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

Chanwoo Park[⋆1], **Xiangyu Liu**[⋆2], **Asuman Ozdaglar**[1], **Kaiqing Zhang**[2]
[1] MIT, [2] University of Maryland, College Park

ABSTRACT

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

1 INTRODUCTION

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

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

[⋆]Equal contribution.

2023; Akata et al., 2023; Fan et al., 2023). These exciting empirical successes call for a rigorous examination and understanding through a theoretical lens of decision-making.

Regret, on the other hand, has been a core metric in (online) decision-making. It measures how "sorry" the decision-maker is, in retrospect, not to have followed the best prediction in hindsight (Shalev-Shwartz, 2012). It provides not only a sensible way to *evaluate* the sophistication level of online decision-makers, but also a quantitative way to measure their *robustness* against arbitrary (and possibly adversarial) environments. More importantly, it inherently offers a connection to modeling and analyzing *strategic behaviors*: the long-run interaction of no-regret learners leads to certain *equilibrium* when they repeatedly play games (Cesa-Bianchi & Lugosi, 2006). In fact, *no-regret* learning has served as a natural model for predicting and explaining human behaviors in strategic decision-making, with experimental evidence (Erev & Roth, 1998; Nekipelov et al., 2015; Balseiro & Gur, 2019). It has thus been posited as an important model of "rational behaviors" in playing games (Blum et al., 2008; Roughgarden, 2015). Hence, it is natural to ask:

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

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

Contributions. First, we carefully examine the performance of several representative pre-trained LLMs in the aforementioned benchmark online decision-making settings, in terms of *regret*. We observe that LLM agents can achieve regret sublinear in time in (non-stochastic) online learning settings, where the loss functions change over time either arbitrarily, or by following some patterns with bounded variation, and in playing both representative and randomly generated repeated games. For the latter, equilibria will emerge as the long-term behavior of the multi-LLM interactions. Second, we provide some theoretical insights into the observed sublinear regret behaviors, based on certain assumptions on the *supervised pre-training* procedure, a common practice in training large models for decision-making, and some hypothetical models for training data generation. In particular, we make a connection of the pre-trained LLMs to the known no-regret algorithm of *follow-the-perturbed-leader* (FTPL) under these assumptions. Third, we also identify (simple) cases where advanced LLMs such as GPT-4 fail to be no-regret. We thus propose a novel *unsupervised* training loss, *regret-loss*, which, in contrast to the supervised pre-training loss, does not require the *labels* of (optimal) actions. We then establish both *statistical* and *optimization* guarantees for regret-loss minimization, which, in particular, show that minimizing such a loss can *automatically* lead to the known no-regret learning algorithm of *follow-the-regularized leader* (FTRL), under single-layer self-attention parameterization. Our further experiments demonstrate the effectiveness of our new loss, especially in addressing the above "regrettable" cases. With the fast development of LLMs, we emphasize that our goal is not to assert whether (current) LLMs are no-regret learners or not, especially given both the positive and negative observations above. Instead, our hope is to introduce and inspire more rigorous metrics and principles into the current evaluation and development of LLM agents, for online and multi-agent strategic decision-making.

2 Preliminaries

Notation. For a finite set $\mathcal{S}$, we use $\Delta(\mathcal{S})$ to denote the simplex over $\mathcal{S}$. We denote $\mathbb{R}^+ := \{x \in \mathbb{R} \,|\, x \geq 0\}$. We define $\mathbf{0}_d$ and $\mathbf{1}_d$ as the d-dimensional all-zero and all-one vector, respectively, and $\boldsymbol{O}_{d\times d}$ and $I_{d\times d}$ as the $d\times d$-dimensional zero matrix and identity matrix, respectively. For a positive integer d, we define $[d] = \{1, 2, \ldots, d\}$. For $p \in \mathbb{R}^d$, $R > 0$ and $C \subseteq \mathbb{R}^d$ being a convex set, define $B(p, R, \|\cdot\|) := \{x \in \mathbb{R}^d \mid \|x - p\| \leq R\}$ and $\texttt{Proj}_{C,\|\cdot\|}(p) = \arg\min_{x\in C} \|x - p\|$. For any $x \in \mathbb{R}^d$, define $\texttt{Softmax}(x) = \left(\frac{e^{x_i}}{\sum_{i\in[d]} e^{x_i}}\right)_{i\in[d]}$. For a vector $v \in \mathbb{R}^n$, we use $\|v\|_p$ to denote its L_p-norm, with $\|v\|$ denoting the L_2-norm by default. We define $\mathbb{1}(\mathcal{E}) = 1$ if some event $\mathcal{E}$ is true, and $\mathbb{1}(\mathcal{E}) = 0$ otherwise. For a random variable X, we use $\text{supp}(X)$ to denote its support.

