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

CONTENTS

A Related Work

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

B Deferred Background

B.1 Notation

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

B.2 Additional Definitions

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

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

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

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

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

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

$$\|\theta\|_{\texttt{TF}} := \max_{l\in[L]} \left\{\|\theta^{(la)}\|_{\texttt{M- (L) SA}} + \|\theta^{(lm)}\|_{\texttt{MLP}}\right\}, \tag{B.1}$$

and $B_{\texttt{TF}} > 0$ 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 $\texttt{clip}$ does not take effect on any of our approximation results.

B.3 In-Context Learning

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

B.4 Online Learning Algorithms

Follow-the-regularized-leader (FTRL). The *follow-the-regularized-leader* 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 & 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 & Schapire, 1997) (also referred to as the Multiplicative Weight Update algorithm (Arora et al., 2012b)) 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 & Vempala, 2005) at each time step t adds a random perturbation vector ϵ_t to the original loss vectors and then selects the best-response action a_t (that is potentially randomized due to ϵ_t) by solving:

$$a_t \in \arg\min_{a\in\mathcal{A}} \ \epsilon_{ta} + \sum_{i=1}^{t-1} \ell_{ia},$$

where the perturbation ϵ_t is *sampled* from a pre-defined distribution. Correspondingly, the *policy* π_t is chosen by following equation:

$$\pi_t = \mathbb{E}\left[\arg\min_{\pi\in\Pi} \langle \epsilon_t, \pi \rangle + \sum_{i=1}^{t-1} \langle \ell_i, \pi \rangle\right]. \tag{B.2}$$

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 & Sandholm (2002) showed 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 & 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.

B.5 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 $f_t(\pi_{\mathscr{A},t}) - f_t(\pi) \leq \langle \nabla f_t(\pi_{\mathscr{A},t}), \pi_{\mathscr{A},t} - \pi \rangle$ for any $\pi \in \Pi$, which indicates

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

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 in that round.

B.6 Six Representative General-Sum Games

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

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.

C Deferred Results and Proofs in Section 3

C.1 Intuition Why Pre-Trained Language Models Might Exhibit No-Regret Behavior

Intuition why pre-trained language models might 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.

C.2 Visualization of Interaction Protocols

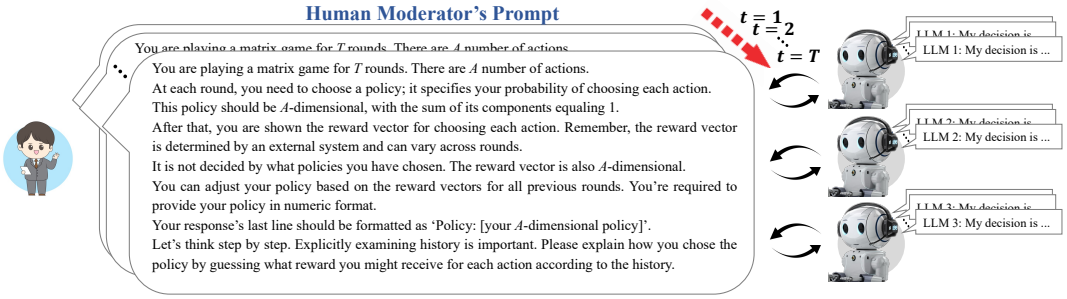

Figure C.1: Demonstration of the prompts and interaction protocol for multi-player repeated games. A human moderator does not provide the game's payoff matrices to the LLMs. Instead, at each round, the human moderator provides each player's own payoff vector history.

C.3 Frameworks for No-Regret Behavior Validation

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

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

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

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 with a finite T. As an alternative, we propose a more tractable hypothesis test, albeit a weaker one, that still captures the essence of our objective:

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

H_1 : The sequence $\left(\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t\right)_{t\in[T]}$ shows a decreasing pattern

where the "decreasing pattern" here refers to the case when *more than* $1/2$ of the elements in the sequence satisfies that $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t > \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t+1]}\right)/(t+1)$. Note that we will only apply the framework when the sequence $\left(\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t\right)_{t\in[T]}$ is non-negative, since a negative regret is even more favorable and directly implies no-regret behaviors.

Ideally, one should check if $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t$ approaches zero or some negative constant 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 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*

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

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

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

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

since $s \geq \frac{T-1}{2}$. □

On the underlying assumption for Equation (C.1). Our *trend-checking* framework was meant to be designed for general sequences $\{a_t\}_{t=1}^T$ for which we *do not know beforehand* how they were generated, since in the online learning setting, by definition, there should be *no* prior assumption on how $\{\text{Regret}_t/t\}_{t=1}^T$ is generated, which very much depends on *both* how the loss sequences and how the policies are generated (by the algorithms).

Our approach implicitly assumes that $(a_{t+1} - a_t)_{t=1}^T$ is mutually independent. We used this assumption since without knowing how $\{\text{Regret}_t/t\}_{t=1}^T$ were generated, one possible (statistical) assumption to model arbitrarily changing sequences is that at each t, some new element is generated randomly and independently, without being affected/biased by any previous elements in the sequence (since we do not know a priori how to model it). Meanwhile, it is possible that the assumption might not hold since it depends on how loss sequences are generated or how LLM behaves. However, it is possible that Equation (C.1) still holds approximately. Specifically, we define

$$\Delta_t = \frac{\text{Regret}_t}{t} - \frac{\text{Regret}_{t+1}}{t+1},$$

and treat $(\Delta_t)_{t=1}^T$ as random variables. We first compute the correlations among those random variables in Figure C.2 using data from Section 3.2, where we can see that the correlations among those random variables are indeed quite small. Meanwhile, this further implies that

$$\mathbb{E}\left[\sum_{t=1}^T \mathbf{1}[\Delta_t > 0]\right] = \sum_{t=1}^T \mathbb{E}\left[\mathbf{1}[\Delta_t > 0]\right],$$
$$\text{Var}\left(\sum_{t=1}^T \mathbf{1}[\Delta_t > 0]\right) \approx \sum_{t=1}^T \text{Var}\left(\mathbf{1}[\Delta_t > 0]\right),$$

i.e., the random variable $\sum_{t=1}^T \mathbf{1}[\Delta_t > 0]$ indeed has the same first-order and second-order moment as in the case where those random variables $\{\mathbf{1}[\Delta_t > 0]\}_{t\in[T]}$ are independent. Therefore, we regard a Binomial distribution (i.e., assuming $\{\mathbf{1}[\Delta_t > 0]\}_{t\in[T]}$ to be independent) to be an approximation for the actual behaviors of $\sum_{t=1}^T \mathbf{1}[\Delta_t > 0]$, which finally gives Equation (C.1). In fact, when binary random variables have weak correlations (but are not necessarily independent), using the Binomial distribution as an approximation for their summation is also used in the Systems Engineering literature (Hoyland & Rausand, 2009).

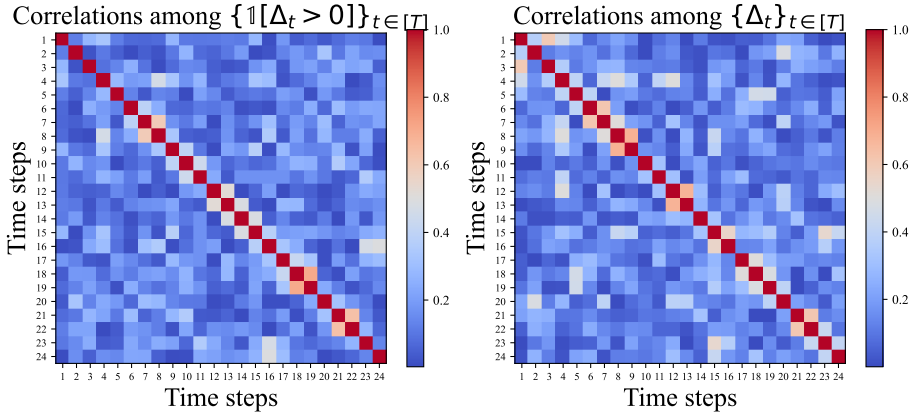

Figure C.2: The absolute value of Pearson correlation coefficient for the random variables $\{\mathbf{1}[\Delta_t > 0]\}_{t\in[T]}$ and $\{\Delta_t\}_{t\in[T]}$ using data obtained in Section 3.2.

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 3: 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.

C.4 Deferred Experiments for Non-stationary Environments in Section 3.2

We 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 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 $\ell_{t_i} \leftarrow 10 \cdot \mathbf{1}_d - \ell_{t_i}$. 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 3. 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 B.4 for a detailed introduction of FTRL/FTPL with Restart.

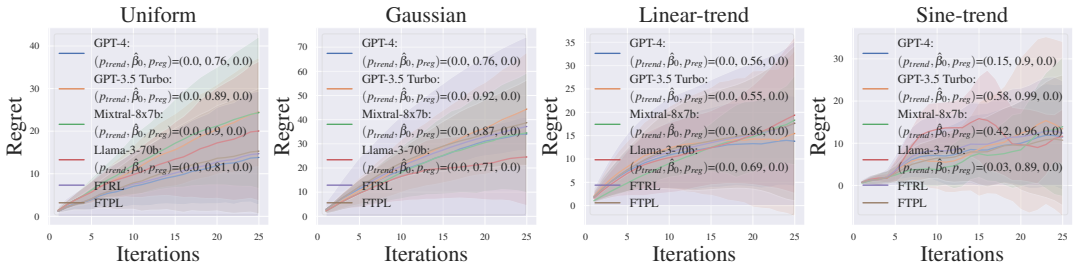

Figure C.3: Regret of pre-trained LLMs for online learning with bandit feedback in 4 different settings. It performs comparably and sometimes even better than well-known no-regret learning algorithms, variants of FTRL and FTPL with bandit-feedback.

C.5 Deferred Experiments for Bandit-feedback Environments in Section 3.2

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 (when the loss is non-negative) 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 C.3 and Table 3, where GPT-4 consistently achieves lower regret.

