Do LLM Agents Have Regret?
A Case Study in Online Learning and Games

Anonymous Authors

March 27, 2024

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 quantitative 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 in online learning and game theory, 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 some theoretical insights into the no-regret behaviors of LLM agents, under certain assumptions on the supervised pre-training and the 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.

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., 2023a). 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., 2023b; Shinn et al., 2023; Wang et al., 2023c; Significant Gravitas, 2023). Specifically, the LLM agent interacts with the (physical) world in a *dynamic/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 and 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: the long-run interaction of no-regret learners leads to certain *equilibria* when they repeatedly play games (Cesa-Bianchi and Lugosi, 2006). In fact, *no-regret* learning has served as a natural model for predicting and explaining human behaviors in strategic decision-making, with experimental evidence (Erev and Roth, 1998; Nekipelov et al., 2015; Balseiro and Gur, 2019). It has thus been posited as an important model of "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 online and strategic decision-making behaviors 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 aforementioned benchmark online decision-making settings, 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 patterns with bounded variation, and in playing both representative and randomly generated repeated games. For the latter, equilibria will emerge as the long-term behavior of the interactions when all LLM agents are no-regret. Second, we provide some theoretical insights into the observed no-regret behaviors, based on some hypothetical model of the human decision-makers who generate the data, and certain assumptions on the *supervised pre-training* procedure, a common practice in training large models for decision-making. In particular, we make a connection of pre-trained LLMs to the known no-regret algorithm of *follow-the-perturbed-leader* (FTPL) under such 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* (Bubeck et al., 2023; Achiam et al., 2023; Wei et al., 2022b,a; Srivastava et al., 2023; Yao et al., 2023a) has inspired a growing line of research on *LLM for (interactive) decision-making*, i.e., an LLM-based autonomous agent interacts with the environment by taking actions repeatedly/sequentially, based on the feedback it perceives. Some promises have been shown from a *planning* perspective (Hao et al., 2023; Valmeekam et al., 2023; Huang et al., 2022b; Shen et al., 2023). In particular, for embodied AI applications, e.g., robotics, LLMs have achieved impressive performance when used as the controller for decision-making (Ahn et al., 2022; Yao et al., 2023b; Shinn et al., 2023; Wang et al., 2023c; Driess et al., 2023; Significant Gravitas, 2023). However, the performance of decision-making has not been rigorously characterized via the regret metric in these works. Very recently, Liu et al. (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., 2024; Li et al., 2023c) showed that multi-LLM *debate* can improve the reasoning and evaluation capabilities of the LLMs. Qian et al. (2023); Schick et al. (2023); Wu et al. (2023) demonstrated the potential of multi-LLM interactions and collaboration in software development, writing, and problem-solving, respectively. Zhang et al. (2024) exhibited a similar potential in embodied cooperative environments. More formally, multi-LLM interactions have also been investigated under a *game-theoretic* framework, to characterize the *strategic* decision-making of LLM agents. Bakhtin et al. (2022); Mukobi et al. (2023) and Xu et al. (2023b,a) have demonstrated the promise of LLMs in playing Diplomacy and WereWolf games, respectively, which are both language-based games with a mixture of competitive and cooperative agents. Note that these works utilized LLM to solve a specific rather than a general game. Related to our work, Brookins and DeBacker (2023); Akata et al. (2023); Lorè and Heydari (2023); Brookins and DeBacker (2023); Fan et al. (2023) have also used (repeated) matrix games as a benchmark to evaluate the reasoning capability and rationality of LLM agents. In contrast to our work, these empirical studies have not formally investigated LLM agents using the metric of *regret*, nor through the lenses of *online learning* and *equilibrium-computation*, which are all fundamental in modeling and analyzing strategic multi-agent interactions. Moreover, our work also provides theoretical results to explain and further enhance the no-regret property of LLM agents.

LLMs & Human/Social behavior. LLMs have also been used to *simulate* the behavior of human beings, for social science and economics studies (Engel et al., 2023). The extent of LLMs simulating human behavior has been claimed as a way to evaluate the level of its intelligence in a controlled environment (Aher et al., 2023; Tsai et al., 2023). For example, Li et al. (2023b); Hong et al. (2024); Zhao et al. (2023) showed that by specifying different "roles" to LLM agents, certain collaborative/-competitive behaviors can emerge. Argyle et al. (2023) showed that LLMs can emulate response distributions from diverse human subgroups, illustrating their adaptability. Horton (2023) argued that an LLM, as a computational model of humans, can be used as *homo economicus* when given endowments, information, preferences, etc., to gain new economic insights by simulating its interaction with other LLMs. Park et al. (2022, 2023) proposed scalable simulators that can generate realistic social behaviors emerging in populated and interactive social systems, and the emerging behaviors of LLM agents in society have also been consistently observed in Chen et al. (2024, 2023). Li et al. (2023d,a) studied the 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 and Levine, 1998) and strategic interactions (Young, 2004; Camerer, 2011).

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

Online learning and games. Online learning has been extensively studied to model the decision-making of an agent who interacts with the environment sequentially, with a potentially arbitrary sequence of loss functions (Shalev-Shwartz, 2012; Hazan, 2016), and has a deep connection to game theory (Cesa-Bianchi and 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 and Schapire, 1997; Arora et al., 2012), EXP3 (Auer et al., 2002), and more generally follow-the-regularized-leader (FTRL) (Shalev-Shwartz and Singer, 2007) and follow-the-perturbed-leader (FTPL) (Kalai and Vempala, 2005). In the bandit literature (Lattimore and Szepesvári, 2020; Bubeck et al., 2012), such a setting without any statistical assumptions on the losses is also referred to as the *adversarial/non-stochastic* setting. Following the conventions in this literature, the online settings we focus on shall not be confused with the stationary and *stochastic*(-bandit)/(-reinforcement learning) settings that have been explored in several other recent works on *Transformers for decision-making*

(Lee et al., 2023; Lin et al., 2024). Centering around the regret metric, our work has also explored the non-stationary bandit setting (Besbes et al., 2014), as well as the repeated game setting where the environment itself consists of strategic agents (Cesa-Bianchi and Lugosi, 2006).

2 Preliminaries

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

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 chosen by the environment, potentially in an adversarial fashion. 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 and Warmuth, 1994; Hazan, 2016), without loss of much generality (c.f. Appendix A.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 ℓ_{ta_t} (we sometimes also interchangeably write it as $\ell_t(a_t)$), the a_tth element of ℓ_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 $r_n : \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to [-B, B]$ 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 $\ell_t := -\mathbb{E}_{a_{-n,t}\sim\pi_{-n,t}}[r_n(\cdot, a_{-n,t})]$ at time t, where $-n$ denotes the index for all the 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 hereafter. 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 $\mathscr{A}$, let $\pi_{\mathscr{A},t}$ denote the decision policy of the agent at time t generated by $\mathscr{A}$. Then, the regret, which is the difference between the accumulated (expected) loss incurred by implementing $\mathscr{A}$ and that incurred by the best-in-hindsight fixed decision, can be defined as

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

In the Experts Problem, the definition can be instantiated as $\text{Regret}_{\mathscr{A}}((\ell_t)_{t\in[T]}) := \sum_{t=1}^{T}\langle \ell_t, \pi_{\mathscr{A},t}\rangle - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle$. With bandit-feedback, a commonly used metric may also take further expectation for $\text{Regret}_{\mathscr{A}}$, over the randomness of the generated policies $(\pi_{\mathscr{A},t})_{t\in[T]}$. An algorithm $\mathscr{A}$ is referred to as being *no-regret*, if $\text{Regret}_{\mathscr{A}}((f_t)_{t\in[T]}) \sim o(T)$, i.e., the regret grows sublinearly in T. Widely-known no-regret algorithms include follow-the-regularized-leader (FTRL) (Shalev-Shwartz and Singer, 2007), follow-the-perturbed-leader (Kalai and Vempala, 2005) (See Appendix A.3 for a detailed introduction).

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 individual time t: $\text{D-Regret}_{\mathscr{A}}((f_t)_{t\in[T]}) := \sum_{t=1}^{T} f_t(\pi_{\mathscr{A},t}) - \sum_{t=1}^{T}\inf_{\pi\in\Pi} f_t(\pi)$, which is a stronger notion than $\text{Regret}_{\mathscr{A}}((f_t)_{t\in[T]})$ in that $\text{Regret}_{\mathscr{A}}((f_t)_{t\in[T]}) \leq \text{D-Regret}_{\mathscr{A}}((f_t)_{t\in[T]})$.

3 Do Pre-Trained LLMs Have Regret? Experimental Validation

In this section, we explore the no-regret behaviors of representative pre-trained LLMs (i.e., GPT-4 Turbo, GPT-4, and GPT-3.5 Turbo), in the context of online learning and games. All experiments with LLMs are conducted using the public OpenAI Python API (Openai, 2023).

Intuition why pre-trained language models may exhibit no-regret behavior. Transformer-based LLMs have demonstrated impressive *in-context-learning* and few-/zero-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 in certain supervised learning problems (Akyürek et al., 2023; Von Oswald et al., 2023; Dai et al., 2023; Ahn et al., 2023; Zhang et al., 2023a; 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 behavior of such meta-learners in online learning, due to their fast adaptability. However, it is not straightforward due to the fundamental difference between multi-task/meta-learning and online learning settings, as well as the difference between *stationary* and *non-stationary/adversarial* environments in decision-making. Next, we provide both experimental and theoretical studies to validate this intuition.

Interaction protocol. To enable the sequential interactions with LLMs, 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 LLMs through online learning or games, we only provide simple prompts that she should utilize the history information, without providing explicit rules of *how* to make use of the history information, nor asking her to *minimize regret* (in any sense). A detailed description and an ablation study of the prompts are deferred to Appendix B.1, and an illustration of the protocol in playing repeated games is given in Figure 3.

3.1 Framework for No-Regret Behavior Validation

Before delving into the results, We note that to the best of our knowledge, we are not aware of any principled framework for validating no-regret behaviors with finite-time experimental data. Therefore, we propose two frameworks to rigorously validate the no-regret behavior of algorithms over a *finite* T, which might be of independent interest.

Trend-checking framework. We propose the following hypothesis test:

$$H_0: \text{The sequence } \left(\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t\right)_{t=1}^{\infty} \text{ either diverges or converges to a positive constant}$$
$$H_1: \text{The sequence } \left(\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t\right)_{t=1}^{\infty} \text{ converges to } 0$$

with H_0 and H_1 denoting the null and alternative hypotheses, respectively. 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:

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

Ideally, one should check if $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t$ approaches zero as t goes to infinity. With a finite T value, 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 $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t - \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t+1]}\right)/(t+1) > 0$, for which we use Proposition 1 below (whose proof is deferred to Appendix B.2) to provide some understanding of (how small) the probability it happens under various counts. For example, with the default choice of $T = 25$ in our experiments later, one can see from Proposition 1 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, denoted as p_{trend}, as the output of this framework.

Proposition 1. (p-value of the null hypothesis)**.** *Define the event*

$$\mathcal{E}(s,T) := \left\{ \text{The number of } \frac{\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)}{t} - \frac{\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t+1]}\right)}{t+1} > 0 \text{ for } t = 1, \ldots, T \text{ is at least } s \geq \frac{T-1}{2} \right\}.$$

Under the assumption that the null hypothesis H_0 holds, the probability of this event happening is bounded as $\mathbb{P}_{H_0}(\mathcal{E}(s,T)) \leq \frac{1}{2^{T-1}} \sum_{t=s}^{T-1} \binom{T-1}{t}$.

Regression-based framework. In complement to the statistical framework above, we propose an alternative approach by fitting the data with regression. In particular, one can use the data

$$\left\{\left(t, \log \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)\right)\right\}_{t\in[T]}$$

to fit a linear function $g(t) = \beta_0 \log t + \beta_1$, 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, i.e., the *sublinear* growth of $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)$ over time. While being simple, it cannot be directly used when $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right) < 0$, so we set $\log \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)$ as -10 if this happens. We define p_{reg} as the p-value of the regression parameter $\widehat{\beta_0}$, and will report the pair of $(\widehat{\beta_0}, p_{reg})$ as the output of this framework.

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 environments.

Online learning in arbitrarily changing environment. We first consider the setting with arbitrarily changing environments, which are instantiated as follows: 1) *Randomly-generated loss sequences*. At every timestep, we generate a random loss vector $\ell_t \sim \text{Unif}([0,10]^d)$ or $\ell_t \sim \mathcal{N}(5\mathbf{1}_d, I)$ with clipping to $[0,10]$ to ensure the boundedness, such that the loss vectors of different timesteps can be arbitrarily different; 2) *Loss sequences with a predictable trend.* Although many real-world environments may change, they often change following 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 regrets than these baselines.

Online learning (in non-stationary environment). We then experiment on the setting where the losses are still changing over time, but their total 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

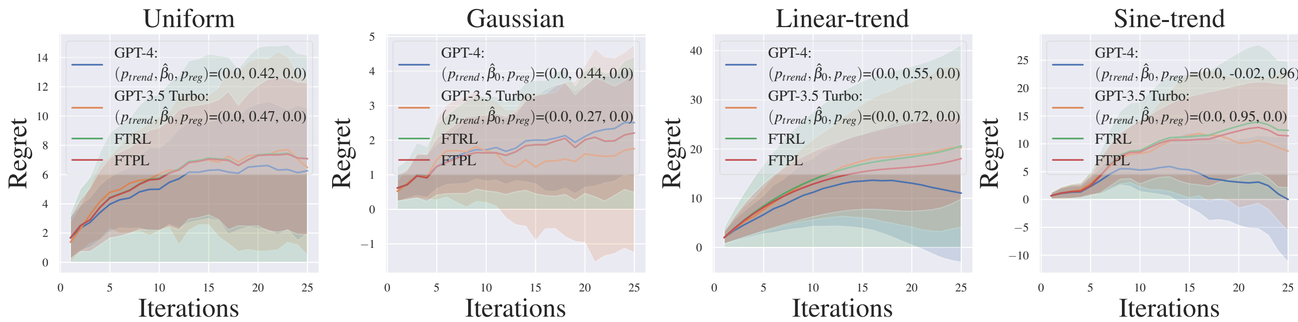

Figure 1: Regret of GPT-3.5 Turbo/GPT-4 for online learning with full-information feedback in 4 different settings. It performs comparably and sometimes even better than well-known no-regret learning algorithms, FTRL and FTPL.

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 $\sum_{t=1}^{T-1} \|\ell_{t+1} - \ell_t\|_\infty = o(T)$; 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 $\ell_{t_i} \leftarrow 10\mathbf{1}_d - \ell_{t_i}$. 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 *FTRL/FTPL with Restart*, a standard variant of FTRL/FTPL for non-stationary online learning (see e.g., Besbes et al. (2014)). We refer to Appendix A.3 for a more detailed introduction of FTRL/FTPL with Restart.

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, then sample one $a_t \sim \pi_t(\cdot)$. In the bandit setting, the LLM agent can only access (a_t, ℓ_{ta_t}). Instead of directly feeding it to the agent, we feed an estimate of the loss vector $\widehat{\ell_t} \in \mathbb{R}^d$, where $\widehat{\ell_t}(a) \leftarrow \frac{\ell_t(a)}{\pi_t(a)} \mathbb{1}(a_t = a)$ for all $j \in [d]$. Note that 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 (Auer et al., 2002). Later, we will also show the benefits of such operations (c.f. Section 4). We compare the performance of pre-trained LLMs with that of the counterparts of FTRL with bandit feedback, e.g., EXP3 (Auer et al., 2002) and the bandit-version of FTPL (Abernethy et al., 2015), in both Figure 2 and Table 1, where GPT-4 consistently achieves lower regret.

3.3 Results: Multi-Player Repeated Games

We now consider the setting when multiple LLMs make online strategic decisions in a *shared* environment repeatedly. Specifically, at each round, the loss vectors each agent receives are determined by both her payoff matrix and the strategies of all other agents. Note that the payoff matrix is not directly revealed to the LLM agent, but she has to make decisions in a completely online fashion based on the payoff vector marginalized by the opponents' strategies (See Figure 3 for a prompt example). This is a typical scenario in learning in (repeated) games (Fudenberg and Levine, 1998).

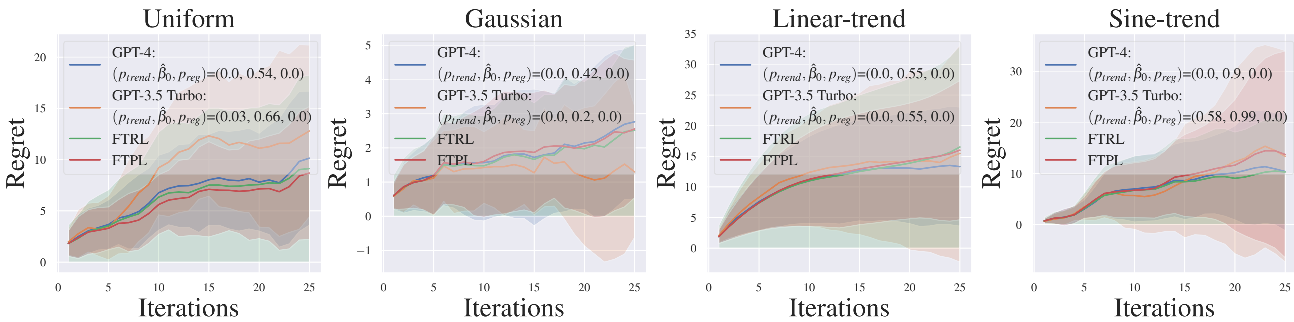

Figure 2: Regret of GPT-3.5 Turbo/GPT-4 for online learning with bandit feedback in 4 different settings. It performs comparably and sometimes even better than well-known no-regret learning algorithms, variants of FTRL and FTPL with bandit-feedback.

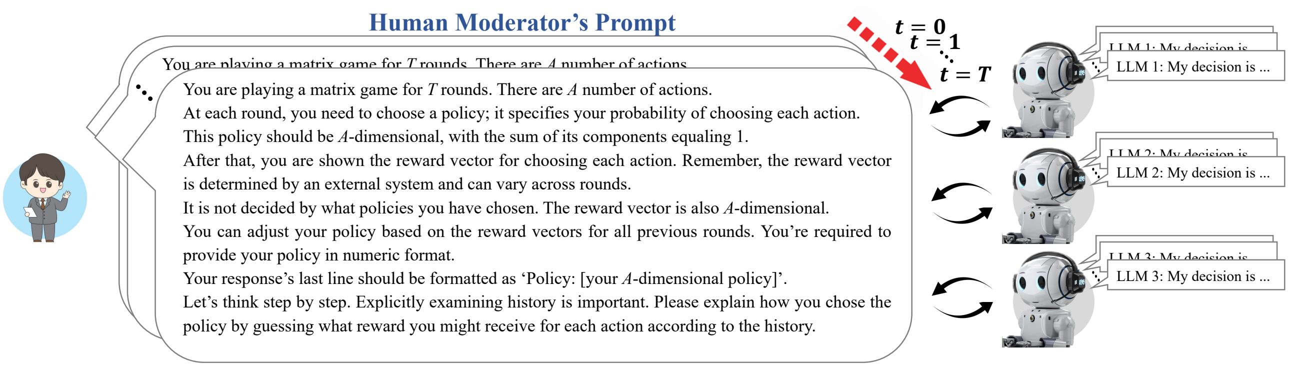

Figure 3: Demonstration of the prompts and interaction protocol for multi-player repeated games. A human moderator does not provide the game's payoff matrices to the LLMs. Instead, at each round, the human moderator provides each player's own payoff vector history.

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

Randomly generated games. To further validate the no-regret behavior of LLMs, 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 Unif([0, 10]). These are larger and more challenging settings than the structured and representative ones above.

We summarize the experimental results in Figure 4, which are similar to the above in the online setting: for all types of games, GPT-4 achieves sublinear regret, which is comparable with that obtained by FTRL for most games. We provide six instances of three-player general-sum games and six instances of four-player general-sum games in Figure 5 and Figure 6, respectively. Occasionally, GPT-4 even provides a negative regret value.

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

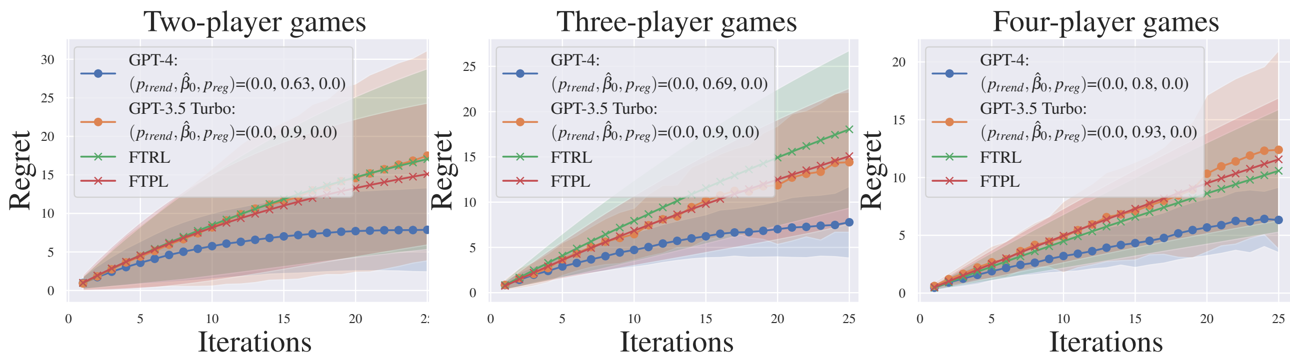

Figure 4: Regret of GPT-3.5 Turbo/GPT-4 for repeated games of 3 different game sizes, where the sublinear regret is validated by both of our statistical frameworks. Due to the symmetry of agents in randomly generated games, we report the regret of one agent for ease of presentation.

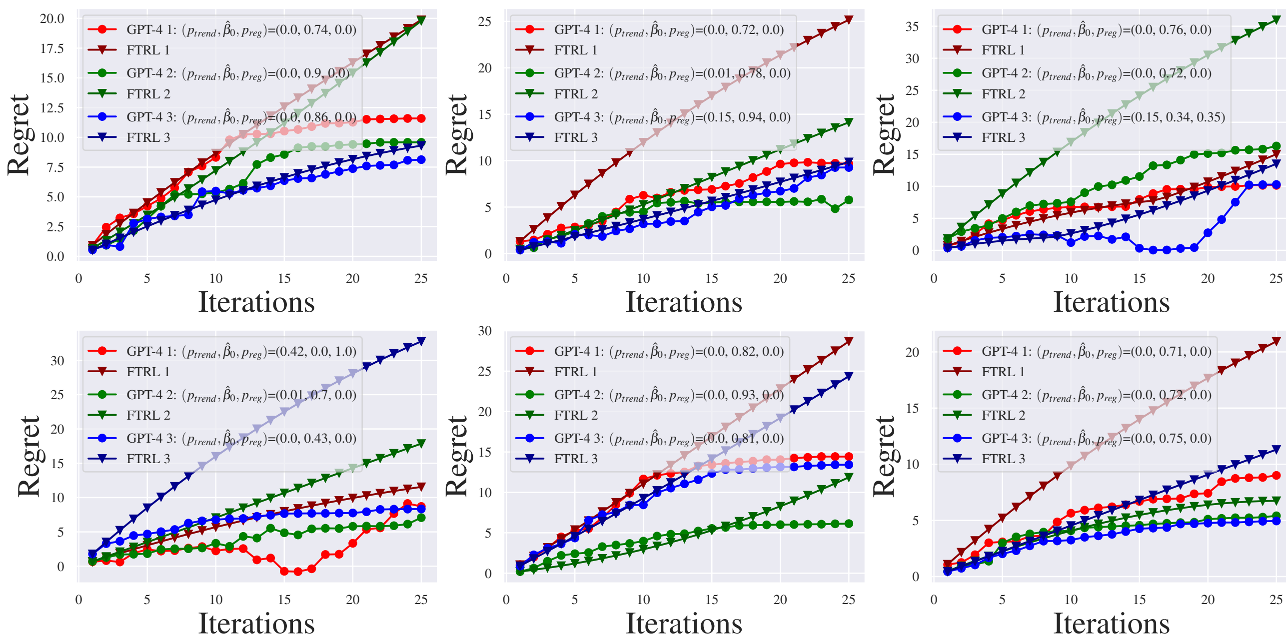

Figure 5: Regret of GPT-4 and the FTRL algorithm in 6 randomly generated three-player general-sum games. GPT-4 has comparable (even better) no-regret properties when compared with the FTRL algorithm.

property might break for LLM agents if the loss vectors are generated in a more adversarial way.

Canonical counterexample for follow-the-leader. To begin with, we consider a well-known example that *follow-the-leader* (FTL) algorithm (Shalev-Shwartz, 2012) suffers from linear regret (Hazan, 2016, Chapter 5), where $\ell_1(1) = 5, \ell_1(2) = 0$ 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 identify the pattern for the loss sequence that the optimal action *alternates*, thus accurately predicting the loss it will receive and achieving nearly zero regret in Figure 7. In other words, GPT-4 agent seems to not fail in the same way as FTL, which is known to be due to the lack of randomness in prediction.

