac{(1-\epsilon)f(h^{-1}((1-\epsilon),k))|\{i\in[N]\mid(h(X_i)<1-\epsilon)\}|}{(1-\epsilon/2)f(h^{-1}((1-\epsilon/2),k))|\{i\in[N]\mid(h(X_i)>1-\epsilon/2)\}|}$$

and we know that

$$\frac{|\{i\in[N]\mid(h(X_i)<1-\epsilon)\}|}{|\{i\in[N]\mid(h(X_i)>1-\epsilon/2)\}|} \xrightarrow{a.s.} \frac{F(1-\epsilon)}{1-F(1-\epsilon/2)}$$

as $N\to\infty$ where F is the cumulative distribution function of random variable $h(X)$. Therefore, we have

$$\begin{aligned}0 \le \lim_{N,k\to\infty}\frac{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)<1-\epsilon)}{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)>1-\epsilon/2)} &\le \lim_{N,k\to\infty}\frac{(1-\epsilon)f(h^{-1}((1-\epsilon),k))|\{i\in[N]\mid(h(X_i)<1-\epsilon)\}|}{(1-\epsilon/2)f(h^{-1}((1-\epsilon/2),k))|\{i\in[N]\mid(h(X_i)>1-\epsilon/2)\}|}\\ &\le_{a.s.} \lim_{N,k\to\infty}\frac{(1-\epsilon)f(h^{-1}((1-\epsilon),k))}{(1-\epsilon/2)f(h^{-1}((1-\epsilon/2),k))}\frac{F(1-\epsilon)}{1-F(1-\epsilon/2)} = 0.\end{aligned}$$

Similarly, we have

$$\lim_{N,k\to\infty}\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)<1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)>1-\epsilon/2)} = 0$$

with probability 1. □

By Lemma 7, we have

$$\lim_{N,k\to\infty}\frac{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)>1-\epsilon)}{\sum_{i=1}^N f(X_i,k)h(X_i)} = \lim_{N,k\to\infty}\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)>1-\epsilon)}{\sum_{i=1}^N f(X_i,k)} = 1$$

with probability 1. Therefore, for any $0 < \epsilon < 1$, we have

$$\lim_{N,k\to\infty} \mathcal{L}(\theta,k,N) = \mathbb{E} \lim_{N,k\to\infty} \left[\frac{\sum_{j\in[N]} h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]})) f(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]} f(R_{\text{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k)} \right]$$

$$= h(\max_{\ell_1,\dots,\ell_T} R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$$

$$\times \mathbb{E} \lim_{N,k\to\infty} \left[\frac{\sum_{j\in[N]} \frac{h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))} f(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k) \mathbb{1}(\frac{h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))} > 1-\epsilon)}{\sum_{j\in[N]} f(R_{\text{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k) \mathbb{1}(\frac{h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))} > 1-\epsilon)} \right]$$

$$\geq (1-\epsilon) h(\max_{\ell_1,\dots,\ell_T} R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$$

where R_{LLM_θ} is a shorthand for $\text{Regret}_{\text{LLM}_\theta}$, which implies $\lim_{N,k\to\infty} \mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$.

Step 3. $\lim_{k\to\infty}\lim_{N\to\infty} \mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T} \textbf{Regret}_{\textbf{LLM}_\theta}((\ell_t)_{t\in[T]}))$

Lastly, if $N \to \infty$, we have

$$\lim_{N\to\infty} \mathcal{L}(\theta,k,N) = \mathbb{E} \lim_{N\to\infty} \left[\frac{\sum_{j\in[N]} h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]})) f(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]} f(R_{\text{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k)} \right]$$

$$= \frac{\mathbb{E} h(R_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]})) f(R_{\text{LLM}_\theta}((\ell_t^{(j)})}{\mathbb{E} f(R_{\text{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k)}$$

since we can use the law of large numbers on each numerator and denominator. Therefore, $\lim_{N\to\infty} \mathcal{L}(\theta,k,N)$ always exists for every k. Now, we use well-known properties of double sequence (Lemma 4), which provides $\lim_{k\to\infty}\lim_{N\to\infty} \mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$. □

Claim 2 (Uniform convergence of $\mathcal{L}(\theta,k,N)$ (with respect to k and N))**.** *$\mathcal{L}(\theta,k,N)$ uniformly converges to $h(\max_{\ell_1,\dots,\ell_T} \textit{Regret}_{\textit{LLM}_\theta}((\ell_t)_{t\in[T]}))$.*

Proof. We will provide a similar analysis with Lemma 7 as follows:

Lemma 8. *For any $0 < \epsilon < 1$ and $0 < \delta < 1$,*

$$\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) < 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) > 1-\epsilon)} = \widetilde{\mathcal{O}}\left(A(k,h,\epsilon) \left(\frac{1}{1-F(1-\epsilon/2)} + \frac{1}{\sqrt{N}} \right) \right)$$

with probability at least $1-\delta$ where X_i's are i.i.d. random variable and $\text{esssup}(h(X_i)) = 1$. *Here,* $A(k,h,\epsilon) = \frac{(1-\epsilon) f(h^{-1}((1-\epsilon),k))}{(1-\epsilon/2) f(h^{-1}((1-\epsilon/2),k))}$.

Proof of Lemma 8. For the same reason as the proof of Lemma 7, we have

$$\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) < 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) > 1-\epsilon/2)} \leq \frac{f(h^{-1}((1-\epsilon),k))|\{i\in[N] \mid (h(X_i) < 1-\epsilon)\}|}{f(h^{-1}((1-\epsilon/2),k))|\{i\in[N] \mid (h(X_i) > 1-\epsilon/2)\}|}.$$

$\frac{1}{N}|\{i \in [N] \mid (h(X_i) < 1-\epsilon)\}| = F(1-\epsilon) + \widetilde{\mathcal{O}}(1/\sqrt{N})$ with probability at least $1-\delta$ due to Hoeffeding's inequality (Lemma 5). Similarly, we have $\frac{1}{N}|\{i \in [N] \mid (h(X_i) > 1-\epsilon/2)\}| = 1 - F(1-\epsilon/2) + \widetilde{\mathcal{O}}(1/\sqrt{N})$ with probability at least $1-\delta$, where F is the cumulative distribution function of random variable $h(X)$. Therefore,

$$\frac{|\{i\in[N] \mid (h(X_i) < 1-\epsilon)\}|}{|\{i\in[N] \mid (h(X_i) > 1-\epsilon/2)\}|} = \frac{F(1-\epsilon)}{1-F(1-\epsilon/2)} + \widetilde{\mathcal{O}}(\sqrt{1/N}) \leq \frac{1}{1-F(1-\epsilon/2)} + \widetilde{\mathcal{O}}(\sqrt{1/N}).$$

with probability at least $1-\delta$. Finally, we have

$$\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) < 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) > 1-\epsilon)} < \frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) < 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i) > 1-\epsilon/2)} \leq A(k,h,\epsilon)\left(\frac{1}{1-F(1-\epsilon/2)} + \widetilde{\mathcal{O}}(\frac{1}{\sqrt{N}}) \right).$$

Note that $\lim_{k\to\infty} A(k,h,\epsilon) = 0$ by the definition of f. □

By Lemma 8, we have

$$\frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)\geq 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)} \geq \frac{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)< 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)> 1-\epsilon/2)} \geq \frac{1}{1+A(k,h,\epsilon)(\frac{1}{1-F(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}.$$

Therefore,

$$\begin{aligned}1 \geq \frac{\sum_{i=1}^N f(X_i,k)h(X_i)}{\sum_{i=1}^N f(X_i,k)} &\geq \frac{\sum_{i=1}^N f(X_i,k)h(X_i)\mathbb{1}(h(X_i)\geq 1-\epsilon)}{\sum_{i=1}^N f(X_i,k)\mathbb{1}(h(X_i)\geq 1-\epsilon)}\frac{1}{1+A(k,h,\epsilon)(\frac{1}{1-F(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\\ &\geq \frac{1-\epsilon}{1+A(k,h,\epsilon)(\frac{1}{1-F(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\end{aligned}$$

with probability at least $1-\delta$.

Now, for any $\epsilon>0$ and $\delta>0$, we have

$$\begin{aligned}0 &\leq h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta,k,N)\\ &\leq h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))\left(1-\frac{(1-\delta)(1-\epsilon)}{1+A(k,h,\epsilon)(\frac{1}{1-F_\theta(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\right)\end{aligned}$$

where F_θ is the cumulative distribution function of the random variable $\frac{h(\text{Regret}_{\text{LLM}_\theta}(\ell_t)_{t\in[T]}))}{h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))}$. Note that

$$1-F_\theta(1-\epsilon/2) = \mathbb{P}(h(\text{Regret}_{\text{LLM}_\theta}(\ell_t)_{t\in[T]})) > (1-\epsilon/2)h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$$

is a continuous function on θ since we assume LLM_θ is a continuous function, $(\ell_t)_{t\in[T]}$ has a continuous distribution, and $\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$ is a continuous function on LLM_θ and $(\ell_t)_{t\in[T]}$. Since we consider the compact Θ (as several literature in Transformer Bai et al. (2023)), we have $p(\epsilon) := \min_{\theta\in\Theta} 1-F_\theta(1-\epsilon/2) > 0$. Therefore,

$$\left(1-\frac{(1-\delta)(1-\epsilon)}{1+A(k,h,\epsilon)(\frac{1}{1-F_\theta(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\right) \leq \left(1-\frac{(1-\delta)(1-\epsilon)}{1+A(k,h,\epsilon)(\frac{1}{p(\epsilon)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\right) \tag{E.1}$$

and we know that $\lim_{N,k\to\infty} 1+A(k,h,\epsilon)(\frac{1}{p(\epsilon)}+\widetilde{\mathcal{O}}(\sqrt{1/N})) = 1$, which is not dependent on θ, we can conclude $\lim_{N,k\to\infty}\sup_{\theta\in\Theta}|h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta,k,N)| = 0$. □

Claim 3 (Double limit of supremum)**.** $\lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta}|\mathcal{L}(\theta,k,N) - h(\max_{\ell_1,\ldots,\ell_T}\textit{Regret}_{\textit{LLM}_\theta}((\ell_t)_{t\in[T]}))| = 0$.

Proof. Since $h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) \geq \mathcal{L}(\theta,k,N)$, we will prove $\lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta}(h(\max_{\ell_1,\ldots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta,k,N)) = 0$.

Lemma 9. $\frac{\sum_{i=1}^N f(X_i,k_1)h(X_i)}{\sum_{i=1}^N f(X_i,k_1)} \leq \frac{\sum_{i=1}^N f(X_i,k_2)h(X_i)}{\sum_{i=1}^N f(X_i,k_2)}$ *holds if* $k_1\leq k_2$.

Proof. By multiplying $(\sum_{i=1}^N f(X_i,k_1))(\sum_{i=1}^N f(X_i,k_2))$ to both sides of the lemma, this is equivalent to $\sum_{1\leq i\neq j\leq N} f(X_i,k_1)h(X_i)f(X_j,k_2) \leq \sum_{1\leq i\neq j\leq N} f(X_i,k_1)h(X_j)f(X_j,k_2)$. This is equivalent to

$$\sum_{1\leq i\neq j\leq N}(f(X_j,k_1)f(X_i,k_2) - f(X_i,k_1)f(X_j,k_2))(h(X_i)-h(X_j)) \geq 0$$

which is true since if $X_i \geq X_j$, $(f(X_j,k_1)f(X_i,k_2) - f(X_i,k_1)f(X_j,k_2)) \geq 0$ due to the log-increasing difference of f. □

Therefore, $\mathcal{L}(\theta, k, N)$ is a monotone nondecreasing function of k if N is fixed, which indicates that $\lim_{k\to\infty} \sup_{\theta\in\Theta}(h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta, k, N))$ exists, as $\mathcal{L}(\theta, k, N)$ is also bounded. Therefore, by Lemma 4 and Claim 2, $\lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta}|\mathcal{L}(\theta, k, N) - h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))|$ exists and this value should be 0. □

Claim 4. $\lim_{N,k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) = \lim_{N\to\infty}\lim_{k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) = \inf_{\theta\in\Theta} h(\max_{\ell_1,\ldots,\ell_T} \textit{Regret}_{\textit{LLM}_\theta}((\ell_t)_{t\in[T]}))$ *holds.*

Proof. Firstly, by Lemma 6, we have $\lim_{N,k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) = \inf_{\theta\in\Theta} h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$. Plus, we already know that $\mathcal{L}(\theta, k, N)$ iis a monotone nondecreasing function of k if N is fixed (Lemma 9) and it is bounded, $\lim_{k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N)$ always exists. Therefore, by Lemma 4, we also have $\lim_{N\to\infty}\lim k\to\infty\inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) = \inf_{\theta\in\Theta} h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$. □

E.3 Deferred Proof of Theorem 5.1 and Corollary 1

Definition 3 (Empirical loss function)**.** *We define the empirical loss $\widehat{\mathcal{L}}$ computed with N_T samples as follows:*

$$\widehat{\mathcal{L}}(\theta, k, N, N_T) := \sum_{s=1}^{N_T}\frac{1}{N_T}\left[\frac{\sum_{j\in[N]} h(R_{\textit{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]}))f(R_{\textit{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]}), k)}{\sum_{j\in[N]} f(R_{\textit{LLM}_\theta}((\ell_{s,i}^{(j)})_{t\in[T]}), k)}\right] \tag{E.2}$$

where $(\ell_{s,t}^{(j)})_{j\in[N],t\in[T]}$ indicates sth sampling of $(\ell_t^{(j)})_{j\in[N],t\in[T]}$ for estimating $\mathcal{L}$.