2.1 Online Learning & Games

Online learning. We consider the online learning setting where an agent interacts with the environment for T rounds, by iteratively making decisions based on the feedback she receives. Specifically, at each time step t, the agent chooses her decision policy $\pi_t \in \Pi$ for some bounded domain Π, and after her commitment to π_t, a bounded loss function $f_t : \Pi \to [-B, B]$ for some constant $B > 0$ is chosen by the environment, potentially in an adversarial fashion. The agent thus incurs a

loss of $f_t(\pi_t)$, and will update her decision to π_{t+1} using the feedback. We focus on the most basic setting where the agent chooses actions from a finite set $\mathcal{A}$ every round, which is also referred to as the *Experts Problem* (Cover, 1966; Vovk, 1990; Littlestone & Warmuth, 1994; Hazan, 2016), without loss of much generality (c.f. Appendix B.5 for a detailed discussion). In this case, Π becomes the simplex over $\mathcal{A}$, i.e., $\Pi = \Delta(\mathcal{A})$, and $f_t(\pi_t) = \langle \ell_t, \pi_t \rangle$ for some loss *vector* $\ell_t \in \mathbb{R}^d$ that may change over time, where $d := |\mathcal{A}|$.

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

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

2.2 Performance Metric: Regret

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

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

In the Experts Problem, the definition is instantiated as $\text{Regret}_{\mathscr{A}}((\ell_t)_{t\in[T]}) := \sum_{t=1}^{T}\langle \ell_t, \pi_{\mathscr{A},t}\rangle - \inf_{\pi\in\Pi}\sum_{t=1}^{T}\langle \ell_t, \pi\rangle$. With bandit-feedback, a common metric may also take further expectation for $\text{Regret}_{\mathscr{A}}$, over the randomness of the policies $(\pi_{\mathscr{A},t})_{t\in[T]}$. An algorithm $\mathscr{A}$ is referred to as being *no-regret*, if $\max_{(f_t)_{t\in[T]}} \text{Regret}_{\mathscr{A}}((f_t)_{t\in[T]}) \sim o(T)$, i.e., the worse-case regret grows sublinearly in T. Known no-regret algorithms include follow-the-regularized-leader (Shalev-Shwartz & Singer, 2007), follow-the-perturbed-leader (Kalai & Vempala, 2005) (see Appendix B.4 for more details).

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

3 Do Pre-Trained LLMs Have Regret? Experimental Validation

In this section, we explore the no-regret behaviors of representative LLMs (i.e., mainly GPT-4 Turbo and GPT-4, together with GPT-3.5 Turbo, Mixtral-8x7b-instruct, and Llama-3-70B-instruct), in the context of online learning and games. All experiments with LLMs are conducted using the public OpenAI (Openai, 2023) or LLM Engine (LLM Engine, 2023) Python API. We provide some hypothetical intuitions as to why pre-trained LLM might be no-regret in Appendix C.1, which will be made concrete next.

Interaction protocol. To enable the sequential interaction with LLMs, we first describe the setup and objective of our experimental study. At each round, we incorporate the entire history of loss vectors of past interactions into our prompts, as concatenated texts, and ask the LLM agent to determine

a policy that guides the decision-making for the next round. Note that since we hope to *evaluate* the sophistication level of pre-trained LLMs through online learning or games, we only provide simple prompts that she should utilize the history information, without providing explicit rules of *how* to make use of the history information, nor asking her to *minimize regret* (in any sense). A detailed description and an ablation study of the prompts are deferred to Appendix C.8, and an illustration of the protocol for playing repeated games is given in Figure C.1.