C.6 Additional Figures for Section 3.3

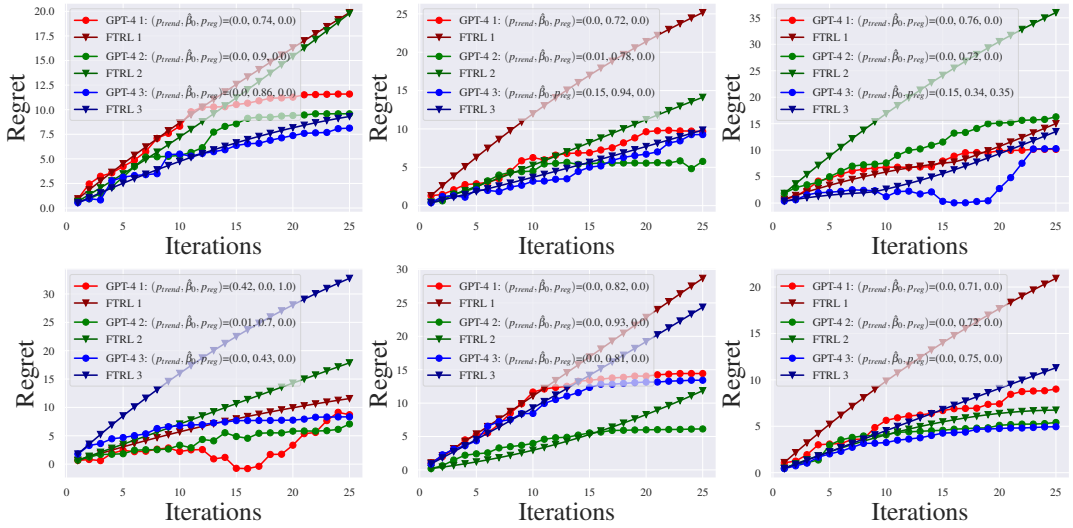

Figure C.4: Regret of GPT-4 and the FTRL algorithm in 6 randomly generated three-player general-sum games. GPT-4 has comparable (even better) no-regret properties when compared with the FTRL algorithm, according to the frameworks in Section 3.1 and the graphic trends..

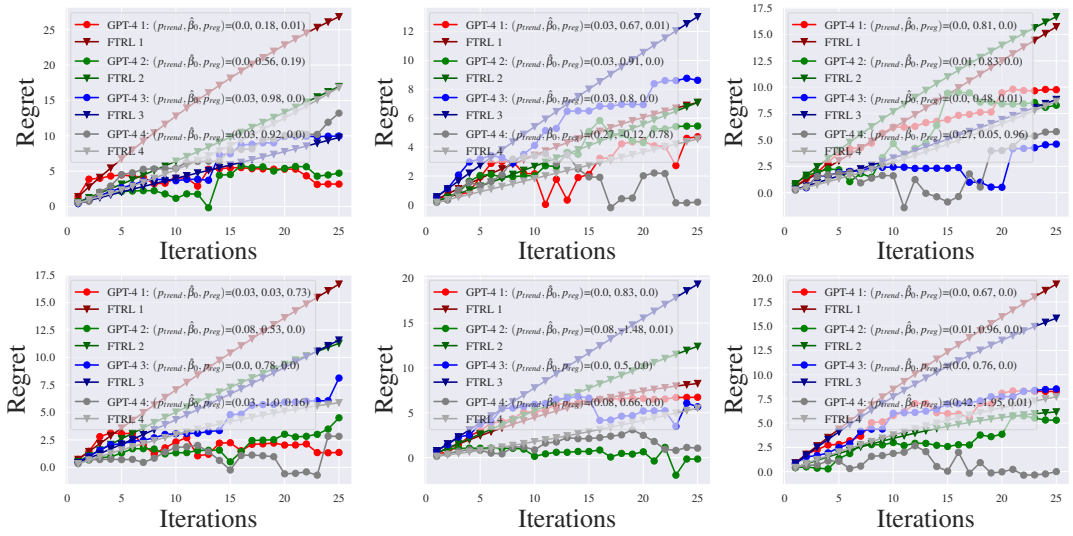

Figure C.5: Regret of GPT-4 and the FTRL algorithm in 6 randomly generated four-player general-sum games. GPT-4 has comparable (even better) no-regret properties when compared with the FTRL algorithm, according to the frameworks in Section 3.1 and the graphic trends.

C.7 Additional Results for Section 3.4

For ***Example 2***, we evaluate LLMs on both the $c = 100$ and $c = 200$ cases. The results and comparisons are presented in Figure C.6 using a temperature of 0 to minimize the randomness for such fixed problem instances, where we can confirm that GPT-4 with raw history identifies the pattern and is able to achieve decreasing, negative regret during the first $c = 100$ or $c = 200$ rounds), whereas FTRL, FTPL, and GPT-4 with only summarized history cannot detect the trend and then make adaptive decisions. Meanwhile, after first c rounds, the LLM with raw history can identify that the pattern for the loss vectors has changed to adjust its policy, and its regret grows more slowly than the LLM with only summarized history.

Such observations further demonstrate the fundamental differences in the stochastic settings considered in Krishnamurthy et al. (2024) and our non-stochastic settings: the summarized history, an essential factor for the successful configuration in Krishnamurthy et al. (2024), can be good statistics in the i.i.d. setting (as a good estimate of the *mean* of the losses), while it loses information and can be highly ineffective in the non-stochastic settings that are highly adversarial (Feder et al., 1992). In contrast, with raw history, GPT-4 was able to better identify the pattern of the sequence and make good predictions to achieve even negative regret values.

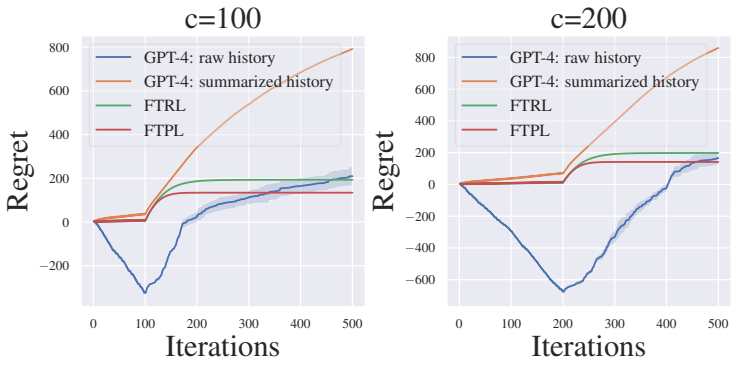

Figure C.6: Comparing LLMs on ***Example 2*** in Section 3.4 with raw history as the input and summarized history as the input.

Explaining the better performance of LLMs on losses with trends via in-context learning. LLMs' in-context-learning capability of being able to *infer* the underlying *trend* in the above case might offer one explanation for the observations above. Specifically, the task of predicting ℓ_{T+1} given past loss sequences $\ell_{1:T}$ could be understood as an in-context learning problem as follows: the demonstration/in-context dataset is given by the following input and label pairs $D = \{x_t, y_t\}_{t\in[T-1]}$, where $x_t = \ell_{1:t}$ and $y_t = \ell_{t+1}$ for each $t \in [T-1]$. Then, LLMs given such demonstration/context D will make prediction based on $x_T = \ell_{1:T}$ (to predict y_T, i.e., the next loss vector ℓ_{T+1}). In other words, in-context learning, in this case, is firstly learning the *trend* from the $T-1$ pairs of inputs and labels, and then making a prediction of the next loss. Hence, when there exists an underlying pattern, in-context-learning can accurately predict the next loss (when raw history is given), and thus achieves good no-regret performance. This perspective may offer an explanation of why LLMs can achieve better performance than FTRL/FTPL when the loss sequences have an obvious trend. Note that, this may also be used to explain why raw-history-based input outperforms the summarized-history-based input in the experiments above – the latter loses such a "context" information, as the mean of the history losses is not sufficient to predict/infer the underlying trend (even when there exists one). Finally, note that, this "trend prediction" explanation does not apply to general loss sequences, for which our explanation in Section 4 that connects LLMs' behaviors to FTPL still applies.

C.8 Ablation Study on the Prompt

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 C.7. 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.

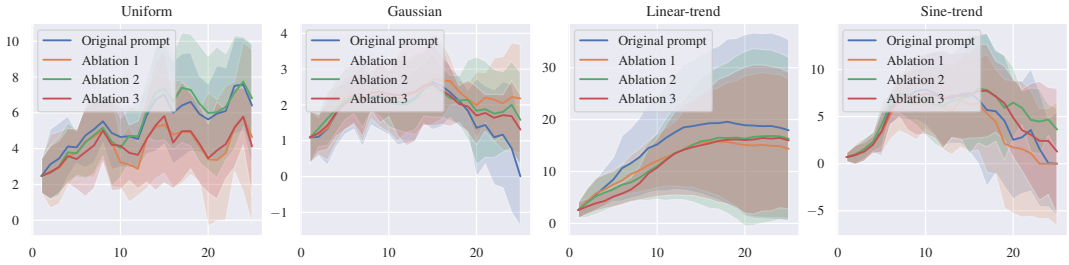

Figure C.7: Ablation study on our prompt design.

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.

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
```

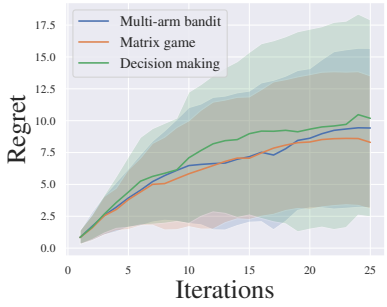

Figure C.8: Regret of GPT-4 for repeated games under 3 different prompt ablations. Its performance is consistent among three different prompts.

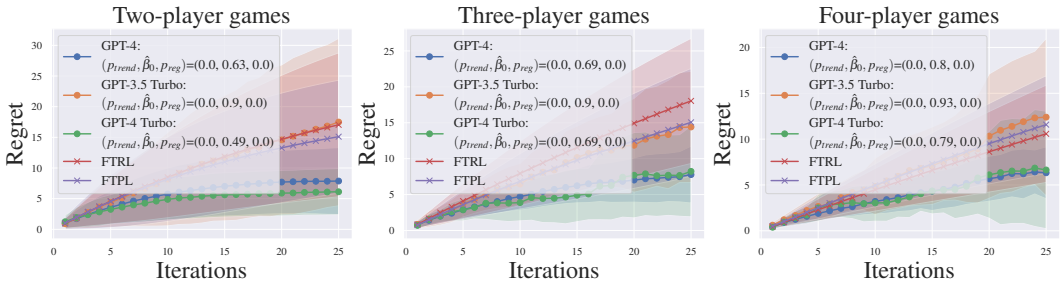

Figure C.9: Comparisons of GPT-4 Turbo with GPT-4 and GPT-3.5 Turbo. Due to the symmetry of agents in randomly generated games, we report the regret of one agent for ease of presentation. We can see that GPT-4 Turbo performs on par with or better than GPT-4.