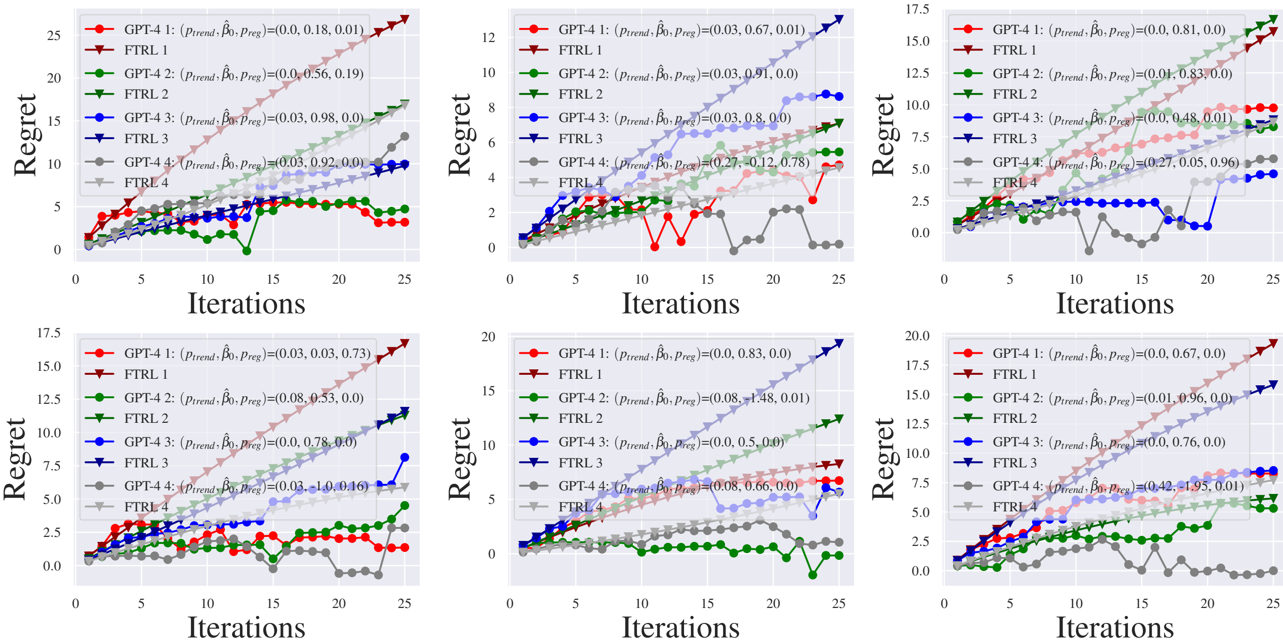

Figure 6: Regret of GPT-4 and the FTRL algorithm in 6 randomly generated four-player general-sum games. GPT-4 has comparable (even better) no-regret properties when compared with the FTRL algorithm, according to the frameworks in Section 3.1 and the graphic trends.

Less predictable loss sequence. Inspired by the counterexample above, 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 average regret over 20 repeated experiments for the later two settings using GPT-4 and more advanced GPT-4 Turbo in Figure 7, where we cannot reject the hypothesis that GPT-4 (Turbo) has linear-regret by either our trend-checking or regression-based framework.

These observations have thus motivated us to design new approaches to better promote the no-regret property of LLM agents, with additional training, 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 have already exhibited good no-regret behaviors oftentimes.

4 Why Are Pre-Trained LLMs (No-)Regret? A Hypothetical Model and Some 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 Turbo 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 literature for modeling human behaviors, and the recent literature on understanding LLMs/Transformers.

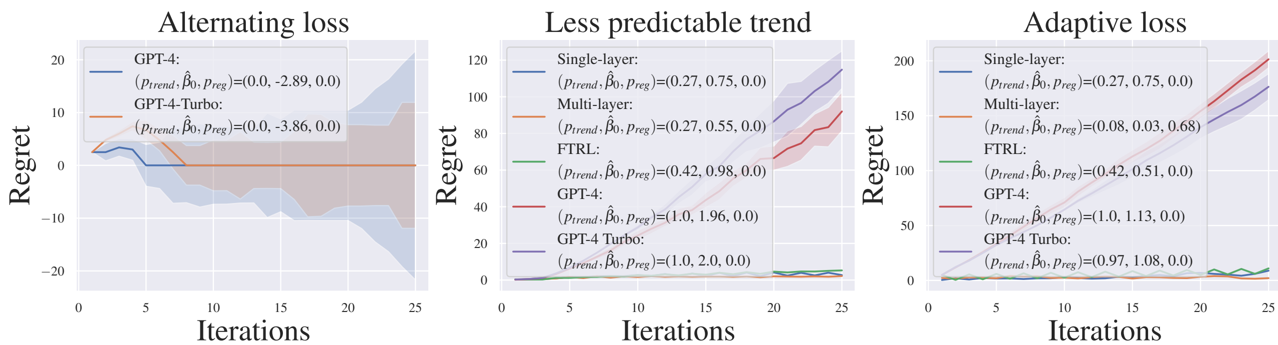

Figure 7: (left) Regret of both GPT-4 and GPT-4 Turbo under the canonical counterexample for FTL (Hazan, 2016, Chapter 5). (mid, right) Failure of GPT-4 and GPT-4 Turbo on two scenarios for regrettable behavior of GPT, while Transformers with regret-loss provide no-regret behaviors.

Dynamic regret		GPT-4	GPT-3.5 Turbo	FTRL	FTPL
Full information	Gradual variation	12.61 ± 7.01 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.0, 0.58, 0.0)$	19.09 ± 11.33 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.0, 0.83, 0.0)$	36.58 ± 24.51	35.19 ± 22.51
	Abrupt variation	30.0 ± 19.91 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.01, 0.87, 0.0)$	33.65 ± 22.51 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.08, 0.96, 0.0)$	36.52 ± 27.68	36.24 ± 28.22
Bandit	Gradual variation	21.39 ± 10.86 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.0, 0.78, 0.0)$	28.42 ± 21.6 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.0, 0.83, 0.0)$	37.64 ± 21.97	36.37 ± 20.7
	Abrupt variation	35.94 ± 28.93 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.42, 0.95, 0.0)$	30.76 ± 25.48 $(p_{trend}, \widehat{\beta}_0, p_{reg}) = (0.92, 1.01, 0.0)$	36.52 ± 27.68	38.82 ± 26.17

Table 1: Dynamic regret of GPT-3.5 Turbo/GPT-4 in a non-stationary environment with either full-information or bandit feedback. Every experiment is conducted with 25 rounds. No-regret behaviors of GPT-3.5 Turbo/GPT-4 are validated by both of our frameworks (low p-values and $\widehat{\beta}_0 < 1$). The only exception is GPT-3.5 Turbo on loss sequence with abrupt variations under bandit feedback. This indicates that GPT-3.5 Turbo may not be capable of dealing with an abruptly changing environment with limited feedback, although the average regret achieved eventually is still lower than that of other baselines.

4.1 Pre-Trained LLMs Have Similar Regret as Humans (Who Generate Data)

We first provide a direct observation based on some existing speculation on the capability of Transformer-based LLMs. 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., 2022; Zhang et al., 2023b; 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:*

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

where P_{data} and P_{LLM} are the pre-training data distribution and the decision policy distribution of the *pre-trained LLM, respectively. Then, the regret of an LLM agent with ϵ-decision error is bounded as:*

$$\text{(D-)Regret}_{LLM}\Big((\ell_t)_{t\in[T]}\Big) \in \left[\text{(D-)Regret}_{data}\Big((\ell_t)_{t\in[T]}\Big) \pm \epsilon\|\ell_t\|\sup_{\pi\in\Pi}\|\pi\|\right],$$

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 achieved if LLM is constructed by a rich function class, e.g., the Transformer architecture (Zhang et al., 2023b; Lin et al., 2024). We defer a proof of the result to Appendix C.

Note that the above observation does not provide the full picture – the $\text{(D-)Regret}_{\text{data}}\Big((\ell_t)_{t\in[T]}\Big)$, which denotes the (dynamic-)regret of using $P_{\text{data}}\Big(\pi \,\Big|\, (\ell_i)_{i\in[t]}\Big)$ for online learning, is not necessarily sublinear in T. We discuss next under what (natural) models and pre-training processes, the no-regret behavior of pre-trained LLMs may emerge.

4.2 A Human Decision-Making Model: Quantal Response

A seminal model for human decision-making behavior is the *quantal response* model, which assumes that humans are often *imperfect* decision-makers, and their bounded rationality can be modeled through unseen latent variables that influence the decision-making process (McFadden, 1976; McKelvey and Palfrey, 1995). Formally, the quantal response is defined as follows:

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

$$P^{\eta}_{quantal}\Big(a \,\Big|\, \ell\Big) = \mathbb{P}\left(a \in \operatorname*{arg\,min}_{a'\in\mathcal{A}} \ z(a')\right), \qquad where \quad z = \ell + \eta\epsilon.$$

In essence, this implies that humans are rational but with respect to (w.r.t.) the latent variable z, a perturbed version of ℓ, instead of ℓ per se. This addition of noise to the actual loss vector characterizes the bounded rationality of humans in decision-making.

The traditional quantal response formulation primarily focused on scenarios with a single loss vector. In online decision-making, given the *history* information, the human at each time t is faced with *multiple* loss vectors. Hence, we adopt the following generalization to model the decision-making behavior in this setting.

Definition 2 (Quantal response against multiple losses). *Given a set of loss vectors $(\ell_i)_{i\in[t]}$, a noise distribution P_{noise}, and $\eta_t > 0$, the generalized quantal response against $(\ell_i)_{i\in[t]}$ is defined as*

$$P^{\eta_t}_{quantal}\Big(a \,\Big|\, (\ell_i)_{i\in[t]}\Big) := P^{\eta_t}_{quantal}\left(a \,\middle|\, \sum_{i=1}^{t} \ell_i\right).$$

For $t = 0$, we define $P^{\eta_t}_{quantal}\Big(a \,\Big|\, (\ell_i)_{i\in[t]}\Big) := P^{\eta_0}_{quantal}\Big(a \,\Big|\, \emptyset\Big) = \frac{1}{d}$.

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, such a response against multiple loss vectors is believed to be natural, and has also been widely adopted in well-known no-regret learning algorithms of *smooth/stochastic fictitious play* (Fudenberg and Kreps, 1993) and *follow-the-perturbed-leader* (Kalai and Vempala, 2005), whose formal definitions can be found in Appendix A.3. Finally, note that the response model in Definition 2 does not necessarily involve a *sequential* decision-making process, i.e., the set of losses may not come from the history of an online learning process. With this (natural) human response model in hand, we hypothesize that powerful pre-trained LLMs also behave similarly as such a model. To illustrate how Definition 2 can potentially capture also the behavior of LLMs instead of only humans, we have also provided examples of LLMs *explaining* how they output their policies in Appendix B.4. It can be observed that the LLM agent tends to use the history of the reward vectors by looking at the "sum/average of the accumulated rewards", and tends to introduce "randomization" in decision-making, in order to be "less predictable and exploitable". These are known to be key to achieving no-regret behaviors in online learning and games (Cesa-Bianchi and Lugosi, 2006). Hence, we analyze a case where pre-training under certain canonical data assumptions can provably lead to the quantal response behaviors and further yield no-regret guarantees, to gain some insights into the empirical observations in Sections 3.2 and 3.3.

4.3 Case Study: Pre-Training under Canonical Data Distribution

Pre-training of LLMs often involves the approach of *next-token prediction*. When applying LLMs to sequential decision-making, the model receives the context of the decision-making sequence and then generates the *action*. This process can be conceptualized as *predicting the optimal action* in the form of the next token prediction. For instance, Yao et al. (2023b); Shinn et al. (2023); Liu et al. (2023a,c) demonstrated how decision-making may be framed in this way. Concretely, LLM receives the context as $(x_1, x_2, \cdots, x_{N_1})$ and output $(x_{N_1+1}, \cdots, x_{N_2})$ for some $N_1, N_2 \in \mathbb{N}^+$ and $N_1 < N_2$, where each $x_i \in \mathcal{V}$ represents one *natural language token* for $i \in [N_2]$, and $\mathcal{V}$ is the finite token set. Afterwards, the human user extracts the corresponding decision action a from the text $(x_{N_1+1}, \cdots, x_{N_2})$.

Meanwhile, in these cases, large models such as Transformers are often (pre-)trained for (sequential) decision-making under several *fixed/stationary* environments (Laskin et al., 2023; Lin et al., 2024; Lee et al., 2023; Reed et al., 2022), which may limit their ability to handle *arbitrary/non-stationary/adversarial* loss sequences in our online learning setup. Thus, it is natural to ask: *Is it possible to have no-regret behaviors emerging as a consequence of this (optimal) action prediction, under only a fixed pre-training distribution of stationary environments?*

Here we analyze a standard pre-training objective on a token sequence distribution $x_{1:N_{t+1}} \sim P_t^{text}$ for given $t \in [T]$, which is the expected log-likelihood maximization for next-token prediction:

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{x_{1:N_{t+1}} \sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log \mathrm{LLM}_\theta \left(x_j \,|\, x_{1:j-1} \right), \tag{1}$$

where we define $\mathrm{LLM}_\theta (x_1 \,|\, x_{1:0}) = \mathrm{LLM}_\theta (x_1)$, and Θ denotes the parameter space of the LLM.

For the pre-training data distribution, intuitively, we model it as follows: there exists a latent variable z, representing the loss of the underlying *static* decision-making problem. The pre-training dataset, however, only contains *partial observations* $x_{1:N_t}$ (a natural language representation of $\ell_{1:t}$) of z due to imperfect data collection. The presence of partial observations could be attributed to the fact that z is only *privately* known by the data-generator (human), representing the intentions of a human being/data-generator. Hence, LLM will only be pre-trained with partial and noisy information about z. Meanwhile, we assume that the optimal action $x_{N_t+1:N_{t+1}}$ (a natural language representation of a) w.r.t. the underlying loss vector z is available in the pre-training dataset as the *label*, which could come from user surveys, personal blogs, or explicit data annotation. We formalize such a scenario in the assumption below:

Assumption 1. *Given $T \in \mathbb{N}^+$, $t \in [T]$, and $N_{t+1} \in \mathbb{N}^+$, there is a latent variable z, $\ell_{1:t}$, $N_1, \cdots N_t \in [N_{t+1}]$, and $N_0 = 0$, such that $\mathbb{P}(z, \ell_{1:t}, x_{1:N_{t+1}}) = \mathbb{P}(z, \ell_{1:t})\mathbb{P}(x_{1:N_t} \mid \ell_{1:t})\mathbb{P}(x_{N_t+1:N_{t+1}} \mid z)$, and*

$$P_t^{text}(x_{1:N_{t+1}}) = \mathbb{P}(x_{1:N_{t+1}}) = \int_z \int_{\ell_{1:t}} \mathbb{P}(z, \ell_{1:t}, x_{1:N_{t+1}}) d\ell_{1:t} dz,$$

where we assume tokens $x_{1:N_t}$ encode the context information, i.e., information for $\ell_{1:t}$, and the user will decode action a from $x_{N_t+1:N_{t+1}}$. More formally, we denote the mappings that decode such semantic information into numeric values as f, g, such that $f(x_{N_{i-1}+1:N_i}) = \ell_i \in \mathbb{R}^d$ for each $i \in [t]$ and $g(x_{N_t+1:N_{t+1}}) = a \in \mathcal{A}$.

To further understand our assumption, we provide the following example and lemma, which recovers the prominent human behavior model introduced in Section 4.2, quantal response, under some canonical data distributions.

Example 1 (An example instantiating Assumption 1)**.** *We consider a common decision-making task that may generate the training data,* recommender systems. *An instance of the text data could be: "On September 29, 2023, user X clicked movie A three times, movie B eight times, and movie C five times". This sentence corresponds to $x_{N_{i-1}+1:N_i}$ for some $i \in [t]$ and serves as a natural language depiction of the numerical ℓ_i. The corresponding label $x_{N_t+1:N_{t+1}}$ can be obtained by some user survey: "User X's favorite movie is movie B". Meanwhile, z represents user X's latent, genuine preference for each movie – information that is private to the user, and cannot be observed or collected in the pre-training dataset. In this example, Assumption 1 suggests that $x_{1:N_t}$, which records the frequency of interactions with each movie, serves as an* imperfect estimate *of the user's latent, genuine preference for the movies, while the text $x_{N_t+1:N_{t+1}}$ depicts the user's favorite movie only based on her latent z.*

Lemma 1. (Alignment of Assumption 1 with quantal response)**.** *Fix $t \in [T]$, $\sigma > 0$. If we model the noise of data collection to be i.i.d. Gaussian distribution in the numeric value space, i.e.,*

$$\mathbb{P}\left(\left\{f(x_{N_{i-1}+1:N_i})\right\}_{i\in[t]} \,\middle|\, z\right) = \prod_{i=1}^{t} \mathbb{P}\left(f(x_{N_{i-1}+1:N_i}) \,\middle|\, z\right) \propto \prod_{i=1}^{t} \exp\left(-\frac{\|f(x_{N_{i-1}+1:N_i}) - z\|_2^2}{2\sigma^2}\right),$$

the prior distribution of the latent variable z is also Gaussian, i.e., $z \sim \mathcal{N}(\mathbf{0}_d, \sigma^2 I)$, and the text labels satisfy that $\mathbb{P}(g(x_{N_t+1:N_{t+1}}) \mid z) = \mathbb{1}\left(g(x_{N_t+1:N_{t+1}}) \in \arg\min_{a\in\mathcal{A}} z_a\right)$, then we have

$$\mathbb{P}\left(g(x_{N_t+1:N_{t+1}}) \,\middle|\, x_{1:N_t}\right) = P_{quantal}^{\sigma\sqrt{t+1}}\left(g(x_{N_t+1:N_{t+1}}) \,\middle|\, \left\{f(x_{N_{i-1}+1:N_i})\right\}_{i\in[t]}\right),$$

with $P_{noise} = \mathcal{N}(\mathbf{0}_d, I)$ in Definition 2, i.e., the action $a = g(x_{N_t+1:N_{t+1}})$ extracted from the text $x_{N_t+1:N_{t+1}}$ is a quantal response w.r.t. the loss vectors $\left(f(x_{N_{i-1}+1:N_i})\right)_{i\in[t]}$.

We defer the proof of the lemma to Appendix C.2. Now based on this lemma, we provide the no-regret guarantees of the corresponding pre-trained LLM.

Theorem 1. (Emergence of no-regret behavior)**.** *Under the assumptions of Lemma 1, suppose the function class of LLM_θ is expressive enough such that for all $t \in [T]$, $\max_{\theta\in\Theta} \mathbb{E}_{x_{1:N_{t+1}} \sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log LLM_\theta\left(x_j \mid x_{1:j-1}\right) = \max_{\{q_j \in \{\mathcal{V}^{j-1} \to \Delta(\mathcal{V})\}\}_{j\in[N_{t+1}]}} \mathbb{E}_{x_{1:N_{t+1}} \sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log q_j\left(x_j \mid x_{1:j-1}\right)$, where we define $q_1(x_1 \mid x_{1:0}) := q_1(x_1)$, and $\theta^\star$ maximizes Equation (1). Then, there exist (simple) algorithms using $LLM_{\theta^\star}$ to achieve no (dynamic) regret for (non-stationary) online learning with full-information/bandit feedback. To be specific, for (2) and (4), by defining the variation bound $\sum_{t=1}^{T-1} \|\ell_{t+1} - \ell_t\|_\infty \leq V_T$ such that $V_T \leq T$ and $V_T = \Theta(T^\rho)$ for some $\rho \in (0, 1)$, it holds that for large enough T, d:*

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

(2) For non-stationary online learning with full-information feedback,

$$\text{D-Regret}_{\text{LLM}_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right) \le \mathcal{O}\left((\log d\ V_T)^{1/3}T^{2/3}\right);$$

(3) For online learning with bandit feedback,

$$\mathbb{E}\left[\text{Regret}_{\text{LLM}_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right)\right] \le \mathcal{O}\left((\log d)^{1/2} d T^{1/2+1/\log T}\log T\right);$$

(4) For non-stationary online learning with bandit feedback,

$$\mathbb{E}\left[\text{D-Regret}_{\text{LLM}_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right)\right] \le \mathcal{O}\left((T^2 d^2 V_T)^{1/3}(\log d)^{1/2} T^{1/\log T}\log T\right).$$

Remark 1 (Implication for playing repeated games)**.** *First, we note that the no-regret guarantee in the online setting is stronger than and thus implies that in the game setting, since regret by definition handles arbitrary/adversarial environments, while in playing games the opponents are not necessarily as adversarial. Second, it is folklore that if all players in the repeated game follow no-regret learning algorithms, then the time-average policies of all players during learning constitute an approximate* ***coarse correlated equilibrium*** *of the game (Cesa-Bianchi and Lugosi, 2006). Hence, the results (1) and (2) in Theorem 1 imply that a coarse correlated equilibrium will emerge in the long run from the interactions of the LLM agents (under certain assumptions as in the theorem).*

We present proofs of the non-asymptotic bounds for (dynamic) regret in Appendix C.4. Furthermore, we demonstrate that the prior distribution of z could also be replaced by a general distribution (c.f. Theorem 5), in order to obtain the above results. We also point out that Assumption 1 may be further relaxed to better match the actual LLMs' pre-training data distributions from diverse sources (c.f. Appendix C.5).

It is important to observe that even when pre-training is conducted solely with *stationary* loss vector generation ($\ell_{1:t}$ are i.i.d. conditioned on z), it can still lead to the *emergence of no-regret behavior* in online learning with potentially adversarial losses. Key in the proof is a connection of pre-trained LLM models to the online learning algorithm of FTPL.

Comparison to Lee et al. (2023); Lin et al. (2024). Intriguingly, similar assumptions and pre-training objectives have also been considered in the very recent work of Lee et al. (2023); Lin et al. (2024) for studying in-context reinforcement learning 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. (2024) generalized the study to the setting of algorithm distillation as in Laskin et al. (2023). However, their results cannot directly imply the no-regret guarantee in our online learning setting, due to the fact that posterior sampling can perform poorly under potentially *adversarial* or *non-stationary* environments (Zimmert and 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 texts (tasks).

Calibrating the degree of bounded rationality of actual LLMs. To further validate our model and data distribution assumptions, we also propose to *calibrate* the parameter $\{\eta_t\}_{t\in[T-1]}$ in Definition 2, the degree of bounded rationality, by estimating the parameters of $\{\eta_t\}_{t\in[T-1]}$ using data from interacting with LLMs (following the same protocol as before), with P_{noise} being a standard normal

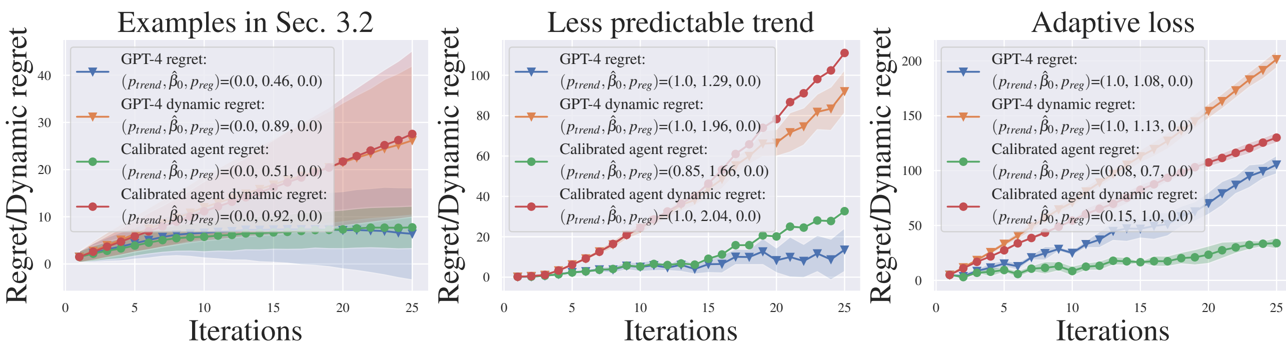

Figure 8: (left) Comparison of GPT-4 with a calibrated agent on the test set, where the calibrated quantal response can perfectly capture the behavior of the GPT-4 agent. (mid, right) The calibrated agent on the less predictable and adaptive loss sequences failed to make accurate predictions for the GPT-4 anymore.

distribution (note that we do not need to calibrate η_0 by Definition 2). Specifically, given n episodes of the LLM agent's behavior $\{(\ell_t^{(j)}, \pi_t^{(j)})_{t\in[T]}\}_{j\in[n]}$, motivated by our Lemma 1 and Theorem 1, we calibrate $\{\eta_t\}_{t\in[T-1]}$ by solving the following problem

$$\sigma^\star \in \arg\min_{\sigma>0} \sum_{t\in[T-1]}\sum_{j\in[n]} \left\|\pi_{t+1}^{(j)} - P_{quantal}^{\sigma\sqrt{t+1}}\left(\cdot \,\middle|\, \ell_{1:t}^{(j)}\right)\right\|_1, \qquad \eta_t^\star = \sigma^\star\sqrt{t+1}, \quad \forall t\in[T-1].$$

We solve this single-variable optimization problem by grid search over $[0,10]$. We then run the generalized quantal response model with the calibrated $\{\eta_t^\star\}_{t\in[T-1]}$ on another *unseen test set*, and compare it with the behavior of the actual LLM agents. We use all the interaction data from Section 3.2 and split it in half for training and testing. In Figure 8, we show the averaged regret for the LLM agent and the calibrated generalized quantal response. It can be seen that calibrated generalized quantal response can *very well capture* the behavior of the LLM agent for all problem instances in Section 3.2, justifying the applicability of our hypothetical model and assumptions.

We also use the same framework to understand the regrettable behaviors in Section 3.4. This analysis uses all the data from Section 3.4. We first find that such fitting procedures do not yield good predictions for LLMs on those counter-examples. Therefore, we resort to a more expressive model by directly fitting each η_t as $\eta_t^\star \in \arg\min_{\eta_t>0}\sum_{j\in[n]}\left\|\pi_{t+1}^{(j)} - P_{quantal}^{\eta_t}\left(\cdot \,\middle|\, \ell_{1:t}^{(j)}\right)\right\|_1$ separately for each $t\in[T-1]$. Even under the expressive model, LLMs fail to follow the generalized quantal response for the counter-examples with less predictable or adaptive loss sequences, as Figure 8 shows the gap between GPT-4 (dynamic) regret and the calibrated agent (dynamic) regret.

Finally, we acknowledge that for most existing pre-trained LLMs like GPT-4, the canonical assumptions above, though may be further relaxed (c.f. Remark 4), may not hold in general. More importantly, the *supervision labels*, i.e., the optimal action given z, may be sometimes imperfect or unavailable during the dataset collection. Hence, it is completely possible to observe regrettable 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 Provably Promoting No-Regret Behavior by an Unsupervised 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 learning* loss that naturally provides no-regret behaviors. This approach is akin to the process of "instruction tuning" (Wei et al., 2021), which was shown to have enhanced LLMs' ability when learning from context, with both theoretical (Ahn et al., 2023; Mahankali et al., 2023; Zhang et al., 2023a) and empirical (Lu et al., 2023) evidence.