3.1 Framework for Sublinear Regret Behavior Validation

Before delving into the results, we note that to the best of our knowledge, we are not aware of any principled framework for validating sublinear growth of the regret with *finite-time* experimental data. Therefore, we propose two frameworks below to rigorously validate the no-regret behaviors of algorithms over a *finite* T, which might be of independent interest. More details can be found in Appendix C.3.

Trend-checking framework. We propose a statistical hypothesis test aligned with our objectives:

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.

Ideally, one should check if $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)/t$ approaches zero (or a negative value) 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. 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 $\mathbb{P}_{H_0}(\mathcal{E}(s,T)) \leq \frac{1}{2^{T-1}} \sum_{t=s}^{T-1} \binom{T-1}{t}$.

Regression-based framework. We propose an alternative approach by fitting the data with regression. In particular, one can use the data $\left\{\left(t, \log \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)\right)\right\}_{t\in[T]}$ to fit a function $g(t) = \beta_0 \log t + \beta_1$, where the estimate of β_0, i.e., $\widehat{\beta}_0$, satisfying $\widehat{\beta}_0 < 1$ may be used to indicate the no-regret behavior, i.e., the *sublinear* growth of $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)$ over time. While being simple, it cannot be directly used when $\text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right) < 0$. Hence, we set $\log \text{Regret}_{\mathscr{A}}\left((f_\tau)_{\tau\in[t]}\right)$ as -10 if this happens. We define p_{reg} as the p-value of the regression parameter $\widehat{\beta}_0$, and will report the pair of $(\widehat{\beta}_0, p_{reg})$ as the output of this framework.

3.2 Results: Online Learning

We now present the experimental results of pre-trained LLMs in online learning in: 1) (arbitrarily) changing environments, 2) non-stationary environments, and 3) bandit-feedback environments. Results for 2) and 3) are deferred to Appendices C.4 and C.5.

Changing environments. We first consider the setting with (arbitrarily) changing environments, which are instantiated as follows: 1) *Randomly-generated loss sequences*. At every timestep, we generate a random loss vector $\ell_t \sim \text{Unif}(\times_{i=1}^d [\min\{x_i, y_i\}, \max\{x_i, y_i\}])$ for $\{x_i, y_i \sim \text{Unif}(0,10)\}_{i\in[d]}$ or $\ell_t \sim \mathcal{N}(\boldsymbol{\mu}_d, I)$ with clipping to $[0,10]$ to ensure boundedness of the loss, where $\boldsymbol{\mu}_d \sim \text{Unif}([0,10]^d)$. Note that we use this as a way to *systematically* generate potentially arbitrary loss sequences, and also note that our regret was defined for each *realization* of the *random loss vectors* (instead of their expectations as in the definition of regret in *stochastic bandit* problems), which can be arbitrarily different across timesteps. 2) *Loss sequences with certain trends.* Although many real-world environments may change, they often change by following certain patterns. Therefore, we consider two representative trends, the *linear* trend and the *periodic* (sinusoid) trend. We sample $a, b \sim \text{Unif}([0,10]^d)$ and let $\ell_t = (b-a)\frac{t}{T} + a$ for the linear trend and $\ell_t = 5(1+\sin(at+b))$ for the periodic trend. In the experiments, we choose $d = 2$. The average regret (over multiple randomly generated instances) performance is presented in Figure 3.1[1], where we compare GPT-4 with well-known no-regret algorithms, FTRL with entropy regularization and FTPL with Gaussian perturbations (with tuned parameters). It is seen that these pre-trained LLMs can achieve sublinear regret in a large portion of the instances, and have sometimes even lower regret values than baselines.

[1]We emphasize that the error bars in the figures are *not* associated with the randomness/variance of the *algorithms/LLM-agents*, but with the randomness/variance of the *generation of environment instances*.