Theorem 5.1. (Generalization gap)**.** *For any $0 < \epsilon < 1/2$, with probability at least $1-\epsilon$, we have*

$$\mathcal{L}\left(\widehat{\theta}_{k,N,N_T}, k, N\right) - \inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) \le \mathcal{O}\left(\frac{1+\log(1/\epsilon)}{\sqrt{N_T}}\right), \tag{5.3}$$

for any N and sufficiently large k, where the empirical loss $\mathcal{L}$ is computed with N_T samples.

Proof. Firstly, we point out that the Transformer structure has a Lipschtiness with respect to the parameter. We adapt the result from (Bai et al., 2023, Section J.1), which is about the Lipschitzness of Transformer:

Proposition 3. *The function $\texttt{TF}_\theta$ is $C_{\texttt{TF}} := L\left((1+B_{\texttt{TF}}^2)(1+B_{\texttt{TF}}^2R^3)\right)^L B_{\texttt{TF}}R(1+B_{\texttt{TF}}R^2+B_{\texttt{TF}}^3R^2)$ - Lipschitz function, i.e.,*

$$\|\texttt{TF}_{\theta_1}(Z) - \texttt{TF}_{\theta_2}(Z)\|_{2,\infty} \le C_{\texttt{TF}}\|\theta_1-\theta_2\|_{\texttt{TF}}.$$

Now, we set C_θ as a Lipshitz constant for the LLM. Now, we prove that regret is also a Lipschtiz function with respect to the Transformer's parameter.

Lemma 10 (Lipschitzness of Regret value)**.** *The function $\textit{Regret}_{g_\theta}$ is $C_{\texttt{Reg}} := B\sqrt{d}\|A\|_{\texttt{op}}TC_\theta$ - Lipschitz function, i.e.,*

$$|\textit{Regret}_{g_{\theta_1}}((\ell_t)_{t=1}^T, T) - \textit{Regret}_{g_{\theta_2}}((\ell_t)_{t=1}^T, T)| \le C_{\texttt{Reg}}\|\theta_1-\theta_2\|_{\texttt{TF}}.$$

Proof.

$$\begin{aligned}
|\text{Regret}_{g_{\theta_1}}((\ell_t)_{t=1}^T, T) - \text{Regret}_{g_{\theta_2}}((\ell_t)_{t=1}^T, T)| &= \left|\sum_{t=1}^T\langle\ell_t, (g_{\theta_1}(Z_{t-1}) - g_{\theta_2}(Z_{t-1}))\rangle\right| \\
&= \left|\sum_{t=1}^T\langle\ell_t, \texttt{Operator}(A\cdot\texttt{LLM}_{\theta_1}(Z_{t-1})_{-1}) - \texttt{Operator}(A\cdot\texttt{LLM}_{\theta_2}(Z_{t-1})_{-1})\rangle\right| \\
&\le B\sqrt{d}\sum_{t=1}^T\|A\cdot\texttt{LLM}_{\theta_1}(Z_{t-1})_{-1} - A\cdot\texttt{LLM}_{\theta_2}(Z_{t-1})_{-1}\|_2 \\
&\le B\sqrt{d}\|A\|_{\text{op}}TC_{\texttt{LLM}}\|\theta_1-\theta_2\|_{\texttt{LLM}} = C_{\texttt{Reg}}\|\theta_1-\theta_2\|_{\texttt{LLM}},
\end{aligned}$$

where $Z_t := (\ell_1, \ldots, \ell_t, c)$ for all $t \in [T]$. Here, the penultimate inequality holds since

- If $\texttt{Operator}$ is a projection to the convex set, then $\|\texttt{Operator}(x)-\texttt{Operator}(y)\|_2 \leq \|x-y\|_2$.
- If $\texttt{Operator}$ is $\texttt{Softmax}$, then $\|\texttt{Softmax}(x)-\texttt{Softmax}(y)\|_2 \leq \|x-y\|_2$ (Gao & Pavel, 2017, Corollary 3)

and $\|\ell_i\|_2 \leq B\sqrt{d}$ as $\|\ell_i\|_\infty \leq B$. Note that the only condition that we require for $\texttt{Operator}$ is nonexpansiveness. □

Now, we will prove the Lipschitzness of

$$c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta) := \frac{\sum_{j\in[N]} h(\text{Regret}_{g_\theta}((\ell_t^{(j)})_{t=1}^T)) f(\text{Regret}_{g_\theta}((\ell_t^{(j)})_{t=1}^T), k)}{\sum_{j\in[N]} f(\text{Regret}_{g_\theta}((\ell_t^{(j)})_{t=1}^T), k)}. \tag{E.3}$$

Claim 5. *For $R>0$, there exists β_R such that if $\beta > \beta_R$, we have*

$$\left|\frac{\sum_{n\in[N]} x_n f(x_n,\beta)}{\sum_{n\in[N]} f(x_n,\beta)} - \frac{\sum_{n\in[N]} y_n f(y_n,\beta)}{\sum_{n\in[N]} f(y_n,\beta)}\right| \leq 2\|x-y\|_\infty$$

for every $x, y \in \mathbb{R}^n$ such that $|x_i| \leq R, |y_i| \leq R$ for all $i \in [N]$.

Proof. If $\beta = \infty$, we have

$$\begin{aligned}\lim_{\beta\to\infty} &\left(\left|\frac{\sum_{n\in[N]} x_n f(x_n,\beta)}{\sum_{n\in[N]} f(x_n,\beta)} - \frac{\sum_{n\in[N]} y_n f(y_n,\beta)}{\sum_{n\in[N]} f(y_n,\beta)}\right| \Big/ \|x-y\|_\infty\right) \\ &= \frac{|\max_{n\in[N]} x_n - \max_{n\in[N]} y_n|}{\|x-y\|_\infty} \leq 1\end{aligned}$$

holds. Moreover, we can think of an optimization problem as follows:

$$\begin{aligned}F(R,\beta) = \max &\left(\left|\frac{\sum_{n\in[N]} x_n f(x_n,\beta)}{\sum_{n\in[N]} f(x_n,\beta)} - \frac{\sum_{n\in[N]} y_n f(y_n,\beta)}{\sum_{n\in[N]} f(y_n,\beta)}\right| \Big/ \|x-y\|_\infty\right) \\ &\text{subject to} \qquad |x_i| \leq R, |y_i| \leq R \text{ for all } i \in [N]\end{aligned}$$

Then, since $\|x\|_\infty \leq R$ and $\|y\|_\infty \leq R$ is a compact set, by Berge's maximum theorem, we have that $F(R,\beta)$ is a continuous function for β. Moreover, we know that $F(R,\infty) \leq 1$, which indicates that we can find $C_\beta(R)$ such that if $\beta > \beta_R$, $F(\beta) \leq 2$. □

Note that Claim 5 does not hold if x_i or y_i is not bounded. Now, we will apply Claim 5 to Equation (E.3). We can guarantee that $\text{Regret}_{g_\theta}((\ell_t)_{t=1}^T) \leq \text{diam}(\Pi)TB$. Since we assumed the continuity of h', and the domain of h is the range of $\text{Regret}_{g_\theta}((\ell_t)_{t=1}^T)$, which is a compact interval, we can assume that h is $C_h(\text{diam}(\Pi)TB)$ Lipschitz continuous.

Lemma 11 (Lipshitzness of c)**.** *Equation (E.3) is $C_{cost} := 2C_h(diam(\Pi)TB)C_{Reg}$ Lipschitz function if $k > k_{diam(\Pi)TB}$, i.e.,*

$$|c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta_1) - c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta_2)| \leq C_{cost}\|\theta_1-\theta_2\|_{LLM}.$$

Proof.

$$\begin{aligned}&|c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta_1) - c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta_2)| \\ &\underset{(i)}{\leq} 2\|h(\text{Regret}_{g_{\theta_1}}((\ell_t^{(j)})_{t=1}^T)) - h(\text{Regret}_{g_{\theta_2}}((\ell_t^{(j)})_{t=1}^T))\|_\infty \\ &\underset{(ii)}{\leq} 2C_h(\text{diam}(\Pi)TB)\|\text{Regret}_{g_{\theta_1}}((\ell_t^{(j)})_{t=1}^T) - \text{Regret}_{g_{\theta_2}}((\ell_t^{(j)})_{t=1}^T)\|_\infty \\ &\underset{(iii)}{\leq} 2C_h(\text{diam}(\Pi)TB)C_{\text{Reg}}\|\theta_1-\theta_2\|_{\text{LLM}} = C_{\text{cost}}\|\theta_1-\theta_2\|_{\text{LLM}}.\end{aligned}$$

Here, (i) holds due to Claim 5, (ii) holds since h is $C_h(\text{diam}(\Pi)TB)$ Lipschitz continuous, and (iii) holds due to Lemma 10. □

For completeness of the paper, we provide the definition of covering set and covering number.

Definition 4 (Covering set and Covering number)**.** *For $\delta > 0$, a metric space $(X, \|\cdot\|)$, and subset $Y \subseteq X$, set $C \subset Y$ is a δ covering when $Y \subseteq \cup_{c\in C} B(c, \delta, \|\cdot\|)$ holds. δ covering number $N(\delta; Y, \|\cdot\|)$ is defined as the minimum cardinality of any covering.*

By (Wainwright, 2019, Example 5.8)), we can verify that the δ-covering number $N(\delta; B(0, r, \|\cdot\|_{\texttt{LLM}}), \|\cdot\|_{\texttt{LLM}})$ as

$$\log N(\delta; B(0, r, \|\cdot\|_{\texttt{LLM}}), \|\cdot\|_{\texttt{LLM}}) \leq d_\theta \log(1 + 2r/\delta).$$

where d_θ is the dimension of LLM's parameter space. For example, if we use the Transformer with parameter space $\Theta_{d.L,M,d',B_{\texttt{TF}}}$,

$$\log N(\delta; B(0, r, \|\cdot\|_{\texttt{LLM}}), \|\cdot\|_{\texttt{LLM}}) \leq L(3Md^2 + 2d(dd' + 3md^2)) \log(1 + 2r/\delta).$$

Therefore, there exists a set Θ_0 with $\log|\Theta_0| = d_\theta \log(1 + 2r/\delta)$ so for any $\theta \in \Theta$, there exists $\theta_0 \in \Theta_0$ with

$$|c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta) - c((\ell_t^{(j)})_{t\in[T],j\in[N]}, k, \theta_0)| \leq C_{\texttt{cost}}\delta.$$

Then, by the standard result of the statistical learning theory, if we trained with N_T samples, for every $0 < \epsilon < 1/2$, with probability at least $1 - \epsilon$, we have

$$L(\widehat{\theta}_{k,N,N_T}, k, N) - \inf_{\theta\in\Theta} L(\theta, k, N) \leq \sqrt{\frac{2(\log|\Theta_0| + \log(2/\epsilon))}{N_T}} + 2C_{\texttt{cost}}\delta$$

if we set $\delta = \Omega(\sqrt{\log(\epsilon)/N_T})$, we obtain

$$L(\widehat{\theta}_{k,N,N_T}, k, N) - \inf_{\theta\in\Theta} L(\theta, k, N) \leq \mathcal{O}\left(\sqrt{\frac{1 + \log(1/\epsilon)}{N_T}}\right)$$

with probability at least $1 - \epsilon$. □

Corollary 1. (Regret)**.** *Suppose h is a non-decreasing function and $\log f$ is a supermodular twice-continuously-differentiable function (i.e., $\frac{\partial^2 \log f}{\partial x \partial k} \geq 0$). For any $0 < \epsilon < 1/2$, with probability at least $1 - \epsilon$, we have*

$$h(\lim_{N\to\infty}\lim_{k\to\infty}\max_{\|\ell_t\|_\infty\leq B} \textit{Regret}_{\textit{LLM}_{\widehat{\theta}_{k,N,N_T}}}((\ell_t)_{t\in[T]})) \leq h(\inf_{\theta\in\Theta}\max_{\|\ell_t\|_\infty\leq B} \textit{Regret}_{\textit{LLM}_\theta}((\ell_t)_{t\in[T]})) + \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{N_T}}\right). \tag{5.4}$$

Proof. The limit of right-hand side remains as $\mathcal{O}\left(\sqrt{\frac{1+\log(1/\epsilon)}{N_T}}\right)$ since we firstly do $\lim_{k\to\inf}$ and then we use $\lim_{N\to\inf}$, as we can guarantee Theorem 5.1 when $k > k_R$. Note that k_R is implicitly related to N.