5.1 A New Unsupervised Training Loss: Regret-Loss

Intuitively, our new training loss is designed to enforce the trained LLM to minimize the regret under an arbitrary sequence of loss vectors. Specifically, we define the training loss as

$$\mathcal{L}(\theta) := \max_{\ell_1,\dots,\ell_T} \quad \text{Regret}_{\text{LLM}_\theta}\Big((\ell_t)_{t\in[T]}\Big) \tag{2}$$

where $\|\ell_t\|_\infty \leq B$ for $t \in [T]$. As discussed in Kirschner et al. (2023), directly minimizing the max regret can be computationally challenging, except for superficially simple problems. Hence, in practice, one may parameterize the LLM and resort to differentiable programming to solve it approximately. However, Equation (2) 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 derivatives can be challenging since we need to calculate $\arg\max_{(\ell_t)_{t\in[T]}} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$ while we have inf in the definition of regret. Therefore, we provide a general framework so that we can approximate Equation (2) by the following surrogate:

$$\mathcal{L}(\theta,k,N) := \mathbb{E}\left[\frac{\sum_{j\in[N]} h(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]} f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}\right], \tag{3}$$

where $k \in \mathbb{N}^+$, $N \in \mathbb{N}^+$, $h : \mathbb{R} \to \mathbb{R}^+$ is a continuous function, with continuous derivative h', and $f(\cdot,k) : \mathbb{R} \to \mathbb{R}^+$ is a continuous function for each $k \in \mathbb{N}^+$, satisfying $\lim_{k\to\infty} \frac{f(R_1,k)}{f(R_2,k)} = \infty \cdot \mathbb{1}(R_1 > R_2) + \mathbb{1}(R_1 = R_2)$, where we use the convention of $\infty \cdot 0 = 0$. These conditions on h, f will be assumed throughout the paper. 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 (3) is thus taken with respect to this distribution. Note that we do not have any statistical assumption on $(\ell_t^{(j)})_{t\in[T],j\in[N]}$ (except being continuous and supported on $[-B,B]^{T\times N}$), in contrast to those in Section 4 to *justify* the no-regret property of pre-trained LLMs.

In Appendix D.2, we prove that under certain regularity conditions of f and h, we have

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

as well as the uniform convergence of $\mathcal{L}(\theta,k,N)$: $\lim_{N,k\to\infty} \sup_{\theta\in\Theta} \left| h(\max_{\ell_1,\dots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta,k,N)\right| = 0$, where Θ is some compact set of the LLM parameter. Hence, one can expect that minimizing the loss function in Equation (3) with large enough k and N may promote the trained LLM to have a small regret value. We will hereafter refer to Equation (3) as the *regret-loss*. Similarly, we can also define *dynamic-regret-loss*, and the results to be presented next can also generalize to this case (c.f. Remark 5 in Appendix D.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 as used in GPT-4 and most existing LLMs (see Proposition 2 for an example with formal statement). This guarantee focuses on their *generalization ability* when trained to minimize the empirical regret loss, which is defined as follows:

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

$$\widehat{\mathcal{L}}(\theta, k, N, N_T) := \frac{1}{N_T}\sum_{s=1}^{N_T}\left[\frac{\sum_{j\in[N]} h\left(\text{Regret}_{\text{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]})\right) f\left(\text{Regret}_{\text{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]}), k\right)}{\sum_{j\in[N]} f\left(\text{Regret}_{\text{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]}), k\right)}\right] \tag{4}$$

where $(\ell_{s,t}^{(j)})_{j\in[N],t\in[T]}$ denotes the s-th sample of $(\ell_t^{(j)})_{j\in[N],t\in[T]}$ for estimating $\mathcal{L}(\theta, k, N)$.

We denote $\widehat{\theta}_{k,N,N_T} \in \arg\min_{\theta\in\Theta} \widehat{\mathcal{L}}(\theta, k, N, N_T)$, and present the generalization guarantee below.

Theorem 2. (Generalization gap). *Suppose LLM_θ is Lipschitz-continuous with respect to the model parameter θ, then for any $0 < \epsilon < 1/2$, with probability at least $1-\epsilon$, we have*

$$\mathcal{L}\left(\widehat{\theta}_{k,N,N_T}, k, N\right) - \inf_{\theta\in\Theta}\mathcal{L}(\theta, k, N) \le \widetilde{\mathcal{O}}\left(\sqrt{\frac{d_\theta + \log(1/\epsilon)}{N_T}}\right), \tag{5}$$

for any N and sufficiently large k, where d_θ is the dimension of the parameter θ.

Through a careful use of Berge's Maximum Theorem (Berge, 1877), we prove that the right-hand side of Equation (5) 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 D.2), we further obtain the following corollary on the regret guarantee:

Corollary 1. (Regret). *Suppose*[♮] *for any $k \in \mathbb{N}^+$, $h, f(\cdot, k)$ are non-decreasing, and $\log f$ is a supermodular function (i.e., $\log f(R_1, k_1) - \log f(R_1, k_2) \ge \log f(R_2, k_1) - \log f(R_2, k_2)$ for $R_1 \ge R_2$ and $k_1 \ge k_2$). Then, with high probably, we have*

$$h\left(\lim_{N\to\infty}\lim_{k\to\infty}\max_{\|\ell_t\|_\infty\le B}\text{Regret}_{\text{LLM}_{\widehat{\theta}_{k,N,N_T}}}\left((\ell_t)_{t\in[T]}\right)\right) \le h\left(\inf_{\theta\in\Theta}\max_{\|\ell_t\|_\infty\le B}\text{Regret}_{\text{LLM}_\theta}\left((\ell_t)_{t\in[T]}\right)\right) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{d_\theta}{N_T}}\right). \tag{6}$$

Corollary 2. (Emerging behavior: Coarse correlated equilibrium). *For a sufficiently large N_T, if each agent in the matrix game plays according to $\text{LLM}_{\widehat{\theta}_{k,N,N_T}}$, then the time-averaged policy for each agent will constitute an approximate coarse correlated equilibrium of the game.*

Proofs of Theorem 2 and Corollary 1 are deferred to Appendix D.3, and Corollary 2 follows from the folklore result discussed in Remark 1. 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 next in Section 5.3), then Corollary 1 means that with a large enough number of samples N_T, the learned $\text{LLM}_{\widehat{\theta}_{k,N,N_T}}$ becomes a *no-regret* learner, i.e., $\text{Regret}_{\text{LLM}_{\widehat{\theta}_{k,N,N_T}}}\left((\ell_t)_{t\in[T]}\right) = o(T)$, since the first term on the right-hand-side of Equation (6) would

[♮] Note that these conditions on h, f are in addition to those specified after Equation (3).

directly be $o(T)$ under the choice of $h(x) = \max\{0, x\}$. For other choices of h, one can use the inverse function of h, i.e., 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 and generality of the 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 the specific parameterization 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 (3) 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., 2023a; Mahankali et al., 2023), and the more practical setting with a finite $N = 1$. Note that in this case, the choice of f (and thus k) is not relevant. Thus, throughout this section, we drop superscript (j) in Equation (3) for notational convenience. We sample ℓ_t for $t \in [T]$ as realizations of some random variable Z. Here, we assume Z is symmetric about zero (i.e., $Z \stackrel{d}{=} -Z$), $\text{Var}(Z) = \Sigma$ is positive definite. We also assume that the support of Z has an interior such that it contains a ball centered at $\mathbf{0}_d$.

Firstly, we consider the following structure of single-layer self-attention model g (see a formal introduction in Appendix A.1):

$$g(Z_t; V, K, Q, v_c, k_c, q_c) := (V\ell_{1:t} + v_c\mathbf{1}_t^\intercal)\texttt{Softmax}\Big((K\ell_{1:t} + k_c\mathbf{1}_t^\intercal)^\intercal \cdot (Qc + q_c)\Big), \tag{7}$$

where $Z_t = (\ell_1, \ldots, \ell_t, c)$, $\ell_{1:t} \in \mathbb{R}^{d\times t}$ denotes a matrix with each column corresponding to each ℓ_i, 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 associated with V, K, Q, and $c \neq \mathbf{0}_d$ is a constant vector. We then have the following result.

Theorem 3. *Consider the policy space $\Pi = B(0, R_\Pi, \|\cdot\|)$ for some $R_\Pi > 0$. The configuration of a single-layer self-attention model in Equation (7) (V, K, Q, v_c, k_c, q_c) such that $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ and*

$$V = -R_\Pi \frac{T}{\sum_{t=1}^{T-1} 1/t} \Sigma^{-1} \mathbb{E}\left[\left\|\sum_{t=1}^{T} \ell_t\right\| \ell_1 \ell_2^\intercal\right] \Sigma^{-1}$$

is a first-order stationary point of Equation (3) with $N = 1$, $h(x) = x^2$. Moreover, if Σ is a diagonal matrix, then plugging this configuration into Equation (7), and projecting the output with $\mathtt{Proj}_{\Pi,\|\cdot\|}$ would perform FTRL with an L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.

In practical training, such stationary points of the loss may be attained by first-order optimization algorithms of (stochastic) gradient descent, the workhorse in machine learning. Moreover, we also consider the single-layer *linear* self-attention model as follows, for which we can strengthen the results above from a stationary-point to an *optimal-solution* argument:

$$g(Z_t; V, K, Q, v_c, k_c, q_c) = \sum_{i=1}^{t} (V\ell_i + v_c)\big((K\ell_i + k_c)^\intercal \cdot (Qc + q_c)\big). \tag{8}$$

Theorem 4. *Consider the policy space $\Pi = B(0, R_\Pi, \|\cdot\|)$ for some $R_\Pi > 0$. The configuration of a single-layer linear self-attention model in Equation (8) (V, K, Q, v_c, k_c, q_c) such that $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ and $V = -2R_\Pi \Sigma^{-1} \mathbb{E}\big(\|\sum_{t=1}^{T} \ell_t\| \ell_1 \ell_2^\intercal\big) \Sigma^{-1}$ is **a global optimal solution** of Equation (3) with $N = 1$, $h(x) =$*

x^2. Moreover, every global optimal configuration of Equation (3) within the parameterization class of Equation (8) has the same output function g. Additionally, if Σ is a diagonal matrix, then plugging any global optimal configuration into Equation (8), and projecting the output with $Proj_{\Pi,\|\cdot\|}$ would perform FTRL with an L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.

Theorem 4 shows the capacity of self-attention Transformer models 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 and Schapire, 1997) that is more compatible with the simplex constraint on π in the Experts Problem. We defer the discussion of this case to Appendix D.7. Through these results, we can also guarantee in the repeated game setting that approximate **coarse correlated equilibria** would emerge in the long run, since each player will exhibit no-regret behavior, using a similar argument as that for Corollary 2.

Remark 2. *The very recent studies by (Ahn et al., 2023; Zhang et al., 2023a; Mahankali et al., 2023) have demonstrated 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 linear self-attention model **implicitly** performs gradient descent. However, in the online learning setting where there are no y-labels, such an implicit gradient descent update-rule is hard to define. Compared to the previous studies, our global optimizer among single-layer linear self-attention models is an* explicit *and* online *gradient descent update for online learning. With a different loss (regret-loss v.s. instruction-tuning-loss), the techniques to obtain the seemingly similar results are also fundamentally different.*

5.4 Experimental Results for Minimizing Regret-Loss

We now provide experimental results for minimizing our *regret-loss*, and evaluate in the following environments: 1) randomly-generated loss sequences (Figure 9); 2) loss sequences with a predictable trend (Figure 10); 3) repeated games (Figure 11); and 4) counterexamples for pre-trained LLMs to be regrettable (Figure 7). Details of the training setup can be found in Appendix D.8. We also provide an ablation study for the training of the loss Equation (3) in Appendix D.9.

Randomly generated loss sequences. We use the same loss vectors as those in Section 3.2 for randomly generated loss functions, and compare the results with that using GPT-4. The results show that with regret-loss, both the trained single-layer self-attention model and the trained Transformers with multi-layer self-attention structures can achieve comparable regrets as FTRL and GPT-4. The results can be found in Figure 9.

Loss sequences with a predictable trend. We investigate the case where the loss sequences have predictable trends such as linear-trend or sine-trend. One might expect that the performance of the trained Transformer would surpass the performance of traditional no-regret learning algorithms such as FTRL, since they may not be an optimal algorithm for the loss sequence with a predictable trend. We modify the training distribution by changing the distribution of random variable Z (which generates the loss vectors ℓ_t) to follow two kinds of trends: linear and sine functions. The results, as illustrated in Figure 10, show that the trained single-layer self-attention model and the trained Transformer with multi-layer self-attention structures with regret-loss outperformed GPT-4 and FTRL in terms of regret, when the loss sequence is a linear trend. Similarly, Figure 10 shows that

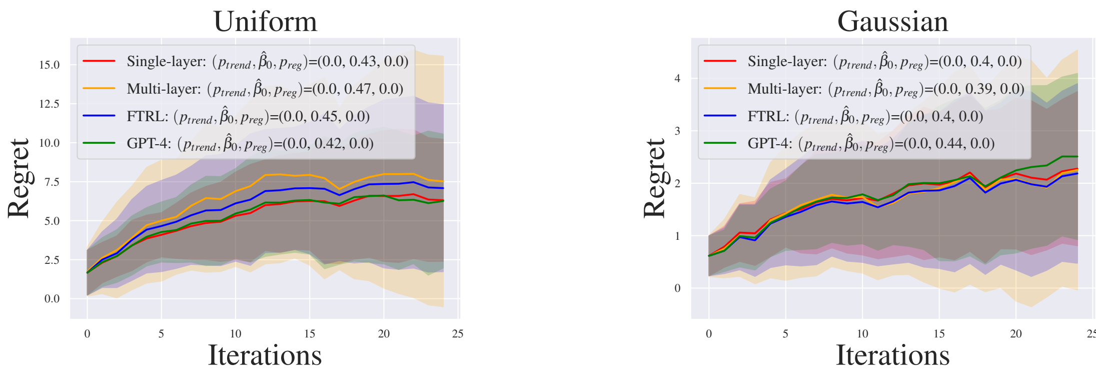

Figure 9: Regret performance for the randomly generated loss sequences that are generated by Gaussian with truncation and uniform distribution. No-regret behaviors of single-layer and multi-layer self-attention models are validated by both of our frameworks (low p-values and $\widehat{\beta_0} < 1$).

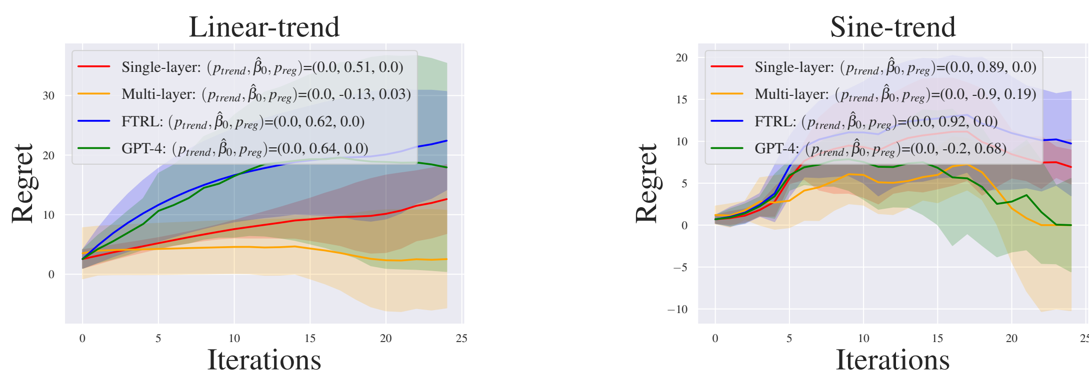

Figure 10: Regret performance for the randomly generated loss sequences that are generated by linear-trend and sine-trend. No-regret behaviors of single-layer and multi-layer self-attention models are validated by both of our frameworks (low p-values and $\widehat{\beta_0} < 1$).

the trained Transformer with multi-layer self-attention structures with regret-loss is comparable to GPT-4 and outperformed FTRL in terms of regret, when the loss sequence is a sine-trend. Note that the training dataset does not contain the sequence of losses. Nonetheless, by focusing on the overall trend during training, we can attain performance that is either superior to or on par with that of FTRL and GPT-4.

Repeated games. We then investigate the case of multi-player repeated games. We study 2x2, 3x3x3, 3x3x3x3 games, where each entry of the payoff matrix is sampled randomly from Unif([0, 10]). The results, as illustrated in Figure 11, show that the trained single-layer self-attention model and the trained Transformer with multi-layer self-attention structures with regret-loss have a similar performance as that of FTRL. However, GPT-4 still outperforms the trained single-layer self-attention model and the trained Transformer with multi-layer self-attention structures in terms of regret. Since for repeated games (in which the environment faced by the agent can be less adversarial than that in the online setting), there might be a better algorithm than FTRL (see e.g., Daskalakis et al. (2021)), while our self-attention models have a similar structure as FTRL (Theorem 3 or Theorem 4). Also, in practical training (with the empirical loss in Equation (4)), we possibly did not find the exact global minimum or stationary point of the *expected* loss in Equation (3). Hence, it is possible that GPT-4 may have lower regret than our trained models with the regret-loss.

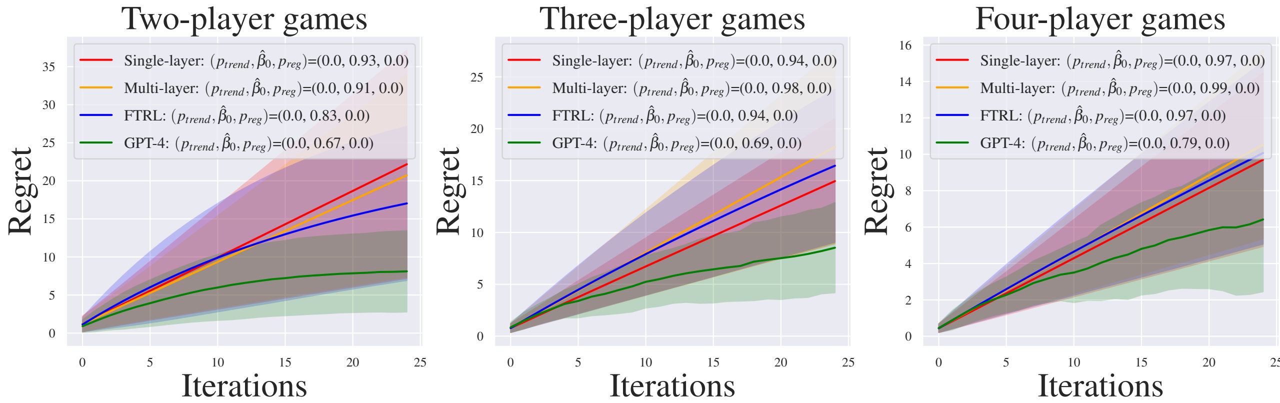

Figure 11: Regret performance for the game with two players, three players, and four players general-sum games. No-regret behaviors of single-layer and multi-layer self-attention models are validated by both of our frameworks (low p-values and $\widehat{\beta_0} < 1$).

Two scenarios that caused regrettable behaviors of GPT-4. Finally, we investigate the cases that have caused GPT-4 to have regrettable performance in Section 3.2. The results, which can be found in Figure 7, show that both the trained single-layer self-attention model and the trained Transformer with regret-loss can achieve comparable no-regret performance as FTRL, and outperforms that of GPT-4. This validates that our new unsupervised training loss can address the regrettable cases, as our theory in Sections 5.2 and 5.3 has predicted.

6 Concluding Remarks

In this paper, we studied the online decision-making and strategic behaviors of LLMs quantitatively, through the metric of regret. We first examined and validated the no-regret behavior of several representative pre-trained LLMs in benchmark settings of online learning and games. As a consequence, (coarse correlated) equilibrium can oftentimes emerge as the long-term outcome of multiple LLMs playing repeated games. We then provide some theoretical insights into the no-regret behavior, by connecting pre-trained LLMs to the follow-the-perturbed-leader algorithm in online learning, under certain assumptions. We also identified (simple) cases where pre-trained LLMs fail to be no-regret, and thus proposed a new unsupervised training loss, *regret-loss*, to provably promote the no-regret behavior of Transformers without the labels of (optimal) actions. We established both experimental and theoretical evidence for the effectiveness of our regret-loss.

As a first attempt toward rigorously understanding the online and strategic decision-making behaviors of LLMs through the metric of regret, our work has opened up fruitful directions for future research:

- There are more than one definitions of (dynamic-)regret in the online learning literature, and we mainly focused on the so-called *external-regret* in the literature. It would be interesting to study the no-regret behavior of LLMs in terms of other regret metrics, e.g., swap-regret (Blum and Mansour, 2007), which may lead to stronger equilibrium notions in playing repeated games.

- Our new regret-loss has exhibited promises in our experiments for training modest-scale Transformers. We are currently generalizing it to training other larger-scale models, such as Foundation Models, for decision-making.

- No-regret behavior can sometimes lead to better outcomes in terms of social efficiency (Blum et al., 2008; Roughgarden, 2015; Nekipelov et al., 2015). It would thus be interesting to further validate the efficiency of no-regret LLM agents in these scenarios, as well as identifying new prompts and training losses for LLMs to promote the efficiency of the outcomes.
- To evaluate the performance quantitatively, we focused on online learning and games with *numeric valued* payoffs. It would be interesting to connect our no-regret-based and game-theoretic framework with existing multi-LLM frameworks, e.g., debate, collaborative problem-solving, and human/social behavior simulation, with potentially new notions of regret (defined in different spaces) as performance metrics.

Acknowledgement

The authors thank Dingwen Kong for discussing the truncation idea in proving Lemma 8. Also, the authors thank Kristian Georgiev and Aleksander Madry for the helpful feedback.

References

Abernethy, J., Lee, C., Sinha, A. and Tewari, A. (2014). Online linear optimization via smoothing. In *Conference on Learning Theory*. PMLR.

Abernethy, J. D., Lee, C. and Tewari, A. (2015). Fighting bandits with a new kind of smoothness. *Advances in Neural Information Processing Systems*, **28**.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aher, G. V., Arriaga, R. I. and Kalai, A. T. (2023). Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*. PMLR.

Ahn, K., Cheng, X., Daneshmand, H. and Sra, S. (2023). Transformers learn to implement preconditioned gradient descent for in-context learning. *Advanced in Neural Information Processing Systems*.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K. et al. (2022). Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Ahsanullah, M., Nevzorov, V. B. and Shakil, M. (2013). *An introduction to order statistics*, vol. 8. Springer.

Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M. and Schulz, E. (2023). Playing repeated games with large language models. *arXiv preprint arXiv:2305.16867*.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T. and Zhou, D. (2023). What learning algorithm is in-context learning? investigations with linear models. *International Conference on Learning Representations*.

Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J. R., Rytting, C. and Wingate, D. (2023). Out of one, many: Using language models to simulate human samples. *Political Analysis*, **31** 337–351.

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

Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, **32** 48–77.

Bai, Y., Chen, F., Wang, H., Xiong, C. and Mei, S. (2023). Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advanced in Neural Information Processing Systems*.

Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H. et al. (2022). Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, **378** 1067–1074.

Balseiro, S. R. and Gur, Y. (2019). Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, **65** 3952–3968.

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

Besbes, O., Gur, Y. and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, **27**.

Blum, A., Hajiaghayi, M., Ligett, K. and Roth, A. (2008). Regret minimization and the price of total anarchy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*.

Blum, A. and Mansour, Y. (2007). From external to internal regret. *Journal of Machine Learning Research*, **8**.

Brookins, P. and DeBacker, J. M. (2023). Playing games with GPT: What can we learn about a large language model from canonical strategic games? *Available at SSRN 4493398*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, **33** 1877–1901.

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

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S. et al. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

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

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

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J. and Liu, Z. (2024). Chateval: Towards better llm-based evaluators through multi-agent debate. *International Conference on Learning Representations*.

Chen, W., Su, Y., Zuo, J., Yang, C., Yuan, C., Qian, C., Chan, C.-M., Qin, Y., Lu, Y., Xie, R. et al. (2024). Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *International Conference on Learning Representations*.

Chen, Y., Liu, T. X., Shan, Y. and Zhong, S. (2023). The emergence of economic rationality of gpt. *Proceedings of the National Academy of Sciences*, **120** e2316205120.

Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z. and Wei, F. (2023). Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023* (A. Rogers, J. Boyd-Graber and N. Okazaki, eds.). Association for Computational Linguistics, Toronto, Canada.
https://aclanthology.org/2023.findings-acl.247

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

Daskalakis, C., Fishelson, M. and Golowich, N. (2021). Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, **34** 27604–27616.

Ding, J., Feng, Y. and Rong, Y. (2022). Myopic quantal response policy: Thompson sampling meets behavioral economics. *arXiv preprint arXiv:2207.01028*.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T. et al. (2023). Palm-e: An embodied multimodal language model. *International Conference on Machine Learning*.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B. and Mordatch, I. (2023). Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Engel, C., Grossmann, M. R. and Ockenfels, A. (2023). Integrating machine behavior into human subject experiments: A user-friendly toolkit and illustrations. *Available at SSRN*.

Erev, I. and Roth, A. E. (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review* 848–881.

Fan, C., Chen, J., Jin, Y. and He, H. (2023). Can large language models serve as rational players in game theory? a systematic analysis. *arXiv preprint arXiv:2312.05488*.

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

Fu, Y., Peng, H., Khot, T. and Lapata, M. (2023). Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*.

Fudenberg, D. and Kreps, D. M. (1993). Learning mixed equilibria. *Games and Economic Behavior*, **5** 320–367.

Fudenberg, D. and Levine, D. K. (1998). *The theory of learning in games*, vol. 2. MIT Press.

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

Garg, S., Tsipras, D., Liang, P. S. and Valiant, G. (2022). What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, **35** 30583–30598.

Giannou, A., Rajput, S., Sohn, J.-y., Lee, K., Lee, J. D. and Papailiopoulos, D. (2023). Looped transformers as programmable computers. *International Conference on Machine Learning*.

Hao, S., Gu, Y., Ma, H., Hong, J., Wang, Z., Wang, D. and Hu, Z. (2023). Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino and K. Bali, eds.). Association for Computational Linguistics, Singapore.
https://aclanthology.org/2023.emnlp-main.507

Hazan, E. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, **2** 157–325.