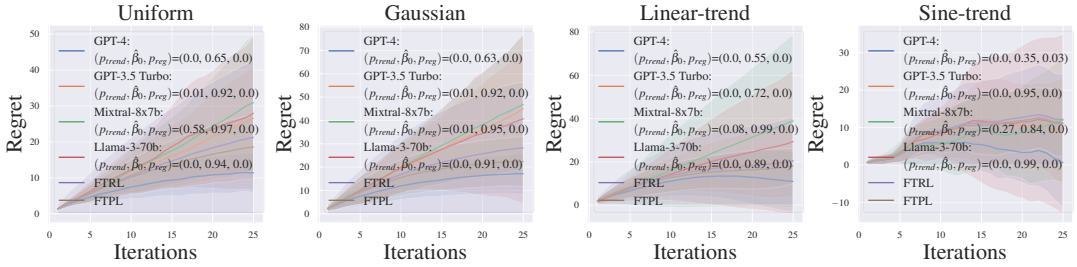

Figure 3.1: Regret of pre-trained LLMs for online learning with full-information feedback. Notably, both commercial and open-source LLMs can achieve sublinear regret as validated by our frameworks and the comparison with FTRL/FTPL, though the performances of weaker models of GPT-3.5 and open-source ones are worse. Interestingly, the GPT-4 model can even outperform well-known no-regret learning algorithms, FTRL and FTPL.

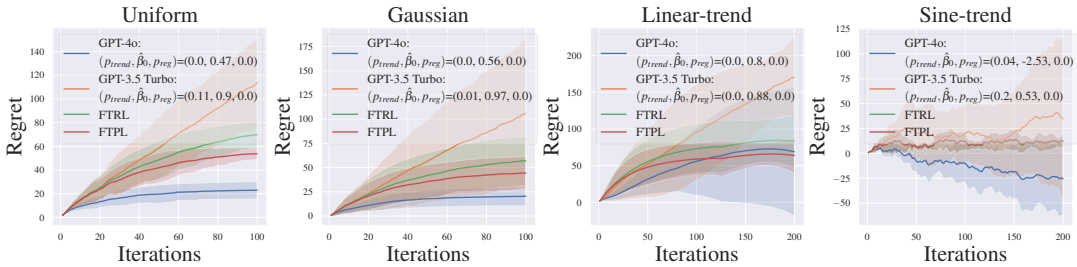

Figure 3.2: Regret of pre-trained LLMs for online learning with full-information feedback, with longer horizons of $T = 100$ and $T = 200$. In most cases, the LLMs can achieve sublinear regret as validated by our frameworks and the comparison with FTRL/FTPL, though the performances of the weaker model of GPT-3.5 is worse.

Behavioral patterns of LLMs. To understand how LLMs make decisions at each time step, we provided example outputs of LLMs *reasoning* how they generate their policies in Appendix C.10. We find that LLMs tend to use the history of the reward vectors by looking at their *sum/average*, and tend to introduce *randomization* in decision-making. These are known to be the keys to achieving no-regret behaviors in online learning (Hazan, 2016; Cesa-Bianchi & Lugosi, 2006).

Longer-horizon results. We also test the robustness and scalability of our empirical findings in more challenging environments. We extend the problem horizon to $T = 100$ for the two settings where loss vectors are generated in a stationary way (i.e., *Uniform* and *Gaussian*), and $T = 200$ for the other two non-stationary settings (i.e., *Linear-trend* and *Sine-trend*). Note that since in each round, we need to feed all the previous history to the LLMs, the API costs in fact scale *quadratically* with respect to the horizon T. Therefore, we replace GPT-4 by its cheaper (and more recent) version of GPT-4o. To further scale to even longer-horizon cases with $T = 500$, we *summarize* the history to reduce the prompt length by providing LLMs with the summation of the history loss associated with each action. Similar summary-based input was also used in the concurrent work Krishnamurthy et al. (2024), where both the *averaged reward* and the *action selection count* of each action were summarized for the (i.i.d.) stochastic bandit setting. The corresponding results are provided in Figure 3.2 and Table 1, where the LLMs can exhibit no-regret behaviors as validated by our frameworks and the comparison with FTRL/FTPL.