Next, we have

$$\begin{aligned}
&\lim_{N\to\infty}\lim_{k\to\infty}|\mathcal{L}(\widehat{\theta}_{K,N}, k, N) - h(\lim_{N\to\infty}\lim_{k\to\infty}\max_{\|\ell_t\|_\infty\leq B} \text{Regret}_{\text{LLM}_{\widehat{\theta}_{N,k}}}((\ell_t)_{t\in[T]})| \\
&\leq \lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta}|\mathcal{L}(\theta, k, N) - h(\lim_{N\to\infty}\lim_{k\to\infty}\max_{\|\ell_t\|_\infty\leq B} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})| = 0
\end{aligned}$$

due to Claim 3.

Finally, we have

$$\lim_{N\to\infty}\lim_{k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) = \inf_{\theta\in\Theta} h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$$

due to Claim 4. □

Remark 4 (Dynamic regret loss)**.** *Similarly, we can define the dynamic regret loss function as follows:*

$$\mathcal{L}(\theta, k, N) := \mathbb{E}\left[\frac{\sum_{j\in[N]} h(\textit{D-Regret}_{\textit{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(\textit{D-Regret}_{\textit{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}), k)}{\sum_{j\in[N]} f(\textit{D-Regret}_{\textit{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}), k)}\right]$$

Then, in disease, we can also show the same result as the regret loss case since regret loss does not utilize the property of the regret except boundedness. To be specific, Lemma 10 holds due to the reason that we can bound the difference of the regret with $\left|\sum_{t=1}^T \langle \ell_t, (g_{\theta_1}(Z_{t-1}) - g_{\theta_2}(Z_{t-1}))\rangle\right|$ *term as well as* $\inf_{\pi_i\in\Pi}\langle \ell_i, \pi_i\rangle$ *is canceled. Moreover, every component of Appendix E.2 holds for the same reason.*

E.4 Deferred Proof of Theorem 5.2

Theorem 5.2. *The configuration in Equation (5.5) and* $\Pi = B(0, R_\Pi, \|\cdot\|)$ *for some* $R_\Pi > 0$*,* (V, K, Q, v_c, k_c, q_c) *such that* $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ *and* $V = -R_\Pi \frac{T}{\sum_{t=1}^{T-1} 1/t}\Sigma^{-1}\mathbb{E}\left[\|\sum_{t=1}^T \ell_t\|\ell_1\ell_2^\intercal\right]\Sigma^{-1}$ *is a first-order stationary point of Equation (5.2) with* $N = 1$*,* $h(x) = x^2$*. Moreover, if* Σ *is a diagonal matrix, then plugging this configuration to Equation (5.5) then* $\texttt{Proj}_{\Pi,\|\cdot\|}$ *would perform FTRL with an* L_2*-regularizer for the loss vectors* $(\ell_t)_{t\in[T]}$*.*

Proof. Define $a := K^\intercal(Qc + q_c) \in \mathbb{R}^d$ and $b_{t-1} := \beta\mathbf{1}_{t-1} := k_c^\intercal(Qc + q_c)\mathbf{1}_{t-1} \in \mathbb{R}^{t-1}$. The loss function (Equation (5.2)) can be written as follows:

$$f(V, a, b, v_c) := \mathbb{E}\left(\sum_{t=1}^T \ell_t^\intercal(V\ell_{1:t-1} + v_c\mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi\|\sum_{t=1}^T \ell_t\|_2\right)^2$$

Step 1. Calculating $\frac{df}{da}$

For $x \in [d]$, we calculate the x directional derivative with the following equation:

$$\begin{aligned}
&\frac{d}{da_x}\ell_t^\intercal(V\ell_{1:t-1} + v_c\mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) \\
&= \frac{d}{da_x}\sum_{i=1}^{t-1}\ell_t^\intercal(V\ell_{1:t-1} + v_c\mathbf{1}_{t-1}^\intercal)e_i\frac{\exp(e_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))} \\
&= \frac{\sum_{i=1}^{t-1}\ell_t^\intercal(V\ell_{1:t-1} + v_c\mathbf{1}_{t-1}^\intercal)e_i\exp(e_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))\frac{de_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1})}{da_x}(\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1})))}{(\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1})))^2} \\
&\quad - \frac{\sum_{i=1}^{t-1}\ell_t^\intercal(V\ell_{1:t-1} + v_c\mathbf{1}_{t-1}^\intercal)e_i\exp(e_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))\left(\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))\frac{de_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1})}{da_x}\right)}{(\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1})))^2}.
\end{aligned}$$

Plugging $a = \mathbf{0}_d$ and $v_c = \mathbf{0}_d$, and $b_{t-1} = \beta\mathbf{1}_{t-1}$ provides

$$\begin{aligned}
&\frac{d}{da_x}\ell_t^\intercal(V\ell_{1:t-1} + v_c\mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1})\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, b=\beta\mathbf{1}_{t-1}} \\
&\quad = \frac{\sum_{i=1}^{t-1}\ell_t^\intercal V\ell_i\ell_{ix}}{(t-1)} - \frac{\sum_{i=1}^{t-1}\ell_t^\intercal V\ell_i\left(\sum_{s=1}^{t-1}\ell_{sx}\right)}{(t-1)^2},
\end{aligned}$$

Using the above calculation, now we calculate df/da_x as follows:

$$
\begin{aligned}
&\frac{df(V,a,b,v_c)}{da_x}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}} \\
&= \mathbb{E}\frac{d}{da_x}\left(\sum_{t=1}^T \ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)^2\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^T \ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}} \\
&\qquad \frac{d}{da_x}\left(\sum_{t=1}^T \ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}}\Bigg] \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^T \frac{1}{t-1}\ell_t^\top V\sum_{i=1}^{t-1}\ell_i+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)\sum_{t=1}^T\left(\frac{\sum_{i=1}^{t-1}\ell_t^\top V\ell_i\ell_{ix}}{(t-1)}-\frac{\sum_{i=1}^{t-1}\ell_t^\top V\ell_i\left(\sum_{s=1}^{t-1}\ell_{sx}\right)}{(t-1)^2}\right)\Bigg] \\
&= 0
\end{aligned}
$$

since the expectation of odd-order polynomial or even-order polynomial times $\|\cdot\|_2$ with respect to symmetric distribution ℓ_t is 0.

Step 2. Calculating $\frac{df}{dv_c}$

We will use the following equation:

$$
\begin{aligned}
&\frac{d}{dv_c}\ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1}) \\
&\quad = \frac{d}{dv_c}\sum_{i=1}^{t-1}\ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)e_i\frac{\exp(e_i^\top(\ell_{1:t-1}^\top a+b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\top(\ell_{1:t-1}^\top a+b_{t-1}))}=\ell_t.
\end{aligned}
$$

Therefore, we can calculate f's derivative over v_c:

$$
\begin{aligned}
&\frac{df(V,a,b,v_c)}{dv_c}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}} \\
&= \mathbb{E}\frac{d}{dv_c}\left(\sum_{t=1}^T \ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)^2\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^T \ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}} \\
&\qquad \frac{d}{dv_c}\left(\sum_{t=1}^T \ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,b_{t-1}=\beta\mathbf{1}_{t-1}}\Bigg] \\
&= \mathbb{E}\left[\left(\sum_{t=1}^T \frac{1}{t-1}\ell_t^\top V\sum_{i=1}^{t-1}\ell_i+R_\Pi\|\sum_{t=1}^T\ell_t\|_2\right)\sum_{t=1}^T\ell_t\right]=0
\end{aligned}
$$

since the expectation of odd-order polynomial or even-order polynomial times $\|\cdot\|_2$ with respect to symmetric distribution ℓ_t is 0.

Step 3. Calculating $\frac{df}{dV}$

We calculate the following equation, which will be used to calculate df/dV:

$$
\begin{aligned}
&\frac{d}{dV}\ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)\texttt{Softmax}(\ell_{1:t-1}^\top a+b_{t-1})\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d} \\
&\quad = \frac{d}{dV}\sum_{i=1}^{t-1}\ell_t^\top (V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\top)e_i\frac{\exp(e_i^\top(\ell_{1:t-1}^\top a+b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\top(\ell_{1:t-1}^\top a+b_{t-1}))}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d} \\
&\quad = \sum_{i=1}^{t-1}\ell_t\ell_i^\top\frac{\exp(e_i^\top(\ell_{1:t-1}^\top a+b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\top(\ell_{1:t-1}^\top a+b_{t-1}))}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d}=\frac{1}{t-1}\sum_{i=1}^{t-1}\ell_t\ell_i^\top.
\end{aligned}
$$

Therefore, if we calculate the derivative over V, then we have

$$
\begin{aligned}
&\frac{df(V,a,b,v_c)}{dV}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d} \\
&= \mathbb{E}\frac{d}{dV}\left(\sum_{t=1}^{T}\ell_t^\intercal(V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a+b_{t-1})+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)^2\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\ell_t^\intercal(V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a+b_{t-1})+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d} \\
&\qquad \frac{d}{dV}\left(\sum_{t=1}^{T}\ell_t^\intercal(V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a+b_{t-1})+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,}\Bigg] \\
&= \mathbb{E}\left[\left(\sum_{t=1}^{T}\frac{1}{t-1}\ell_t^\intercal V\sum_{i=1}^{t-1}\ell_i+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\sum_{t=1}^{T}\sum_{i=1}^{t-1}\frac{1}{t-1}\ell_t\ell_i^\intercal\right] \\
&= \mathbb{E}\left[\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\left(\frac{1}{t-1}\ell_t^\intercal V\ell_i\right)\left(\frac{1}{t-1}\ell_t\ell_i^\intercal\right)+R_\Pi T\|\sum_{t=1}^{T}\ell_t\|_2\ell_t\ell_i^\intercal\right)\right] \\
&= \mathbb{E}\left[\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\sum_{x=1}^{d}\sum_{y=1}^{d}v_{xy}\ell_{tx}\ell_{iy}\left(\frac{1}{t-1}\right)^2[\ell_{tz}\ell_{iw}]_{(z,w)}+R_\Pi T\|\sum_{t=1}^{T}\ell_t\|_2\ell_t\ell_i^\intercal\right)\right] \\
&= \sum_{t=1}^{T}\sum_{i=1}^{t-1}\sum_{x=1}^{d}\sum_{y=1}^{d}\frac{1}{(t-1)^2}[\sigma_{xz}v_{xy}\sigma_{yw}]_{(z,w)}+\mathbb{E}\left[R_\Pi T\|\sum_{t=1}^{T}\ell_t\|_2\ell_t\ell_i^\intercal\right] \\
&= \left(\sum_{t=1}^{T-1}\frac{1}{t}\right)\Sigma V\Sigma+\mathbb{E}\left[R_\Pi T\|\sum_{t=1}^{T}\ell_t\|_2\ell_t\ell_i^\intercal\right].
\end{aligned}
$$