Hofbauer, J. and Sandholm, W. H. (2002). On the global convergence of stochastic fictitious play. *Econometrica*, **70** 2265–2294.

Hong, S., Zheng, X., Chen, J., Cheng, Y., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., Ran, C. et al. (2024). Metagpt: Meta programming for multi-agent collaborative framework. *nternational Conference on Learning Representations*.

Horton, J. J. (2023). Large language models as simulated economic agents: What can we learn from homo silicus? Tech. rep., National Bureau of Economic Research.

Huang, W., Abbeel, P., Pathak, D. and Mordatch, I. (2022a). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*. PMLR.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y. et al. (2022b). Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

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

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

Kasprzak, M. J., Giordano, R. and Broderick, T. (2022). 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*.

Kirschner, J., Bakhtiari, A., Chandak, K., Tkachuk, V. and Szepesvari, C. (2023). Regret minimization via saddle point optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D., Hansen, S., Filos, A., Brooks, E. et al. (2023). In-context reinforcement learning with algorithm distillation. *International Conference on Learning Representations*.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lee, J. N., Xie, A., Pacchiano, A., Chandak, Y., Finn, C., Nachum, O. and Brunskill, E. (2023). Supervised pretraining can learn in-context reinforcement learning. *Neural Information Processing Systems*.

Li, C., Su, X., Fan, C., Han, H., Xue, C. and Zheng, C. (2023a). Quantifying the impact of large language models on collective opinion dynamics. *arXiv preprint arXiv:2308.03313*.

Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D. and Ghanem, B. (2023b). Camel: Communicative agents for" mind" exploration of large scale language model society. *Neural Information Processing Systems*.

Li, R., Patel, T. and Du, X. (2023c). Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Li, S., Yang, J. and Zhao, K. (2023d). 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*.

Li, Y., Ildiz, M. E., Papailiopoulos, D. and Oymak, S. (2023e). Transformers as algorithms: Generalization and stability in in-context learning. *International Conference on Machine Learning*.

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

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z. and Shi, S. (2023). Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Lin, L., Bai, Y. and Mei, S. (2024). Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *International Conference on Learning Representations*.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, **108** 212–261.

Liu, H., Sferrazza, C. and Abbeel, P. (2023a). Chain of hindsight aligns language models with feedback. *arXiv preprint arXiv:2302.02676*, **3**.

Liu, Y., Van Roy, B. and Xu, K. (2023b). Nonstationary bandit learning via predictive sampling. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Liu, Z., Hu, H., Zhang, S., Guo, H., Ke, S., Liu, B. and Wang, Z. (2023c). Reason for future, act for now: A principled architecture for autonomous llm agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Lorè, N. and Heydari, B. (2023). Strategic behavior of large language models: Game structure vs. contextual framing. *arXiv preprint arXiv:2309.05898*.

Lu, S., Bigoulaeva, I., Sachdeva, R., Madabushi, H. T. and Gurevych, I. (2023). Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*.

Mahankali, A., Hashimoto, T. B. and Ma, T. (2023). One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *International Conference on Learning Representations*.

Mao, W., Zhang, K., Zhu, R., Simchi-Levi, D. and Başar, T. (2020). Model-free non-stationary RL: Near-optimal regret and applications in multi-agent RL and inventory control. *arXiv preprint arXiv:2010.03161*.

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

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

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H. and Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Y. Goldberg, Z. Kozareva and Y. Zhang, eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates. https://aclanthology.org/2022.emnlp-main.759

Mukobi, G., Erlebach, H., Lauffer, N., Hammond, L., Chan, A. and Clifton, J. (2023). Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*.

Nekipelov, D., Syrgkanis, V. and Tardos, E. (2015). Econometrics for learning agents. In *ACM Conference on Economics and Computation*.

Openai (2023). Gpt-4 technical report.

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

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P. and Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST '23, Association for Computing Machinery, New York, NY, USA.
https://doi.org/10.1145/3586183.3606763

Park, J. S., Popowski, L., Cai, C., Morris, M. R., Liang, P. and Bernstein, M. S. (2022). Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*.

Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., Liu, Z. and Sun, M. (2023). Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M. and de Freitas, N. (2022). A generalist agent. *Transactions on Machine Learning Research*. Featured Certification, Outstanding Certification.
https://openreview.net/forum?id=1ikK0kHjvj

Robinson, D. and Goforth, D. (2005). *The topology of the 2x2 games: a new periodic table*, vol. 3. Psychology Press.

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

Roughgarden, T., Syrgkanis, V. and Tardos, E. (2017). The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, **59** 59–101.

Schick, T., Dwivedi-Yu, J., Jiang, Z., Petroni, F., Lewis, P., Izacard, G., You, Q., Nalmpantis, C., Grave, E. and Riedel, S. (2023). Peer: A collaborative language model. *International Conference on Learning Representations*.

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

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

Shalev-Shwartz, S. and Singer, Y. (2007). A primal-dual perspective of online learning algorithms. *Machine Learning*, **69** 115–142.

Shen, Y., Song, K., Tan, X., Li, D., Lu, W. and Zhuang, Y. (2023). Hugginggpt: Solving AI tasks with chatgpt and its friends in huggingface. *Neural Information Processing Systems*.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K. R. and Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Significant Gravitas (2023). Autogpt.
https://github.com/Significant-Gravitas/AutoGPT

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A. et al. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Swan, M., Kido, T., Roland, E. and Santos, R. P. d. (2023). Math agents: Computational infrastructure, mathematical embedding, and genomics. *arXiv preprint arXiv:2307.02502*.

Tsai, C. F., Zhou, X., Liu, S. S., Li, J., Yu, M. and Mei, H. (2023). Can large language models play text games well? current state-of-the-art and open questions. *arXiv preprint arXiv:2304.02868*.

Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S. and Kambhampati, S. (2023). 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*.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, vol. 3. Cambridge university press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A. and Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*. PMLR.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48. Cambridge university press.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L. and Anandkumar, A. (2023a). Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wang, X., Zhu, W. and Wang, W. Y. (2023b). Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *International Conference on Machine Learning 2023 Workshop ES-FoMO*.

Wang, Z., Cai, S., Liu, A., Ma, X. and Liang, Y. (2023c). Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *Advances in neural information processing systems*.

Wei, C.-Y. and Luo, H. (2021). Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. In *Conference on learning theory*. PMLR.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M. and Le, Q. V. (2021). Finetuned language models are zero-shot learners. *International Conference on Learning Representations*.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D. et al. (2022a). Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D. et al. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, **35** 24824–24837.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X. and Wang, C. (2023). Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Xie, S. M., Raghunathan, A., Liang, P. and Ma, T. (2022). An explanation of in-context learning as implicit bayesian inference. *International Conference on Learning Representations*.

Xiong, K., Ding, X., Cao, Y., Liu, T. and Qin, B. (2023). Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (H. Bouamor, J. Pino and K. Bali, eds.). Association for Computational Linguistics, Singapore.
https://aclanthology.org/2023.findings-emnlp.508

Xu, Y., Wang, S., Li, P., Luo, F., Wang, X., Liu, W. and Liu, Y. (2023a). Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*.

Xu, Z., Yu, C., Fang, F., Wang, Y. and Wu, Y. (2023b). Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y. and Narasimhan, K. (2023a). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y. (2023b). React: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*.

Young, H. P. (2004). *Strategic learning and its limits*. OUP Oxford.

Zhang, H., Du, W., Shan, J., Zhou, Q., Du, Y., Tenenbaum, J. B., Shu, T. and Gan, C. (2024). Building cooperative embodied agents modularly with large language models. *International Conference on Learning Representations*.

Zhang, R., Frei, S. and Bartlett, P. L. (2023a). Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*.

Zhang, Y., Zhang, F., Yang, Z. and Wang, Z. (2023b). What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*.

Zhao, Q., Wang, J., Zhang, Y., Jin, Y., Zhu, K., Chen, H. and Xie, X. (2023). Competeai: Understanding the competition behaviors in large language model-based agents. *arXiv preprint arXiv:2310.17512*.

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

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*.

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

Contents

1 Introduction **1**
1.1 Related Work . . . 2

2 Preliminaries **5**
2.1 Online Learning & Games . . . 5
2.2 Performance Metric: Regret . . . 6

3 Do Pre-Trained LLMs Have Regret? Experimental Validation **6**
3.1 Framework for No-Regret Behavior Validation . . . 7
3.2 Results: Online Learning . . . 8
3.3 Results: Multi-Player Repeated Games . . . 9
3.4 Pre-Trained LLM Agents May Still Have Regret . . . 10

4 Why Are Pre-Trained LLMs (No-)Regret? A Hypothetical Model and Some Theoretical Insights **12**
4.1 Pre-Trained LLMs Have Similar Regret as Humans (Who Generate Data) . . . 13
4.2 A Human Decision-Making Model: Quantal Response . . . 14
4.3 Case Study: Pre-Training under Canonical Data Distribution . . . 15

5 Provably Promoting No-Regret Behavior by an Unsupervised Loss **18**
5.1 A New Unsupervised Training Loss: Regret-Loss . . . 19
5.2 Guarantees via Regret-Loss Minimization . . . 20
5.3 Minimizing Regret-Loss Can Automatically Produce Known Online Learning Algorithms . . . 21
5.4 Experimental Results for Minimizing Regret-Loss . . . 22

6 Concluding Remarks **24**

A Deferred Background **36**
A.1 Additional Definitions for Appendix . . . 36
A.2 In-Context Learning . . . 36
A.3 Online Learning Algorithms . . . 37
A.4 Why Focusing on Linear Loss Function? . . . 38
A.5 Six Representative General-Sum Games . . . 38

B Deferred Results and Proofs in Section 3 **39**
B.1 Ablation Study on Prompts . . . 39
B.2 Proof for Proposition 1 . . . 42
B.3 Results of GPT-4 Turbo . . . 43
B.4 LLM Agents' Explanation on Their Output Policies . . . 43

C Deferred Results and Proofs in Section 4 44
C.1 Deferred Proof of Observation 1 . . . 44
C.2 Deferred Proof of Lemma 1 . . . 45
C.3 Relationship between FTPL and Definition 2 . . . 46
C.4 Deferred Proof of Theorem 1 . . . 47
C.5 Extending Theorem 1 with Relaxed Assumptions . . . 51
C.5.1 Relaxation under More General Data Distributions . . . 51
C.5.2 Relaxation under Decision-Irrelevant Pre-Training Data . . . 52

D Deferred Results and Proofs in Section 5 53
D.1 Basic Lemmas . . . 53
D.2 Deferred Proof for the Arguments in Section 5.1 . . . 53
D.3 Deferred Proofs of Theorem 2 and Corollary 1 . . . 59
D.4 Deferred Proof of Theorem 3 . . . 62
D.5 Deferred Proof of Theorem 4 . . . 66
D.6 Empirical Validation of Theorem 3 and Theorem 4 . . . 71
D.6.1 Empirical Validation of Theorem 3 . . . 71
D.6.2 Empirical Validation of Theorem 4 . . . 72
D.7 Discussions on the Production of FTRL with Entropy Regularization . . . 72
D.7.1 Numerical Analysis of Step 2 and Step 4 . . . 77
D.7.2 Empirical Validation . . . 77
D.8 Training Details of Section 5.4 . . . 77
D.9 Ablation Study on Training Equation (3) . . . 77

A Deferred Background

A.1 Additional Definitions 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 $\widehat{z}_i = (V z_{1:i-1})\sigma((K z_{1:i-1})^\intercal (Q z_i))$, where $z_{1:i-1}$ denotes the 1 to $i-1$ columns of Z, σ is either the $\mathtt{Softmax}$ or $\mathtt{ReLU}$ 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 $\widehat{z}_i = (V z_{1:i-1})((K z_{1:i-1})^\intercal (Q z_i))$. We write this (linear) self-attention layer's output as $\mathtt{(L)SA}_{(V,Q,K)}(Z)$. We define an M-head self-attention layer with $\theta = \{(V_m, Q_m, K_m)\}_{m \in [M]}$ as $\mathtt{M\text{-}(L)SA}_\theta(Z) := \sum_{m=1}^M \mathtt{(L)SA}_{(V_m, Q_m, K_m)}(Z)$. We define $\|\cdot\|_{\mathtt{M\text{-}(L)SA}}$ as $\|\theta\|_{\mathtt{M\text{-}(L)SA}} := \max_{m \in [M]} \left\{\|Q_m\|_{\mathrm{op}}, \|K_m\|_{\mathrm{op}}\right\} + \sum_{m=1}^M \|V_m\|_{\mathrm{op}}$.

Transformers. For a multi-layer 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 $\mathtt{Softmax}$ or $\mathtt{ReLU}$. We write the output of an MLP layer with parameter θ as $\mathtt{MLP}_\theta(Z)$. Defining $\|\cdot\|_{\mathtt{MLP}}$ as $\|\theta\|_{\mathtt{MLP}} := \|W_1\|_{\mathrm{op}} + \|W_2\|_{\mathrm{op}}$ and $\mathtt{ResNet}(f, Z) := Z + f(Z)$, we can define an L-layer Transformer with parameter $\theta = (\theta^{(lm)}, \theta^{(la)})_{l \in [L]}$ as

$$\mathtt{TF}_\theta(Z) := Z^{(L)},$$

where the output $Z^{(L)}$ is defined iteratively from $Z^{(0)} = \mathtt{clip}_R(Z) := \min(-R, \max(R, Z))$ and

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

for some $R > 0$. We define a class of Transformers with certain parameters as $\Theta_{d,L,M,d',B_{\mathtt{TF}}} := \{\theta = (\theta^{(lm)}, \theta^{(la)})_{l \in [L], m \in [M]} : \|\theta\|_{\mathtt{TF}} \leq B_{\mathtt{TF}}\}$, where M is the number of heads of self-attention,

$$\|\theta\|_{\mathtt{TF}} := \max_{l \in [L]} \left\{\|\theta^{(la)}\|_{\mathtt{M\text{-}(L)SA}} + \|\theta^{(lm)}\|_{\mathtt{MLP}}\right\}, \tag{9}$$

and $B_{\mathtt{TF}} > 0$ is some constant. When it is clear from the context, we may omit the subscripts and write it as Θ for simplicity. We assume R to be sufficiently large such that $\mathtt{clip}$ does not take effect on any of our approximation results.

A.2 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 $\mathrm{LLM}((z_t)_{t \in [T]}, x_{T+1})$ provides the inference of y_{T+1}, where $z_t = (x_t, y_t)$.

A.3 Online Learning Algorithms

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 (Freund and Schapire, 1997).

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 square, 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:

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

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 (Freund and Schapire, 1997) (also referred to as the 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 $R(\pi) = \sum_{j \in [d]} \pi_j \log \pi_j$ (where 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 $j \in [d]$.

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 and Vempala, 2005) updates the policy π at each time step t by incorporating a perturbation vector ϵ_t. This perturbation is *sampled* from a pre-defined distribution. The policy π_t for the next time step is chosen by solving the following optimization problem:

$$\pi_t = \mathbb{E}\left[\arg\min_{\pi\in\Pi}\langle \epsilon_t, \pi\rangle + \sum_{i=1}^{t-1}\langle \ell_i, \pi\rangle\right]. \tag{10}$$

Here ϵ_t introduces randomness to the decision-making.

Relationship between FTRL and FTPL. The FTRL and FTPL algorithms are deeply related. For example, FTPL with perturbations of Gumbel distribution and FTRL with Entropy Regularization (i.e., Hedge) are equivalent. In general, for the FTPL algorithm with any perturbation distribution, one can always find an FTRL algorithm with a particular regularization such that their update rule is equivalent. However, this relationship does not hold vice versa. For example, Hofbauer and Sandholm (2002) shows that for FTRL with log barrier regularization, there does not exist an equivalent perturbation distribution for FTPL.

Restarting techniques for non-stationary online learning. For non-stationary online learning problems, one common technique is *restarting*: one restarts the standard online learning algorithm periodically (Besbes et al., 2014) (see also e.g., Wei and Luo (2021); Mao et al. (2020)). After each restarting operation, the algorithm will ignore the previous history and execute as if it is the beginning of the interaction with the environment. Since the variation of the loss sequences is bounded, loss sequences between two consecutive restarting operations can be regarded as being *almost stationary*, which makes achieving an overall sublinear dynamic regret guarantee possible.

A.4 Why Focusing on Linear Loss Function?

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

$$\text{Regret}_{\mathscr{A}}\left((f_t)_{t\in[T]}\right) \le \sum_{t=1}^{T}\mathbb{E}[\langle \nabla f_t(\pi_{\mathscr{A},t}), \pi_{\mathscr{A},t}\rangle] - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\mathbb{E}[\langle \nabla f_t(\pi_{\mathscr{A},t}), \pi\rangle].$$

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

A.5 Six Representative General-Sum Games

In game theory, there are six representative two-player general-sum games (Robinson and 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.

In contrast, **the prisoner's dilemma**, depicted by payoff matrices $A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$, 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 the **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.

The **cyclic game**, with matrices $A = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & 4 \\ 2 & 1 \end{pmatrix}$, presents 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.

The **biased game**, depicted by $A = \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}$, inherently favors one player, often reflecting situations where external factors or inherent advantages influence outcomes, leading to consistently unequal payoffs.

Finally, the **second-best game**, with payoff 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 interactions. From cooperative to competitive and fair to biased scenarios, these matrices provide a rich landscape for exploring the nuances of decision-making behavior in game theory.

B Deferred Results and Proofs in Section 3

B.1 Ablation Study on Prompts

Ablation study on online learning. 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 by using different prompts in Figure 12. Specifically, for **Ablation1**, we remove examples to illustrate the game rules. For **Ablation2**, we remove the number of iterations. For **Ablation3**, we incorporate some *hints* for the LLM for decision-making, including the hints to suggest it to pay attention to the loss history, to behave more greedily at the end of an episode, and also to explain the reason of its decision step-by-step. The latter hint is a popular technique in prompt engineering known as the *Chain-of-Thought* prompting (Wei et al., 2022b). Finally, we recall that d is the number of actions in all prompts.

Original prompt

```
You are solving a decision-making problem for 25 rounds.

There are $d$ number of action (which is 0 to $d-1$).

At each round, you need to choose a policy, it specifies your probability to choose each action.

This policy should be $d$-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.
```

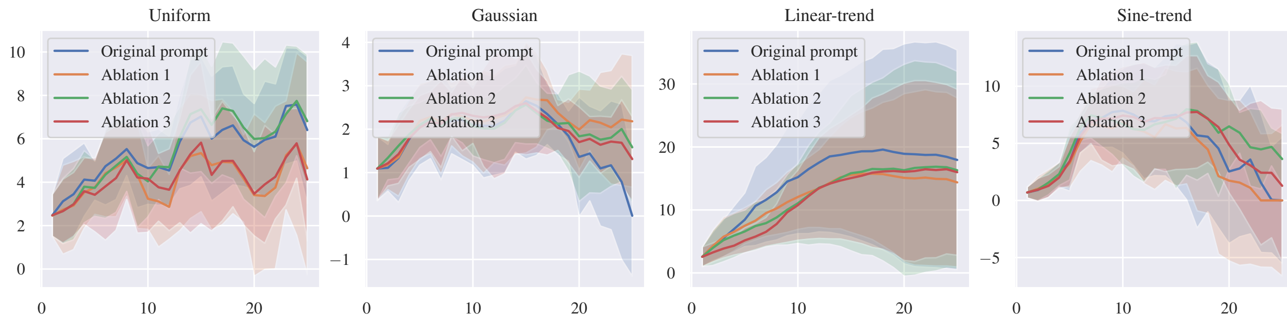

Figure 12: Ablation study on our prompt design.

```
It is not decided by what policies you have chosen. The reward vector is also $d$-dimensional.

It represents the reward of choosing action from 0 to $d-1$.

For example, a reward vector of [0.8, 3.2] means reward for action_0 is 0.8
and the reward for action_1 is 3.2.

Then your reward for this round will be calculated according to the reward of each action and
your probability of choosing each action.

For example, if you choose the policy [0.2, 0.8] and get the reward vector [1, 2],
then your expected reward is 0.2*1 + 0.8*2=1.8

Your goal is to maximize your accumulative expected reward.

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 $d$-dimensional policy]'.
```

Ablation1: no examples

```
You are solving a decision-making problem for 25 rounds.

There are $d$ number of action (which is 0 to $d-1$).

At each round, you need to choose a policy, it specifies your probability to choose each action.

This policy should be $d$-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 $d$-dimensional.

It represents the reward of choosing action from 0 to $d-1$.

Then your reward for this round will be calculated according to the reward of each action and
your probability of choosing each action.

Your goal is to maximize your accumulative expected reward.
```

```
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 $d$-dimensional policy]'.
```

Ablation2: no round information

```
You are solving a decision-making problem.

There are $d$ number of action (which is 0 to $d-1$).

At each round, you need to choose a policy, it specifies your probability to choose each action.

This policy should be $d$-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 $d$-dimensional.

It represents the reward of choosing action from 0 to $d-1$.

For example, a reward vector of [0.8, 3.2] means reward for action_0 is 0.8
and the reward for action_1 is 3.2.

Then your reward for this round will be calculated according to the reward of each action and
your probability of choosing each action.

For example, if you choose the policy [0.2, 0.8] and get the reward vector [1, 2],
then your expected reward is 0.2*1 + 0.8*2=1.8

Your goal is to maximize your accumulative expected reward.

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 $d$-dimensional policy]'.
```

Ablation3: adding hints

```
You are solving a decision-making problem for 25 rounds.

There are $d$ number of action (which is 0 to $d-1$).

At each round, you need to choose a policy, it specifies your probability to choose each action.

This policy should be $d$-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 $d$-dimensional.
```

```
It represents the reward of choosing action from 0 to $d-1$.

For example, a reward vector of [0.8, 3.2] means reward for action_0 is 0.8
and the reward for action_1 is 3.2.

Then your reward for this round will be calculated according to the reward of each action and
your probability of choosing each action.

For example, if you choose the policy [0.2, 0.8] and get the reward vector [1, 2],
then your expected reward is 0.2*1 + 0.8*2=1.8

Your goal is to maximize your accumulative expected reward.

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 $d$-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.

You should explore for first several rounds and behave greedily for later rounds, for example,
choosing one action with probability more than 0.99.

Please also explain whether you are behaving more greedily and less greedily
by explicitly considering the policy you just used for last round.
```

We can see in Figure 12 that the performances of LLM agents are consistent under different variants of the prompts.

Ablation study on repeated games. For the game setting, we also investigate whether explicitly informing LLM agents that they are ``playing a repeated matrix game with some other opponents'' would affect the performance. Therefore, we evaluate three different prompts by informing LLM agents that they are `playing a matrix game`, `solving multi-arm bandit`, or solving general `decision-making` problems, in the first line of the prompt. We show the performance of such three prompts in Figure 13, where it is seen that LLM agents' performance on repeated games is consistent among these variants of the prompts.

B.2 Proof for Proposition 1

Proof. Under the null hypothesis H_0, the probability p that $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t - \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t+1]}\right)/(t+1) > 0$ 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 $s \geq \frac{T-1}{2}$. $\square$

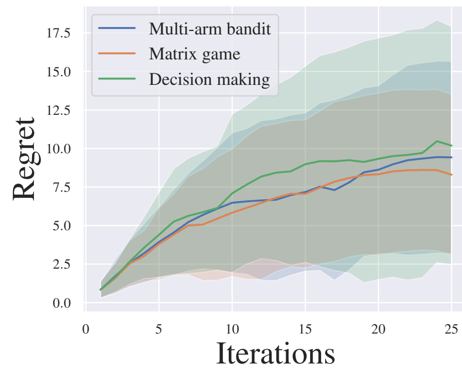

Figure 13: Regret of GPT-4 for repeated games under 3 different prompt ablations. Its performance is consistent among three different prompts.

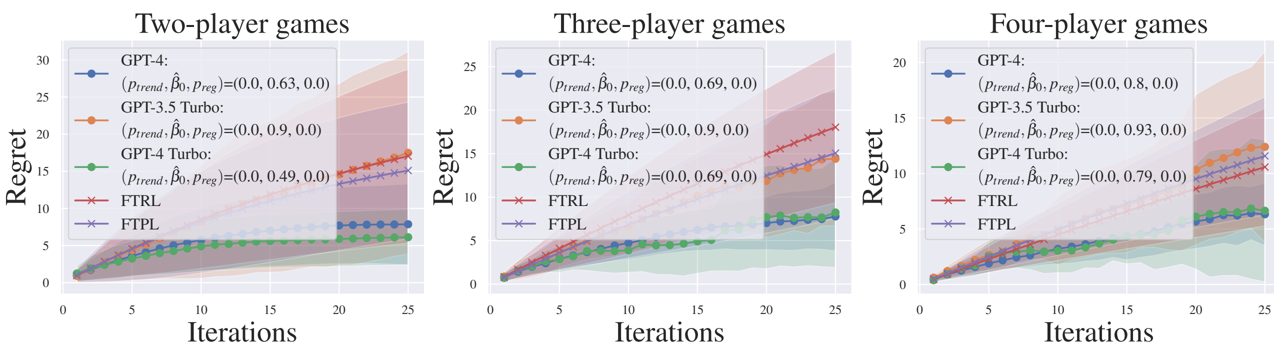

Figure 14: Comparisons of GPT-4 Turbo with GPT-4 and GPT-3.5 Turbo. Due to the symmetry of agents in randomly generated games, we report the regret of one agent for ease of presentation. We can see that GPT-4 Turbo performs on par with or better than GPT-4.

B.3 Results of GPT-4 Turbo

When preparing our draft, we are aware of a more advanced version of GPT-4, GPT-4 Turbo. Although GPT-4 has successfully exhibited no-regret behaviors in most problems, we also evaluate and report the performance of its more advanced version here. Specifically, we compare GPT-4 and GPT-4 Turbo for the repeated game scenario in Figure 14, where we validate that GPT-4 Turbo performs either comparably with or better than GPT-4.

B.4 LLM Agents' Explanation on Their Output Policies

We here present two examples of LLMs' explanations of their output policies, showing how they *reason* in their decision-making.