$(p_{trend}, \widehat{\beta}_o, p_{reg})$	GPT-4o	FTRL	FTPL
Uniform	(0.0, 0.85, 0.0)	(0.0, 0.6, 0.0)	(0.0, 0.52, 0.0)
Gaussian	(0.0, 0.86, 0.0)	(0.0, 0.64, 0.0)	(0.0, 0.68, 0.0)
Linear-trend	(0.02, 0.83, 0.5)	(0.02, 0.76, 0.1)	(0.01, 0.79, 0.0)
Sine-trend	(0.09, 0.28, 0.0)	(0.01, 0.24, 0.0)	(0.01, 0.26, 0.0)

Table 1: Longer-horizon ($T = 500$). GPT-4o model can still exhibit sublinear regret behaviors as validated by our frameworks and the comparison with FTRL/FTPL.

3.3 Results: Multi-Player Repeated Games

We now consider the setting when multiple LLMs make online decisions in a *shared* environment repeatedly. Specifically, at each round, the loss vectors each agent receives are determined by both her payoff matrix and the strategies of all other agents. Note that the payoff matrix is not directly

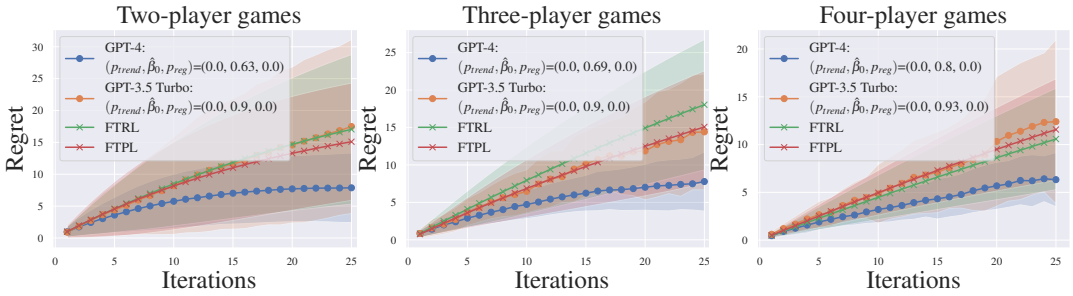

Figure 3.3: Regret of pre-trained LLMs for repeated games of different sizes, n most cases, both commercial and open-source LLMs can achieve sublinear regret as validated by our frameworks and the comparison with FTRL/FTPL. We report the regret of one agent for ease of presentation.

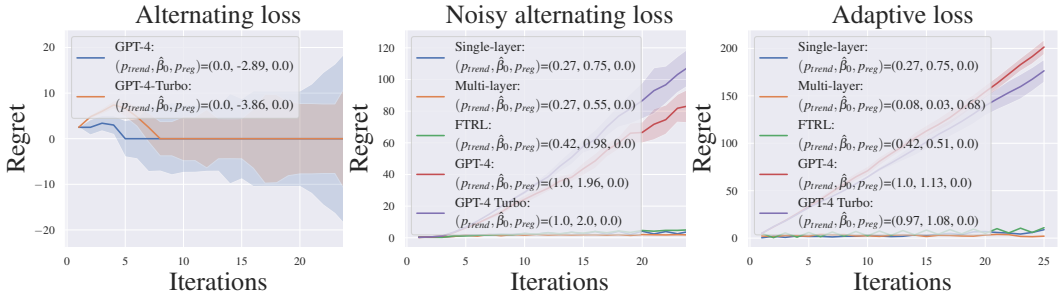

Figure 3.4: (left) Regret of GPT-4 (Turbo) under the canonical counterexample for FTL (Hazan, 2016, Chapter 5). (mid, right) Failure of GPT-4 (Turbo) on two scenarios with regrettable behaviors, while Transformers trained by our new regret-loss ($N = 1$) in Section 5 can achieve sublinear regret.

revealed to the LLM agent, but she has to make decisions in a completely online fashion based on the payoff vector marginalized by the opponents' strategies (see Figure C.1 for an example of the prompt). This is a typical scenario in learning in (repeated) games (Cesa-Bianchi & Lugosi, 2006).

Representative games. We first test LLMs on 6 representative general-sum games (*win-win, prisoner's dilemma, unfair, cyclic, biased,* and *second best*) studied in Robinson & Goforth (2005) (c.f. Appendix B.6). For each type of the game, we conduct 20 repeated experiments.