Therefore, if $V=R_\Pi\frac{T}{\sum_{t=1}^{T-1}1/T}\Sigma^{-1}\mathbb{E}\left[\|\sum_{t=1}^{T}\ell_t\|_2\ell_t\ell_i^\intercal\right]\Sigma^{-1}$, $\frac{df}{dV}=\boldsymbol{O}_{d\times d}$. Lastly, we have

$$
\begin{aligned}
\frac{df}{dK}&=\frac{df}{da}\frac{da}{dK}=\mathbf{0}_d\frac{da}{dK}=\boldsymbol{O}_{d\times d} \\
\frac{df}{dQ}&=\frac{df}{da}\frac{da}{dQ}=\mathbf{0}_d\frac{da}{dQ}=\boldsymbol{O}_{d\times d} \\
\frac{df}{dq_c}&=\frac{df}{da}\frac{da}{dq_c}=\mathbf{0}_d\frac{da}{dq_c}=\mathbf{0}_d
\end{aligned}
$$

which means that such configurations are the first-order stationary points. □

E.5 Deferred Proof of Theorem 5.3

Theorem 5.3. *The configuration of a single-layer linear self-attention model in Equation (5.6) (V,K,Q,v_c,k_c,q_c) such that $K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d$ and $\Pi=B(0,R_\Pi,\|\cdot\|)$ for some $R_\Pi>0$, $V=-2R_\Pi\Sigma^{-1}\mathbb{E}\left(\|\sum_{t=1}^{T}\ell_t\|\ell_1\ell_2^\intercal\right)\Sigma^{-1}$ is **a global optimal solution** of Equation (5.2) with $N=1$, $h(x)=x^2$. Moreover, every global optimal configuration of Equation (5.2) within the parameterization class of Equation (5.6) has the same output function g. If Σ is a diagonal matrix, plugging any global optimal configuration to Equation (5.6) then $\texttt{Proj}_{\Pi,\|\cdot\|}$ would perform FTRL with an-L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.*

Proof. The output of the single-layer self-attention structure is as follows:

$$
\begin{aligned}
&g(Z_t;V,K,Q,v_c,k_c,q_c) \\
&\quad=\sum_{i=1}^{t}\left(V\ell_i\ell_i^\intercal(K^\intercal(Qc+q_c))+(Vk_c^\intercal(Qc+q_c)+v_c(Qc+q_c)^\intercal K)\,\ell_i+v_ck_c^\intercal(Qc+q_c)\right)
\end{aligned}
\tag{E.4}
$$

which can be expressed with a larger class

$$g(Z_t, \mathbb{A}, \beta, \mathbb{C}, \delta) := \sum_{i=1}^{t} (\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta) \tag{E.5}$$

where $\mathbb{A} \in \mathbb{R}^{d\times d}$, $\beta, \mathbb{C}, \delta \in \mathbb{R}^d$. Then, if a minimizer of

$$f(\mathbb{A}, \beta, \mathbb{C}, \delta) := \mathbb{E}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta)\rangle - \inf_{\pi\in\Pi}\left\langle \sum_{t=1}^{T}\ell_t, \pi\right\rangle\right)^2$$

can be expressed with $\mathbb{A} = V, \beta = K^\intercal(Qc + q_c), \mathbb{C} = Vk_c^\intercal(Qc + q_c) + v_c(Qc + q_c)^\intercal K, \beta = v_c k_c^\intercal(Qc + q_c)$, we can conclude that corresponding V, Q, K, v_c, q_c, k_c are also a minimizer of

$$\mathbb{E}\left(\sum_{t=1}^{T}\langle \ell_t, g(Z_{t-1})\rangle - \inf_{\pi\in\Pi}\left\langle \sum_{t=1}^{T}\ell_i, \pi\right\rangle\right)^2$$

since corresponding V, Q, K, v_c, q_c, k_c constitute a minimizer among a larger class. Now, since $\Pi = B(\mathbf{0}_d, B, \|\cdot\|)$, we can rewrite f as

$$f(\mathbb{A}, \beta, \mathbb{C}, \delta) = \mathbb{E}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta)\rangle + R_\Pi\left\|\sum_{t=1}^{T}\ell_i\right\|_2\right)^2. \tag{E.6}$$

Step 1. Finding condition for $\frac{df}{d\delta} = 0$

Due to the Leibniz rule, if we calculate the derivative of Equation (E.6) over δ, we have

$$\begin{aligned}
\frac{df(\mathbb{A}, \beta, \mathbb{C}, \delta)}{d\delta} &= \frac{d}{d\beta}\mathbb{E}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta)\rangle + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)^2 \\
&= \mathbb{E}\frac{d}{d\delta}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta)\rangle + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)^2 \\
&= \mathbb{E}\sum_{t=1}^{T}(t-1)\ell_t\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta) + R_\Pi\|\sum_{t=1}^{T}\ell_t\|\right).
\end{aligned} \tag{E.7}$$

Since the expectation of odd-order polynomial or even-order polynomial times $\|\cdot\|_2$ with respect to symmetric distribution ℓ_t is 0, we have

$$\mathbb{E}\sum_{t=1}^{T}(t-1)\ell_t R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2 = 0, \qquad \mathbb{E}\sum_{t=1}^{T}(t-1)\ell_t\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i = 0.$$

Now, we calculate

$$\begin{aligned}
\mathbb{E}\sum_{t=1}^{T}(t-1)\ell_t\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{A}\ell_i\ell_i^\intercal\beta &= \mathbb{E}\sum_{t_1=1}^{T}\sum_{t=1}^{T}\sum_{i=1}^{t-1}(t_1-1)\ell_{t_1}\ell_t^\intercal\mathbb{A}\ell_i\ell_i^\intercal\beta \\
&\underset{(i)}{=} \mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{t-1}(t-1)\ell_t\ell_t^\intercal\mathbb{A}\ell_i\ell_i^\intercal\beta = \mathbb{E}\sum_{t=1}^{T}(t-1)^2\ell_t\ell_t^\intercal\mathbb{A}\Sigma\beta = \frac{1}{6}n(2n^2-3n+1)\Sigma\mathbb{A}\Sigma\beta
\end{aligned}$$

since (i) holds since if $t_1 \neq t$, due to the independence of ℓ_t, ℓ_{t_1}, we can use $\mathbb{E}\ell_t = 0$. Lastly,

$$\mathbb{E}\sum_{t=1}^{T}(t-1)\ell_t\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\delta = \mathbb{E}\sum_{t_1=1}^{T}\sum_{t=1}^{T}(t_1-1)(t-1)\ell_{t_1}\ell_t^\intercal\delta = \frac{1}{6}n(2n^2-3n+1)\Sigma\delta.$$

Plugging above equations to Equation (E.7), we have

$$\frac{df(\mathbb{A}, \beta, \mathbb{C}, \delta)}{d\delta} = \frac{1}{6}n(2n^2-3n+1)(\Sigma\mathbb{A}\Sigma\beta + \Sigma\delta).$$

Due to the optimality condition, we have

$$\mathbb{A}\Sigma\beta + \delta = 0. \tag{E.8}$$

Step 2. Plugging the optimality condition from $\frac{df}{d\delta}$ to Equation (E.6)

Plugging Equation (E.8) to Equation (E.6), f can be written as

$$\begin{aligned}
f(\mathbb{A},\beta,\mathbb{C},-\mathbb{A}\Sigma\beta) &= \mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\left(\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta+\mathbb{C}\ell_i\right)+R_\Pi\left\|\sum_{t=1}^{T}\ell_t\right\|_2\right)^2 \\
&= \underbrace{\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right)^2}_{(i)} + \mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\right)^2 + \mathbb{E}\left(R_\Pi\left\|\sum_{t=1}^{T}\ell_t\right\|_2\right)^2 \\
&\quad + \underbrace{2\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right)\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\right)}_{(ii)} \\
&\quad + \underbrace{2\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right)\left(R_\Pi\left\|\sum_{t=1}^{T}\ell_t\right\|_2\right)}_{(iii)} \\
&\quad + 2\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\right)\left(R_\Pi\left\|\sum_{t=1}^{T}\ell_t\right\|_2\right).
\end{aligned}$$

For the part (i), we have

$$\begin{aligned}
\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right)^2 &= \mathbb{E}\left[\sum_{t_1=1}^{T}\sum_{i_1=1}^{t_1-1}\sum_{t=1}^{T}\sum_{i=1}^{t-1}\beta^\intercal(\ell_{i_1}\ell_{i_1}^\intercal-\Sigma)\mathbb{A}^\intercal\ell_{t_1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] \\
&\underset{(1)}{=} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i_1=1}^{t-1}\sum_{i=1}^{t-1}\beta^\intercal(\ell_{i_1}\ell_{i_1}^\intercal-\Sigma)\mathbb{A}^\intercal\ell_t\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] \\
&\underset{(2)}{=} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{t-1}\beta^\intercal(\ell_i\ell_i^\intercal-\Sigma)\mathbb{A}^\intercal\ell_i\ell_i^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] \\
&= \mathbb{E}\left[\frac{(T-1)T}{2}\beta^\intercal(\ell_i\ell_i^\intercal-\Sigma)\mathbb{A}^\intercal\Sigma\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] = \mathbb{E}\left[\frac{(T-1)T}{2}\|\sqrt{\Sigma}\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\|^2\right]
\end{aligned} \tag{E.9}$$

Here, (1) holds because if $t_1 \neq t$, we know that $\mathbb{E}\ell_{t_1} = \mathbb{E}\ell_t = 0$ and they are independent, and (2) holds because if $i_1 \neq i$, We can calculate $\mathbb{E}(\ell_{i_1}\ell_{i_1}^\intercal - \Sigma) = \boldsymbol{O}_{d\times d}$ In addition, we can easily check that (ii) and (iii) are 0 as they are a polynomial of odd degrees and we have $Z \overset{d}{=} -Z$. Note that equation E.9 is minimized when $\mathbb{P}(\sqrt{\Sigma}\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta = \mathbf{0}_d) = 1$. If $A \neq \boldsymbol{O}_{d\times d}$, assume that singular value decomposition of $A = U\Lambda V$ such that Λ is a diagonal matrix that the first diagonal element is non-zero, and U, V are orthogonal matrices. Then, we want to find β that $\sqrt{\Sigma}U\Lambda V\ell_i\ell_i^\intercal\beta = \mathbf{0}_d$ for any ℓ_i such that $\mathbb{P}(\ell_i) \neq 0$. Since Σ and U are invertible, we only need to consider $\Lambda V\ell_i\ell_i^\intercal\beta = \mathbf{0}_d$. Since Λ's first diagonal component is non-zero, we will consider equation $\boldsymbol{e}_1\Lambda V\ell_i\ell_i^\intercal\beta = 0$ where $\boldsymbol{e}_1 \in \mathbb{R}^d$ is $(1,0,\ldots,0)^\intercal$. This is equivalent to $V_1\ell_i\ell_i^\intercal\beta = 0$ where V_1 is the first row of V which is non-zero vector. Since ℓ_i's support is $\mathbb{R}^d$, $V_1\ell_i$ can be any value, so $\ell_i^\intercal\beta = 0$ for all $\ell_i \in \text{supp}(Z)$, which indicates $\beta = \mathbf{0}_d$.

Therefore, if we want to minimize Equation (E.9), $A = \boldsymbol{O}_{d\times d}$ or $\beta = \mathbf{0}_d$ holds. In both cases, Equation (E.5) can be re-written as

$$g(Z_t;\mathbb{A},\beta,\mathbb{C},\delta) := \sum_{i=1}^{t}\mathbb{C}\ell_i,$$

and this is covered by the original parametrization (Equation (E.4)) with $K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d$.

Step 3. Calculating $\frac{df}{d\mathbb{C}}$

Now, we do optimization over $\mathbb{C}$. So we have the following minimization problem with respect to $\mathbb{C}$:

$$\begin{aligned}
f(\mathbb{C}) &:= \mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i + R_\Pi\|\sum_{t=1}^{T}\ell_t\|\right)^2 \\
&= \underbrace{\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\right)^2}_{(i)} + 2\mathbb{E}\left(\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\right)R_\Pi\|\sum_{t=1}^{T}\ell_t\|\right) + \mathbb{E}\left(R_\Pi\|\sum_{t=1}^{T}\ell_t\|\right)^2 \\
&= \frac{T(T-1)}{2}\operatorname{Tr}\left(\mathbb{C}^\intercal\Sigma\mathbb{C}\Sigma\right) + 2\mathbb{E}\left(B\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\|\sum_{j=1}^{T}\ell_j\|\right) + \mathbb{E}\left(R_\Pi\|\sum_{t=1}^{T}\ell_t\|\right)^2.
\end{aligned}$$

Here, (i) can be calculated as follows:

$$\begin{aligned}
\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{C}\ell_i\right)^2 &= \mathbb{E}\left(\sum_{t_1=1}^{T}\sum_{i_1=1}^{t_1-1}\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_{i_1}^\intercal\mathbb{C}^\intercal\ell_{t_1}\ell_i^\intercal\mathbb{C}\ell_i\right) \\
&\underset{(1)}{=} \mathbb{E}\left(\sum_{t=1}^{T}\sum_{i_1=1}^{i-1}\sum_{i=1}^{t-1}\ell_{i_1}^\intercal\mathbb{C}^\intercal\ell_i\ell_i^\intercal\mathbb{C}\ell_i\right) = \mathbb{E}\left(\sum_{t=1}^{T}\sum_{i_1=1}^{i-1}\sum_{i=1}^{t-1}\ell_{i_1}^\intercal\mathbb{C}^\intercal\Sigma\mathbb{C}\ell_i\right) \\
&\underset{(2)}{=} \mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_k^\intercal\mathbb{C}^\intercal\Sigma\mathbb{C}\ell_i\right) \underset{(3)}{=} \mathbb{E}\operatorname{Tr}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\mathbb{C}^\intercal\Sigma\mathbb{C}\ell_i\ell_k^\intercal\right) = \frac{T(T-1)}{2}\operatorname{Tr}\left(\mathbb{C}^\intercal\Sigma\mathbb{C}\Sigma\right),
\end{aligned}$$

since (1) holds because if $t_1\neq t$, we already know that $\mathbb{E}\ell_t=\mathbb{E}\ell_{t_1}=0$, (2) holds due to a similar reason, and (3) comes from $\operatorname{Tr}(AB)=\operatorname{Tr}(BA)$.