Output Example 1

```
Looking at the history of the reward vectors, it is clear to see that the reward for choosing
action 0 is consistently higher than that for choosing action 1. This pattern is evident across
all 5 rounds of the game.

In the first round, the difference in rewards is 3.52, this remained relatively stable with
the difference in the last round being 3.52 as well. It's indicative that action 0 consistently
brings a higher return than action 1.
```

```
Therefore, I'll lean more towards selecting action 0 in my policy.

However, acting deterministically, that is always choosing action 0, would introduce the risk
of becoming too predictable and letting the opponent exploit this predictability

So, in order to maintain some unpredictability while still capitalizing on the higher average
return of action 0, I'll set my policy as follows: select action 0 with a probability of 0.8
and action 1 with a probability of 0.2.

This way, the majority of the time I'm choosing the action that gives a higher reward, but I'm
also occasionally choosing the other action to keep the opponent guessing, and avoid
any predictable patterns that can be exploited

So, my policy is as follows: Policy: [0.8, 0.2]
```

Output Example 2

```
The total reward from action_1 is 130.92 and the average reward is around 10.07.

On the other hand, the total reward from action_0 is 33.418 and the average reward is around 8.354.

Although action_0 has a less average reward, its highest reward is 26.977,
which is considerably high and shows its potential.

For this round, I want to continue to explore action_0 but also significantly exploit action 1
as it reliably gives higher reward.

So, I'd like to decrease the probability of selecting action_0 compared to last time
but still keep it in the policy as a possible option.

The policy for the previous round was not provided but assuming it was more balanced,
I'm now going to lean more towards action_1.

So, the policy will be: [0.3, 0.7]. Here, I'm behaving more greedily compared to earlier rounds,
as we are choosing the action which has been giving us higher rewards with more probability.
```

As observed from the output, the LLM agent tends to use the history of the reward vectors by looking at the "sum/average of the accumulated rewards", and tends to introduce randomization in decision-making, in order to be "less predictable and exploitable". These are several key components in achieving no-regret in online learning and games (Cesa-Bianchi and Lugosi, 2006), explaining the empirical evidence we had in Section 3.

C Deferred Results and Proofs in Section 4

C.1 Deferred Proof of Observation 1

Proof of Observation 1. For given $(\ell_t)_{t\in[T]}$,

$$\sum_{t=1}^{T}\int_{\pi_t\in\Pi} P_{\text{LLM}}(\pi_t \mid (\ell_i)_{i\in[t-1]})\langle \ell_t,\pi_t\rangle d\pi_t \leq \sum_{t=1}^{T}\int_{\pi_t\in\Pi}\Big(P_{\text{data}}(\pi_t \mid (\ell_i)_{i\in[t-1]})+\epsilon\Big)\langle \ell_t,\pi_t\rangle d\pi_t$$

holds, where we use the convention of $P_{\text{LLM}}(\pi_t \mid (\ell_0)) := P_{\text{LLM}}(\pi_t)$ and $P_{\text{data}}(\pi_t \mid (\ell_0)) := P_{\text{data}}(\pi_t)$. Hence,

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

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 result for the dynamic-regret case, we can simply change the term $\inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t,\pi\rangle$ in the above derivation to $\sum_{t=1}^{T}\inf_{\pi\in\Pi}\langle \ell_t,\pi\rangle$. □

C.2 Deferred Proof of Lemma 1

Lemma 1. (Alignment of Assumption 1 with quantal response)**.** *Fix $t\in[T]$, $\sigma>0$. If we model the noise of data collection to be i.i.d. Gaussian distribution in the numeric value space, i.e.,*

$$\mathbb{P}\left(\left\{f(x_{N_{i-1}+1:N_i})\right\}_{i\in[t]}\,\middle|\, z\right)=\prod_{i=1}^{t}\mathbb{P}\left(f(x_{N_{i-1}+1:N_i})\,\middle|\, z\right)\propto\prod_{i=1}^{t}\exp\left(-\frac{\|f(x_{N_{i-1}+1:N_i})-z\|_2^2}{2\sigma^2}\right),$$

the prior distribution of the latent variable z is also Gaussian, i.e., $z\sim\mathcal{N}(\mathbf{0}_d,\sigma^2 I)$, and the text labels satisfy that $\mathbb{P}(g(x_{N_t+1:N_{t+1}})|z)=\mathbb{1}\left(g(x_{N_t+1:N_{t+1}})\in\arg\min_{a\in\mathcal{A}} z_a\right)$, then we have

$$\mathbb{P}\left(g(x_{N_t+1:N_{t+1}})\,\middle|\, x_{1:N_t}\right)=P_{quantal}^{\sigma\sqrt{t+1}}\left(g(x_{N_t+1:N_{t+1}})\,\middle|\,\left\{f(x_{N_{i-1}+1:N_i})\right\}_{i\in[t]}\right),$$

with $P_{noise}=\mathcal{N}(\mathbf{0}_d,I)$ in Definition 2, i.e., the action $a=g(x_{N_t+1:N_{t+1}})$ extracted from the text $x_{N_t+1:N_{t+1}}$ is a quantal response w.r.t. the loss vectors $\left(f(x_{N_{i-1}+1:N_i})\right)_{i\in[t]}$.

Proof. Note that

$$\mathbb{P}(z|x_{1:N_t})=\int_{\ell_{1:t}}\mathbb{P}(z,\ell_{1:t}|x_{1:N_t})d\ell_{1:t}=\int_{\ell_{1:t}}\mathbb{P}(\ell_{1:t}|x_{1:N_t})\mathbb{P}(z|x_{1:N_t},\ell_{1:t})d\ell_{1:t}.$$

For $\mathbb{P}(\ell_{1:t}|x_{1:N_t})$, since we have assumed the existence of function f to decode $\ell_{1:t}$ from $x_{1:N_t}$, it holds that

$$\mathbb{P}(\ell_{1:t}|x_{1:N_t})=\prod_{i=1}^{t}\delta\left(\ell_i-f(x_{N_{i-1}+1:N_i})\right),$$

where we use δ to denote the d-dimensional Dirac-delta function. For $\mathbb{P}(z|x_{1:N_t},\ell_{1:t})$, by Assumption 1, it holds that

$$\mathbb{P}(z,x_{1:N_t},\ell_{1:t})=\mathbb{P}(z,\ell_{1:t})\mathbb{P}(x_{1:N_t}|\ell_{1:t}),$$

which leads to $\mathbb{P}(x_{1:N_t}|\ell_{1:t}) = \mathbb{P}(x_{1:N_t}|\ell_{1:t}, z)$ by Bayes rule. This implies that the random variable $x_{1:N_t}$ and z are independent conditioned on $\ell_{1:t}$. Therefore, it holds that $\mathbb{P}(z|x_{1:N_t}, \ell_{1:t}) = \mathbb{P}(z|\ell_{1:t})$. Finally, we can compute

$$\begin{aligned}\mathbb{P}(z|x_{1:N_t}) &= \int_{\ell_{1:t}} \mathbb{P}(z, \ell_{1:t}|x_{1:N_t}) d\ell_{1:t} = \int_{\ell_{1:t}} \prod_{i=1}^{t} \delta(\ell_i - f(x_{N_{i-1}+1:N_i}))\mathbb{P}(z|\ell_{1:t}) d\ell_{1:t} \\ &= \mathbb{P}\left(z|\left(\ell_i = f(x_{N_{i-1}+1:N_i})\right)_{i\in[t]}\right).\end{aligned}$$

Based on this, we conclude that

$$\begin{aligned}\mathbb{P}(g(x_{N_t+1:N_{t+1}})|x_{1:N_t}) &= \int_z \mathbb{P}(g(x_{N_t+1:N_{t+1}})|z, x_{1:N_t})\mathbb{P}(z|x_{1:N_t}) dz \\ &= \int_z \mathbb{P}(g(x_{N_t+1:N_{t+1}})|z)\mathbb{P}(z|\{\ell_i = f(x_{N_{i-1}+1:N_i})\}_{i\in[t]}) dz \\ &= \mathbb{P}\left(g(x_{N_t+1:N_{t+1}})|\left(\ell_i = f(x_{N_{i-1}+1:N_i})\right)_{i\in[t]}\right)\end{aligned}$$

where the first equality is by the independence between $x_{N_t+1:N_{t+1}}$ and $x_{1:N_t}$ conditioned on z, due to Assumption 1. Therefore, it suffices to consider the probability of $\mathbb{P}(a|\ell_{1:t})$ only, in order to analyze $\mathbb{P}(g(x_{N_t+1:N_{t+1}})|x_{1:N_t})$, where we recall the definition that $a = g(x_{N_t+1:N_{t+1}})$. Since $z \sim \mathcal{N}(\mathbf{0}_d, \sigma^2 I)$, and $\ell_i \mid z \sim \mathcal{N}(z, \sigma^2 I)$, we have

$$z \mid \ell_{1:t} \sim \mathcal{N}\left(\frac{1}{t+1}\sum_{i\in[t]} \ell_i, \frac{\sigma^2}{t+1} I\right), \tag{11}$$

by the posterior distribution of Gaussian distribution. Now we conclude that

$$\begin{aligned}\mathbb{P}(a|\ell_{1:t}) &= \int_z \mathbb{P}(a|z, \ell_{1:t})\mathbb{P}(z|\ell_{1:t}) dz = \int_z \mathbb{P}(a|z)\mathbb{P}(z|\ell_{1:t}) dz \\ &= \int_z \mathbb{1}(a \in \arg\min_{a'\in\mathcal{A}} z_{a'})\mathbb{P}(z|\ell_{1:t}) dz = \int_z \mathbb{1}\left(a \in \arg\min_{a'\in\mathcal{A}}\left(\frac{\sigma}{\sqrt{t+1}}\epsilon + \frac{1}{t+1}\sum_{i\in[t]}\ell_i\right)_{a'}\right)\mathbb{P}(\epsilon) d\epsilon \\ &= \int_z \mathbb{1}\left(a \in \arg\min_{a'\in\mathcal{A}}\left(\sigma\sqrt{t+1}\epsilon + \sum_{i\in[t]}\ell_i\right)_{a'}\right)\mathbb{P}(\epsilon) d\epsilon = \mathbb{P}\left(a \in \arg\min_{a'\in\mathcal{A}}\left(\sigma\sqrt{t+1}\epsilon + \sum_{i\in[t]}\ell_i\right)_{a'}\right) \\ &= P_{quantal}^{\sigma\sqrt{t+1}}(a|\ell_{1:t}),\end{aligned}$$

where $\mathbb{P}(\epsilon) = \mathcal{N}(\mathbf{0}_d, I)$. This completes the proof. ☐

C.3 Relationship between FTPL and Definition 2

Fact 1. *Performing generalized quantal response of Definition 2 at every iteration $t \in [T]$ w.r.t. history loss vectors $\ell_{1:t-1}$ is essentially executing an FTPL algorithm.*

Proof. Before we move to the proof, we will define the random variable which has distribution P_{noise} as Z_{noise}. Note that at round $t \geq 2$ (as the policy at round $t = 1$ is fixed), we have

$$P_{\text{quantal}}^{\eta_{t-1}}(a|\ell_{1:t-1}) := \mathbb{P}\left(a \in \arg\min_{a'\in\mathcal{A}}\left(\sum_{i=1}^{t-1}\ell_i + \eta_{t-1}\epsilon\right)(a')\right) \tag{12}$$

which is exactly the case when ϵ_t in Equation (10) satisfies $\epsilon_t \stackrel{d}{=} \eta_{t-1}\epsilon$. ☐

C.4 Deferred Proof of Theorem 1

Theorem 1. (Emergence of no-regret behavior). *Under the assumptions of Lemma 1, suppose the function class of LLM_θ is expressive enough such that for all $t \in [T]$, $\max_{\theta\in\Theta} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log LLM_\theta\left(x_j \mid x_{1:j-1}\right) = \max_{\{q_j\in\{\mathcal{V}^{j-1}\to\Delta(\mathcal{V})\}\}_{j\in[N_{t+1}]}} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log q_j\left(x_j \mid x_{1:j-1}\right)$, where we define $q_1(x_1 \mid x_{1:0}) := q_1(x_1)$, and $\theta^\star$ maximizes Equation (1). Then, there exist (simple) algorithms using $LLM_{\theta^\star}$ to achieve no (dynamic) regret for (non-stationary) online learning with full-information/bandit feedback. To be specific, for (2) and (4), by defining the variation bound $\sum_{t=1}^{T-1}\|\ell_{t+1}-\ell_t\|_\infty \le V_T$ such that $V_T \le T$ and $V_T = \Theta(T^\rho)$ for some $\rho \in (0,1)$, it holds that for large enough T, d:*

(1) For online learning with full-information feedback, $Regret_{LLM_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right) \le \mathcal{O}\left(\sqrt{T\log d}\right)$;

(2) For non-stationary online learning with full-information feedback,

$$D\text{-}Regret_{LLM_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right) \le \mathcal{O}\left((\log d\ \ V_T)^{1/3}T^{2/3}\right);$$

(3) For online learning with bandit feedback,

$$\mathbb{E}\left[Regret_{LLM_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right)\right] \le \mathcal{O}\left((\log d)^{1/2} d T^{1/2+1/\log T}\log T\right);$$

(4) For non-stationary online learning with bandit feedback,

$$\mathbb{E}\left[D\text{-}Regret_{LLM_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right)\right] \le \mathcal{O}\left((T^2d^2V_T)^{1/3}(\log d)^{1/2}T^{1/\log T}\log T\right).$$

Proof. Note that

$$\begin{aligned}\max_{\{q_j\in\{\mathcal{V}^{j-1}\to\Delta(\mathcal{V})\}\}_{j\in[N_{t+1}]}} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log q_j\left(x_j \mid x_{1:j-1}\right) &= \max_{q\in\Delta(\mathcal{V}^{N_{t+1}})} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \log q(x_{1:N_{t+1}}) \\ &= \max_{q\in\Delta(\mathcal{V}^{N_{t+1}})} -\mathrm{KL}(P_t^{text}\,||\,q) + \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}}[P_t^{text}(x_{1:N_{t+1}})],\end{aligned}$$

where $\mathrm{KL}(q\,||\,p)$ denotes the Kullback–Leibler divergence between two distributions p,q. Now we define $\mathrm{LLM}_\theta(x_{1:N_{t+1}}) = \prod_{t=1}^{N_{t+1}} \mathrm{LLM}_\theta(x_j \mid x_{1:j-1})$. It is easy to verify that $\mathrm{LLM}_\theta(x_{1:N_{t+1}}) \in \Delta(\mathcal{V}^{N_{t+1}})$, i.e., it also defines a valid joint distribution over tokens. Therefore, we have

$$\max_{\theta\in\Theta} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log \mathrm{LLM}_\theta\left(x_j \mid x_{1:j-1}\right) = \max_{\theta\in\Theta} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \log \mathrm{LLM}_\theta(x_{1:N_{t+1}}).$$

Now, due to our assumption that

$$\max_{\theta\in\Theta} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log \mathrm{LLM}_\theta\left(x_j \mid x_{1:j-1}\right) = \max_{\{q_j\in\{\mathcal{V}^{j-1}\to\Delta(\mathcal{V})\}\}_{j\in[N_{t+1}]}} \mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log q_j\left(x_j \mid x_{1:j-1}\right),$$

we conclude that

$$\min_{\theta\in\Theta} \mathrm{KL}(P_t^{text}\,||\,\mathrm{LLM}_\theta) = \min_{q\in\Delta(\mathcal{V}^{N_{t+1}})} \mathrm{KL}(P_t^{text}\,||\,q) = 0,$$

which implies that $\mathrm{LLM}_{\theta^\star} = P_t^{text}$. Correspondingly, if we define $\mathrm{LLM}_{\theta^\star}(x_{N_t+1:N_{t+1}} \mid x_{1:N_t})$ to be the distribution induced by the joint distribution $\mathrm{LLM}_{\theta^\star}(x_{1:N_{t+1}})$, it holds that

$$\mathrm{LLM}_{\theta^\star}(x_{N_t+1:N_{t+1}} \mid x_{1:N_t}) = \mathbb{P}(x_{N_t+1:N_{t+1}} \mid x_{1:N_t}).$$

In other words, intuitively, $\text{LLM}_{\theta^\star}$ has learned the corresponding *pre-training* distribution perfectly. Note that this has been a common assumption in the Bayesian perspective of ICL (Xie et al., 2022; Lee et al., 2023; Zhang et al., 2023b). Therefore, to analyze the actions taken by $\text{LLM}_{\theta^\star}$, it suffices to consider $\mathbb{P}(g(x_{N_t+1:N_{t+1}})\,|\,x_{1:N_t})$, which is equal to $P^{\sigma\sqrt{t+1}}_{quantal}\left(g(x_{N_t+1:N_{t+1}})\,\Big|\,\left\{f(x_{N_{i-1}+1:N_i})\right\}_{i\in[t]}\right)$ by Lemma 1. Therefore, we proved that $\text{LLM}_{\theta^\star}$ is essentially mimicking the well-known no-regret algorithm, FTPL with perturbation distribution as $\mathcal{N}(\mathbf{0}_d, \sigma^2 tI)$ for round $t\in[T]$, according to Equation (12) of Fact 1, for which we can establish the corresponding regret guarantee for each case:

(1) Combining the above result with Lemma 2, we can derive the regret bound for online learning with full-information feedback.

(2) Combining the above result with Lemma 2 and Lemma 4, we get that

$$\text{D-Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]}) \le \min_{\Delta_T\in[T]} \frac{2T}{\Delta_T} C\sqrt{\Delta_T \log d} + 2\Delta_T V_T,$$

for some constant C. We firstly consider the following problem

$$\min_{u>0} \frac{2T}{u} C\sqrt{u\log d} + 2uV_T,$$

where the optimal solution is $u^\star = \left(\frac{C^2T^2\log d}{4V_T^2}\right)^{1/3}$. Therefore, if we have $u^\star\in[1,T]$, we can choose $\Delta_T = \lceil u^\star\rceil$, which results in a regret bound of

$$\text{D-Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]}) \le \frac{2T}{\sqrt{u^\star}} C\sqrt{\log d} + 4u^\star V_T = \mathcal{O}\Big((\log d\ \ V_T)^{1/3} T^{2/3}\Big).$$

Now we check the conditions for $u^\star\in[1,T]$. It is direct to see that since $V_T\le T$, $u^\star\ge 1$ holds as long as d is sufficiently large. To ensure $u^\star\le T$, we get the condition $V_T\ge C\sqrt{\frac{\log d}{4T}}$, which holds as long as T is large enough.

(3) Combining the above result with Lemma 3, we can prove a regret guarantee for online learning with bandit feedback.

(4) Combining this result with Lemma 3 and Lemma 4, it holds that

$$\mathbb{E}[\text{D-Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]})] \le \min_{\Delta_T\in[T]} \frac{2T}{\Delta_T} C(\log d)^{\frac{1}{2}} d\Delta_T^{\frac{1}{2}+\frac{1}{\log T}} \log\Delta_T + 2\Delta_T V_T,$$

for some constant C. By adopting a similar analysis as that of (2), we choose $u^\star = \left(\frac{C'T^2d^2}{V_T^2}\right)^{1/3}$ for some constant C'. If $u^\star\in[1,T]$, we choose $\Delta_T = \lceil u^\star\rceil$ and derive the following regret:

$$\mathbb{E}[\text{D-Regret}_{\text{LLM}_{\theta^\star}}((\ell_i)_{i\in[T]})] \le \mathcal{O}\Big((T^2d^2V_T)^{1/3}(\log d)^{1/2}T^{1/\log T}\log T\Big).$$

Now we check the condition of $u^\star\in[1,T]$. Note that since $V_T\le T$, $u^\star\ge 1$ holds as long as d is sufficiently large. For $u^\star\le T$, we have $V_T\ge\sqrt{\frac{C'd^2}{T}}$, which holds as long as T is large enough.

Now, we present Lemma 2 - Lemma 4. Before proceeding, we assume $\|\ell_t\|_\infty\le B=1$ for simplicity of presentations hereafter. The results and proof are not affected by the constant bound B.

Lemma 2 (Regret guarantee of FTPL with full-information feedback)**.** *Suppose the noise distribution of FTPL satisfies that $\epsilon_t\sim\mathcal{N}(\mathbf{0}_d,\zeta_t^2 I)$ in Equation (10) and $\zeta_t=\sigma\sqrt{t}$, then for online learning with full-information feedback,*

$$\textit{Regret}_{FTPL}((\ell_i)_{i\in[T]}) \le 4\left(\sigma+\frac{1}{\sigma}\right)\sqrt{T\log d} = \mathcal{O}(\sqrt{T\log d}).$$

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

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

Therefore, plugging $\zeta_t = \sigma\sqrt{t}$ and $\|\ell_t\|_\infty^2 \le 1$ provides

$$\mathrm{Regret}_{\mathrm{FTPL}}((\ell_i)_{i\in[T]}) \le \sqrt{2\log d}\left(\sigma\sqrt{T} + \sum_{t=1}^{T}\frac{1}{\sigma\sqrt{t}}\right) \le 4\left(\sigma + \frac{1}{\sigma}\right)\sqrt{T\log d},$$

completing the proof. □

Lemma 3 (Regret guarantee of FTPL with bandit feedback)**.** *Suppose the noise distribution of FTPL satisfies that $\epsilon_t \sim \mathcal{N}(\mathbf{0}_d, \zeta_t^2 I)$ in Equation (10) and $\zeta_t = \sigma\sqrt{t}$, then for online learning with bandit feedback,*

$$\mathbb{E}[\mathit{Regret}_{\mathit{FTPL}}((\ell_i)_{i\in[T]})] \le \mathcal{O}((\log d)^{\frac{1}{2}} d T^{\frac{1}{2}+\frac{1}{\log T}} \log T).$$

Proof. The proof of the bandit problem is more complex. We first define the following notation. We denote $G_t = \sum_{t'=1}^{t} -\ell_{t'}$, $\widehat{G}_t = \sum_{t'=1}^{t} -\widehat{\ell}_{t'}$, $\Phi(G) = \max_\pi \langle \pi, G\rangle$, $\Phi_t(G) = \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0}_d, I)}\Phi(G+\zeta_t\epsilon)$, and D_{Φ_t} to be the Bregman divergence with respect to Φ_t, where we recall the construction of the empirical estimator $\widehat{\ell}_{t'}$ of $\ell_{t'}$ in Section 3.2. By Li and Tewari (2017), $\pi_t = \nabla\Phi_t(\widehat{G}_t)$. Now due to the convexity of Φ,

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

Therefore,

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

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,

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

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

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

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

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

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

Lemma 4. *Denote the variation of loss vectors as $L_T = \sum_{t=1}^{T-1} \|\ell_{t+1} - \ell_t\|_\infty$. Suppose there exists an algorithm $\mathscr{A}$ for online learning with full-information feedback with regret guarantee that $\text{Regret}_{\mathscr{A}}((\ell_i)_{i\in[T]}) \leq f(T,d)$ for some function f, where T denotes the horizon and d denotes the policy dimension. Then, there exists another algorithm $\mathscr{A}'$ that can achieve*

$$\text{D-Regret}_{\mathscr{A}'}((\ell_i)_{i\in[T]}) \leq \min_{\Delta_T \in [T]} \left(\frac{T}{\Delta_T} + 1\right) f(\Delta_T, d) + 2\Delta_T L_T.$$

Similarly, suppose there exists an algorithm $\mathscr{B}$ for online learning with bandit feedback with regret guarantee that $\mathbb{E}\left[\text{Regret}_{\mathscr{B}}((\ell_i)_{i\in[T]})\right] \leq g(T,d)$ for some function g; then there exists another algorithm $\mathscr{B}'$ that can achieve

$$\mathbb{E}[\text{D-Regret}_{\mathscr{B}'}((\ell_i)_{i\in[T]})] \leq \min_{\Delta_T \in [T]} \left(\frac{T}{\Delta_T} + 1\right) g(\Delta_T, d) + 2\Delta_T L_T.$$

Proof. We denote $\mathscr{A}'$ as the algorithm that restarts $\mathscr{A}$ every Δ_T iterations. We break the time index $[T]$ into m batches $\mathcal{T}_{1:m}$ of size Δ_T (except for, possibly the last batch). Denote $\ell_i^\star := \min_{j\in[d]} \ell_{ij}$. By Equation (6) of Besbes et al. (2014), it holds that for each $k \in [m]$

$$\min_{j\in[d]} \left(\sum_{t\in\mathcal{T}_k} \ell_t\right)_j - \sum_{t\in\mathcal{T}_k} \ell_t^\star \leq 2\Delta_T L_k,$$

where we define $L_k = \sum_{t\in\mathcal{T}_k} \|\ell_{t+1} - \ell_t\|_\infty$. Therefore, we have

$$\begin{aligned}\text{D-Regret}_{\mathscr{A}'}((\ell_i)_{i\in[T]}) &\leq \min_{j\in[d]} \left(\sum_{t\in[T]} \ell_t\right)_j - \sum_{t\in[T]} \ell_t^\star + \sum_{k\in[m]} \text{Regret}_{\mathscr{A}}((\ell_i)_{i\in[\mathcal{T}_k]}) \\ &\leq 2\Delta_T (\sum_{k\in[m]} L_k) + (T/\Delta_T + 1) g(\Delta_T, d).\end{aligned} \tag{13}$$

By Equation (4) of Besbes et al. (2014) that $\sum_{k\in[m]} L_k \leq L_T$ and this inequality holds for any $\Delta_T \in [T]$, we proved $\text{D-Regret}_{\mathscr{A}'}((\ell_i)_{i\in[T]}) \leq \min_{\Delta_T\in[T]} \left(\frac{T}{\Delta_T} + 1\right) f(\Delta_T, d) + 2\Delta_T L_T$.

Similarly, if we take the expectation for Equation (13), it holds that

$$\begin{aligned}\mathbb{E}[\text{D-Regret}_{\mathscr{B}'}((\ell_i)_{i\in[T]})] &\leq \min_{j\in[d]} \left(\sum_{t\in[T]} \ell_t\right)_j - \sum_{t\in[T]} \ell_t^\star + \sum_{k\in[m]} \mathbb{E}[\text{Regret}_{\mathscr{B}}((\ell_i)_{i\in[\mathcal{T}_k]})] \\ &\leq \min_{\Delta_T \in [T]} \left(\frac{T}{\Delta_T} + 1\right) g(\Delta_T, d) + 2\Delta_T L_T,\end{aligned}$$

thus completing the proof. □

Combining the results above completes the proof for Theorem 1. □

C.5 Extending Theorem 1 with Relaxed Assumptions

C.5.1 Relaxation under More General Data Distributions

We first remark on the possibility of relaxing the Gaussian assumptions on the data distributions.