```
greedily  by explicitly considering the policy you just used for
last round.
```

We can see in Figure C.7 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 C.8, where it is seen that LLM agents' performance on repeated games is consistent among these variants of the prompts.

C.9 Results for GPT-4 Turbo

When preparing our paper, 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 C.9, where we validate that GPT-4 Turbo performs either comparably with or better than GPT-4.

C.10 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 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 & Lugosi, 2006), explaining the empirical evidence we had in Section 3.

C.11 CASE STUDIES ON REAL-WORLD APPLICATIONS

In this subsection, we evaluate the sequential decision-making abilities of LLMs in realistic scenarios from the perspective of regret and dynamic regret. While several studies have explored sequential decision-making using synthetic scenarios (Krishnamurthy et al., 2024; Wu et al., 2024b; Xia et al., 2024; Akata et al., 2023) or real-world data scenarios (Liu et al., 2023b; Wang et al., 2023c; Wu et al., 2024a), none have explicitly analyzed regret or dynamic regret. As a result, the *theoretical optimality* of such a sequential decision-making process remains unclear.

Our first case study investigates single-agent sequential decision-making using real-world data, leveraging the same dataset and experimental setup as (Liu et al., 2023b). The second case study explores a two-player negotiation scenario, providing insights into dynamic interactions and their impact on decision-making performance.

C.11.1 SEQUENTIAL RECOMMENDATION

We consider the task of sequential recommendation, a task that people have been employing LLMs to solve with success (Liu et al., 2023b; Wang et al., 2023c; Wu et al., 2024a). Note that how existing literature (Liu et al., 2023b) uses LLMs to solve this task fits exactly into our online learning framework, where humans feed a history of items the user have interacted with to the LLM and then ask the LLM to recommend the item (or several items) the user may want to interact next. The entire process carries on repeatedly.

Formally, the problem is as follows. Given a sequence of history items the user has interacted with $(x_1, x_2, \cdots, x_{t-1})$, where each $x_i \in D$ for $i \in [t-1]$ and D is the collection of all items, the LLM needs to recommend n items that the user might interact with in the next step t. Typically, the LLM should also give a priority on the n items it recommends. For simplicity here, we here assume they are of equal priority. In other words, at step t, the LLM will take an action $a_t \subseteq D$ with $|a_t| = n$, hoping what the user will interact at step t belongs to a_t. Hence, the loss is given by $\ell_t(a_t, x_t) := \mathbf{1}[x_t \notin a_t]$. Correspondingly, the regret by our definition is given by

$$\text{Regret}(x_{1:T}) = \sum_{t=1}^{T} \ell_t(x_t, a_t) - \min_{a} \sum_{t=1}^{T} \ell_t(x_t, a).$$

We refer to (Liu et al., 2023b) for a more detailed introduction. Meanwhile, we use the real-world data and follow the experimental setup of (Liu et al., 2023b).

In the left one of Figure C.10, we can observe that LLMs can achieve expressively low and sublinear regret on such a real-world application with real-world data. As a comparison, in the right one of Figure C.10, we replace the real-world data with synthetic data generated in a uniformly random way (it is worth mentioning that the prompt setting still follows the setup of sequential recommendation of Liu et al. (2023b)), where we can see that LLMs can still be no-regret. However, interestingly, LLMs perform better on real-world data, which validates that real-world applications can exhibit certain trends/structures, for which LLMs can exploit and achieve superior performance as we have shown in our paper through synthetic problems with trends.

C.11.2 INTERACTIVE NEGOTIATION

The experiment was designed to simulate negotiation scenarios between two LLMs, designated as LLM A and LLM B, across multiple turns. The primary objectives were to analyze multi-agent sequential decision-making processes and quantify regret. For each repetition, an LLM generated unique negotiation topics. Based on these topics, the LLM also created the context, objectives, and relevant background information to design engaging and interactive negotiation scenarios.

Negotiation Process. The negotiation process was executed in a turn-based manner, with each turn comprising three steps:

1. **Intention Generation**: Each LLM defined its goal for the turn, specifying what it aimed to achieve with its response.

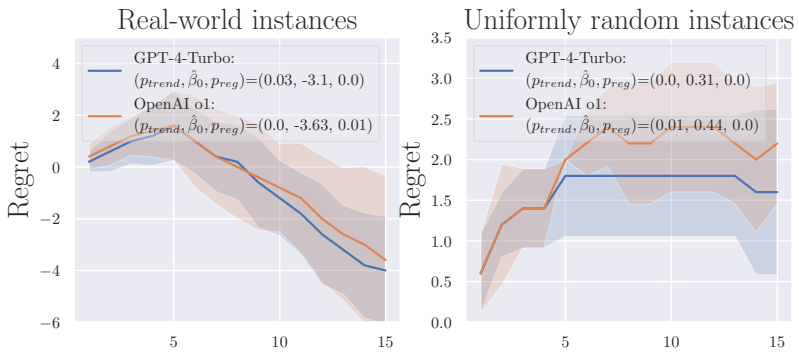

Figure C.10: We evaluate GPT-4-Turbo and OpenAI o1 on both real-world data and uniformly random synthetic data, where we can see both models can still achieve sublinear regret.

2. **Response Generation**: Based on the defined intention and the dialogue history, each LLM generated a response.
3. **Alternative Response Generation**: Three distinct alternative replies were produced for each original response. These alternatives represented diverse negotiation strategies or perspectives while preserving the original intention.

Response Evaluation. After the dialogue concluded, all responses—both original and alternatives—were evaluated using a scoring scale from 1 to 10 based on the following criteria for each turn:

- **Clarity**: How clear and understandable the reply is.
- **Relevance**: How pertinent the reply is to the negotiation topic and the defined intention.
- **Engagement**: How engaging or persuasive the reply is in fostering further dialogue.
- **Alignment with the Stated Intention**: How well the conversation aligns with the turn's stated intention following the reply. For alternative replies, this was assessed by hypothetically replacing the original reply with an alternative and evaluating the alignment based on the entire conversation.

Each response was scored using an LLM as the evaluator. Although human evaluation would be preferable, the use of an LLM as a scorer was chosen for scalability. This approach is common in the LLM domain and is sometimes referred to as G-eval (where "G" stands for GPT) (Liu et al., 2023c).

Dynamic Regret Analysis. Finally, dynamic regret was calculated to measure suboptimality by comparing the scores of the original replies against the highest-scoring alternative responses. Since calculating regret typically requires hindsight knowledge of the best possible responses, which requires rollout of every possible dialogues, we decide to analyze on dynamic regret. Dynamic regret analysis provided a quantitative measure of decision-making effectiveness across turns. This analysis offered insights into how regret dynamics can inform improved decision-making strategies in real-world negotiation contexts.

Example. Here is an example from our simulation:

Step 1: Generate Topics and Backgrounds. Topics and backgrounds were generated using a language model.

> *Topic*: The Trade-Off Negotiation Between Eco-Tech Innovator and Traditional Manufacturing Tycoon
>
> *Background of Player A:* Eco-Tech Innovator (Jordan Green). Jordan Green is the CEO of a rapidly growing startup, EcoWave Technologies, which specializes in developing sustainable

energy solutions and eco-friendly manufacturing processes. With a background in environmental science and engineering, Jordan is passionate about reducing carbon footprints and promoting renewable energy sources. Their innovative products, such as biodegradable materials and energy-efficient machinery, have garnered attention and accolades within the green tech community. However, despite the startup's promise, EcoWave faces challenges in scaling production and reaching wider markets due to limited financial resources and manufacturing capabilities.

Background of Player B: Traditional Manufacturing Tycoon (Robert Steele). Robert Steele is the owner of Steele Industries, a well-established manufacturing company known for its mass production of consumer goods. With decades of experience in the industry, Robert has built a reputation for efficiency and profitability, but his company has been criticized for its environmental impact and reliance on outdated practices. Facing increasing pressure from regulators and consumers to adopt sustainable practices, Robert is aware that failing to adapt could threaten the future of his business. However, he is also concerned about the costs and risks associated with transitioning to new technologies and processes, which could disrupt his current operations.

Detailed Situation: Jordan and Robert have agreed to meet to discuss a potential partnership. Robert is interested in integrating EcoWave's sustainable technologies into his manufacturing processes, which could help Steele Industries improve its environmental footprint and public image. However, he is also hesitant about the costs associated with adopting these new technologies and is particularly concerned about how this might affect his established workforce and production timelines. On the other hand, Jordan sees this as a golden opportunity to scale EcoWave's innovative solutions and gain access to Robert's extensive distribution networks. However, Jordan is wary of compromising on key values regarding environmental integrity and sustainable practices, fearing that a partnership with a traditional manufacturer could dilute EcoWave's mission. The negotiation aims to balance Robert's need for cost-effectiveness and operational continuity with Jordan's commitment to sustainability and innovation. Both players must navigate their differing priorities, explore potential compromises, and consider the long-term implications of their decisions on their businesses and the environment.

Step 2: Generate Conversation. We iterated between **Step 2-1** and **Step 2-2** to create a conversation with a length of 20 turns. In this setup, each LLM (A and B) contributed 10 responses based on the evolving conversation history.

Step 2-1: Intention Generation and Reply Generation. We provided the initial turn's intention for LLM A:

Intention: In my next response, I aim to outline potential trade-offs that both I and Robert might consider in order to reach a mutually beneficial agreement. I will also suggest ways we can ensure that the partnership aligns with their core values of sustainability and operational efficiency. This will help set the stage for a constructive dialogue that addresses our respective concerns and priorities.

Reply of Player A based on the intention: Thank you for meeting with me, Robert. I believe we have a unique opportunity to collaborate that could benefit both EcoWave and Steele Industries. I understand your concerns about the costs and risks of integrating new technologies, but I think we can explore some trade-offs. For instance, if we can demonstrate the long-term cost savings and efficiency gains from adopting sustainable practices, it might help alleviate some of your concerns. Additionally, I'm open to discussing a phased implementation approach that allows for gradual integration without disrupting your current operations. How do you feel about that, and what trade-offs might you be willing to consider to make this partnership work?

Step 2-2: Alternative Generation. The LLM also generated three alternative responses based on the aforementioned intention.

Alternative 1: I appreciate your concerns, Robert, and it's great to see your commitment to a smooth transition. Instead of jumping straight into a pilot project, how about we organize a joint workshop with your key team members? This would allow us to collaboratively explore

areas where EcoWave's technologies can fit into your existing operations. We can analyze potential impacts, address any workforce concerns, and identify specific processes that could benefit from our innovations. This way, we ensure that everyone is on board and feels included in the transition. What are your thoughts on setting up a workshop to discuss this further?

Alternative 2: I appreciate your concerns, Robert, and I think it's important for us to find a starting point that feels comfortable for you. Instead of a full-scale integration right away, how about we explore a collaborative approach where we can jointly assess some specific production lines that may benefit from our technologies? We could conduct a thorough analysis of these areas together, identifying potential efficiency improvements and cost benefits through a detailed study. This way, we can base our decisions on solid data while ensuring that your existing operations remain stable. Additionally, I'd be more than willing to work closely with your team to address any training needs that may arise. Would this kind of collaborative assessment resonate with you?

Alternative 3: I appreciate your insights, Robert, and I recognize the importance of ensuring a smooth transition. Instead of jumping straight into a pilot project, how about we start by conducting a joint assessment of your current operations? This would allow us to pinpoint specific areas where EcoWave's technologies could fit seamlessly. By collaborating on this assessment, we can address your concerns and tailor our solutions to your unique processes. Additionally, I'd be happy to share case studies from other manufacturers who have successfully integrated our technologies, demonstrating the benefits they achieved without significant disruptions. What do you think about this collaborative approach?

Step 3: Evaluation. Each reply and its alternatives were evaluated using G-eval.

Experiment Setting. We set the maximum number of turns to 10 and evaluated two language models: GPT-4-Turbo and OpenAI's o1 model. For each model, we tested performance across 10 different scenarios.

Results. Interestingly, both models demonstrated no-dynamic-regret behavior within the regression framework. Both models also exhibited low p-values, with GPT-4-Turbo achieving 0.09 and OpenAI o1 achieving 0.02. The detailed results are shown in Figure C.11.

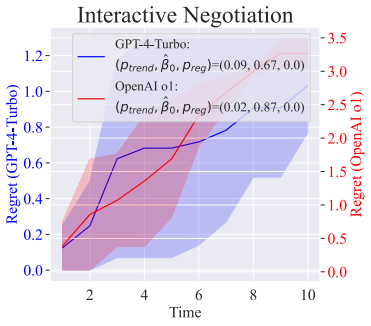

Figure C.11: We evaluate GPT-4-Turbo and OpenAI o1 on two-player Negotiation, where we can see both model can achieve no-regret in the regression framework.

D Deferred Results and Proofs in Section 4

D.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:*

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

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:*

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

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).

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

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

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,

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

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 $\sum_{t=1}^{T}\inf_{\pi\in\Pi}\langle \ell_t, \pi\rangle$. □

D.2 Background and Motivations for (Generalized) Quantal Response

Formally, the quantal response is defined as follows:

Definition D.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*

$$P^{\eta}_{quantal}(a \mid \ell) = \mathbb{P}\left(a \in \arg\min_{a'\in\mathcal{A}} \; z(a')\right), \qquad \textit{where} \quad z = \ell + \eta\epsilon.$$

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.*

Further motivations for generalized quantal response. Note that a *dynamic* version of quantal response in Definition 4.1 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 & Kreps, 1993) and *follow-the-perturbed-leader* (Kalai & Vempala, 2005), whose formal definitions can be found in Appendix B.4. Finally, note that the response model in Definition 4.1 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.

D.3 THE EXAMPLE INSTANTIATING ASSUMPTION 1

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.*

D.4 ALIGNMENT OF ASSUMPTION 1 WITH QUANTAL RESPONSE

Before presenting the technical lemma, based on Assumption 1, we denote the (potentially unkown) mappings that decode semantic information in Assumption 1 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 $g(x_{N_t+1:N_{t+1}}) = a \in \mathcal{A}$.

Lemma 1. *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.,*

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

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 4.1, 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

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

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

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

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. □

D.5 Relationship between FTPL and Definition 4.1

Fact 1. *Performing generalized quantal response of Definition 4.1 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

$$P_{\text{quantal}}^{\eta_{t-1}}(a \,|\, \ell_{1:t-1}) := \mathbb{P}\left(a \in \arg\min_{a'\in\mathcal{A}}\left(\sum_{i=1}^{t-1}\ell_i + \eta_{t-1}\epsilon\right)(a')\right) \tag{D.2}$$

which is exactly the case when ϵ_t in Equation (B.2) satisfies $\epsilon_t \overset{d}{=} \eta_{t-1}\epsilon$. □

D.6 Formal Statement and Proof of Theorem 4.1

Theorem D.1. (Emergence of no-regret behavior)**.** *Under the assumptions of Lemma 1, suppose the function class of LLM$_\theta$ is expressive enough such that for all $t \in [T]$, $\max_{\theta\in\Theta}\mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}}\sum_{j=1}^{N_{t+1}}\log \text{LLM}_\theta\left(x_j \,|\, x_{1:j-1}\right) = \max_{\{q_j\in\{\mathcal{V}^{j-1}\to\Delta(\mathcal{V})\}\}_{j\in[N_{t+1}]}}\mathbb{E}_{x_{1:N_{t+1}}\sim P_t^{text}}\sum_{j=1}^{N_{t+1}}\log q_j\left(x_j \,|\, x_{1:j-1}\right)$, where we define $q_1(x_1 \,|\, x_{1:0}) := q_1(x_1)$, and $\theta^\star$ maximizes Equation (4.1). 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, $\mathbb{E}\left[\text{Regret}_{\text{LLM}_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right)\right] \le \mathcal{O}\left((\log d)^{1/2}dT^{1/2+1/\log T}\log T\right)$;

(4) For non-stationary online learning with bandit feedback, $\mathbb{E}\left[\text{D-Regret}_{\text{LLM}_{\theta^\star}}\left((\ell_t)_{t\in[T]}\right)\right] \le \mathcal{O}\left((T^2d^2V_T)^{1/3}(\log d)^{1/2}T^{1/\log T}\log T\right)$.

Proof. Note that

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

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

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

Now, due to our assumption that

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

we conclude that

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

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

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

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 $\mathbb{P}(g(x_{N_t+1:N_{t+1}}) \mid x_{1:N_t})$, which is equal to $P^{\sigma\sqrt{t+1}}_{quantal}\left(g(x_{N_t+1:N_{t+1}}) \,\middle|\, \left\{f(x_{N_{i-1}+1:N_i})\right\}_{i\in[t]}\right)$ 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 (D.2) 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

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

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

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

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

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

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:

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

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

Now, we present Lemma 2 - Lemma 4. Before proceeding, we assume $\|\ell_t\|_\infty \leq B = 1$ 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 (B.2) and $\zeta_t = \sigma\sqrt{t}$, then for online learning with full-information feedback,*

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

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

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

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

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

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 (B.2) and $\zeta_t = \sigma\sqrt{t}$, then for online learning with bandit feedback,*

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

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 & Tewari (2017), $\pi_t = \nabla \Phi_t(\widehat{G}_t)$. Now due to the convexity of Φ,

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

Therefore,

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

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,

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

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