We calculate $\frac{df(\mathbb{C})}{d\mathbb{C}}$:

$$\frac{df(\mathbb{C})}{d\mathbb{C}} = T(T-1)\Sigma\mathbb{C}\Sigma + 2R_\Pi\mathbb{E}\left(\|\sum_{j=1}^{T}\ell_j\|\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t\ell_i^\intercal\right),$$

So the optimal $\mathbb{C}=-\frac{2R_\Pi}{T(T-1)}\Sigma^{-1}\mathbb{E}\left(\|\sum_{j=1}^{T}\ell_j\|\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t\ell_i^\intercal\right)\Sigma^{-1}$.

Now, we see the special case of $\Sigma=I$, then we have $\mathbb{C}=-R_\Pi\mathbb{E}\left(\|\sum_{j=1}^{T}\ell_j\|\ell_t\ell_i^\intercal\right)$. If we calculate (a,b)-coordinate of $\mathbb{C}$, we need to calculate

$$\mathbb{E}_l\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2}\ell_{ia}\ell_{kb}\right]$$

If $a\neq b$, then since Z is symmetric, we can think event about $(\ell_{ta})_{t=1}^T$ become $(-\ell_{ta})_{t=1}^T$, so it becomes zero. Therefore, we only need to consider $a=b$ case, which is $\mathbb{E}_l\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2}\ell_{ia}\ell_{ka}\right]$, and it will be the same value among $a\in[d]$ if ℓ_i's coordinate is also independent.

Now, we calculate the scale of $\mathbb{E}_l\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2}\ell_{id}\ell_{kd}\right]$. Firstly, we have $\sum_{s=1}^{T}\ell_{so}/\sqrt{T}\xrightarrow{d}\mathcal{N}(0,1)$ by central limit theorem, and by Slutsky theorem, we have $(\sum_{s=1}^{T}\ell_{so}/\sqrt{T})^2\xrightarrow{d}\chi^2(1)$, so we have

$$\frac{\sum_{o=1}^{d-1}(\sum_{s=1}^{T}\ell_{so})^2}{T\sqrt{d}} - \sqrt{d}\xrightarrow{d}\mathcal{N}(1,2).$$

If we define $Z = \frac{\sum_{o=1}^{d-1}(\sum_{s=1}^{T}\ell_{so})^2}{T\sqrt{d}}$ and $W = \sum_{s\neq i,k}\ell_{sd}/\sqrt{T} \xrightarrow{d} \mathcal{N}(0,1)$, we have

$$\mathbb{E}_l\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2\ell_{id}\ell_{kd}}\right] = \mathbb{E}_{Z,W,\ell_{id},\ell_{kd}}\left[\sqrt{T\sqrt{d}Z + (\sqrt{T}W + \ell_{id} + \ell_{kd})^2}\ell_{id}\ell_{kd}\right]$$

$$= \mathbb{E}_{Z,W,\ell_{id}>\ell_{kd}\geq 0}\left[\sqrt{T\sqrt{d}Z + (\sqrt{T}W + \ell_{id} + \ell_{kd})^2}\ell_{id}\ell_{kd} - \sqrt{T\sqrt{d}Z + (\sqrt{T}W + \ell_{id} - \ell_{kd})^2}\ell_{id}\ell_{kd}\right]$$

$$= \mathbb{E}_{Z,W,\ell_{id}>\ell_{kd}\geq 0}\left[\frac{4(\sqrt{T}W + \ell_{id})\ell_{kd}}{\sqrt{T\sqrt{d}Z + (\sqrt{T}W + \ell_{id} + \ell_{kd})^2} + \sqrt{T\sqrt{d}Z + (\sqrt{T}W + \ell_{id} - \ell_{kd})^2}}\ell_{id}\ell_{kd}\right]$$

Assuming that $T, d \to \infty$, we can estimate this value with

$$\mathbb{E}_{Z,W,\ell_{id}>\ell_{kd}\geq 0}\left[\frac{4(\sqrt{T}W)\ell_{kd}^2\ell_{id} + \text{const}}{2\sqrt{Td}}\right] = \Theta(1/\sqrt{Td}).$$

Therefore, the output of the single-layer self-attention provides us with online gradient descent with step-size $\Theta(R_\Pi/\sqrt{Td})$. In the online gradient descent literature, we usually set the gradient step size as $\Theta(R_\Pi/\sqrt{Td})$ (Hazan, 2016, Theorem 3.1), so it is consistent with existing online learning literature too. □

Remark 5. *The studies by (Ahn et al., 2023; Zhang et al., 2023b; Mahankali et al., 2023) demonstrate that if $Z_t = ((x_1, y_1), \ldots, (x_t, y_t), (x_{t+1}, 0))$ and the 'instruction tuning' loss (i.e., $\mathbb{E}[\|\widehat{y}_{t+1} - y_{t+1}\|^2]$) is being minimized with a single-layer linear self-attention model, then a global optimizer among single-layer linear self-attention models yields the output $\widehat{y}_{n+1} = \eta\sum_{i=1}^{n} y_i x_i^\intercal x_{n+1}$. This output can be interpreted as a gradient descent algorithm, indicating that a single-layer self-attention model **implicitly** performs gradient descent. However, in the online learning setting where there are no y labels, implicit gradient descent is hard to define. Compared to the previous studies, our global optimizer among single-layer linear self-attention models is an explicit online gradient descent algorithm for online learning. Additionally, we employ a distinct loss function specifically designed for no-regret learning.*

E.6 Empirical Validation of Theorem 5.2 and Theorem 5.3

We will provide empirical validation of Theorem 5.2 and Theorem 5.3. We provide the training details and the results.

E.6.1 Empirical Validation of Theorem 5.2

Our model architecture is defined as follows: the number of layers T is set to 30 and the dimensionality d to 32, with the loss vector l_i's distribution Z following a standard normal distribution $\mathcal{N}(0,1)$. During training, we conducted 40,000 epochs with a batch size of 512. We employed the Adam optimizer, setting the learning rate to 0.001. We initialized the value, query, and key vectors (v_c, q_c, k_c) as zero vectors.

Our empirical analysis aims to demonstrate that the optimized model inherently emulates online gradient descent. To illustrate this, we will focus on two key convergence properties: $K^\intercal Q$ approaching the zero matrix $\boldsymbol{O}_{d\times d}$ and V converging to $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, where a and b are constants in $\mathbb{R}$. The conditions $K^\intercal Q = \boldsymbol{O}_{d\times d}$ and $V = a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$ imply that the function $g(Z_t; V, Q, K) = \sum_{i=1}^{t}(b-a)l_i$, effectively emulating the process of an online gradient descent method. We repeated 10 times. For verifying $K^\intercal Q = \boldsymbol{O}_{d\times d}$, we will measure Frobenius norm $(\|\cdot\|_{2,2})$ of $K^\intercal Q$. Also for measuring the closeness of V and $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, we will measure $\min_{a,b\in\mathbb{R}}\|V - (a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d})\|_{2,2}/b$. The results are demonstrated in the first plot of Figure 11.

E.6.2 Empirical Validation of Theorem 5.3

We will focus on two key convergence properties: $K^\intercal(Q\mathbf{1} + q_c)$ approaching the zero vector $\mathbf{0}_d$ and V converging to $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, where a and b are constants in $\mathbb{R}$. The conditions $K^\intercal(Q\mathbf{1}+q_c) = \mathbf{0}_d$ and $V = a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$ imply that the function $g(Z_t; V, Q, K) = \sum_{i=1}^{t}(b-a)l_i$, effectively

emulating the process of an online gradient descent method. We repeated 10 times. For verifying $K^\intercal(Q\mathbf{1} + q_c) = \mathbf{0}_d$, we will measure 2-norm of $K^\intercal(Q\mathbf{1} + q_c)$. Also for measuring the closeness of V and $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, we will measure $\min_{a,b\in\mathbb{R}} \|V - (a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d})\|_{2,2}/b$. The results are demonstrated in the second plot of Figure 11.

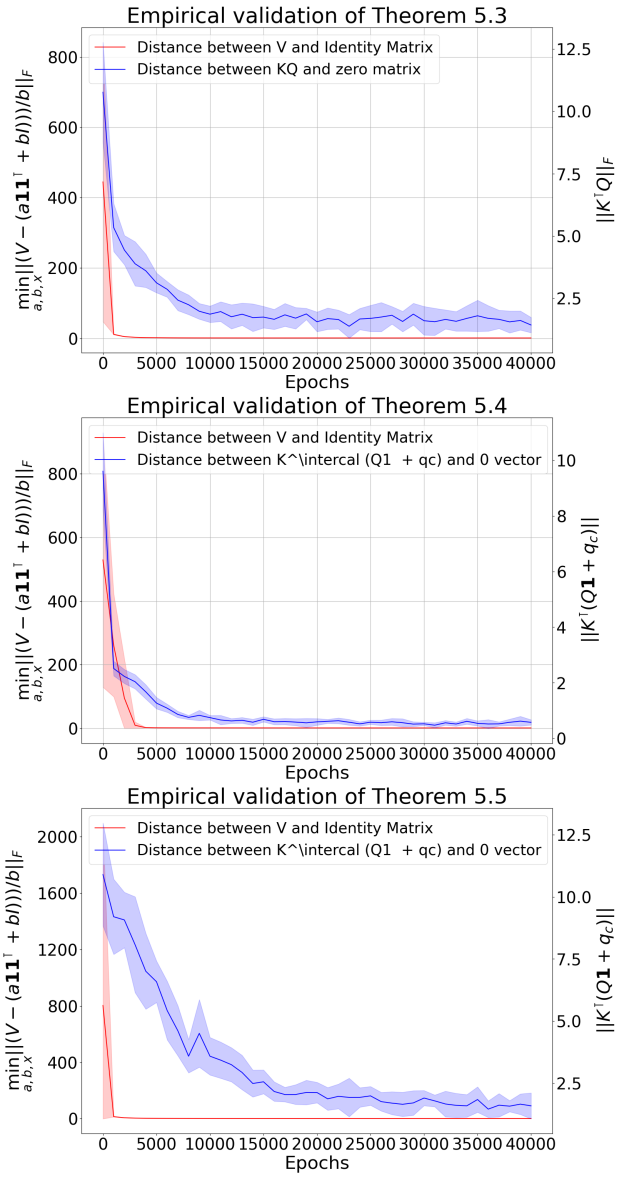


Figure 11: Empirical validation of Theorem 5.2 (top), Theorem 5.3 (middle), and Conjecture 1 (bottom). The observed convergence in Theorem 5.2 and Conjecture 1's result suggests that configuration in Theorem 5.2 and Conjecture 1 are not only the local optimal point, but it has potential for the global optimizer.

E.7 Discussions and Validations on the Production of FTRL with Entropy Regularization

Now, we will consider projecting a single-layer linear self-attention model into a constrained domain such as a simplex; which includes the setting for the expert problem setting in online learning. To this end, we consider the following parameterization by adding an additional *non-linear* structure

for the single-layer linear self-attention:

$$g(Z_t; V, K, Q, v_c, k_c, q_c) \tag{E.10}$$
$$= \texttt{Operator}\left(\sum_{i=1}^{t}(V\ell_i + v_c)((K\ell_i + k_c))^\intercal \cdot (Qc + q_c))\right),$$

where the $\texttt{Operator}$ denotes projection to the convex set.

Conjecture 1. *Assume $\Sigma = I$. Then, the configuration that $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ and $V = \widetilde{\Omega}\left(-\frac{1}{\sqrt{nd}}\right) I_{d\times d}$ is a first-order stationary point of Equation (5.2) with $N = 1$ and $h(x) = x^2$ when LLM$_\theta$ is parameterized with Equation (E.10), $\texttt{Operator} = \texttt{Softmax}$, and $\Pi = \Delta(\mathcal{A})$. This configuration performs FTRL with an entropy regularizer which is a no-regret algorithm.*

We provide a possible idea for proving the conjecture, together with its numerical validation. Also, we have observed in Figure 11 that Theorem 5.2 and Conjecture 1 might be also a global optimal point, as training results provide the configuration that Theorem 5.2 and Conjecture 1 have suggested.