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

We summarize the experimental results in Figure 3.3, which are similar to the above in the online setting: for all types of games, pre-trained LLMs can achieve sublinear regret, which is often lower than that obtained by FTRL/FTPL for most games. We provide six instances of three-player general-sum games and six instances of four-player general-sum games in Figure C.4 and Figure C.5, respectively. Occasionally, GPT-4 even provides a negative regret value.

3.4 Pre-Trained LLM Agents Can Still Have Regret

The experiments above may suggest the no-regret behaviors of LLMs in online learning and game playing. However, is this capability *universal*? We show that the no-regret property can break for LLM agents if the loss vectors are generated in a more adversarial way.

Canonical counterexamples for follow-the-leader. First, we consider two well-known examples that the *follow-the-leader* (FTL) algorithm (Shalev-Shwartz, 2012) suffers from *linear regret*.

Example 1: $\ell_1(1) = 5, \ell_1(2) = 0$ and $\ell_t(2 - t\%2) = 10, \ell_t(1 + t\%2) = 0$ for $t \geq 2$ (Hazan, 2016).

Example 2: $\ell_t(2 - t\%2) = 10, \ell_t(1 + t\%2) = 0$ for $1 \leq t \leq c$ and $\ell_t(1) = 10$, $\ell_t(2) = 0$ for $c + 1 \leq t \leq T(= 500)$, for some integer c satisfying $0 < c < T$ (Feder et al., 1992).

Here, $\%$ denotes the modulo operation. Interestingly, for ***Example 1***, GPT-4 agent can easily identify the pattern for the loss sequence that the optimal action *alternates*, thus accurately predicting the loss it will receive and achieving low regret in Figure 3.4. For ***Example 2***, the GPT-4 agent with *raw history* input also provides an impressively lower (negative) regret than FTRL and FTPL

(Figure C.6). The GPT-4 agent with *summarized history* input, in contrast, suffers from much larger regret than FTRL and FTPL. We defer the detailed comparison between using raw history and summarized history to Figure C.6, and an explanation of LLMs' behaviors via predicting the *trend* of the loss instances to Appendix C.7. In summary, the GPT-4 agent may predict such worst-case sequences well, and does not fail in the same way as FTL, which is known to suffer from a lack of randomness in decisions.

Additionally, the results on ***Example 2*** also imply that summary-based history input can perform worse than the raw-history-based one in the adversarial setting we consider, while the former was claimed to be the key in succeeding in the i.i.d. stochastic bandit setting (Krishnamurthy et al., 2024). The regret values with these two types of input differ significantly, with a p-value of 1.2×10^{-157} under a one-sided independent t-test. These results further illustrate the fundamental differences between the settings considered in Krishnamurthy et al. (2024) and ours.

Noisy alternating loss sequence. Inspired by the above, we design a new loss sequence that is *similar but less predictable*, by adding some noise to the canonical counterexample. Specifically, we construct the following (simple) loss sequence with 2 actions such that $\ell_t(1 + t\%2) = \min(25/t, 10), \ell_t(2 - t\%2) \sim \text{Unif}([9, 10])$ for $t \in [25]$.

Adaptive loss sequence. We also develop a simpler but more *adaptive* loss sequence that takes the full power of the adversary in our online learning setup. After the GPT-4 agent provides π_t, we choose ℓ_t with $\ell_t(\arg\max_i \pi_{ti}) = 10$ and $\ell_t(3 - \arg\max_i \pi_{ti}) = 0$.

We also report the average regret over 20 repeated experiments for the later two settings using GPT-4 and more advanced GPT-4 Turbo in Figure 3.4, where we cannot reject the hypothesis that GPT-4 (Turbo) has linear regret by either our trend-checking or regression-based framework. These observations have thus motivated us to design new approaches to further promote the no-regret behaviors of the models, with additional training, as to be detailed in Section 5. Before it, we first provide some theoretical insights into the observed sublinear regret behaviors.