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

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

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

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*

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

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

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

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 $k \in [m]$

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

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

$$\begin{aligned}\text{D-Regret}_{\mathscr{A}'}((\ell_i)_{i\in[T]}) &\le \min_{j\in[d]} \left(\sum_{t\in[T]} \ell_t\right)_j - \sum_{t\in[T]} \ell_t^\star + \sum_{k\in[m]} \text{Regret}_{\mathscr{A}}((\ell_i)_{i\in[\mathcal{T}_k]}) \qquad \text{(D.3)}\\ &\le 2\Delta_T(\sum_{k\in[m]} L_k) + (T/\Delta_T + 1)g(\Delta_T, d).\end{aligned}$$

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

Similarly, if we take the expectation for Equation (D.3), it holds that

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

thus completing the proof. □

Combining the results above completes the proof for Theorem 4.1. □

D.6.1 Implications of Theorem 4.1 for Repeated Games

Remark D.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 & Lugosi, 2006). Hence, the results (1) and (2) in Theorem 4.1 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).*

D.7 Extending Theorem 4.1 with Relaxed Assumptions

D.7.1 Relaxation under More General Data Distributions

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

Remark D.2 (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 (D.1), it would suffice. It is a combined effect of both the prior and the conditional distributions.*

More formally, we can extend Theorem 4.1 to the case with a much more general prior task distribution than the Gaussian one, where the key is that Equation (D.1) only needs to hold approximately.

Theorem D.2. *In Theorem 4.1, 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 4.1 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 4.1 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

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

where C is the normalization constant and

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

Therefore, we conclude that 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

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

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 $\mathbb{P}(z \mid (\ell_i)_{i\in[t]})$, and FTPL to be the algorithm w.r.t. the noise distribution $\mathcal{N}(\widehat{z}, \frac{\sigma^2}{t})$. Therefore, we have

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

In other words, using $\mathbb{P}(z \mid (\ell_i)_{i\in[t]})$ 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. □

D.7.2 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 D.3 (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 C.1), to induce it to make decisions. This will query the* conditional *distribution of*

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

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

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

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

D.8 Comparison with Lee et al. (2023); Lin et al. (2024); Liu et al. (2023e)

Similar assumptions and pre-training objectives have also been considered in the very recent work of Lee et al. (2023); Lin et al. (2024); Liu et al. (2023e) for studying in-context reinforcement learning property of Transformers/LLM-agents 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). Liu et al. (2023e) adopted the similar data generation assumption as Lee et al. (2023) without assuming optimal labels are available in the pre-training datasets, but leverages external oracles for *planning*. Consequently, the resulting LLM agent would still perform the posterior sampling algorithm. However, these results cannot directly imply the no-regret guarantee in our online learning setting, due to the known fact that

posterior sampling can perform poorly under potentially *adversarial* or *non-stationary* environments (Zimmert & Seldin, 2021; Liu et al., 2023d). In contrast, we here establish the equivalence of the pre-trained LLM to the FTPL algorithm (under different pre-training data distribution specifications), with the ability to handle arbitrary loss sequences, even though the LLMs are only trained on a fixed/stationary distribution of texts (tasks).

D.9 Details of Estimating the Parameters of Our Hypothetical Model

To further validate our model and data distribution assumptions, we also propose to estimate the parameter $\{\eta_t\}_{t\in[T-1]}$ in Definition 4.1, using data from interacting with LLMs (following the same protocol as before), with P_{noise} being a standard normal distribution (note that we do not need to estimate η_0 by Definition 4.1). 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 4.1, we estimate $\{\eta_t\}_{t\in[T-1]}$ 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 estimated $\{\eta_t^\star\}_{t\in[T-1]}$ 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.

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 noisy alternating or adaptive loss sequences, as Figure 4.1 shows the gap between GPT-4 (dynamic) regret and the our model's (dynamic) regret.

E Deferred Results and Proofs in Section 5

E.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.*

E.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:*

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

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

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

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

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

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

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

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

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. □

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

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

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

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

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

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

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

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

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

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

$$\operatorname{esssup}\left(h\left(\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right)\right) = \max_{\ell_1,\dots,\ell_T} h\left(\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right) = h\left(\max_{\ell_1,\dots,\ell_T}\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right). \tag{E.1}$$

Equation (E.1) will be used frequently in the overall proof in Appendix E.2.

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

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

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

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

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

Claim 2 (Uniform convergence of $\mathcal{L}(\theta, k, N)$ (with respect to k and N))**.** *$\mathcal{L}(\theta, k, N)$ uniformly converges to $h(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))$ 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, $\text{esssup}(H(X_i)) = 1$, $H : \mathbb{R} \to \mathbb{R}^+$ is a continuous non-decreasing function, $A(k,t,\epsilon) := \frac{(1-\epsilon)f((t/\text{esssup}(t(X)))^{-1}(1-\epsilon),k)}{(1-\epsilon/2)f((t/\text{esssup}(t(X)))^{-1}(1-\epsilon/2),k)}$, 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

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

□

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 $H(R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})) = \frac{h(R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))}{h(\max_{\ell_1,\ldots,\ell_T} R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]}))}$, we have

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

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

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

with probability at least $1-\delta$.

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

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

Note that

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

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 $\text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$ 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 $p(\epsilon) := \min_{\theta\in\Theta} 1 - F_{h,R_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})}(1-\epsilon/2) > 0$. Therefore,

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

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

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

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

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

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

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 $\sum_{1\le i\neq j\le N} f(X_i,k_1)h(X_i)f(X_j,k_2) \le \sum_{1\le i\neq j\le N} f(X_i,k_1)h(X_j)f(X_j,k_2)$. This is equivalent to

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

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

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. □

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

E.3 Definition of the Empirical Loss Function

Definition E.1 (Empirical loss function)**.** *We define the empirical loss $\widehat{\mathcal{L}}$ computed with N_T samples as follows:*

$$\widehat{\mathcal{L}}(\theta,k,N,N_T) := \frac{1}{N_T}\sum_{s=1}^{N_T}\left[\frac{\sum_{j\in[N]} h\left(\text{Regret}_{\text{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]})\right) f\left(\text{Regret}_{\text{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]}),k\right)}{\sum_{j\in[N]} f\left(\text{Regret}_{\text{LLM}_\theta}((\ell_{s,t}^{(j)})_{t\in[T]}),k\right)}\right] \tag{E.3}$$

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 $\mathcal{L}(\theta,k,N)$.

E.4 Deferred Proofs of Theorem E.1 and Theorem 5.1

Theorem E.1. (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*