We will investigate the case $\Pi = B(0, B, \|\cdot\|_2)$ and $\texttt{Operator}(p) = \texttt{Proj}_{B,\|\cdot\|_2}(p)$. also, we will consider the case $\Sigma = I$. To be specific, we will consider

$$f(V, a, \beta, v_c) = \mathbb{E}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)} - \min_s\sum_{t=1}^{T}\ell_{ts}\right)^2$$

and will try to prove that $a = \mathbf{0}_d, v_c = v\mathbf{1}_d, V = kI$ is a first order stationary point.

Step 1. Calculating $\frac{df}{dv_c}$

We use the following formula; for $x \in [d]$, we have

$$\frac{d}{dv_{cx}}\exp\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$$
$$= \exp\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\frac{d}{dv_{cx}}\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$$
$$= \exp\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\sum_{i=1}^{t}(a^\intercal\ell_i e_x + \beta)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$$
$$= t\beta\exp(v\beta)\exp(\beta k\sum_{i=1}^{t}\ell_{iy}),$$

so we have

$$\frac{d}{dv_{cx}}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)} - \min_s\sum_{t=1}^{T}\ell_{ts}\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$$
$$= \beta\exp(v\beta)$$
$$\sum_{t=1}^{T}t\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right) - \sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}\beta V\ell_j\right)\right)^2}$$
$$= 0.$$

Therefore,

$$\begin{aligned}
&\frac{df(V,a,\beta,v_c)}{dv_{cx}}\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right) \\
&\quad \frac{d}{dv_{cx}}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)\Bigg]\Bigg|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI} \\
&= 0.
\end{aligned}$$

Step 2. Calculating $\frac{df}{dV}$

The following formula will be used for calculating $\frac{df}{dV}$; for $r,c\in[d]$, we have

$$\begin{aligned}
&\frac{d}{dV_{rc}}\exp\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a+(\beta V+v_ca^\intercal)\ell_i+v_c\beta)\right)\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI} \\
&\quad= \exp\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a+(\beta V+v_ca^\intercal)\ell_i+v_c\beta)\right)\frac{d}{dV_{rc}}\left(e_y^\intercal\sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a+(\beta V+v_ca^\intercal)\ell_i+v_c\beta)\right)\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=k} \\
&\quad= \exp\left(\sum_{i=1}^{t}k\beta\ell_{iy}+v\beta\right)\sum_{i=1}^{t}\beta\mathbf{1}(y=r)\ell_{ic}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\frac{df(V,a,\beta,v_c)}{dV_{rc}}\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right) \\
&\quad \frac{d}{dV_{rc}}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)\Bigg]\Bigg|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}+v\beta\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta V\ell_{jy}+v\beta\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right) \\
&\quad \Bigg(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{j=1}^{t-1}\beta\mathbf{1}(s=r)\ell_{jc}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}+v\beta\right)\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)\right)^2} \\
&\qquad -\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}+v\beta\right)\sum_{y=1}^{d}\left(\sum_{j=1}^{t-1}\beta\mathbf{1}(y=r)\ell_{jc}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)\right)^2}\Bigg)\Bigg] \\
&= \beta\mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta V\ell_{jy}\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right) \\
&\qquad \underbrace{\left(\frac{\sum_{t=1}^{T}\sum_{j=1}^{t-1}\sum_{y=1}^{d}\ell_{tr}\ell_{jc}\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jr}\right)\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jy}\right)\right)^2}\right.}_{(i)}
\end{aligned}$$

$$\underbrace{- \frac{\sum_{t=1}^{T}\sum_{j=1}^{t-1}\sum_{y=1}^{d} \ell_{ty}\ell_{jc} \exp\left(\beta k \sum_{j=1}^{t-1} \ell_{jr}\right) \exp\left(\beta k \sum_{j=1}^{t-1} \ell_{jy}\right)}{\left(\sum_{y=1}^{d} \exp\left(\beta k \sum_{j=1}^{t-1} \ell_{jy}\right)\right)^2}}_{(ii)}\Bigg)\Bigg].$$

We can observe the followings: 1) if $r_1 \neq c_1$ and $r_2 \neq c_2$, $\frac{df}{dV_{r_1c_1}} = \frac{df}{dV_{r_2c_2}}$ holds, and 2) $\frac{df}{dV_{r_1r_1}} = \frac{df}{dV_{r_2r_2}}$.

Step 3. Calculating $\frac{df}{d\beta}$

The following formula will be used for calculating $\frac{df}{d\beta}$;

$$\begin{aligned}
&\frac{d}{d\beta} \exp\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&\quad = \exp\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right) \frac{d}{d\beta}\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&\quad = tv\beta \exp\left(\sum_{i=1}^{t} k\beta\ell_{iy} + v\beta\right).
\end{aligned}$$

so we have

$$\begin{aligned}
&\frac{d}{d\beta}\left(\sum_{t=1}^{T}\sum_{s=1}^{d} \ell_{ts} \frac{\exp\left(e_s^\intercal \sum_{j=1}^{t-1} (V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)}{\sum_{y=1}^{d} \exp\left(e_y^\intercal \sum_{j=1}^{t-1} (V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)} - \min_s \sum_{t=1}^{T} \ell_{ts}\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&= v\beta \exp(v\beta) \\
&\quad \sum_{t=1}^{T} t \sum_{s=1}^{d} \ell_{ts} \frac{\sum_{y=1}^{d} \exp\left(\sum_{j=1}^{t-1} \beta k \ell_{jy}\right) \exp\left(\sum_{j=1}^{t-1} \beta k \ell_{js}\right) - \sum_{y=1}^{d} \exp\left(\sum_{j=1}^{t-1} \beta k \ell_{js}\right) \exp\left(\sum_{j=1}^{t-1} \beta k \ell_{jy}\right)}{\left(\sum_{y=1}^{d} \exp\left(e_y^\intercal \sum_{j=1}^{t-1} \beta V \ell_j\right)\right)^2} \\
&= 0.
\end{aligned}$$

Step 4. Calculating $\frac{df}{da}$

Note that

$$\begin{aligned}
&\frac{d}{da_x} \exp\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&\quad = \exp\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right) \frac{d}{da_x}\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&\quad = \exp\left(e_y^\intercal \sum_{i=1}^{t} (V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right) \sum_{i=1}^{t} \left(e_y^\intercal V\ell_i\ell_i^\intercal e_x + e_y^\intercal v_c\ell_i^\intercal e_x\right)\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&\quad = \exp\left(\sum_{i=1}^{t} \beta k\ell_{iy} + v\beta\right) \sum_{i=1}^{t} (k\ell_{iy}\ell_{ix} + v\ell_{ix}).
\end{aligned}$$

Therefore,

$$\frac{df(V,a,\beta,v_c)}{da_x}\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}$$

$$= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)$$

$$\frac{d}{da_x}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)\Bigg]\Bigg|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}$$

$$= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)$$

$$\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{j=1}^{t-1}(k\ell_{js}\ell_{jx}+v\ell_{jx})\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right.$$

$$\left.-\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\left(\sum_{j=1}^{t-1}(k\ell_{jy}\ell_{jx}+v\ell_{jx})\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right)\Bigg]$$

$$= \mathbb{E}\Bigg[k\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)$$

$$\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{j=1}^{t-1}\ell_{js}\ell_{jx}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right.$$

$$\left.-\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\left(\sum_{j=1}^{t-1}\ell_{jy}\ell_{jx}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right)\Bigg]$$

Note that the value does not depend on x, which means that $\frac{df}{da} = c\mathbf{1}_d$ for some constant c.

E.7.1 Numerical Analysis of Step 2 and Step 4.

In steps 2 and 4, we were not able to show that a k whose value becomes zero exists, so we will provide empirical evidence. First, we attach here the estimated $\frac{df}{dV_{rc}}$ ($r \neq c$), $\frac{df}{dV_{rr}}$, $\frac{df}{da_x}$) and $\frac{df}{da_x}$ graph with respect to k value when $\ell_{ts} \sim \text{Unif}([0,1])$ for all $t \in [T], s \in [d]$. While the graph of $\frac{df}{dV}$ is not stable, we can see that k for $\frac{df}{dV_{rc}} = 0$, $\frac{df}{dV_{rr}} = 0$ and $\frac{df}{da_x} = 0$ is very similar in Figure 12. We used Monte Carlo estimation for 1,000,000 times.

E.7.2 Empirical Validation

Our model architecture is defined as follows: the number of layers T is set to 30 and the dimensionality d to 32, with the loss vector l_i's distribution Z following a standard normal distribution $\mathcal{N}(0,1)$. During training, we conducted 40,000 epochs with a batch size of 512. We employed the Adam optimizer, setting the learning rate to 0.001. We focus on two key convergence properties: $K^\intercal(Q\mathbf{1}+q_c)$ approaching the zero vector $\mathbf{0}_d$ and V converging to $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, where a and b are constants in $\mathbb{R}$. The conditions $K^\intercal(Q\mathbf{1}+q_c) = \mathbf{0}_d$ and $V = a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$ imply that the function $g(Z_t;V,Q,K) = \sum_{i=1}^{t}(b-a)l_i$, effectively emulating the process of an online gradient descent method. We repeated 10 times. For verifying $K^\intercal(Q\mathbf{1}+q_c) = \mathbf{0}_d$, we will measure 2-norm of $K^\intercal(Q\mathbf{1}+q_c)$. Also for measuring the closeness of V and $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, we will measure $\min_{a,b\in\mathbb{R}}\|V-(a\mathbf{1}_d\mathbf{1}_d^\intercal+bI_{d\times d})\|_{2,2}/b$. The results are demonstrated in the third plot of Figure 11.

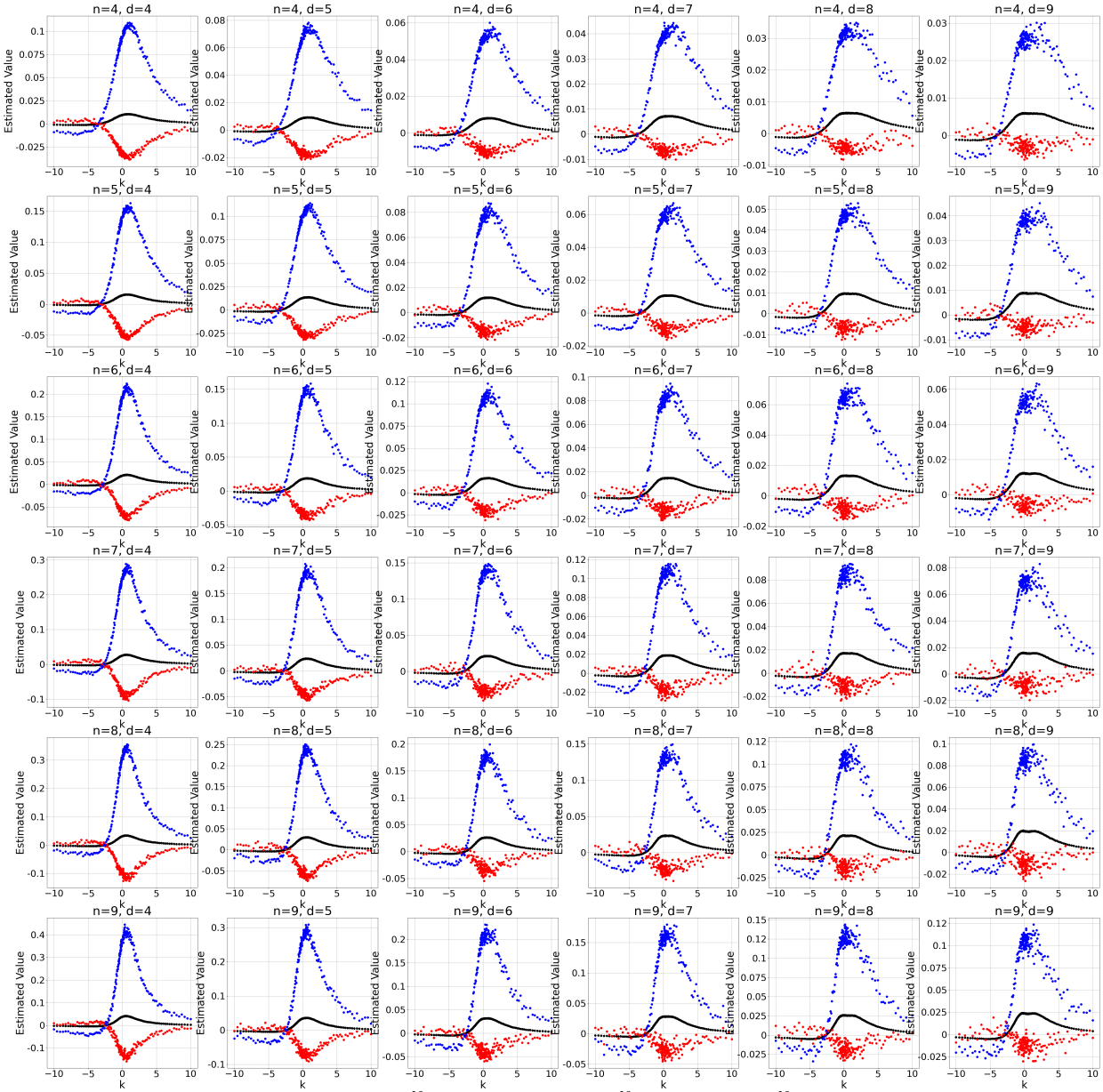


Figure 12: Calculation of $20\frac{df}{dV_{rc}}$ $(r \neq c)$(red), $20\frac{df}{dV_{rr}}$ (blue), and $\frac{df}{da_x}$ (black). We experimented with $n \in [4, 9]$ and $d \in [4, 9]$. The figure might indicate that βk that makes the derivative zero would coincide.