Remark 3 (Relaxing the Gaussian distribution assumption)**.** *In the proof of Lemma 1, to obtain the result that the action is a quantal response w.r.t.* $\ell_{1:T}$*, one does not necessarily require* both *the prior distribution of* z *and the conditional distribution of* ℓ_i *given* z *to be Gaussian. Instead, for* any *joint distribution* $\mathbb{P}(z, \ell_{1:T})$*, as long as its posterior distribution satisfies Equation (11), it would suffice. It is a combined effect of both the prior and the conditional distributions.*

More formally, we can extend Theorem 1 to the case with a much more general prior task distribution than the Gaussian one, where the key is that Equation (11) only needs to hold approximately.

Theorem 5. *In Theorem 1, we can relax the assumption on* $\mathbb{P}(z)$ *to one where we only require* $\mathbb{P}(z)$ *to be i.i.d for each coordinate of* z *and* $0 < \mathbb{P}(z_j) < \infty$, $|\nabla \mathbb{P}(z_j)| < \infty$ *for any* $j \in [d]$, $z_j \in \mathbb{R}$*, and the bounds for (1) and (2) of Theorem 1 still hold, with only a degradation of* $\mathcal{O}(d^2 \log T)$*.*

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 referred to as the *Bernstein–von Mises theorem* (Van der Vaart, 2000).

Proof. Since we extend Theorem 1 to settings with general task prior distribution only requiring the coordinates to be i.i.d, from now on, we consider the j-th coordinate only. To begin with, fix $t \in [T]$, we define the log-likelihood of the posterior as

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

Then, the 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 $\widehat{J}_t : \mathbb{R} \to \mathbb{R}$ as:

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

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 $\widehat{M}_1 = \max_{z_j \in [-\delta, 1+\delta]} \frac{1}{\mathbb{P}(z_j)}$
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 $M_1 = \sup_{z_j \in [-\delta, 1+\delta]} \left| \frac{\nabla \mathbb{P}(z_j)}{\mathbb{P}(z_j)} \right|$, $\widetilde{M}_1 = \sup_{z_j \in [-\delta, 1+\delta]} |\mathbb{P}(z_j)|$ since we have assumed that $0 < \mathbb{P}(z_j) < \infty$, $|\nabla \mathbb{P}(z_j)| < \infty$.

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

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

where C is the normalization constant and

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

Therefore, we conclude that the TV distance between z (conditioned on $(\ell_i)_{i\in[t]}$) and $\mathcal{N}\left(\widehat{z}, \frac{\sigma^2}{t}\right)$ satisfies that

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

due to the independence of $(z_j)_{j\in[d]}$ conditioned on $\ell_{1:t}$. Now we denote algorithm $\widehat{\text{FTPL}}$ to be the FTPL algorithm w.r.t. the noise distribution $\mathbb{P}(z\,|\,(\ell_i)_{i\in[t]})$, and FTPL to be the algorithm w.r.t. the noise distribution $\mathcal{N}(\widehat{z}, \frac{\sigma^2}{t})$. Therefore, we have

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

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

$$\left|\text{D-Regret}_{\text{FTPL}}((\ell)_{i\in[T]}) - \text{D-Regret}_{\widehat{\text{FTPL}}}((\ell)_{i\in[T]})\right| \le \mathcal{O}(d^2\log T),$$

which completes the proof. □

C.5.2 Relaxation under Decision-Irrelevant Pre-Training Data

We then remark on the possible relaxation when the training data may not all come from decision-making tasks.

Remark 4 (Pre-training with relaxed data assumptions)**.** *Note that the pre-training (text) data are so far assumed to be related to* decision-making *problems (though not necessarily* sequential ones*), see Assumption 1 and Example 1 for instance. It can also be generalized to the text datasets involving* Question-Answering *(Q-A), a typical task in natural language processing, where the* true/fact *answer, sampled answers from different human users (with possibly wrong or biased answers), correspond to the latent z (and associated maximizer a) and $\ell_{1:t}$, respectively. Moreover, in practice, the pre-training data may also involve* non-decision-making/Q-A *texts, given the diversity of the datasets. For such scenarios, we will make the assumptions on the data distribution* conditioned on the prompt for decision-making*. Specifically, when interacting with the LLM, human users will provide prompts (see e.g., our Figure 3), to induce it to make decisions. This will query the* conditional *distribution of*

$$\mathbb{P}\Big(g(x_{N_t+1:N_{t+1}})\,\Big|\,x_{1:N_t}, \textit{decision-making prompt}\Big)$$

to generate the control action. Correspondingly, Assumption 1 will thus only need to be made on

$$\mathbb{P}\Big(z, \ell_{1:t}, x_{1:N_{t+1}}, \textit{decision-making prompt}\Big),$$

while we do not need to make such assumptions on other prompts, e.g., corpora that are not related to decision-making.

D Deferred Results and Proofs in Section 5

D.1 Basic Lemmas

Lemma 5 (Double iterated limit)**.** *For a sequence $(a_{mn})_{m,n\in\mathbb{N}^+}$, suppose that $\lim_{m,n\to\infty} a_{mn} = L$. Then the following are equivalent:*

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

Lemma 6 (Hoeffding'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 $t > 0$,*

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

Lemma 7 (Uniform convergence $\Longrightarrow$ Interchanging limit and infimum)**.** *If $(f_n : X \to \mathbb{R})_{n\in\mathbb{N}^+}$ is a sequence of continuous functions that uniformly converge to a function $f : X \to \mathbb{R}$ on the domain X, then $\lim_{n\to\infty}\inf_{x\in X} f_n(x) = \inf_{x\in X} f(x)$ holds.*

D.2 Deferred Proof for the Arguments in Section 5.1

In this section, we prove some properties of $\mathcal{L}(\theta, k, N)$ under certain regularity conditions of f, h. Throughout this subsection, we will assume the following condition holds.

Condition 1. *For $h : \mathbb{R} \to \mathbb{R}^+$ and $f : \mathbb{R}\times\mathbb{N}^+ \to \mathbb{R}^+$, suppose $h(\cdot)$ and $f(\cdot, k)$ are both continuous and non-decreasing functions for any $k \in \mathbb{N}^+$. The derivative $h' : \mathbb{R} \to \mathbb{R}$ is also a continuous function. Moreover, f satisfies that $\log f(R_1, k_1) - \log f(R_1, k_2) \geq \log f(R_2, k_1) - \log f(R_2, k_2)$ for $R_1 \geq R_2$ and $k_1 \geq k_2$, i.e., $\log f$ is supermodular. Lastly, f is a function such that $\lim_{k\to\infty}\frac{f(R_1,k)}{f(R_2,k)} = \infty\cdot\mathbb{1}(R_1 > R_2) + \mathbb{1}(R_1 = R_2)$, with the convention of $\infty\cdot 0 = 0$. Lastly, $(\ell_t^{(j)})_{t\in[T], j\in[N]}$ are continuous random variables supported on $[-B, B]^{T\times N}$.*

Claim 1 (Iterated limit of $\mathcal{L}(\theta,k,N)$ is the same as double limit of $\mathcal{L}(\theta,k,N)$)**.** *It holds that:*

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

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

Firstly, as both h and f are non-negative (Condition 1), and $\mathbb{E}_{(\ell_t^{(j)})_{t\in[T],j\in[N]}}\left[h(\max_{j\in[N]}\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))\right]$ exists, we have by dominated convergence theorem that

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

where R_{LLM_θ} denotes an abbreviation of $\text{Regret}_{\text{LLM}_\theta}$. By (Ahsanullah et al., 2013, Chapter 11), 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 $N\to\infty$. Hence, we have $\lim_{N\to\infty}\lim_{k\to\infty}\mathcal{L}(\theta,k,N)=h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$ holds.

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

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

Lemma 8. *For any* $0<\epsilon<1$, *it follows that*

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

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 variables,* $\text{esssup}(H(X_i))=1$, *and* $H:\mathbb{R}\to\mathbb{R}^+$ *is a continuous non-decreasing function.*

Proof of Lemma 8. Since $f(\cdot,k),H$ are non-negative and non-decreasing functions, 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

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

By a similar argument, 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. $\square$

One key idea in the proof above is the use of some *truncation* level ϵ for $H(X)$ with $\operatorname{esssup}(H(X)) = 1$. By Lemma 8, we have

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

since

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

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

$$\begin{aligned}
&\lim_{N,k\to\infty} \mathcal{L}(\theta,k,N) = \mathbb{E}\lim_{N,k\to\infty}\left[\frac{\sum_{j\in[N]} h(R_{\mathrm{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(R_{\mathrm{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]} f(R_{\mathrm{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k)}\right] \\
&= h\left(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})\right) \\
&\quad \times \mathbb{E}\lim_{N,k\to\infty}\left[\frac{\sum_{j\in[N]} \frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))} f(R_{\mathrm{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)\mathbb{1}(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}>1-\epsilon)}{\sum_{j\in[N]} f(R_{\mathrm{LLM}_\theta}((\ell_i^{(j)})_{t\in[T]}),k)\mathbb{1}(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}>1-\epsilon)}\right] \\
&\ge (1-\epsilon)h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))
\end{aligned}$$

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

$$\mathcal{L}(\theta,k,N) \le h\left(\max_{\ell_1,\dots,\ell_T}\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})\right)$$

by definition of $\mathcal{L}$, the fact that h is non-decreasing, and by setting $\epsilon\to 0$ to obtain

$$\mathcal{L}(\theta,k,N) \ge h\left(\max_{\ell_1,\dots,\ell_T}\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})\right).$$

Here, we used the fact that $(\ell_t)_{t\in[T]}$ has a continuous distribution, $\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})$ is a continuous function, and the non-decreasing property and continuity of h (Condition 1), which lead to:

$$\operatorname{esssup}\left(h\left(\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})\right)\right) = \max_{\ell_1,\dots,\ell_T} h\left(\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})\right) = h\left(\max_{\ell_1,\dots,\ell_T}\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})\right). \tag{14}$$

Equation (14) will be used frequently in the overall proof in Appendix D.2.

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

Lastly, if $N \to \infty$, similarly by dominated convergence theorem we have

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

Thus, $\lim_{N\to\infty}\mathcal{L}(\theta,k,N)$ always exists for every k. Now, we use the known property of double iterated limit (Lemma 5), and obtain that $\lim_{k\to\infty}\lim_{N\to\infty}\mathcal{L}(\theta,k,N) = h(\max_{\ell_1,\dots,\ell_T}\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))$. $\square$

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

Proof. We will provide a similar analysis as Lemma 8 as follows:

Lemma 9. *For any $0<\epsilon<1$, $0<\delta<1$, and $k\in\mathbb{N}^+$, we have*

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

with probability at least $1-\delta$, where X_i's are i.i.d. random variables, $\mathrm{esssup}(H(X_i))=1$, $H:\mathbb{R}\to\mathbb{R}^+$ is a continuous non-decreasing function, $A(k,t,\epsilon) := \frac{(1-\epsilon)f((t/\mathrm{esssup}(t(X)))^{-1}(1-\epsilon),k)}{(1-\epsilon/2)f((t/\mathrm{esssup}(t(X)))^{-1}(1-\epsilon/2),k)}$, for any non-decreasing function $t:\mathbb{R}\to\mathbb{R}^+$, and $F_{t,X}$ is a cumulative distribution function of random variable $t(X)/esssup(t(X))$.

Proof of Lemma 9. With the same argument as the proof of Lemma 8, we have

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

It holds that $\frac{1}{N}|\{i\in[N]\mid(H(X_i)<1-\epsilon)\}| = F_{H,X}(1-\epsilon)+\widetilde{\mathcal{O}}(1/\sqrt{N})$ with probability at least $1-\delta/2$ due to Hoeffding's inequality (Lemma 6). Similarly, we have $\frac{1}{N}|\{i\in[N]\mid(H(X_i)>1-\epsilon/2)\}| = 1-F_{H,X}(1-\epsilon/2)+\widetilde{\mathcal{O}}(1/\sqrt{N})$ with probability at least $1-\delta/2$. Therefore,

$$\frac{|\{i\in[N]\mid(H(X_i)<1-\epsilon)\}|}{|\{i\in[N]\mid(H(X_i)>1-\epsilon/2)\}|} = \frac{F_{H,X}(1-\epsilon)}{1-F_{H,X}(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}) \le \frac{1}{1-F_{H,X}(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}),$$

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

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

$\square$

Note that $\lim_{k\to\infty} A(k,H,\epsilon)=0$, since $\lim_{k\to\infty}\frac{f(R_1,k)}{f(R_2,k)}=\infty\cdot\mathbb{1}(R_1>R_2)+\mathbb{1}(R_1=R_2)$. By Lemma 9 with $H(R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))=\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}$, we have

$$\frac{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)\mathbb{1}\left(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}\geq 1-\epsilon\right)}{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)}$$
$$=\frac{1}{1+\frac{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)\mathbb{1}\left(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}<1-\epsilon\right)}{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)\mathbb{1}\left(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}\geq 1-\epsilon\right)}}\geq\frac{1}{1+A(k,H,\epsilon)(\frac{1}{1-F_{H,R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})}(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))},$$

where we recall the shorthand notation of $R_{\mathrm{LLM}_\theta}=\mathrm{Regret}_{\mathrm{LLM}_\theta}$. Note that $A(k,H,\epsilon)=A(k,h,\epsilon)$ and $F_{H,R_{\mathrm{LLM}_\theta}}=F_{h,R_{\mathrm{LLM}_\theta}}$ hold by the definitions of $F_{t,X}$ and $A(k,t,\epsilon)$ in Lemma 9. Therefore,

$$\begin{aligned}
1&\geq\frac{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}}{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)}\\
&\geq\frac{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}\mathbb{1}(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}\geq 1-\epsilon)}{\sum_{i=1}^N f(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}),k)\mathbb{1}(\frac{h(R_{\mathrm{LLM}_\theta}((\ell_t^{(i)})_{t\in[T]}))}{h(\max_{\ell_1,\dots,\ell_T} R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]}))}\geq 1-\epsilon)}\\
&\quad\times\frac{1}{1+A(k,h,\epsilon)(\frac{1}{1-F_{h,R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})}(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\\
&\geq\frac{1-\epsilon}{1+A(k,h,\epsilon)(\frac{1}{1-F_{h,R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})}(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}
\end{aligned}$$

with probability at least $1-\delta$.

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

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

Note that

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

is a continuous function of θ, since we assume LLM_θ is a continuous function of θ, $(\ell_t)_{t\in[T]}$ has a continuous distribution, and $\mathrm{Regret}_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})$ is a continuous function of LLM_θ and $(\ell_t)_{t\in[T]}$. Since we consider a compact Θ (as several recent works on analyzing Transformers (Bai et al., 2023; Lin et al., 2024)), we have $p(\epsilon):=\min_{\theta\in\Theta} 1-F_{h,R_{\mathrm{LLM}_\theta}((\ell_t)_{t\in[T]})}(1-\epsilon/2)>0$. Therefore,

$$\left(1-\frac{(1-\delta)(1-\epsilon)}{1+A(k,h,\epsilon)(\frac{1}{1-F_{h,R_{\mathrm{LLM}_\theta}}(1-\epsilon/2)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\right)\leq\left(1-\frac{(1-\delta)(1-\epsilon)}{1+A(k,h,\epsilon)(\frac{1}{p(\epsilon)}+\widetilde{\mathcal{O}}(\sqrt{1/N}))}\right),\tag{15}$$

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. Thus, we can conclude that $\lim_{N,k\to\infty}\sup_{\theta\in\Theta}|h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) - \mathcal{L}(\theta,k,N)| = 0$, as we can choose arbitrarily small ϵ,δ. $\square$

Claim 3 (Double iterated limit of supremum)**.** *It holds that:*

$$\lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta}\left|\mathcal{L}(\theta,k,N) - h\left(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{LLM_\theta}((\ell_t)_{t\in[T]})\right)\right| = 0.$$

Proof. Since $h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) \geq \mathcal{L}(\theta,k,N)$, we will prove

$$\lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta} h\left(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right) - \mathcal{L}(\theta,k,N) = 0.$$

Lemma 10. $\frac{\sum_{i=1}^N f(X_i,k_1)h(X_i)}{\sum_{i=1}^N f(X_i,k_1)} \leq \frac{\sum_{i=1}^N f(X_i,k_2)h(X_i)}{\sum_{i=1}^N f(X_i,k_2)}$ *holds if* $0 < k_1 \leq k_2$ *for any real-valued* $(X_i)_{i\in[N]}$.

Proof. By multiplying $(\sum_{i=1}^N f(X_i,k_1))(\sum_{i=1}^N f(X_i,k_2))$ on both sides of the formula, we know that it 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

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

which is true since if $X_i \geq X_j$, $(f(X_i,k_1)f(X_j,k_2) - f(X_j,k_1)f(X_i,k_2)) \leq 0$ due to the log-increasing difference of f (Condition 1), as $\log f(X_j,k_1) - \log f(X_j,k_2) \geq \log f(X_i,k_1) - \log f(X_i,k_2)$ if $X_i \geq X_j$. $\square$

Therefore, $\mathcal{L}(\theta,k,N)$ is a non-decreasing function of k if N is fixed, which indicates that

$$\lim_{k\to\infty}\sup_{\theta\in\Theta} h\left(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right) - \mathcal{L}(\theta,k,N)$$

exists, as $\mathcal{L}(\theta,k,N)$ is also bounded. Therefore, by Lemma 5 and Claim 2, we know that

$$\lim_{N\to\infty}\lim_{k\to\infty}\sup_{\theta\in\Theta}\left|\mathcal{L}(\theta,k,N) - h\left(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right)\right|$$

exists and this value should be 0. $\square$

Claim 4. *It holds that*

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

Proof. Firstly, by Lemma 7, we have $\lim_{N,k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta,k,N) = \inf_{\theta\in\Theta} h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$. Plus, we already know that $\mathcal{L}(\theta,k,N)$ is a monotonically non-decreasing function of k for any fixed N (Lemma 10), and it is bounded, $\lim_{k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta,k,N)$ always exists. Therefore, by Lemma 5 , we also have $\lim_{N\to\infty}\lim_{k\to\infty}\inf_{\theta\in\Theta}\mathcal{L}(\theta,k,N) = \inf_{\theta\in\Theta} h(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$. $\square$

D.3 Deferred Proofs of Theorem 2 and Corollary 1

Theorem 2. (Generalization gap). *Suppose* LLM_θ *is Lipschitz-continuous with respect to the model parameter* θ*, then for any* $0 < \epsilon < 1/2$*, with probability at least* $1-\epsilon$*, we have*

$$\mathcal{L}\left(\widehat{\theta}_{k,N,N_T}, k, N\right) - \inf_{\theta\in\Theta} \mathcal{L}(\theta, k, N) \leq \widetilde{\mathcal{O}}\left(\sqrt{\frac{d_\theta + \log(1/\epsilon)}{N_T}}\right), \tag{5}$$

for any N *and sufficiently large* k*, where* d_θ *is the dimension of the parameter* θ*.*

Before proving the theorem, we remark on what LLM structure enjoys the Lipschitz-continuity. We provide two auxiliary results in the following proposition. The first result is from (Bai et al., 2023, Section J.1), which is about the Lipschitzness of Transformers. The second result is regarding processing the output of Transformers. In particular, the output of Transformers is usually not directly used, but passed through some matrix multiplication (by some matrix A), followed by some projection $\texttt{Operator}$ (to be specified later).

Proposition 2. *The* L*-layer Transformer* TF_θ *as defined in Appendix A.1 is* C_{TF}*-Lipschitz continuous with respect to* θ *with* $C_{TF} := L\left((1+B_{TF}^2)(1+B_{TF}^2R^3)\right)^L B_{TF}R(1+B_{TF}R^2+B_{TF}^3R^2)$*, i.e.,*

$$\|TF_{\theta_1}(Z) - TF_{\theta_2}(Z)\|_{2,\infty} \leq C_{TF}\|\theta_1-\theta_2\|_{TF}$$

where $\|\cdot\|_{TF}$ *is as defined in Equation (9), and* R, Z, B_{TF} *are as introduced in Appendix A.1. Moreover, the function* $\texttt{Operator}(A\cdot TF_\theta(\cdot)_{-1})$ *is* $\|A\|_{op}C_{TF}$*-Lipschitz continuous with respect to* θ*, i.e.,*

$$\|\texttt{Operator}(A\cdot TF_{\theta_1}(Z)_{-1}) - \texttt{Operator}(A\cdot TF_{\theta_2}(Z)_{-1})\|_2 \leq \|A\|_{op}C_{TF}\|\theta_1-\theta_2\|_{TF}.$$

Here, $\texttt{Operator}$ *is either the projection operator onto some convex set, or the* $\texttt{Softmax}$ *function.*

Proof. The first result is from (Bai et al., 2023, Section J.1). The second result comes from

- If $\texttt{Operator}$ is a projection onto 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 and Pavel, 2017, Corollary 3).

Note that the only condition that we require for $\texttt{Operator}$ is its non-expansiveness. □

Proof of Theorem 2. Let C_{LLM} denote the Lipschitz-continuity constant for LLM_θ with respect to some norm $\|\cdot\|_{\mathrm{LLM}}$, where $\|\cdot\|_{\mathrm{LLM}}$ denotes any norm defined on the parameter space of LLM (e.g., the norm $\|\cdot\|_{\mathrm{TF}}$ above in Proposition 2). Now, we prove that regret is also a Lipschitz-continuous function with respect to the LLM's parameter.

Lemma 11 (Lipschitzness of regret). *The function* $Regret_{LLM_\theta}$ *is* $C_{Reg} := BC_{LLM}T$*-Lipschitz continuous with respect to* θ*, i.e.,*

$$\left|\mathrm{Regret}_{LLM_{\theta_1}}((\ell_t)_{t\in[T]}) - \mathrm{Regret}_{LLM_{\theta_2}}((\ell_t)_{t\in[T]})\right| \leq C_{Reg}\|\theta_1-\theta_2\|_{LLM}.$$

Proof. By definition, we have

$$\begin{aligned}\left|\mathrm{Regret}_{\mathrm{LLM}_{\theta_1}}((\ell_t)_{t\in[T]}) - \mathrm{Regret}_{\mathrm{LLM}_{\theta_2}}((\ell_t)_{t\in[T]})\right| &= \left|\sum_{t=1}^T \langle \ell_t, \mathrm{LLM}_{\theta_1}(Z_{t-1}) - \mathrm{LLM}_{\theta_2}(Z_{t-1})\rangle\right| \\ &= B\sum_{t=1}^T \|\mathrm{LLM}_{\theta_1}(Z_{t-1}) - \mathrm{LLM}_{\theta_2}(Z_{t-1})\| \\ &\leq BC_{\mathrm{LLM}}T\|\theta_1-\theta_2\|_{\mathrm{LLM}}\end{aligned}$$

where $Z_t := (\ell_1, \ldots, \ell_t, c)$ for all $t\in[T]$ and $Z_0 = (c)$ where c is a d-dimensional vector. □

Now, we will prove the Lipschitzness of

$$C\left((\ell_t^{(j)})_{t\in[T],j\in[N]},k,\theta\right):=\frac{\sum_{j\in[N]}h(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]}f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[t]}),k)} \tag{16}$$

with respect to the model parameter θ.

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

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

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

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

holds. Moreover, consider the following constrained optimization problem:

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

whose optimum is denoted as $F(R,\beta)$. Then, since $\|x\|_\infty\leq R$ and $\|y\|_\infty\leq R$ is a compact set, by Berge's Maximum Theorem (Berge, 1877), 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 a large enough β_R such that if $\beta>\beta_R$, $F(R,\beta)\leq 2$. □

Note that Claim 5 does not hold if either x_i or y_i is unbounded. Now, we will apply Claim 5 to Equation (16). We can guarantee that $\left|\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right|\leq\text{diam}(\Pi,\|\cdot\|_2)TB$.

Also, note that the domain of $h:\mathbb{R}\to\mathbb{R}^+$ is effectively *constrained* to the range that $\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$ can achieve, which means that we can regard h as $h:[-\text{diam}(\Pi,\|\cdot\|_2)TB,\text{diam}(\Pi,\|\cdot\|_2)TB]\to\mathbb{R}^+$. Due to the continuity of h', and the fact that h has a compact domain, we know that $h(\cdot)$ is C_h-Lipschitz continuous for some $C_h>0$ on this interval of $[-\text{diam}(\Pi,\|\cdot\|_2)TB,\text{diam}(\Pi,\|\cdot\|_2)TB]$.

Lemma 12 (Lipschitzness of C in Equation (16))**.** *The function C in Equation (16) is $C_{cost}:=2C_hC_{Reg}$-Lipschitz continuous with respect to θ, if $k>k_{diam(\Pi,\|\cdot\|_2)TB}$ for some $k_{diam(\Pi,\|\cdot\|_2)TB}>0$, i.e.,*

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

Proof.