$$\mathcal{L}\left(\widehat{\theta}_{k,N,N_T},k,N\right) - \inf_{\theta\in\Theta}\mathcal{L}(\theta,k,N) \le \widetilde{\mathcal{O}}\left(\sqrt{\frac{d_\theta+\log(1/\epsilon)}{N_T}}\right), \tag{E.4}$$

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 (E.4) 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.

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 `Operator` (to be specified later).

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

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

where $\|\cdot\|_{\texttt{TF}}$ is as defined in Equation (B.1), and $R, Z, B_{\texttt{TF}}$ are as introduced in Appendix B.2. Moreover, the function $\texttt{Operator}(A\cdot\texttt{TF}_\theta(\cdot)_{-1})$ is $\|A\|_{op}C_{\texttt{TF}}$-Lipschitz continuous with respect to θ, i.e.,

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

Here, `Operator` is either the projection operator onto some convex set, or the `Softmax` function.

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

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

Note that the only condition that we require for `Operator` is its non-expansiveness. □

Proof of Theorem E.1. Let C_{LLM} denote the Lipschitz-continuity constant for LLM_θ with respect to some norm $\|\cdot\|_{\text{LLM}}$, where $\|\cdot\|_{\text{LLM}}$ denotes any norm defined on the parameter space of LLM (e.g., the norm $\|\cdot\|_{\texttt{TF}}$ 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 $\text{Regret}_{\text{LLM}_\theta}$ is $C_{\texttt{Reg}} := BC_{\text{LLM}}T$-Lipschitz continuous with respect to θ, i.e.,*

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

Proof. By definition, we have

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

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

Now, we will prove the Lipschitzness of

$$C\left((\ell_t^{(j)})_{t\in[T],j\in[N]},k,\theta\right) := \frac{\sum_{j\in[N]}h(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}))f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[T]}),k)}{\sum_{j\in[N]}f(\text{Regret}_{\text{LLM}_\theta}((\ell_t^{(j)})_{t\in[t]}),k)} \tag{E.5}$$

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*

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

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

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

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

holds. Moreover, consider the following constrained optimization problem:

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

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 (E.5). 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 (E.5))**.** *The function C in Equation (E.5) 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 E.2 (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 $N(\delta; B(0, r, \|\cdot\|_{\text{LLM}}), \|\cdot\|_{\text{LLM}})$ can be bounded by

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

where d_θ is the dimension of the LLM's whole parameter. For example, if we use the $\|\cdot\|_{\text{TF}}$ and consider the Transformer model as defined in Appendix B.2, for any $r > 0$,

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

Since we consider a compact Θ (as several recent works on analyzing Transformers (Bai et al., 2023; Lin et al., 2024)), let $R_\Theta := \text{diam}(\Theta, \|\cdot\|_{\text{LLM}})$ (which corresponds to B_{TF} for the Transformer models as defined in Appendix B.2, with $\|\cdot\|_{\text{LLM}} = \|\cdot\|_{\text{TF}}$), 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

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

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

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

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. □

Theorem 5.1. (Regret). *Suppose*[3] *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 probability, we have*

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

Proof. The limit on the right-hand side of Equation (E.4) 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 E.1 holds for large enough k and any N. Next, we have

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

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

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

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

[3]Note that these conditions on h, f are in addition to those specified after Equation (5.2).

As a result, the coarse correlated equilibrium will emerge as the long-term interactions of multiple such learned LLMs, as stated in the following corollary.

Corollary 1. (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.*

Remark E.1 (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:*

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

Then, 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

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

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 E.2 also hold for similar reasons.

E.5 Detailed Explanation of Optimizing Equation (5.2) with Single-layer Self-attention Model

We consider the following structure of single-layer self-attention model g (see a formal introduction in Appendix B.2):

$$g(Z_t; V, K, Q, v_c, k_c, q_c) := (V\ell_{1:t} + v_c\mathbf{1}_t^\intercal)\texttt{Softmax}\left((K\ell_{1:t} + k_c\mathbf{1}_t^\intercal)^\intercal \cdot (Qc + q_c)\right), \tag{E.6}$$

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

Theorem E.2. *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 (E.6) (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 (5.2) with $N = 1$, $h(x) = x^2$. Moreover, if Σ is a diagonal matrix, then plugging this configuration into Equation (E.6), and projecting the output with $\texttt{Proj}_{\Pi,\|\cdot\|}$ 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.

E.6 Deferred Proof of Theorem E.2

Theorem E.2. *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 (E.6) (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 (5.2) with $N = 1$, $h(x) = x^2$. Moreover, if Σ is a diagonal matrix, then plugging this configuration into Equation (E.6), and projecting the output with $\texttt{Proj}_{\Pi,\|\cdot\|}$ 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 (5.2)) can be written as follows:

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

where for $t = 1$, we use the output of the single-layer self-attention as v_c and we will write it as $(V\ell_{1:0} + v_c \mathbf{1}_0^\intercal)\texttt{Softmax}(\ell_{1:0}^\intercal a + b_0)$ 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 $t \geq 2$:

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

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

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

For $t = 1$, as $\ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) = \ell_1^\intercal v_c$, $\frac{\partial}{\partial a_x} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1})\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}} = 0$, 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:

$$\begin{aligned}
&\frac{\partial f(V, a, (b_t)_{t\in[T-1]}, v_c)}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}} \\
&= \mathbb{E}\frac{\partial}{\partial a_x}\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \| \sum_{t=1}^{T} \ell_t \|_2\right)^2 \Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}} \\
&= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \| \sum_{t=1}^{T} \ell_t \|_2\right)\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}} \\
&\qquad \frac{\partial}{\partial a_x}\left(\sum_{t=1}^{T} \ell_t^\intercal (V\ell_{1:t-1} + v_c \mathbf{1}_{t-1}^\intercal)\texttt{Softmax}(\ell_{1:t-1}^\intercal a + b_{t-1}) + R_\Pi \| \sum_{t=1}^{T} \ell_t \|_2\right)\Big|_{a=\mathbf{0}_d, v_c=\mathbf{0}_d, (b_t=\beta\mathbf{1}_t)_{t\in[T-1]}}\Bigg] \\
&= \mathbb{E}\left[\left(\sum_{t=1}^{T} \ell_t^\intercal V \sum_{i=1}^{t-1} \frac{1}{t-1}\ell_i + R_\Pi \| \sum_{t=1}^{T} \ell_t \|_2\right) \sum_{t=1}^{T}\left(\sum_{i=1}^{t-1} \frac{\ell_t^\intercal V \ell_i \ell_{ix}}{(t-1)} - \sum_{i=1}^{t-1} \frac{\ell_t^\intercal V \ell_i \left(\sum_{s=1}^{t-1} \ell_{sx}\right)}{(t-1)^2}\right)\right] \quad \text{(E.7)} \\
&= 0,
\end{aligned}$$

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:

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

This yields Equation (E.7)=0.

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

We will use the following equation for $t\geq 2$:

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

For $t=1$, we define $\frac{\partial}{\partial v_c}\ell_1^\intercal(V\ell_{1:0}+v_c\mathbf{1}_0^\intercal)\texttt{Softmax}(\ell_{1:0}^\intercal a+b_0)=\ell_1$, 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:

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

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 $t\geq 2$:

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

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}\bigg|_{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)\texttt{Softmax}(\ell_{1:t-1}^\intercal a+b_{t-1})+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)^2\bigg|_{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)\texttt{Softmax}(\ell_{1:t-1}^\intercal a+b_{t-1})+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\bigg|_{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)\texttt{Softmax}(\ell_{1:t-1}^\intercal a+b_{t-1})+R_\Pi\|\sum_{t=1}^{T}\ell_t\|_2\right)\bigg|_{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

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

which means that such configurations are first-order stationary points of Equation (5.2) with $N=1$, $h(x)=x^2$, and $\Pi = B(0,R_\Pi,\|\cdot\|)$. □

E.7 Deferred Proof of Theorem 5.2

Theorem 5.2. *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 (5.3) (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 (5.2) with $N=1$, $h(x)=x^2$. Moreover, every global optimal configuration of Equation (5.2) within the parameterization class of Equation (5.3) has the same output function g. Additionally, if Σ is a diagonal matrix, then plugging any global optimal configuration into Equation (5.3), and projecting the output with $\texttt{Proj}_{\Pi,\|\cdot\|}$ is equivalent to FTRL with an L_2-regularizer.*