E.8 Training Details on Section 5.4

We provide the training details on Section 5.4. For the multi-layer transformer training, we used 4 layers, 1 head Transformer. For both single-layer and multi-layer, we employed the Adam optimizer, setting the learning rate to 0.001. During training, we conducted 2,000 epochs with a batch size 512. Moreover, when we trian for the loss sequences with predictable trend, we used 4 layers, 1 head Transformer. For both single-layer and multi-layer, we employed the Adam optimizer, setting the learning rate to 0.001. During training, we conducted 9,000 epochs with a batch size 512.

E.9 Detailed Experimental Settings in Section 5.4

E.9.1 Randomly-generated loss sequences

We used the same loss vector with Section 3.2's randomly generated loss function to compare the result with GPT-4. The results show that the trained single-layer self-attention model or trained Transformer with regret-loss has comparable regret with FTRL and GPT-4's regret, and it can be checked in Figure 13.

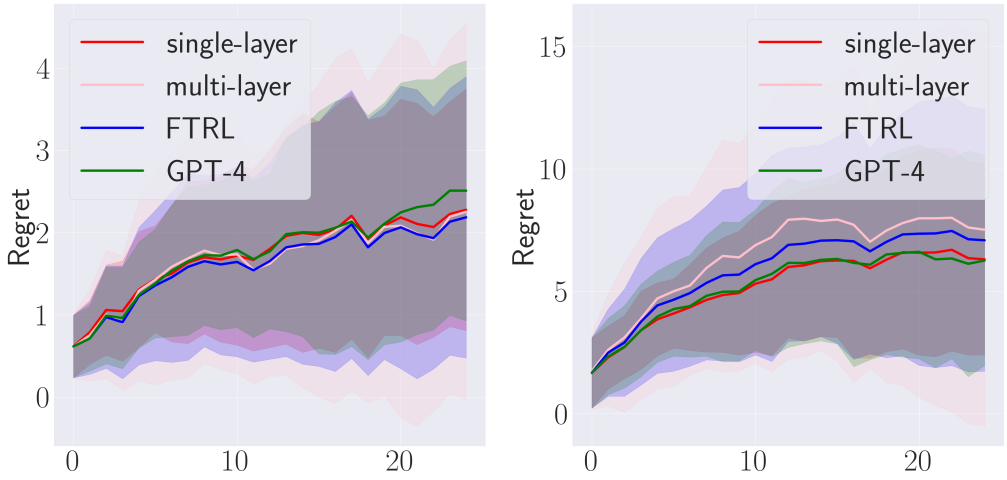


Figure 13: Result of the Randomly-generated loss sequences with Gaussian (left) and uniform with truncation (right). The p-value and $\widehat{\beta}_0$ value of the Gaussian loss sequences with trained Transformer / single-layer self-attention results were $p = 0.0, \widehat{\beta}_0 = 0.4$, $p = 0.0, \widehat{\beta}_0 = 0.39$, repsectively. The p-value and $\widehat{\beta}_0$ value of uniform loss sequences with trained Transformer / single-layer self-attention results were $p = 0.0, \widehat{\beta}_0 = 0.43$, $p = 0.0, \widehat{\beta}_0 = 0.47$, repsectively.

E.9.2 Loss sequences with a predictable trend

We investigate the case of loss sequences with predictable trends such as linear trends or sinusoid trends. We might expect that the performance of the trained Transformer would surpass the performance of traditional no-regret algorithms since FTRL would not be an optimal algorithm for the loss sequence with a predictable trend. We modified the training distribution of random variable Z to follow two kinds of trends: linear and sinusoid functions. The results show that the trained single-layer self-attention model or trained Transformer with regret-loss outperformed GPT-4 in the metric of regret when the loss sequence is a linear trend, and it can be checked in Figure 14.

E.9.3 Repeated games

We investigate the case with a multi-player repeated game; 2x2, 3x3x3, 3x3x3x3 games. The results show that the trained single-layer self-attention model or trained Transformer with regret-loss has a similar performance with FTRL; it can be checked in Figure 15.

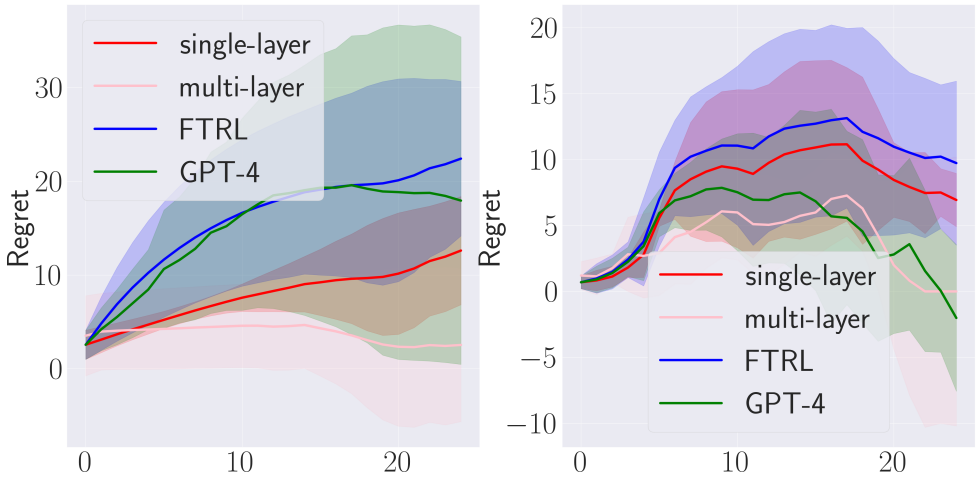


Figure 14: Result of the loss sequences with linear trends (left) and sinusoid trend (right). The p-value and $\widehat{\beta}_0$ value of the Gaussian loss sequences with trained Transformer / single-layer self-attention results were $p = 0.0, \widehat{\beta}_0 = 0.51$, $p = 0.0, \widehat{\beta}_0 = -0.13$, repsectively. The p-value and $\widehat{\beta}_0$ value of uniform loss sequences with trained Transformer / single-layer self-attention results were $p = 0.0, \widehat{\beta}_0 = 0.89$, $p = 0.0, \widehat{\beta}_0 = -0.9$, repsectively.

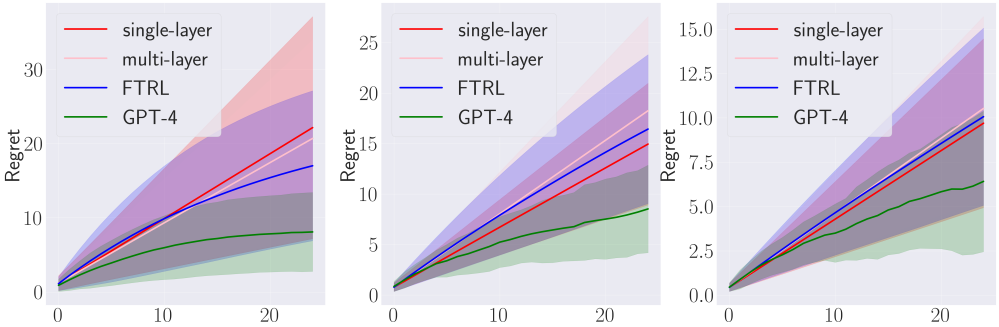


Figure 15: Result of the game with two-player (left) ($(p, \widehat{\beta}_0) = (0.0, 0.69)$), three-player (middle) ($(p, \widehat{\beta}_0) = (0.0, 0.94)$), and four-player (right) ($(p, \widehat{\beta}_0) = (0.0, 0.98)$).

E.9.4 Two Scenarios for Regrettable Behavior of GPT-4

We used the same loss vector as in Section 3.2. The results show that the trained single-layer self-attention model or training Transformer with regret-loss can achieve comparable regret performance as FTRL and outperform GPT-4, and it can also be checked with Figure 4.

E.10 Ablation Study on Training Equation (5.2)

In this section, we provide an ablation study that changing N and k in Equation (5.2). To be specific, we will set $N = 1, 2, 4$, $f(x, k) = \max(x, 0)^k$, $h(x) = \max(x, 0)^2$, and $k = 0, 1, 2$. For the multi-layer transformer training, we used 4 layers, 1 head Transformer. For both single-layer and multi-layer, we employed the Adam optimizer, setting the learning rate to 0.001. During training, we conducted 2,000 epochs with a batch size 512. We experimented on the randomly-generated loss sequences. Especially, we used the uniform loss sequence ($\ell_t \sim \text{Unif}([0, 10]^2)$ (Figure 16 and Figure 17) and the Gaussian loss sequence ($\ell_t \sim \mathcal{N}(5, \mathbf{1}_2, I)$ (Figure 18 and Figure 19). The result shows that it might be available to train with several different setting.

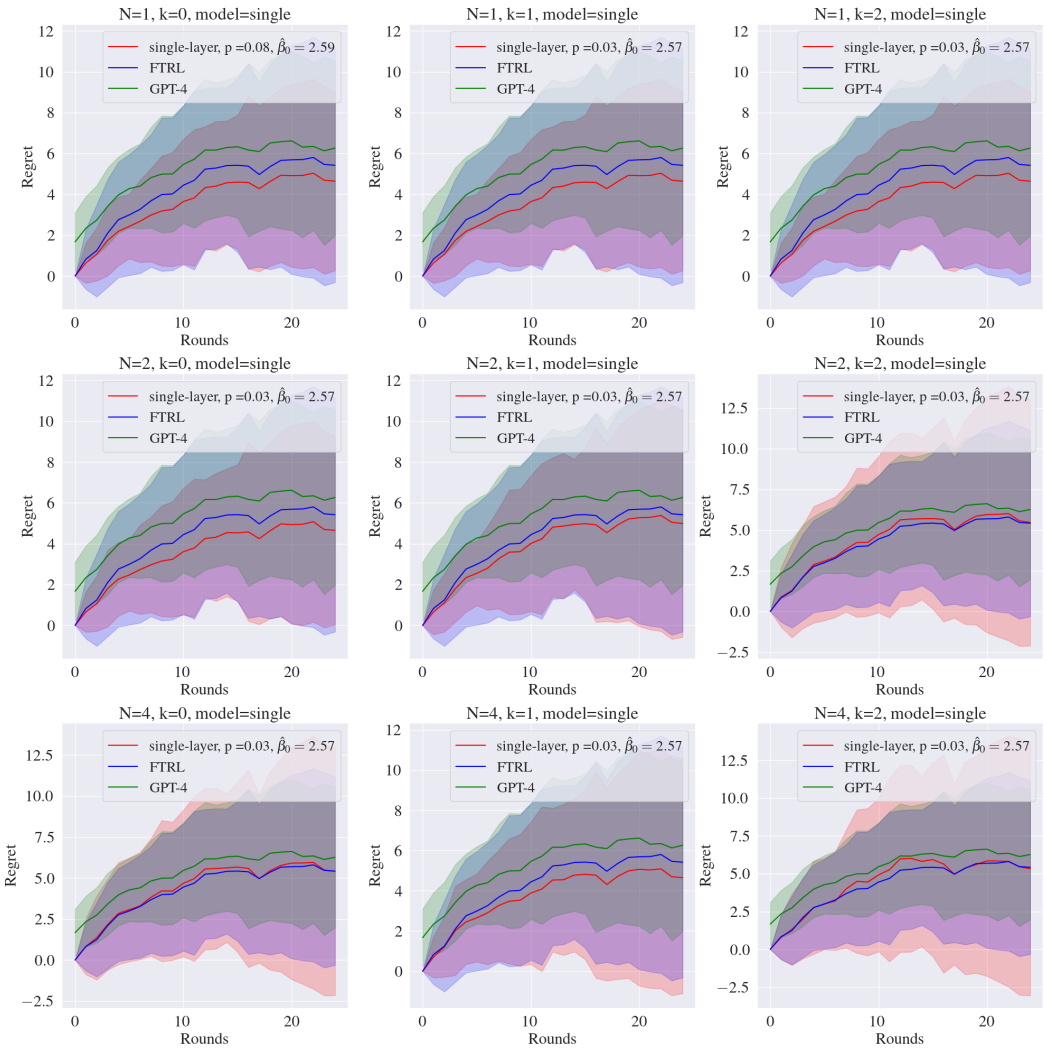


Figure 16: Ablation study for the uniform loss sequence trained with single-layer self-attention layer with `Softmax` projection. p value is around 0.03 to 0.08, and it shows the no-regret behavior by our trend-checking framework.