4 Why Do Pre-Trained LLMs (Not) Have Regret? A Hypothetical Model and Some Theoretical Insights

We now provide some plausible explanations about the observed no-regret behaviors of pre-trained LLMs, which are highly *hypothetical* by nature, since to the best of our knowledge, the details of pre-training these popular LLMs (e.g., GPT-3.5 Turbo and GPT-4), regarding data distribution, training algorithm, etc., have not been revealed. We instead make the explanations based on some existing assumptions in the literature for modeling human behaviors, and the recent literature on understanding LLMs and Transformers.

4.1 A (Human) Decision-Making Model: Quantal Response

A seminal model for human decision-making behaviors is the *quantal response* model, which assumes that humans are often imperfect decision-makers, and their *bounded rationality* can be modeled through unseen *latent variables* that influence the decision-making process (McFadden, 1976; McKelvey & Palfrey, 1995), for which we defer the formal definition and introduction to Appendix D.2. In online decision-making, given the *history* information with *multiple* loss vectors, we adopt the following generalization of the quantal response model.

Definition 4.1 (Quantal response against multiple losses)**.** *Given a set of losses $(\ell_i)_{i\in[t]}$, a noise distribution $\epsilon \sim P_{noise}$, and $\eta_t > 0$, the generalized quantal response against $(\ell_i)_{i\in[t]}$ is defined as*

$$P^{\eta_t}_{quantal}\left(a \,\middle|\, (\ell_i)_{i\in[t]}\right) := P^{\eta_t}_{quantal}\left(a \,\middle|\, \sum_{i=1}^{t} \ell_i\right) = \mathbb{P}\left(a \in \arg\min_{a'\in\mathcal{A}} \; z(a')\right), \text{ where } z = \eta_t \epsilon + \sum_{i=1}^{t} \ell_i.$$

In simpler terms, the generalized quantal response is defined as the standard quantal response against the *summation* of the losses. Such a model has been investigated in the learning-in-games and behavioral economics literature (see Appendix D.2 for more details). Such a definition is also aligned with our empirical findings on LLMs' behavioral patterns in Section 3.2: i) evaluating the summation/average; ii) introducing randomization in decision-making. To gain more insights into these empirical findings, we next analyze a case where pre-training under certain assumptions provably leads to the quantal response behaviors and further yields no-regret guarantees.

4.2 Case Study: Pre-Training under Canonical Data Distribution

Pre-training of LLMs is predominantly based on *next-token prediction*. When applying LLMs to sequential decision-making, the model receives the context of the decision-making task as

$(x_1, x_2, \cdots, x_N)$ and then generates $(x_{N+1}, \cdots, x_M)$ encoding the *action* for some $N, M \in \mathbb{N}^+$ and $N < M$, where each $x_i \in \mathcal{V}$ represents one *natural language token* for $i \in [M]$, and $\mathcal{V}$ is the finite token set. This process can be conceptualized as *predicting the optimal action* in the form of the next token prediction (Yao et al., 2023b; Shinn et al., 2023; Liu et al., 2023a;e). Note that this training procedure may also appear in the form of *supervised fine-tuning (SFT)* for downstream tasks of decision-making or question-answering, where optimal action labels may be easier to obtain (Cobbe et al., 2021; Li et al., 2022; Lewkowycz et al., 2022). Meanwhile, large models are often (pre-)trained under several *fixed/stationary* environments (Laskin et al., 2023; Lin et al., 2024; Lee et al., 2023; Reed et al., 2022), which may limit their ability to handle *arbitrary/non-stationary/adversarial* loss sequences in online learning. Thus, it is natural to ask: *Is it possible to have no-regret behaviors emerging as a consequence of this (optimal) action prediction, under only a* fixed *pre-training distribution of the environments?*

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

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{x_{1:N_{t+1}} \sim P_t^{text}} \sum_{j=1}^{N_{t+1}} \log \mathrm{LLM}_\theta \left(x_j \mid x_{1:j-1} \right), \qquad (4.1)$$

where we define $\mathrm{LLM}_\theta \left(x_1 \mid x_{1:0} \right) = \mathrm{LLM}_\theta \left(x_1 \right)$.