This theorem involves the analysis of a *non-convex optimization* problem through stationary point analysis. We identified the set of stationary points. By constructing the optimization problem as shown in Equation (E.13), we significantly reduced the candidate set for optimal points using our novel argument on the expected value of a nonnegative definite matrix. The main challenge here was to address the global optimization problem in a non-convex setting, which required the exploitation of the particular Transformer architecture.

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

$$\begin{aligned} &g(Z_t; V, K, Q, v_c, k_c, q_c) \\ &\quad = \sum_{i=1}^{t} \left(V\ell_i\ell_i^\intercal (K^\intercal (Qc + q_c)) + (Vk_c^\intercal (Qc + q_c) + v_c (Qc + q_c)^\intercal K)\, \ell_i + v_c k_c^\intercal (Qc + q_c)\right), \end{aligned} \tag{E.8}$$

which can be expressed with a larger class

$$g(Z_t, \mathbb{A}, \beta, \mathbb{C}, \delta) := \sum_{i=1}^{t} (\mathbb{A}\ell_i\ell_i^\intercal \beta + \mathbb{C}\ell_i + \delta), \tag{E.9}$$

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

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

can be expressed 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

$$f(\mathbb{A}, \beta, \mathbb{C}, \delta) = \mathbb{E}\left(\sum_{t=1}^{T} \langle \ell_t, \sum_{i=1}^{t-1} (\mathbb{A}\ell_i\ell_i^\intercal \beta + \mathbb{C}\ell_i + \delta)\rangle + R_\Pi \left\| \sum_{t=1}^{T} \ell_i \right\|_2\right)^2. \tag{E.10}$$

Step 1. Finding condition for $\frac{\partial f}{\partial \delta} = 0$.

Due to the Leibniz rule, if we calculate the partial derivative of Equation (E.10) w.r.t. δ, we have

$$\begin{aligned} \frac{\partial f(\mathbb{A}, \beta, \mathbb{C}, \delta)}{\partial \delta} &= \frac{\partial}{\partial \delta} \mathbb{E}\left(\sum_{t=1}^{T} \langle \ell_t, \sum_{i=1}^{t-1} (\mathbb{A}\ell_i\ell_i^\intercal \beta + \mathbb{C}\ell_i + \delta)\rangle + R_\Pi \| \sum_{t=1}^{T} \ell_t \|_2\right)^2 \\ &= \mathbb{E}\frac{\partial}{\partial \delta}\left(\sum_{t=1}^{T} \langle \ell_t, \sum_{i=1}^{t-1} (\mathbb{A}\ell_i\ell_i^\intercal \beta + \mathbb{C}\ell_i + \delta)\rangle + R_\Pi \| \sum_{t=1}^{T} \ell_t \|_2\right)^2 \\ &= \mathbb{E}\sum_{t=1}^{T} \ell_t \left(\sum_{t=1}^{T}\sum_{i=1}^{t-1} (t-1)\ell_t^\intercal (\mathbb{A}\ell_i\ell_i^\intercal \beta + \mathbb{C}\ell_i + \delta) + R_\Pi \| \sum_{t=1}^{T} \ell_t \|\right). \end{aligned} \tag{E.11}$$

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 (E.11), 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

$$\mathbb{A}\Sigma\beta+\delta=0. \tag{E.12}$$

Step 2. Plugging the optimality condition for $\frac{\partial f}{\partial\delta}$ into Equation (E.10).

Plugging Equation (E.12) to Equation (E.10), f can be written as

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

For the part (i), we have

$$\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i=1}^{t-1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right)^2=\mathbb{E}\left[\sum_{t_1=1}^{T}\sum_{i_1=1}^{t_1-1}\sum_{t=1}^{T}\sum_{i=1}^{t-1}\beta^\intercal(\ell_{i_1}\ell_{i_1}^\intercal-\Sigma)\mathbb{A}^\intercal\ell_{t_1}\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right]$$
$$\underset{(1)}{=}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i_1=1}^{t-1}\sum_{i=1}^{t-1}\beta^\intercal(\ell_{i_1}\ell_{i_1}^\intercal-\Sigma)\mathbb{A}^\intercal\ell_t\ell_t^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right]$$
$$\underset{(2)}{=}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{t-1}\beta^\intercal(\ell_i\ell_i^\intercal-\Sigma)\mathbb{A}^\intercal\ell_i\ell_i^\intercal\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\beta\right]$$
$$=\frac{(T-1)T}{2}\beta^\intercal\mathbb{E}\left[(\ell_i\ell_i^\intercal-\Sigma)\mathbb{A}^\intercal\Sigma\mathbb{A}(\ell_i\ell_i^\intercal-\Sigma)\right]\beta \tag{E.13}$$
$$=\frac{(T-1)T}{2}\beta^\intercal\mathbb{E}\left[(\sqrt{\Sigma}A(\ell_i\ell_i^\intercal-\Sigma))^\intercal(\sqrt{\Sigma}A(\ell_i\ell_i^\intercal-\Sigma))\right]\beta.$$

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 (E.13) 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 $B(\mathbf{0}_d, c_z, \|\cdot\|) \subseteq \text{supp}(Z)$, 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

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

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 (E.13), either $A = \boldsymbol{O}_{d\times d}$ or $\beta = \mathbf{0}_d$ should hold. In both cases, Equation (E.9) can be re-written as

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

and this is covered by the original parametrization (Equation (E.8)) 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:

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

Here, (i) can be calculated as follows:

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

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 $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$.

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}.

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

Taking $d \to \infty$, we have

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

which further implies

$$\sqrt{Td}\frac{4(\sqrt{T}W + \ell_{i1})\ell_{k1}}{\sqrt{T(d-1)Z + (\sqrt{T}W + \ell_{i1} + \ell_{k1})^2} + \sqrt{T(d-1)Z + (\sqrt{T}W + \ell_{i1} - \ell_{k1})^2}}\ell_{i1}\ell_{k1}$$
$$\overset{d}{\to}\sqrt{Td}\frac{4(\sqrt{T}W + \ell_{i1})\ell_{k1}}{2\sqrt{Td}}\ell_{i1}\ell_{k1} = 2(\sqrt{T}W + \ell_{i1})\ell_{i1}\ell_{k1}$$

as $d \to \infty$. Therefore,

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

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. □

E.8 Empirical Validation of Theorem E.2 and Theorem 5.2

We now provide empirical validations for Theorem E.2 and Theorem 5.2. We provide the training details and the results as follows.

E.8.1 Empirical Validation of Theorem E.2

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

Our empirical analysis aims to demonstrate that the optimized model inherently emulates online gradient descent. To illustrate this, we will focus on two key convergence properties: $K^\intercal Q$ approaching the zero matrix $\boldsymbol{O}_{d\times d}$ and V converging to $a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$, where a and b are constants in $\mathbb{R}$. The conditions $K^\intercal Q = \boldsymbol{O}_{d\times d}$ and $V = a\mathbf{1}_d\mathbf{1}_d^\intercal + bI_{d\times d}$ imply that the function $g(Z_t; V, Q, K) = \sum_{i=1}^t (b-a)\ell_i$, 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 E.1.

E.8.2 Empirical Validation of Theorem 5.2

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 E.1.

E.9 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:

$$g(Z_t; V, K, Q, v_c, k_c, q_c) = \texttt{Operator}\left(\sum_{i=1}^t (V\ell_i + v_c)((K\ell_i + k_c))^\intercal \cdot (Qc + q_c)\right), \quad \text{(E.14)}$$

where the `Operator` denotes projection to the convex set.

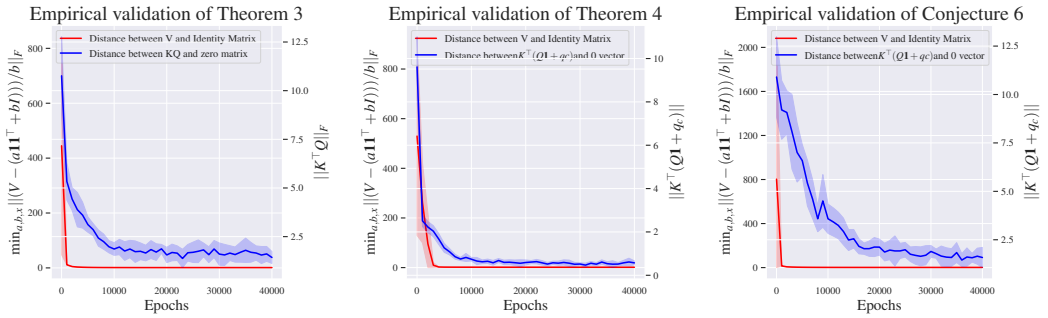

Figure E.1: Empirical validation of Theorem E.2 (top), Theorem 5.2 (middle), and Conjecture 3 (bottom). The observed convergence in Theorem E.2 and Conjecture 3's result suggests that configuration in Theorem E.2 and Conjecture 3 are not only the local optimal point, but it has the potential as being the global optimizer.

Conjecture 3. *Assume* $\Sigma = I$. *Then, the configuration that* $K^\intercal(Qc + q_c) = v_c = \mathbf{0}_d$ *and* $V = \widetilde{\Omega}\left(-\frac{1}{\sqrt{nd}}\right) I_{d\times d}$ *is a first-order stationary point of Equation* (5.2) *with* $N = 1$ *and* $h(x) = x^2$ *when* LLM_θ *is parameterized with Equation* (E.14), `Operator` $=$ `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 E.1 that Theorem E.2 and Conjecture 3 might also be a global optimizer, as training results have provided the configuration that Theorem E.2 and Conjecture 3 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

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

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

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

Therefore,

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

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

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

Therefore,

$$\frac{\partial f(V,a,\beta,v_c)}{\partial V_{rc}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$$
$$= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal \sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal \sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)} - \min_s \sum_{t=1}^{T}\ell_{ts}\right)$$
$$\frac{\partial}{\partial V_{rc}}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal \sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal \sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a + (\beta V + v_c a^\intercal)\ell_j + v_c\beta)\right)} - \min_s \sum_{t=1}^{T}\ell_{ts}\right)\Bigg]\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$$
$$= \mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js} + v\beta\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta V\ell_{jy} + v\beta\right)} - \min_s \sum_{t=1}^{T}\ell_{ts}\right)$$

$$\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{j=1}^{t-1}\beta\mathbf{1}(s=r)\ell_{jc}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}+v\beta\right)\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)\right)^2}\right.$$
$$\left.-\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}+v\beta\right)\sum_{y=1}^{d}\left(\sum_{j=1}^{t-1}\beta\mathbf{1}(y=r)\ell_{jc}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}+v\beta\right)\right)^2}\right)\Bigg]$$
$$=\beta\mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta V\ell_{jy}\right)}-\min_{s}\sum_{t=1}^{T}\ell_{ts}\right)$$
$$\Bigg(\underbrace{\frac{\sum_{t=1}^{T}\sum_{j=1}^{t-1}\sum_{y=1}^{d}\ell_{tr}\ell_{jc}\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jr}\right)\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jy}\right)\right)^2}}_{(i)}$$
$$\left.\underbrace{-\frac{\sum_{t=1}^{T}\sum_{j=1}^{t-1}\sum_{y=1}^{d}\ell_{ty}\ell_{jc}\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jr}\right)\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\beta k\sum_{j=1}^{t-1}\ell_{jy}\right)\right)^2}}_{(ii)}\right)\Bigg].$$