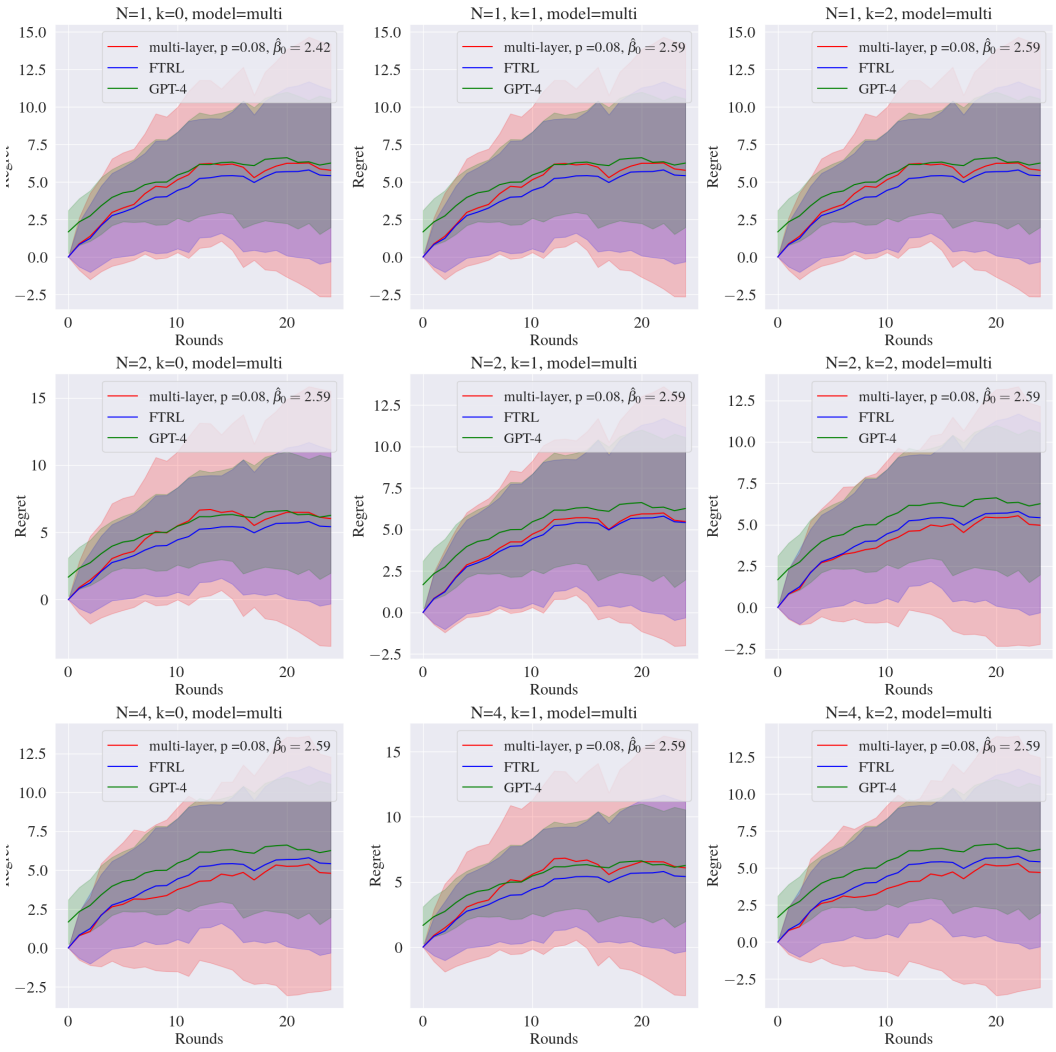


Figure 17: Ablation study for the uniform loss sequence trained with multi-layer self-attention layer with `Softmax` projection. p value is around 0.03 to 0.08, and it shows the no-regret behavior by our trend-checking framework.

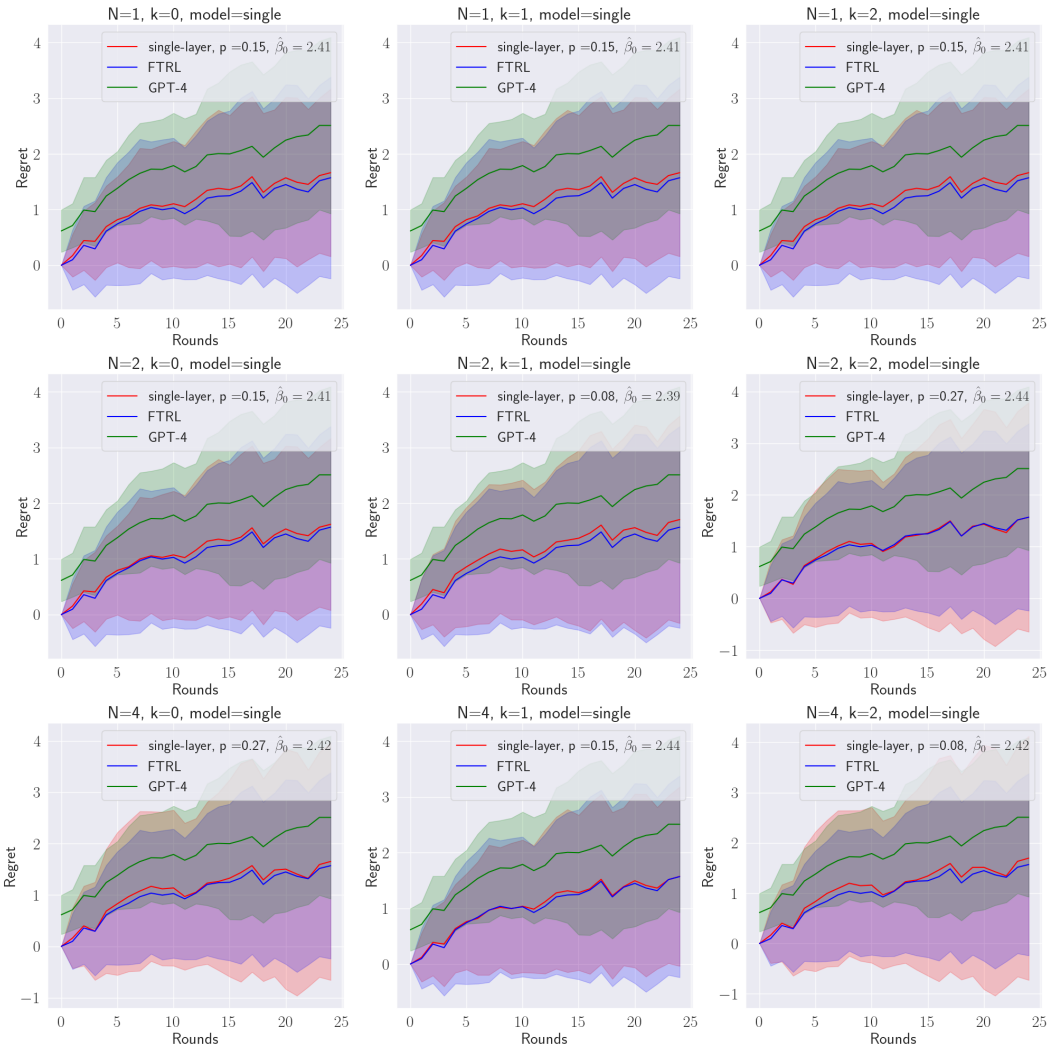


Figure 18: Ablation study for the Gaussian loss sequence trained with single-layer self-attention layer with `Softmax` projection. p value is around 0.08 to 0.27, and it shows the no-regret behavior by our trend-checking framework.

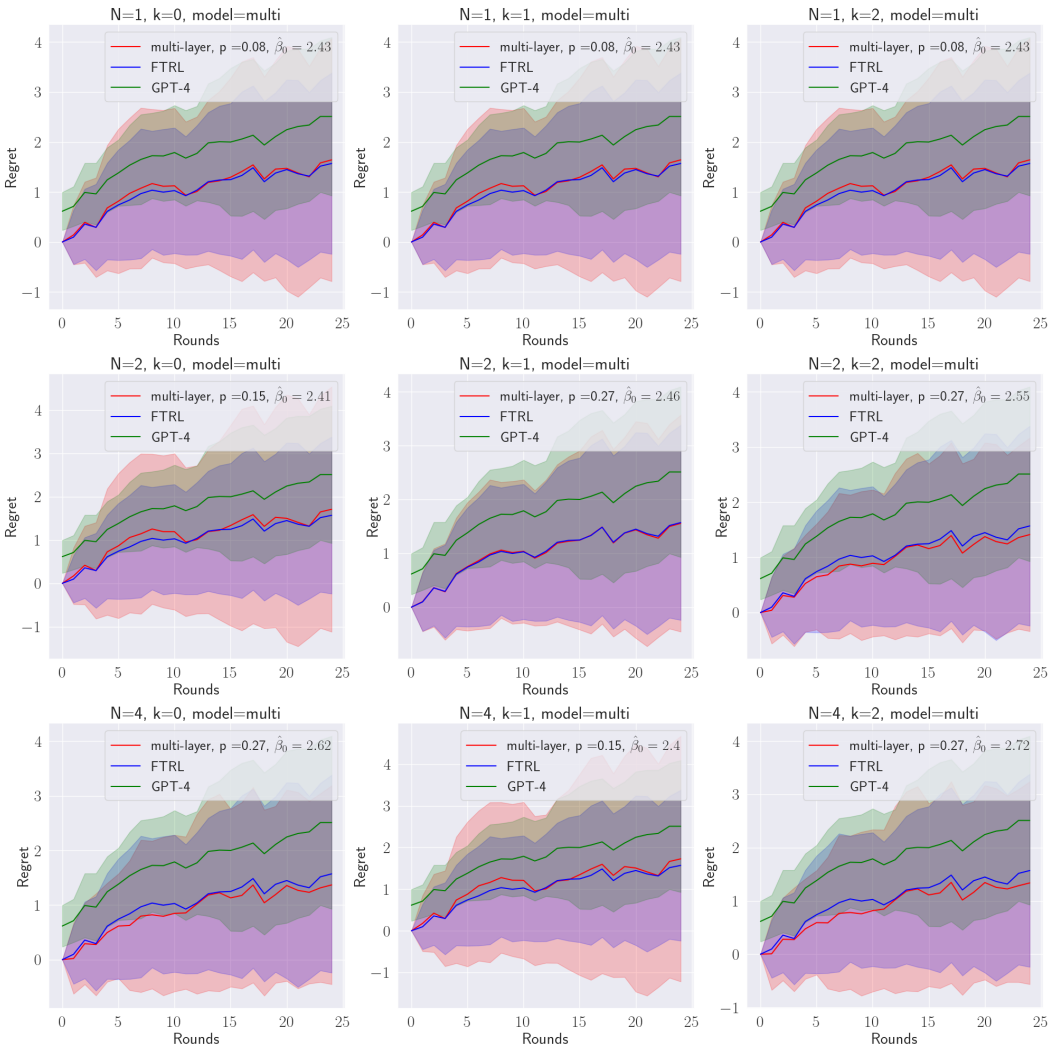


Figure 19: Ablation study for the Gaussian loss sequence trained with single-layer self-attention layer with `Softmax` projection. p value is around 0.08 to 0.27, and it shows the no-regret behavior by our trend-checking framework.