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

Here, (i) holds due to Claim 5, (ii) holds since h is C_h-Lipschitz continuous on the range of $\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$, and (iii) holds due to Lemma 11. □

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 of Y 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 set.*

By (Wainwright, 2019, Example 5.8), for any $r > 0$, we can verify that the δ-covering number $N(\delta; B(0, r, \|\cdot\|_{\mathrm{LLM}}), \|\cdot\|_{\mathrm{LLM}})$ can be bounded by

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

where d_θ is the dimension of the LLM's whole parameter. For example, if we use the $\|\cdot\|_{\mathrm{TF}}$ and consider the Transformer model as defined in Appendix A.1, for any $r > 0$,

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

Since we consider a compact Θ (as several recent works on analyzing Transformers (Bai et al., 2023; Lin et al., 2024)), let $R_\Theta := \mathrm{diam}(\Theta, \|\cdot\|_{\mathrm{LLM}})$ (which corresponds to B_{TF} for the Transformer models as defined in Appendix A.1, with $\|\cdot\|_{\mathrm{LLM}} = \|\cdot\|_{\mathrm{TF}}$), then there exists a set Θ_0 with $\log|\Theta_0| = d_\theta \log(1 + 2R_\Theta/\delta)$ such that for any $\theta \in \Theta$, there exists a $\theta_0 \in \Theta_0$ with

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

Then, by the standard result from statistical learning theory (Wainwright, 2019, Chapter 5), when trained with N_T samples, for every $0 < \epsilon < 1/2$, with probability at least $1 - \epsilon$, we have

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

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

$$\mathcal{L}(\widehat{\theta}_{k,N,N_T}, k, N) - \inf_{\theta \in \Theta} \mathcal{L}(\theta, k, N) \leq \widetilde{\mathcal{O}}\left(\sqrt{\frac{\log|\Theta_0| + \log(1/\epsilon)}{N_T}} \right)$$

with probability at least $1 - \epsilon$, completing the proof. □

Corollary 1. (Regret). *Suppose*[†] *for any $k \in \mathbb{N}^+$, $h, f(\cdot, k)$ are non-decreasing, and $\log f$ is a supermodular function (i.e., $\log f(R_1, k_1) - \log f(R_1, k_2) \geq \log f(R_2, k_1) - \log f(R_2, k_2)$ for $R_1 \geq R_2$ and $k_1 \geq k_2$). Then, with high probably, we have*

$$h\left(\lim_{N \to \infty} \lim_{k \to \infty} \max_{\|\ell_t\|_\infty \leq B} \mathrm{Regret}_{\mathrm{LLM}_{\widehat{\theta}_{k,N,N_T}}}\left((\ell_t)_{t \in [T]} \right) \right) \leq h\left(\inf_{\theta \in \Theta} \max_{\|\ell_t\|_\infty \leq B} \mathrm{Regret}_{\mathrm{LLM}_\theta}\left((\ell_t)_{t \in [T]} \right) \right) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{d_\theta}{N_T}} \right). \quad (6)$$

Proof. The limit on the right-hand side of Equation (5) remains as $\widetilde{\mathcal{O}}\left(\sqrt{\frac{d_\theta + \log(1/\epsilon)}{N_T}} \right)$, since we firstly take $\lim_{k \to \infty}$ and then take $\lim_{N \to \infty}$, thanks to the fact that Theorem 2 holds for large enough k and

[†] Note that these conditions on h, f are in addition to those specified after Equation (3).

any N. Next, we have

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

due to the continuity of h and Claim 3. Finally, we have

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

due to Claim 4, which, combined with the fact that h is non-decreasing, completes the proof. □

Remark 5 (Dynamic-regret loss). *So far, we have focused on the canonical online learning setting with regret being the metric. One can also generalize the results to the non-stationary setting, with dynamic regret being the metric. Specifically, one can define the* dynamic-regret-loss *function as follows:*

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

Then, one can also establish similar results as before, since the analysis does not utilize other properties of the regret except its boundedness, and the Lipschitz-continuity of LLM with respect to θ. To be specific, Lemma 11 holds due to the reason that we can bound the difference of the regret with the term

$$\left|\sum_{t=1}^{T}\langle\ell_t,(LLM_{\theta_1}(Z_{t-1})-LLM_{\theta_2}(Z_{t-1}))\rangle\right|,$$

as well as the fact that $\inf_{\pi_i\in\Pi}\langle\ell_i,\pi_i\rangle$ will be canceled. One can verify that all the arguments in Appendix D.2 also hold for similar reasons.

D.4 Deferred Proof of Theorem 3

Theorem 3. *Consider the policy space $\Pi=B(0,R_\Pi,\|\cdot\|)$ for some $R_\Pi>0$. The configuration of a single-layer self-attention model in Equation (7) (V,K,Q,v_c,k_c,q_c) such that $K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d$ and*

$$V=-R_\Pi\frac{T}{\sum_{t=1}^{T-1}1/t}\Sigma^{-1}\mathbb{E}\left[\left\|\sum_{t=1}^{T}\ell_t\right\|\ell_1\ell_2^\intercal\right]\Sigma^{-1}$$

is a first-order stationary point of Equation (3) with $N=1$, $h(x)=x^2$. Moreover, if Σ is a diagonal matrix, then plugging this configuration into Equation (7), and projecting the output with $Proj_{\Pi,\|\cdot\|}$ would perform FTRL with an L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.

Proof. We will locally use $\mathcal{A} = [d]$ without losing generality as $\mathcal{A}$ is finite with $|\mathcal{A}| = d$, and will interchangeably use $\ell_i(j)$ and ℓ_{ij} for notational convenience. Define $a := K^\intercal(Qc + q_c) \in \mathbb{R}^d$ and $b_{t-1} := \beta \mathbf{1}_{t-1} := k_c^\intercal(Qc + q_c)\mathbf{1}_{t-1} \in \mathbb{R}^{t-1}$. With $N = 1$, $h(x) = x^2$, and the choice of Π, the loss function (Equation (3)) can be written as follows:

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

where for $t = 1$, we use the output of the single-layer self-attention as v_c and we will write it as $(V\ell_{1:0} + v_c\mathbf{1}_0^\intercal)\mathrm{Softmax}(\ell_{1:0}^\intercal a + b_0)$ for notational consistency with $t \geq 2$. Also, we will define empty sum $\sum_{i=1}^{0} a_i = 0$ for any sequence $(a_i)_{i\in\mathbb{N}^+}$.

Step 1. Calculating $\frac{\partial f}{\partial a}$.

For $x \in [d]$, we calculate the corresponding directional derivative with the following equation for $t \geq 2$:

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

Plugging $a = \mathbf{0}_d$ and $v_c = \mathbf{0}_d$, and $(b_t = \beta\mathbf{1}_t)_{t\in[T-1]}$ provides

$$\begin{aligned}
&\frac{\partial}{\partial a_x} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathrm{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1})\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}} \\
&= \sum_{i=1}^{t-1} \frac{\ell_t^\intercal V\ell_i \ell_{ix}}{(t-1)} - \sum_{i=1}^{t-1} \frac{\ell_t^\intercal V \ell_i \left(\sum_{s=1}^{t-1}\ell_{sx}\right)}{(t-1)^2}.
\end{aligned}$$

For $t = 1$, as $\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathrm{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) = \ell_1^\intercal v_c$, $\frac{\partial}{\partial a_x}\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathrm{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1})\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}} = 0$, so we can use the same formula as $t \geq 2$ with empty sum $\sum_{i=1}^{t-1}$. Using the above calculation, we can further compute $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}}$ as follows:

$$
\begin{aligned}
&\frac{\partial f(V, a, (b_t)_{t\in[T-1]}, v_c)}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&= \mathbb{E}\frac{\partial}{\partial a_x}\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)^2\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&\qquad \frac{\partial}{\partial a_x}\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}}\Bigg] \\
&= \mathbb{E}\left[\left(\sum_{t=1}^{T} \ell_t^\intercal V \sum_{i=1}^{t-1}\frac{1}{t-1}\ell_i + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\sum_{t=1}^{T}\left(\sum_{i=1}^{t-1}\frac{\ell_t^\intercal V \ell_i \ell_{ix}}{(t-1)} - \sum_{i=1}^{t-1}\frac{\ell_t^\intercal V\ell_i\left(\sum_{s=1}^{t-1}\ell_{sx}\right)}{(t-1)^2}\right)\right] \qquad (17) \\
&= 0,
\end{aligned}
$$

where we used the fact that ℓ_i is drawn from a symmetric distribution, and flipping the sign of the variable as $-\ell_i$ yields the same distribution, which leads to the following:

$$
\begin{aligned}
&\mathbb{E}\left[\left(\sum_{t=1}^{T} \ell_t^\intercal V \sum_{i=1}^{t-1}\frac{1}{t-1}\ell_i + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\sum_{t=1}^{T}\left(\sum_{i=1}^{t-1}\frac{\ell_t^\intercal V \ell_i \ell_{ix}}{(t-1)} - \sum_{i=1}^{t-1}\frac{\ell_t^\intercal V\ell_i\left(\sum_{s=1}^{t-1}\ell_{sx}\right)}{(t-1)^2}\right)\right] \\
&\quad = \mathbb{E}\left[\left(\sum_{t=1}^{T} \ell_t^\intercal V \sum_{i=1}^{t-1}\frac{1}{t-1}\ell_i + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\sum_{t=1}^{T}\left(-\sum_{i=1}^{t-1}\frac{\ell_t^\intercal V \ell_i \ell_{ix}}{(t-1)} + \sum_{i=1}^{t-1}\frac{\ell_t^\intercal V\ell_i\left(\sum_{s=1}^{t-1}\ell_{sx}\right)}{(t-1)^2}\right)\right].
\end{aligned}
$$

This yields Equation (17)=0.

Step 2. Calculating $\frac{\partial f}{\partial v_c}$.
We will use the following equation for $t \geq 2$:

$$
\begin{aligned}
&\frac{\partial}{\partial v_c}\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) \\
&\qquad = \frac{\partial}{\partial v_c}\sum_{i=1}^{t-1}\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)e_i\frac{\exp(e_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))} = \ell_t.
\end{aligned}
$$

For $t = 1$, we define $\frac{\partial}{\partial v_c}\ell_1^\intercal (V\ell_{1:0} + v_c \mathbf{1}_0^\intercal)\mathtt{Softmax}(\ell_{1:0}^\intercal a + b_0) = \ell_1$, so that we can use the same formula as $t \geq 2$. Therefore, we can calculate $\frac{\partial f}{\partial v_c}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}}$ as follows:

$$
\begin{aligned}
&\frac{\partial f(V, a, (b_t)_{t\in[T-1]}, v_c)}{\partial v_c}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&= \mathbb{E}\frac{\partial}{\partial v_c}\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)^2\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&\qquad \frac{\partial}{\partial v_c}\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}}\Bigg] \\
&= \mathbb{E}\left[\left(\sum_{t=2}^{T} \ell_t^\intercal V \sum_{i=1}^{t-1}\frac{1}{t-1}\ell_i + R_\Pi \|\sum_{t=1}^{T}\ell_t\|_2\right)\sum_{t=1}^{T}\ell_t\right] = 0.
\end{aligned}
$$

The last line is due to the same reason as the last part of Step 1.

Step 3. Calculating $\frac{\partial f}{\partial V}$.

We calculate the following equation, which will be used to calculate $\frac{\partial f}{\partial V}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}}$ for $t \geq 2$:

$$
\begin{aligned}
&\frac{\partial}{\partial V}\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\mathtt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1})\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&\quad = \frac{\partial}{\partial V}\sum_{i=1}^{t-1}\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)e_i \frac{\exp(e_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} \\
&\quad = \sum_{i=1}^{t-1}\ell_t \ell_i^\intercal \frac{\exp(e_i^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))}{\sum_{s=1}^{t-1}\exp(e_s^\intercal(\ell_{1:t-1}^\intercal a + b_{t-1}))}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta \mathbf{1}_t)_{t\in[T-1]}} = \sum_{i=1}^{t-1}\frac{1}{t-1}\ell_t\ell_i^\intercal.
\end{aligned}
$$

For $t = 1$, note that $\frac{\partial}{\partial V}\ell_t^\intercal v_c = \boldsymbol{O}_{d\times d}$, so we can use the same formula as $t \geq 2$ with empty sum $\sum_{i=1}^{t-1}$.

Therefore, we have

$$
\begin{aligned}
&\frac{\partial f(V,a,(b_t)_{t\in[T-1]},v_c)}{\partial V}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]}}\\
&=\mathbb{E}\frac{\partial}{\partial V}\left(\sum_{t=1}^{T}\ell_t^\intercal(V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\intercal)\mathtt{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,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]}}\\
&=\mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\ell_t^\intercal(V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\intercal)\mathtt{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,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]}}\\
&\qquad\frac{\partial}{\partial V}\left(\sum_{t=1}^{T}\ell_t^\intercal(V\ell_{1:t-1}+v_c\mathbf{1}_{t-1}^\intercal)\mathtt{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,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]}}\Bigg]\\
&=\mathbb{E}\left[\left(\sum_{t=1}^{T}\ell_t^\intercal V\sum_{i=1}^{t-1}\frac{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 $V^\star = R_\Pi\frac{T}{\sum_{t=1}^{T-1}1/t}\Sigma^{-1}\mathbb{E}\left[\|\sum_{t=1}^{T}\ell_t\|_2\ell_t\ell_i^\intercal\right]\Sigma^{-1}$, then $\frac{\partial f}{\partial V}\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]},V=V^\star} = \boldsymbol{O}_{d\times d}$. Lastly, we have

$$
\begin{aligned}
\frac{\partial f}{\partial K}\Big|_{K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d,V=V^\star}&=\left(\frac{\partial f}{\partial a}\frac{\partial a}{\partial K}\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]},V=V^\star}=\boldsymbol{O}_{d\times d}\\
\frac{\partial f}{\partial Q}\Big|_{K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d,V=V^\star}&=\left(\frac{\partial f}{\partial a}\frac{\partial a}{\partial Q}\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]},V=V^\star}=\boldsymbol{O}_{d\times d}\\
\frac{\partial f}{\partial q_c}\Big|_{K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d,V=V^\star}&=\left(\frac{\partial f}{\partial a}\frac{\partial a}{\partial q_c}\right)\Big|_{a=\mathbf{0}_d,v_c=\mathbf{0}_d,(b_t=\beta\mathbf{1}_t)_{t\in[T-1]},V=V^\star}=\mathbf{0}_d
\end{aligned}
$$

which means that such configurations are first-order stationary points of Equation (3) with $N=1$, $h(x)=x^2$, and $\Pi=B(0,R_\Pi,\|\cdot\|)$. □

D.5 Deferred Proof of Theorem 4

Theorem 4. *Consider the policy space $\Pi=B(0,R_\Pi,\|\cdot\|)$ for some $R_\Pi>0$. The configuration of a single-layer linear self-attention model in Equation (8) (V,K,Q,v_c,k_c,q_c) such that $K^\intercal(Qc+q_c)=v_c=\mathbf{0}_d$ and $V=-2R_\Pi\Sigma^{-1}\mathbb{E}\left(\|\sum_{t=1}^{T}\ell_t\|\ell_1\ell_2^\intercal\right)\Sigma^{-1}$ is **a global optimal solution** of Equation (3) with $N=1$, $h(x)=$*

x^2. *Moreover, every global optimal configuration of Equation (3) within the parameterization class of Equation (8) has the same output function g. Additionally, if Σ is a diagonal matrix, then plugging any global optimal configuration into Equation (8), and projecting the output with $Proj_{\Pi,\|\cdot\|}$ would perform FTRL with an L_2-regularizer for the loss vectors $(\ell_t)_{t\in[T]}$.*

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

$$
\begin{aligned}
&g(Z_t; V, K, Q, v_c, k_c, q_c) \\
&= \sum_{i=1}^{t}\Big(V\ell_i\ell_i^\intercal(K^\intercal(Qc+q_c)) + \Big(Vk_c^\intercal(Qc+q_c) + v_c(Qc+q_c)^\intercal K\Big)\ell_i + v_c k_c^\intercal(Qc+q_c)\Big),
\end{aligned} \tag{18}
$$

which can be expressed with a larger class

$$
g(Z_t, \mathbb{A}, \beta, \mathbb{C}, \delta) := \sum_{i=1}^{t}(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta), \tag{19}
$$

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

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

can be expressed as $\mathbb{A} = V, \beta = K^\intercal(Qc+q_c), \mathbb{C} = Vk_c^\intercal(Qc+q_c) + v_c(Qc+q_c)^\intercal K, \beta = v_c k_c^\intercal(Qc+q_c)$, then we can conclude that the 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 the corresponding V, Q, K, v_c, q_c, k_c constitute a minimizer among a larger class. Now, since $\Pi = B(\mathbf{0}_d, R_\Pi, \|\cdot\|)$, we can rewrite f as

$$
f(\mathbb{A}, \beta, \mathbb{C}, \delta) = \mathbb{E}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}\Big(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta\Big)\rangle + R_\Pi\left\|\sum_{t=1}^{T}\ell_i\right\|_2\right)^2. \tag{20}
$$

Step 1. Finding condition for $\frac{\partial f}{\partial \delta} = 0$.
Due to the Leibniz rule, if we calculate the partial derivative of Equation (20) w.r.t. δ, we have

$$
\begin{aligned}
\frac{\partial f(\mathbb{A}, \beta, \mathbb{C}, \delta)}{\partial \delta} &= \frac{\partial}{\partial \delta}\mathbb{E}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}\Big(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta\Big)\rangle + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)^2 \\
&= \mathbb{E}\frac{\partial}{\partial \delta}\left(\sum_{t=1}^{T}\langle \ell_t, \sum_{i=1}^{t-1}\Big(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta\Big)\rangle + R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)^2 \\
&= \mathbb{E}\sum_{t=1}^{T}\ell_t\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}(t-1)\ell_t^\intercal\Big(\mathbb{A}\ell_i\ell_i^\intercal\beta + \mathbb{C}\ell_i + \delta\Big) + R_\Pi\|\sum_{t=1}^{T}\ell_t\|\right).
\end{aligned} \tag{21}
$$

Since the expectation of either odd-order polynomial or even-order polynomial times $\|\cdot\|_2$ is 0, due to that ℓ_t follows a symmetric distribution, we have

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

Now, we calculate

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

where (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,

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

Plugging the above equations into Equation (21), we have

$$\frac{\partial f(\mathbb{A},\beta,\mathbb{C},\delta)}{\partial\delta}=\frac{1}{6}T(2T^2-3T+1)(\Sigma\mathbb{A}\Sigma\beta+\Sigma\delta).$$

Due to the optimality condition, we have

$$\mathbb{A}\Sigma\beta+\delta=0. \tag{22}$$

Step 2. Plugging the optimality condition for $\frac{\partial f}{\partial\delta}$ into Equation (20).
Plugging Equation (22) to Equation (20), 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

$$
\begin{aligned}
\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal \mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right)^2 &= \mathbb{E}\left[\sum_{t_1=1}^{T}\sum_{i_1=1}^{t_1-1}\sum_{t=1}^{T}\sum_{i=1}^{t-1}\beta^\intercal(\ell_{i_1}\ell_{i_1}^\intercal-\Sigma)\mathbb{A}^\intercal\ell_{t_1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] \\
&\underset{(1)}{=} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i_1=1}^{t-1}\sum_{i=1}^{t-1}\beta^\intercal(\ell_{i_1}\ell_{i_1}^\intercal-\Sigma)\mathbb{A}^\intercal\ell_t\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] \\
&\underset{(2)}{=} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{t-1}\beta^\intercal(\ell_i\ell_i^\intercal-\Sigma)\mathbb{A}^\intercal\ell_i\ell_i^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right] \\
&= \frac{(T-1)T}{2}\beta^\intercal\mathbb{E}\Big[(\ell_i\ell_i^\intercal-\Sigma)\mathbb{A}^\intercal\Sigma\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\Big]\beta \qquad (23)\\
&= \frac{(T-1)T}{2}\beta^\intercal\mathbb{E}\Big[(\sqrt{\Sigma}A(\ell_i\ell_i^\intercal-\Sigma))^\intercal(\sqrt{\Sigma}A(\ell_i\ell_i^\intercal-\Sigma))\Big]\beta.
\end{aligned}
$$

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 polynomials of odd degrees and we have $Z \overset{d}{=} -Z$. Note that Equation (23) is minimized when $\mathbb{P}(\sqrt{\Sigma}\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta = \mathbf{0}_d) = 1$.

If $\mathbb{A} \neq \boldsymbol{O}_{d\times d}$, suppose that the singular value decomposition of $A = U\Lambda V$ yields that Λ is a diagonal matrix whose first diagonal element is non-zero, and U, V are orthogonal matrices. Then, we want to find β that $\sqrt{\Sigma}U\Lambda V(\ell_i\ell_i^\intercal - \Sigma)\beta = \mathbf{0}_d$ for any ℓ_i such that $p(\ell_i) \neq 0$, where p indicates the probability density function of loss vectors. Since Σ and U are invertible, we only need to consider $\Lambda V(\ell_i\ell_i^\intercal - \Sigma)\beta = \mathbf{0}_d$. Since Λ's first diagonal component is non-zero, we will consider equation $e_1^\intercal\Lambda V(\ell_i\ell_i^\intercal-\Sigma)\beta = 0$. This is equivalent to $V_1(\ell_i\ell_i^\intercal-\Sigma)\beta = 0$, where V_1 is the first row of V, and is a non-zero vector.

Now, we will generally consider $a_{x,y}(v) := vv^\intercal x - y$ where $x, y, v \in \mathbb{R}^d$ and $a_{x,y} : B(\mathbf{0}_d, 2\epsilon_1, \|\cdot\|) \to \mathbb{R}^d$ function. Then, we can check that the Jacobian of $a_{x,y}(v)$ is $vx^\intercal + (v\cdot x)I$, and we can find that the determinant of the Jacobian is nonzero when $v = \epsilon_1 x$ if $x \neq \mathbf{0}_d$. Therefore, the volume of $(V_1(\ell_i\ell_i^\intercal - \Sigma))$ for $\ell_i \in B(\mathbf{0}_d, c_z, \|\cdot\|)$ is greater than the volume of $(V_1(vv^\intercal - \Sigma))$ for $v \in B(\epsilon_1 V_1^\intercal, \epsilon_2, \|\cdot\|)$, where c_z is a constant such that $B(\mathbf{0}_d, c_z, \|\cdot\|) \subseteq \operatorname{supp}(Z)$, and $\epsilon_1, \epsilon_2 > 0$ satisfy that $\epsilon_1|V_1| + \epsilon_2 < c_z$. Here, we define $\epsilon_2 > 0$ sufficiently small so that the determinant of Jacobian$(vv^\intercal V_1^\intercal - \Sigma V_1^\intercal) > 0$ for $v \in B(\epsilon_1 V_1^\intercal, \epsilon_2, \|\cdot\|)$, and $v \to vv^\intercal V_1^\intercal - \Sigma V_1^\intercal$ is a one-to-one correspondence, by inverse function theorem. Therefore, the volume of $(V_1(vv^\intercal - \Sigma))$ for $v \in B(\epsilon_1 V_1^\intercal, \epsilon_2, \|\cdot\|)$ can be calculated as

$$
[\text{Volume } (V_1(vv^\intercal - \Sigma)) \text{ for } v \in B(\epsilon_1 V_1^\intercal, \epsilon_2, \|\cdot\|)] = \int_{v\in B(\epsilon_1 V_1^\intercal, \epsilon_2, \|\cdot\|)} \Big|\det(\text{Jacobian}(V_1(vv^\intercal-\Sigma)))\Big| dv > 0.
$$

Therefore, Volume$(V_1(vv^\intercal - \Sigma))$ where $v \in B(\epsilon_1 V_1^\intercal, \epsilon_2, \|\cdot\|)$ is non-zero, so that we can find d loss vectors $\{\ell_i\}_{i\in[d]}$ such that the vectors $\{V_1(\ell_i\ell_i^\intercal - \Sigma)\}_{i\in[d]}$ are linearly independent. Hence, if we want to minimize Equation (23), either $A = \boldsymbol{O}_{d\times d}$ or $\beta = \mathbf{0}_d$ should hold. In both cases, Equation (19) 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 (18)) with $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$.

Step 3. Calculating $\frac{\partial f}{\partial \mathbb{C}}$.

Now, we optimize over $\mathbb{C}$, by minimizing the following objective:

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

Here, (i) can be calculated as follows:

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

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 $\frac{\partial f(\mathbb{C})}{\partial \mathbb{C}}$:

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

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

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

If $a \neq b$, then since Z is symmetric, the term above becomes zero. Therefore, we only need to consider the case when $a = b$, which is $\mathbb{E}_\ell\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2}\ell_{ia}\ell_{ka}\right]$, and it will be the same value for all $a \in [d]$ since ℓ_i's coordinates are independent.

Now, we calculate the scale of $\mathbb{E}_\ell\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2}\ell_{i1}\ell_{k1}\right]$. We have $Z := \frac{\sum_{o=1}^{d-1}(\sum_{s=1}^{T}\ell_{so})^2}{T(d-1)} \overset{a.s.}{\to} 1$ as $d \to \infty$ (by the law of large numbers) and we define $W := \sum_{s\neq i,k}\ell_{s1}/\sqrt{T}$ which is independent of ℓ_{i1}

and ℓ_{k1}.

$$
\begin{aligned}
\mathbb{E}_{\ell}\left[\sqrt{\sum_{o=1}^{d}(\sum_{s=1}^{T}\ell_{so})^2}\ell_{i1}\ell_{k1}\right] &= \mathbb{E}_{Z,W,\ell_{i1},\ell_{k1}}\left[\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}+\ell_{k1})^2}\ell_{i1}\ell_{k1}\right] \\
&= \mathbb{E}_{Z,W,\ell_{i1},\ell_{k1}\geq 0}\left[\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}+\ell_{k1})^2}\ell_{i1}\ell_{k1}-\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}-\ell_{k1})^2}\ell_{i1}\ell_{k1}\right] \\
&= \mathbb{E}_{Z,W,\ell_{i1},\ell_{k1}\geq 0}\left[\frac{4(\sqrt{T}W+\ell_{i1})\ell_{k1}}{\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}+\ell_{k1})^2}+\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}-\ell_{k1})^2}}\ell_{i1}\ell_{k1}\right].
\end{aligned}
$$

Taking $d \to \infty$, we have

$$
\frac{\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}+\ell_{k1})^2}+\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}-\ell_{k1})^2}}{2\sqrt{Td}} \xrightarrow{d} 1,
$$

which further implies

$$
\begin{aligned}
&\sqrt{Td}\frac{4(\sqrt{T}W+\ell_{i1})\ell_{k1}}{\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}+\ell_{k1})^2}+\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}-\ell_{k1})^2}}\ell_{i1}\ell_{k1} \\
&\xrightarrow{d} \sqrt{Td}\frac{4(\sqrt{T}W+\ell_{i1})\ell_{k1}}{2\sqrt{Td}}\ell_{i1}\ell_{k1} = 2(\sqrt{T}W+\ell_{i1})\ell_{i1}\ell_{k1}
\end{aligned}
$$

as $d \to \infty$. Therefore,

$$
\begin{aligned}
&\lim_{d\to\infty}\mathbb{E}_{Z,W,\ell_{i1},\ell_{k1}\geq 0}\left[\sqrt{Td}\frac{4(\sqrt{T}W+\ell_{i1})\ell_{k1}}{\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}+\ell_{k1})^2}+\sqrt{T(d-1)Z+(\sqrt{T}W+\ell_{i1}-\ell_{k1})^2}}\ell_{i1}\ell_{k1}\right] \\
&= \mathbb{E}_{Z,W,\ell_{i1},\ell_{k1}\geq 0}\left[2(\sqrt{T}W+\ell_{i1})\ell_{i1}\ell_{k1}\right] = \mathbb{E}_{\ell_{i1},\ell_{k1}\geq 0}\left[\ell_{i1}^2\ell_{k1}\right]
\end{aligned}
$$

which is a constant. The last equality came from the fact that W, ℓ_{i1}, ℓ_{k1} are independent random variables, and expectation of ℓ_{i1} is zero. Therefore, the output of the single-layer linear self-attention provides us with online gradient descent with step-size $\Theta(R_{\Pi}/\sqrt{Td})$. In the online learning literature, we usually set the gradient step size as $\Theta(R_{\Pi}/\sqrt{Td})$ (Hazan, 2016, Theorem 3.1), which is consistent with the result above. □

D.6 Empirical Validation of Theorem 3 and Theorem 4

We now provide empirical validations for Theorem 3 and Theorem 4. We provide the training details and the results as follows.