We can observe the followings: 1) if $r_1\neq c_1$ and $r_2\neq c_2$, $\left.\frac{\partial f}{\partial V_{r_1c_1}}\right|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}=\left.\frac{\partial f}{\partial V_{r_2c_2}}\right|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}$ holds, and 2) $\left.\frac{\partial f}{\partial V_{r_1r_1}}\right|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}=\left.\frac{\partial f}{\partial V_{r_2r_2}}\right|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}$.

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

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

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

Further, we have

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

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

Note that

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

Therefore,

$$\frac{\partial f(V,a,\beta,v_c)}{\partial a_x}\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}$$
$$=\mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)$$
$$\frac{\partial}{\partial a_x}\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(e_s^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}{\sum_{y=1}^{d}\exp\left(e_y^\intercal\sum_{j=1}^{t-1}(V\ell_j\ell_j^\intercal a+(\beta V+v_ca^\intercal)\ell_j+v_c\beta)\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)\Bigg]\Big|_{a=\mathbf{0}_d,v_c=v\mathbf{1}_d,V=kI}$$
$$=\mathbb{E}\Bigg[\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)$$
$$\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{j=1}^{t-1}(k\ell_{js}\ell_{jx}+v\ell_{jx})\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right.$$
$$\left.-\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\left(\sum_{j=1}^{t-1}(k\ell_{jy}\ell_{jx}+v\ell_{jx})\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right)\Bigg]$$
$$=\mathbb{E}\Bigg[k\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)}{\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}-\min_s\sum_{t=1}^{T}\ell_{ts}\right)$$
$$\left(\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\sum_{j=1}^{t-1}\ell_{js}\ell_{jx}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right.$$
$$\left.-\sum_{t=1}^{T}\sum_{s=1}^{d}\ell_{ts}\frac{\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{js}\right)\sum_{y=1}^{d}\left(\sum_{j=1}^{t-1}\ell_{jy}\ell_{jx}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)}{\left(\sum_{y=1}^{d}\exp\left(\sum_{j=1}^{t-1}\beta k\ell_{jy}\right)\right)^2}\right)\Bigg]$$

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

E.9.1 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 $\ell_{ts} \sim \text{Unif}([0,1])$ 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 E.2. We used the Monte Carlo estimation of $1,000,000$ times.

E.9.2 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 E.1.

E.10 Comparison with In-Context-Learning Analyses in Supervised Learning

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.

E.11 Details of Section 5.4

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 E.3.

Loss sequences with certain trends. 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 E.4, 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 E.4 shows that 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

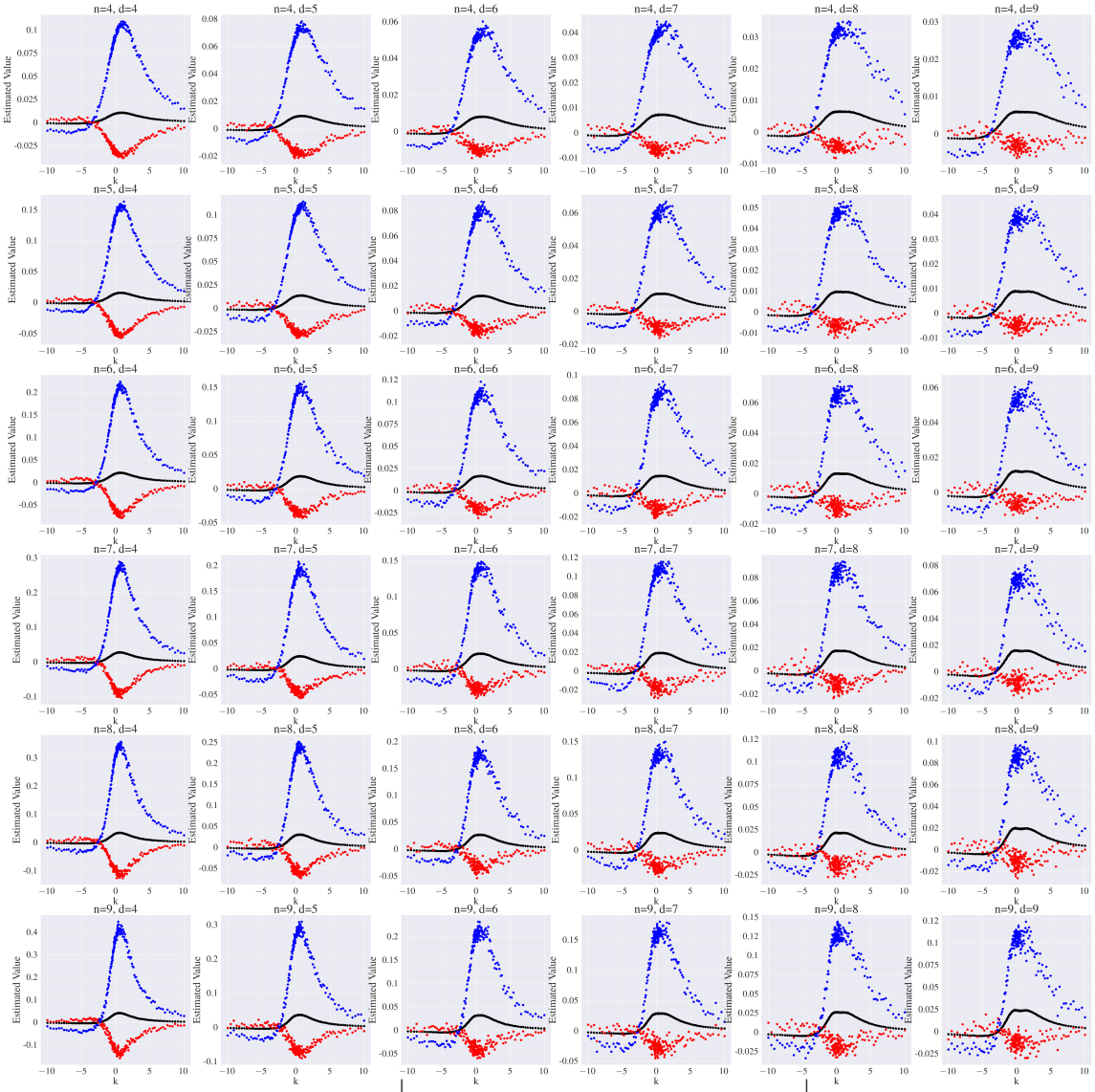

Figure E.2: Calculation of $20\frac{\partial f}{\partial V_{rc}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ $(r \neq c)$(red), $20\frac{\partial f}{\partial V_{rr}}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ (blue), and $\frac{\partial f}{\partial a_x}\Big|_{a=\mathbf{0}_d, v_c=v\mathbf{1}_d, V=kI}$ (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.

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.

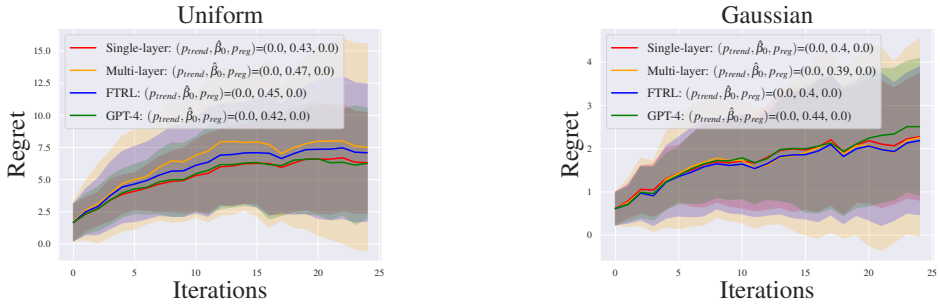

Figure E.3: Regret performance for the randomly generated loss sequences that are generated by Gaussian with truncation and uniform distribution. No-regret behaviors of single-layer and multi-layer self-attention models are validated by both of our frameworks (low p-values and $\widehat{\beta}_0 < 1$).

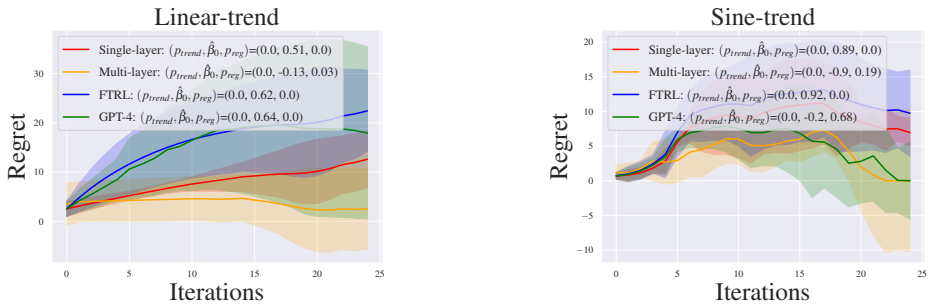

Figure E.4: Regret performance for the randomly generated loss sequences that are generated by linear-trend and sine-trend. No-regret behaviors of single-layer and multi-layer self-attention models are validated by both of our frameworks (low p-values and $\widehat{\beta}_0 < 1$).

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 $\mathrm{Unif}([0, 10])$. The results, as illustrated in Figure E.5, 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 E.2 or Theorem 5.2). Also, in practical training (with the empirical loss in Equation (E.3)), we possibly did not find the exact global minimum or stationary point of the *expected* loss in Equation (5.2). Hence, it is possible that GPT-4 may have lower regret than our trained models with the regret-loss.

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 3.4, 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 Section 5.2 and 5.3 has predicted.

Remark on performance discrepancy between single-agent and multi-agent settings. Why does GPT-4 exhibit better regret performance compared to single/multi-layer models in the single-agent setting, yet underperform in the multi-agent setting? What factors contribute to this discrepancy in its effectiveness across different settings?

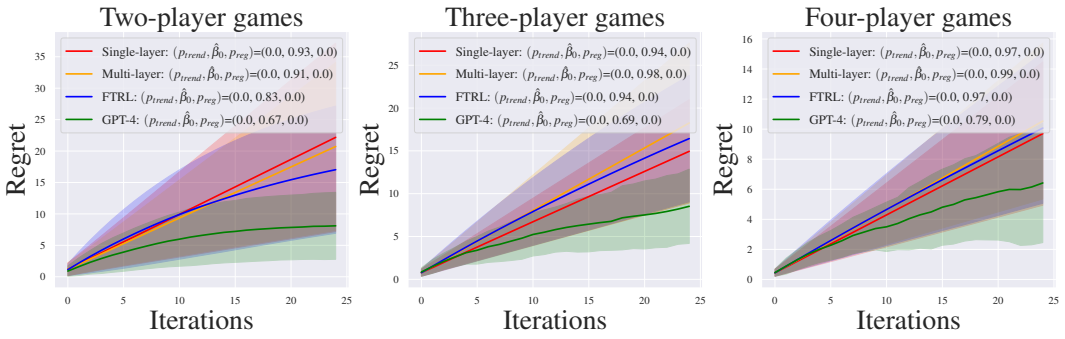

Figure E.5: Regret performance for the game with two players, three players, and four players general-sum games. No-regret behaviors of single-layer and multi-layer self-attention models are validated by both of our frameworks (low p-values and $\widehat{\beta}_0 < 1$).

In certain scenarios, LLMs can outperform FTRL/FTPL algorithms and single/multi-layer models. This phenomenon is primarily observed when the loss sequence exhibits discernible trends, as seen in the single-agent setting. In Section 3.4, we explored this behavior using canonical counterexamples for the follow-the-leader algorithm. Specifically, when the loss sequences display obvious or predictable patterns, LLMs can effectively infer the next loss vector based on historical data, enabling near-optimal decisions. This phenomenon can be further formalized through the lens of in-context learning. Conversely, FTRL/FTPL algorithms, constrained by their update rules, tend to produce near-uniform policies in such cases, as do single/multi-layer Transformer models. In Appendix C.7, we provide ablation studies to support these observations, demonstrating that LLMs leverage trends in the loss sequences by comparing their performance when provided with raw versus summarized historical data. When the loss sequences are summarized (e.g., through aggregation), the resulting loss vectors no longer reflect the trend, leading to significantly diminished performance by the LLMs. These findings have been clarified and emphasized in the updated manuscript.

In contrast, in multi-agent or game settings, the loss sequence trends depend on the behavior of other agents, rendering them inherently less predictable as all agents continually update their behavior policies. This increased unpredictability likely accounts for the comparable or inferior performance of LLMs relative to FTRL/FTPL algorithms or single/multi-agent-trained Transformer models in such settings.

E.11.1 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.

E.12 Ablation Study on Training Equation (5.2)

In this section, we provide an ablation study that changes N and k in Equation (5.2). To be specific, we will set $N = 1, 2, 4$, $f(x, k) = \max(x, 0)^k$, $h(x) = \max(x, 0)^2$, and $k = 1, 2$. 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 ($\ell_t \sim \text{Unif}([0, 10]^2)$), with the results in Figure E.6 and Figure E.7; and the Gaussian loss sequence ($\ell_t \sim \mathcal{N}(5 \cdot \mathbf{1}_2, I)$), with the results in Figure E.8 and Figure E.9.

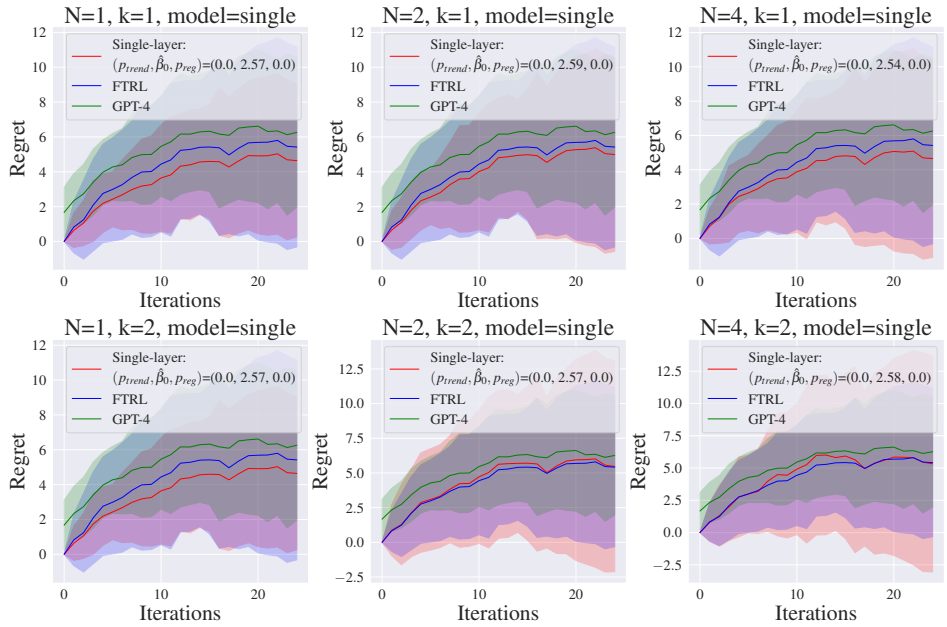

Figure E.6: Ablation study for the uniform loss sequence trained with single-layer self-attention layer and `Softmax` projection.

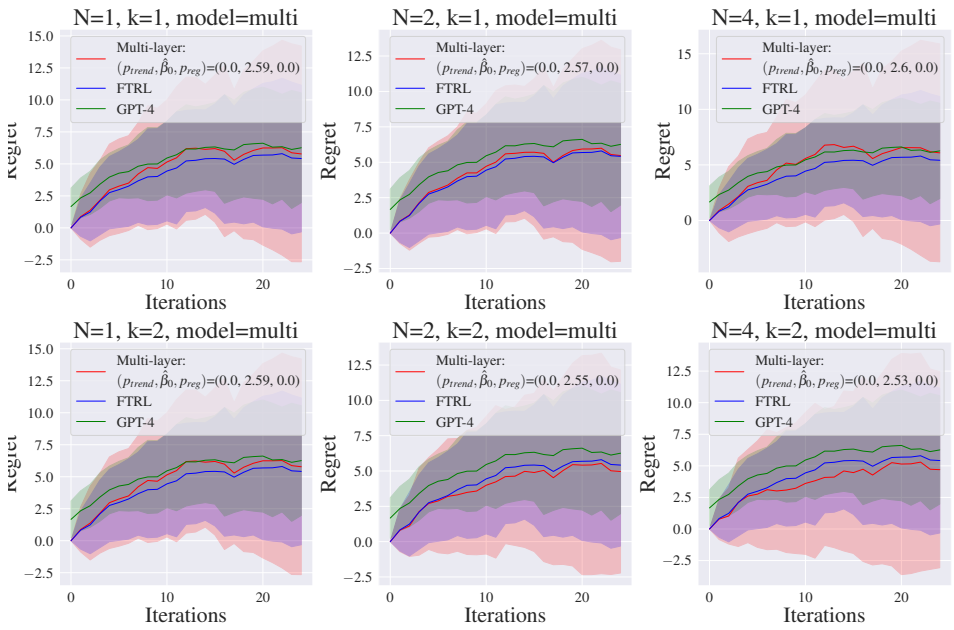

Figure E.7: Ablation study for the uniform loss sequence trained with multi-layer self-attention layer and `Softmax` projection.

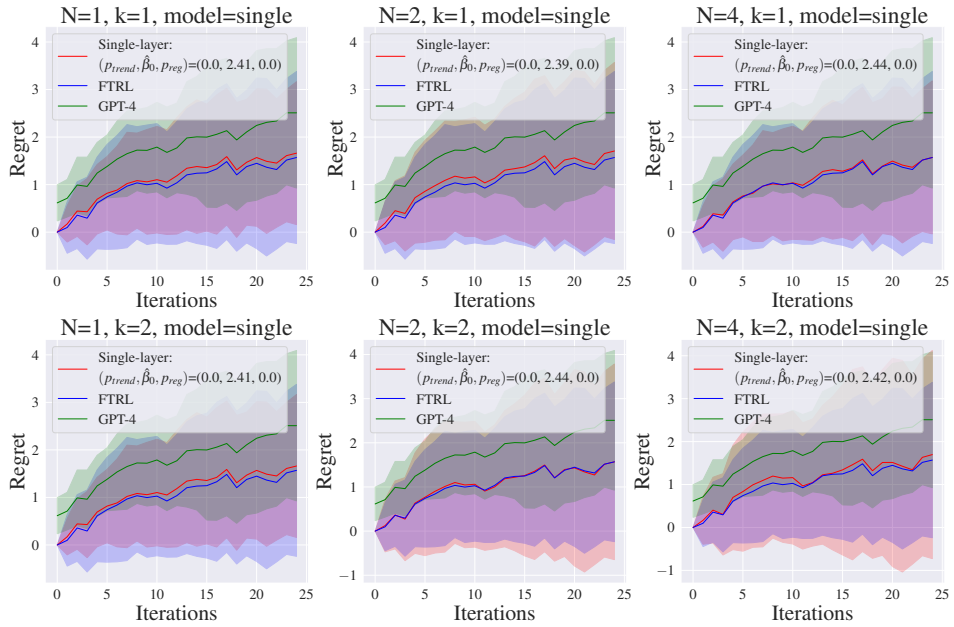

Figure E.8: Ablation study for the Gaussian loss sequence trained with single-layer self-attention layer and `Softmax` projection.

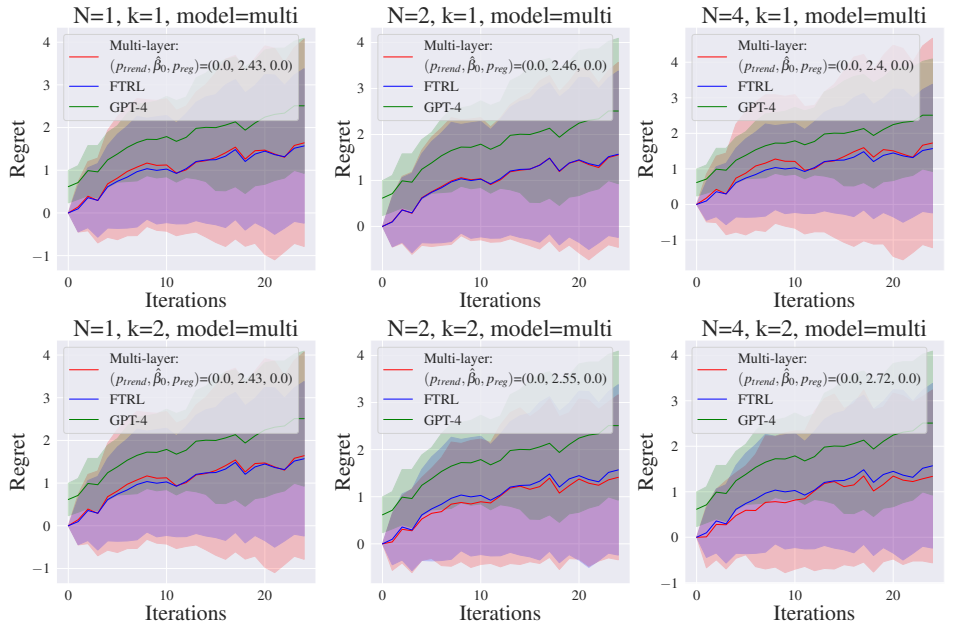

Figure E.9: Ablation study for the Gaussian loss sequence trained with single-layer self-attention layer and `Softmax` projection.

F LIMITATIONS AND 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, We provide the following limitations and list some potential 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. There are some other regret metrics we have studied, e.g., swap-regret (Blum & Mansour, 2007), which may lead to stronger equilibrium notions in playing repeated games, or policy regret (Arora et al., 2012a), which accounts for adaptive adversaries.
- Our new regret-loss has exhibited promises in our experiments for training modest-scale Transformers. One limitation is that we have not trained other larger-scale models, such as Foundation Models, for decision-making, which is an important ongoing effort.
- Our Theorem 4.1 towards explaining why LLMs achieved sublinear regret is highly hypothetical. Given that LLMs are such complex, random, and black-box systems, there are definitely behaviors that our Theorem 4.1 cannot fully capture, and there do exist other possible explanations. For example, an alternative in-context-learning-based explanation may be used to account for the *improved* performance of LLMs on specific loss sequences *with trends*. Specifically, LLMs may interpret past loss sequences as *demonstrations* to identify the latent trends, make accurate predictions on the next loss, and make optimal decisions. However, this explanation may not generalize to the loss sequences *without* obvious trends, complementing our explanations based on the connection to no-regret learning algorithms, which apply to general loss sequences (see Appendix C.7 for more discussions). Hence, it would be interesting to propose and validate other hypotheses for the observed behaviors of LLMs.
- 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 identify 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.