D.6.1 Empirical Validation of Theorem 3

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 $g(Z_t; V, Q, K) = \sum_{i=1}^{t}(b-a)\ell_i$, effectively emulating the process of an online gradient descent method. We repeated 10 times of the experiments. For verifying $K^\intercal Q = \boldsymbol{O}_{d\times d}$, we will measure Frobenius norm ($\|\cdot\|_F$) 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 $\min_{a,b\in\mathbb{R}} \|V - (a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d})\|_F / b$. The results are demonstrated in the first plot of Figure 15.

D.6.2 Empirical Validation of Theorem 4

We now focus on two key convergence properties: $K^\intercal(Q\mathbf{1}_d + q_c)$ 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 $K^\intercal(Q\mathbf{1}_d + q_c) = \mathbf{0}_d$ and $V = a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$ imply that the function $g(Z_t; V, Q, K) = \sum_{i=1}^{t}(b-a)\ell_i$, effectively emulating the process of an online gradient descent method. We repeated 10 times. For verifying $K^\intercal(Q\mathbf{1}_d + q_c) = \mathbf{0}_d$, we will measure 2-norm of $K^\intercal(Q\mathbf{1}_d + q_c)$. Also for measuring the closeness of V and $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, we will measure $\min_{a,b\in\mathbb{R}} \|V - (a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d})\|_F / b$. The results are demonstrated in the second plot of Figure 15.

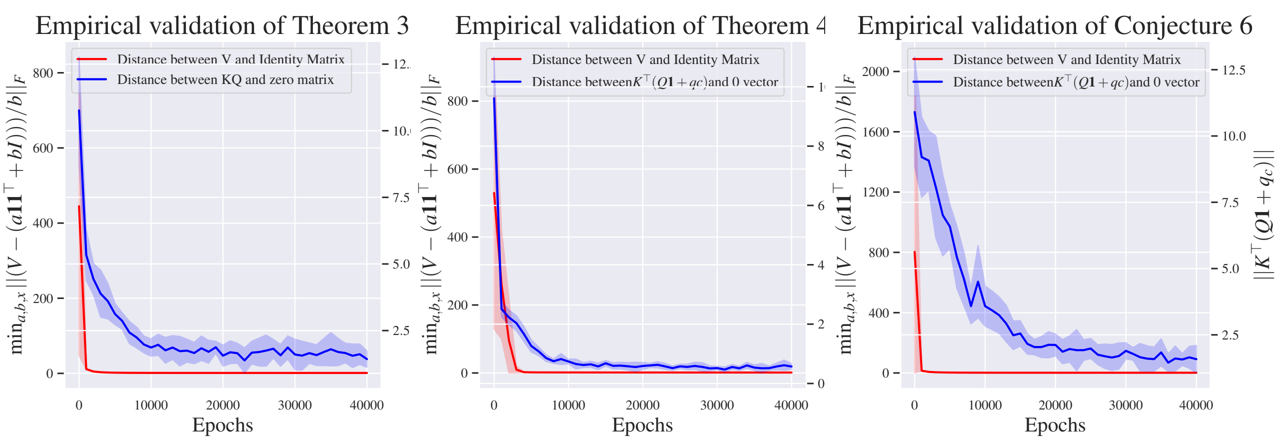

Figure 15: Empirical validation of Theorem 3 (top), Theorem 4 (middle), and Conjecture 6 (bottom). The observed convergence in Theorem 3 and Conjecture 6's result suggests that configuration in Theorem 3 and Conjecture 6 are not only the local optimal point, but it has the potential as being the global optimizer.

D.7 Discussions 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 is more amenable to the Experts Problem setting. To this end, we consider the following parameterization by adding an additional *non-linear* structure for the single-layer linear self-attention:

$$g(Z_t; V, K, Q, v_c, k_c, q_c) = \texttt{Operator}\left(\sum_{i=1}^{t}(V\ell_i + v_c)((K\ell_i + k_c))^\intercal \cdot (Qc + q_c)\right), \tag{24}$$

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

Conjecture 6. *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 (3) with $N = 1$ and $h(x) = x^2$ when LLM_θ is parameterized with Equation (24), $\mathtt{Operator} = \mathtt{Softmax}$, and $\Pi = \Delta(\mathcal{A})$. This configuration performs FTRL with an entropy regularizer which is a no-regret algorithm.*

We provide an idea for proving the conjecture, together with its numerical validation. Also, we have observed in Figure 15 that Theorem 3 and Conjecture 6 might also be a global optimizer, as training results have provided the configuration that Theorem 3 and Conjecture 6 have suggested.

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 $\frac{\partial f}{\partial v_c}$.

We use the following formula: for $x \in [d]$ and $t \geq 2$, we have

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

and for $t = 1$, $\frac{\partial}{\partial v_{cx}}\exp\left(e_y^\intercal \sum_{i=1}^{t}(V\ell_i\ell_i^\intercal a + (\beta V + v_c a^\intercal)\ell_i + v_c\beta)\right)\Big|_{a=\mathbf{0}_d, v_c = v\mathbf{1}_d, V=kI} = 0$, so we can use the same formula with $t \geq 2$. Thus, we have

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

Therefore,

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

Step 2. Calculating $\frac{\partial f}{\partial V}$.

The following formula will be used for calculating $\frac{\partial f}{\partial V}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$: for $r,c\in[d]$, we have

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

Therefore,

$$\begin{aligned}
&\frac{\partial f(V,a,\beta,v_c)}{\partial V_{rc}}\bigg|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&= \mathbb{E}\Bigg[\Bigg(\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}\Bigg) \\
&\qquad \frac{\partial}{\partial V_{rc}}\Bigg(\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}\Bigg)\Bigg]\Bigg|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} \\
&= \mathbb{E}\Bigg[\Bigg(\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}\Bigg) \\
&\qquad \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\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[\Bigg(\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}\Bigg)
\end{aligned}$$

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

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

We can observe the followings: 1) if $r_1 \neq c_1$ and $r_2 \neq c_2$, $\frac{\partial f}{\partial V_{r_1 c_1}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} = \frac{\partial f}{\partial V_{r_2 c_2}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ holds, and 2) $\frac{\partial f}{\partial V_{r_1 r_1}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} = \frac{\partial f}{\partial V_{r_2 r_2}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$.

Step 3. Calculating $\frac{\partial f}{\partial \beta}$.

The following formula will be used for calculating $\frac{\partial f}{\partial \beta}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$:

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

Further, we have

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

Step 4. Calculating $\frac{\partial f}{\partial a}$.

Note that

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

Therefore,

$$
\begin{aligned}
&\frac{\partial f(V,a,\beta,v_c)}{\partial a_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)\\
&\qquad\frac{\partial}{\partial a_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]\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(\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)\\
&\qquad\Bigg(\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}\\
&\qquad\quad-\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}\Bigg)\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)\\
&\qquad\Bigg(\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}\\
&\qquad\quad-\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}\Bigg)\Bigg]
\end{aligned}
$$

Note that the value does not depend on x, which means that $\frac{\partial f}{\partial a}\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}=\widetilde{c}\mathbf{1}_d$ for some constant $\widetilde{c}$.

D.7.1 Numerical Analysis of Step 2 and Step 4

In Steps 2 and 4 above, we were not able to show that a k whose value becomes zero exists. We hence provide some empirical evidence here. First, we attach the estimated $\frac{\partial f}{\partial V_{rc}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ $(r \neq c)$, $\frac{\partial f}{\partial V_{rr}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$, $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ and $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ graph with respect to k value when $\ell_{ts} \sim \mathrm{Unif}([0,1])$ for all $t \in [T], s \in [d]$. While the graph of $\frac{\partial f}{\partial V}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ is not stable, we can see that k for $\frac{\partial f}{\partial V_{rc}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} = 0$, $\frac{\partial f}{\partial V_{rr}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} = 0$ and $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI} = 0$ is very similar in Figure 16. We used the Monte Carlo estimation of $1{,}000{,}000$ times.

D.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 15.

D.8 Training Details of Section 5.4

We provide the training details of 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 trained for the loss sequences with the 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 of 512.

D.9 Ablation Study on Training Equation (3)

In this section, we provide an ablation study that changes N and k in Equation (3). To be specific, we will set $N = 1,2,4$, $f(x,k) = \max(x,0)^k$, $h(x) = \max(x,0)^2$, and $k = 1,2$. For the multi-layer Transformer training, we used 4 layers and 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 of 512. We experimented on the randomly generated loss sequences. Especially, we used the uniform loss sequence ($\ell_t \sim \mathrm{Unif}([0,10]^2)$), with the results in Figure 17 and Figure 18; and the Gaussian loss sequence ($\ell_t \sim \mathcal{N}(5\cdot\mathbf{1}_2, I)$), with the results in Figure 19 and Figure 20.

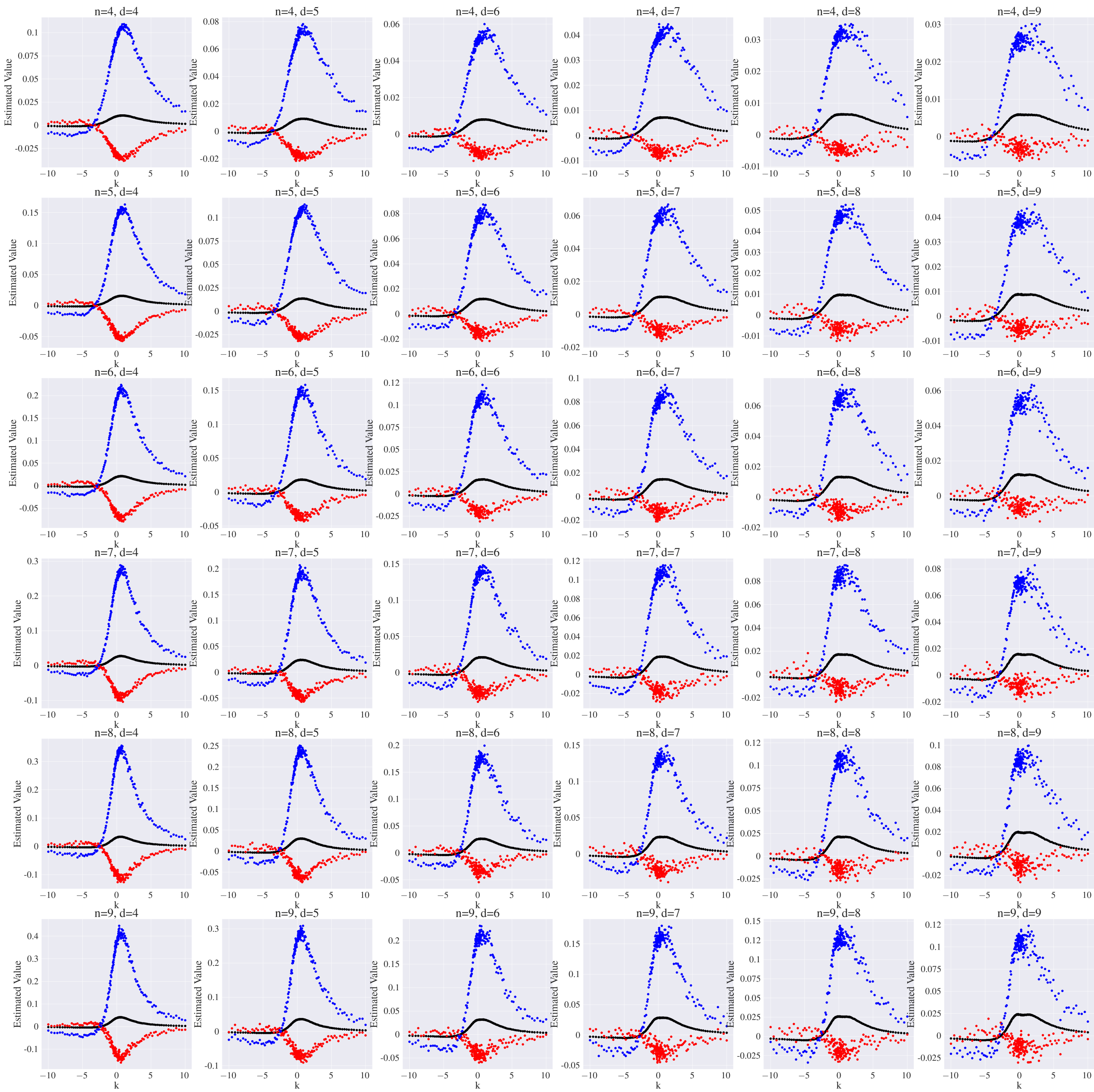

Figure 16: Calculation of $20\frac{\partial f}{\partial V_{rc}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ $(r \neq c)$(red), $20\frac{\partial f}{\partial V_{rr}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ (blue), and $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ (black). We experimented with $n \in [4,9]$ and $d \in [4,9]$. The figure might indicate that βk that makes the derivative zero of $\frac{\partial f}{\partial V_{rc}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ $(r \neq c)$, $\frac{\partial f}{\partial V_{rr}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$, and $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ would coincide.

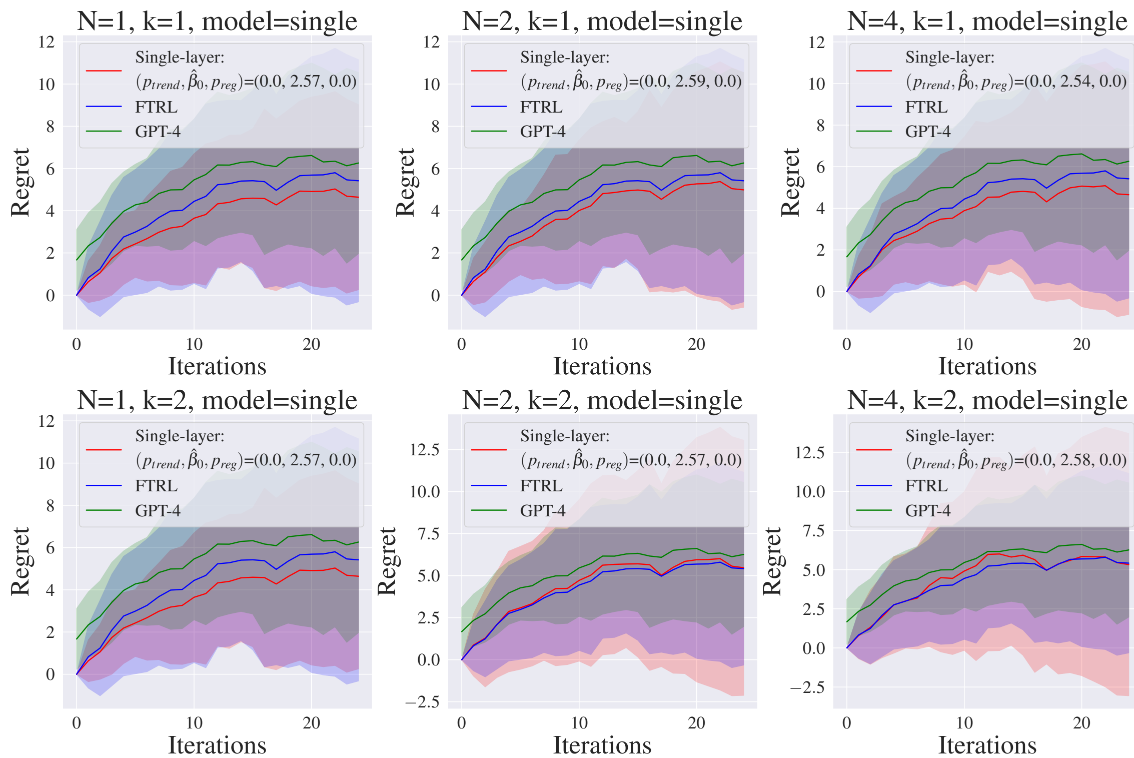

Figure 17: Ablation study for the uniform loss sequence trained with single-layer self-attention layer and Softmax projection.

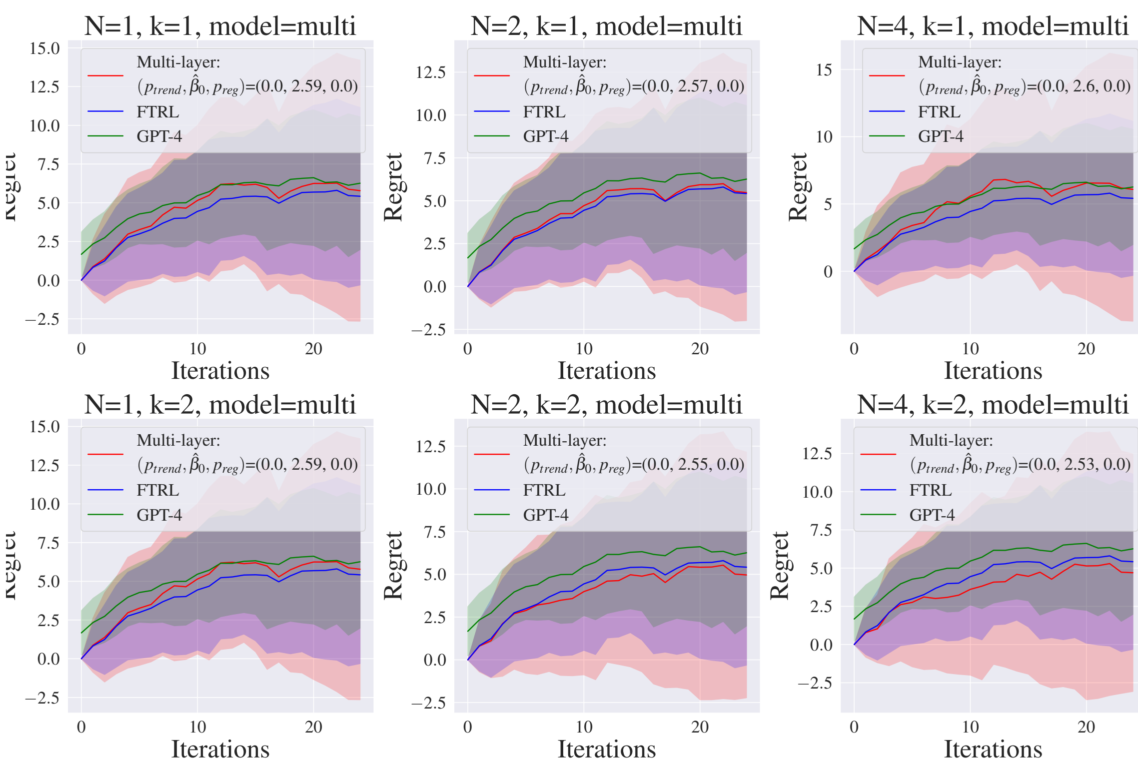

Figure 18: Ablation study for the uniform loss sequence trained with multi-layer self-attention layer and Softmax projection.

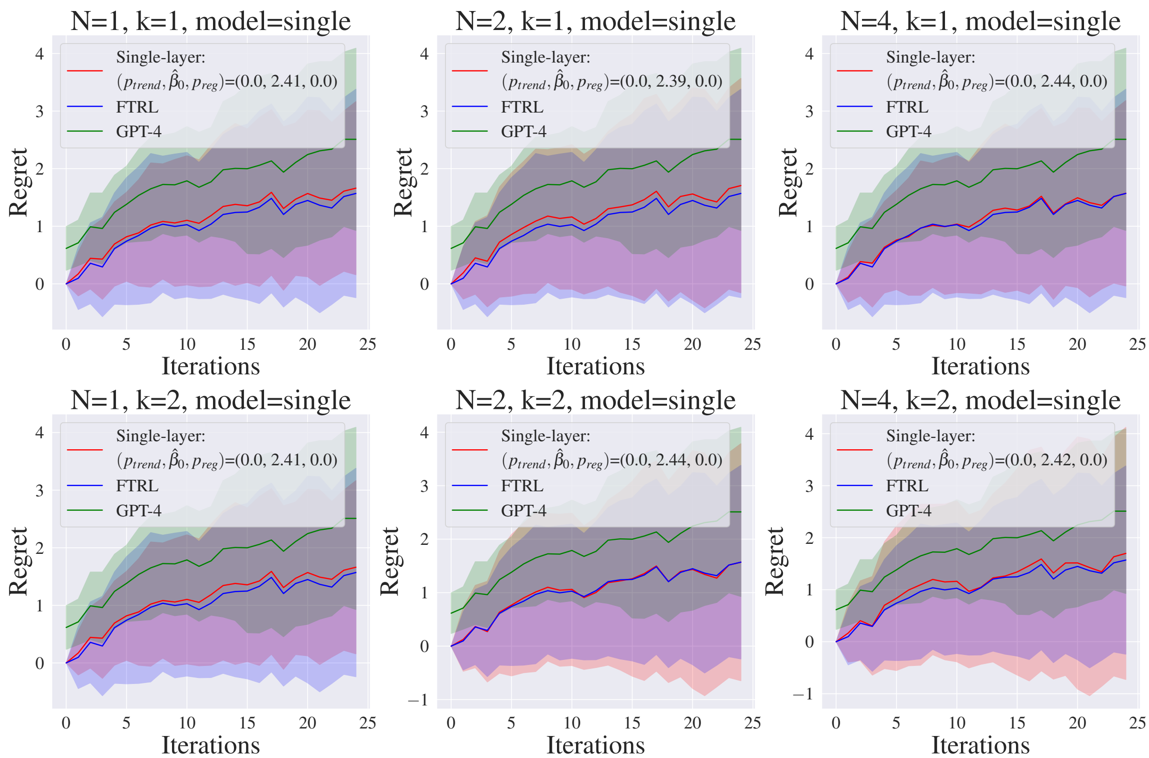

Figure 19: Ablation study for the Gaussian loss sequence trained with single-layer self-attention layer and Softmax projection.

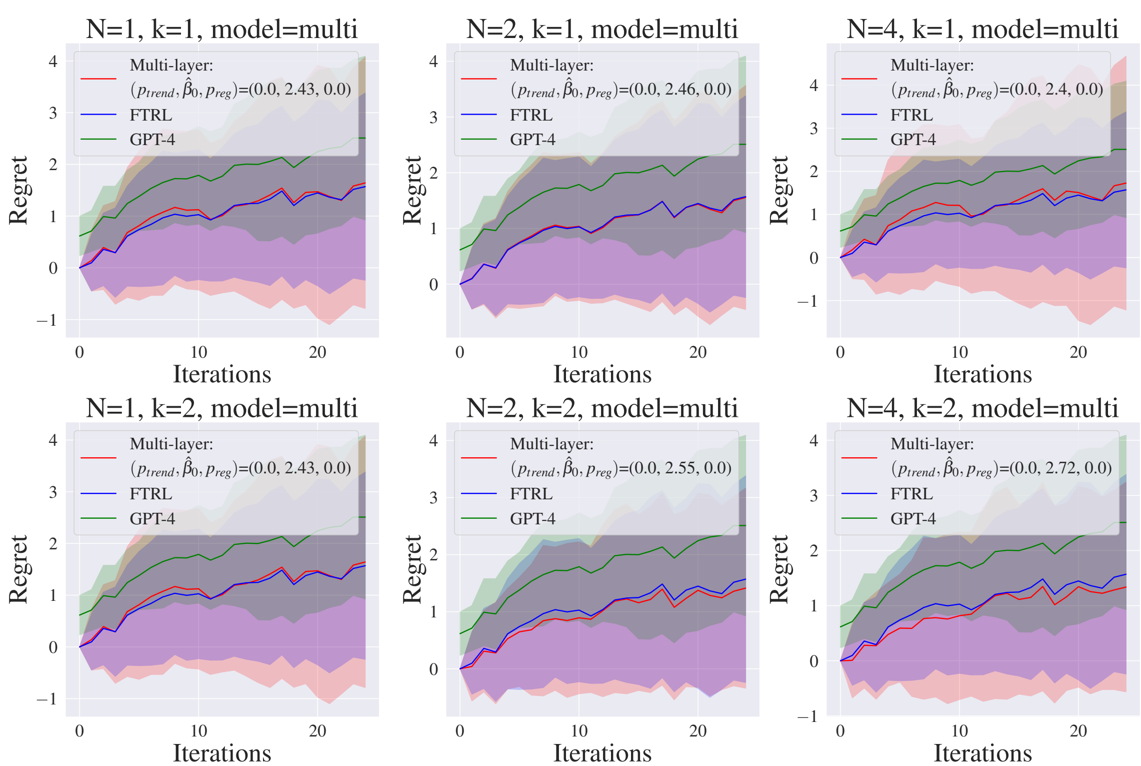

Figure 20: Ablation study for the Gaussian loss sequence trained with single-layer self-attention layer and Softmax projection.