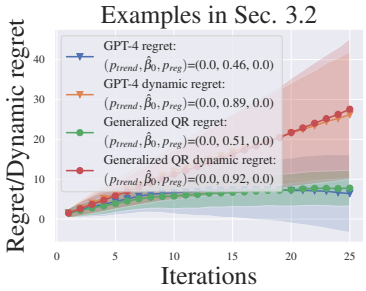

Figure 4.1: Comparison of GPT-4 with the generalized QR model, where the model can very well capture the behavior of the GPT-4 agent for examples in Section 3.2.

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

Assumption 1 (Pre-training distribution)**.** *Given $T \in \mathbb{N}^+$, $t \in [T]$, $N_{t+1} \in \mathbb{N}^+$, there are latent variables $(z, \ell_{1:t})$, $N_1, \cdots, N_t \in [N_{t+1}]$, $N_0 = 0$, such that $\mathbb{P}(z, \ell_{1:t}, x_{1:N_{t+1}}) = \mathbb{P}(z, \ell_{1:t}) \mathbb{P}(x_{1:N_t} \mid \ell_{1:t}) \mathbb{P}(x_{N_t+1:N_{t+1}} \mid z)$, and $P_t^{text}(x_{1:N_{t+1}}) := \mathbb{P}(x_{1:N_{t+1}}) = \int_z \int_{\ell_{1:t}} \mathbb{P}(z, \ell_{1:t}, x_{1:N_{t+1}}) d\ell_{1:t} dz$. Intuitively, tokens $\{x_{N_{i-1}+1:N_i}\}_{i \in [t]}$ encode the context, i.e., information for $\ell_{1:t}$, and the user will decode action a from $x_{N_t+1:N_{t+1}}$.*

To further understand our assumption, we provide an example in Appendix D.3, showing how a natural text corpus may satisfy it. Similar assumptions that suppose the existence of such latent variables in generating the pre-training datasets have also been made recently in Lee et al. (2023); Lin et al. (2024); Liu et al. (2023e), for understanding the in-context decision-making behaviors of LLMs/Transformers through posterior sampling, for which we defer a detailed comparison to Appendix D.8. In particular, we show in Theorem 4.1 that if the noise, i.e., $\ell_i - z$ is modeled as Gaussian distributions and $x_{N_t+1:N_{t+1}}$ encodes the optimal action for z, the pre-trained LLM provably recovers the prominent human behavior model in Section 4.1, the quantal response model.

Theorem 4.1 (Informal: Emergence of no-regret behavior)**.** *Suppose Assumption 1 holds with both the prior distribution of z and the conditional distribution of $\{\ell_i \mid z\}_{i \in [t]}$ being Gaussian, and $x_{N_t+1:N_{t+1}}$ encodes the optimal action for z. Then, with the function class of LLM_θ being expressive enough, and $\theta^\star$ being a maximizer of Equation (4.1), the behavior of $\mathrm{LLM}_{\theta^\star}$ follows Definition 4.1. Furthermore, the use of $\mathrm{LLM}_{\theta^\star}$ can achieve no (dynamic) regret for (non-stationary) online learning with full-information/bandit feedback for arbitrary loss vectors (with bounded variation).*

The formal statement and proof are deferred to Appendix D.6. The results show that even when pre-training is conducted solely with loss vectors generated from *stationary* distributions ($\ell_{1:t}$ are i.i.d. conditioned on z), it can still enable the *emergence of no-regret behaviors* in online learning against *potentially adversarial losses*. Key in the proof is the connection of pre-trained LLM models to the online learning algorithm of FTPL. Furthermore, Assumption 1 can be relaxed to better match

the actual LLMs' pre-training data distributions from diverse sources (c.f. Appendix D.7), and the prior distribution of z could also be replaced by a general distribution (c.f. Theorem D.2). Finally, we point out its implications for playing games in Appendix D.6.1.

How well can our hypothetical model class predict actual LLMs' behaviors? To further verify our theoretically-justified model in Theorem 4.1, we propose to *estimate* the parameters of $\{\eta_t\}_{t=0}^{T-1}$ in Definition 4.1 using the interaction data with actual LLMs, and use the estimated model to predict LLMs' behaviors on some test set. In Figure 4.1, we show the averaged regret for the LLMs and our estimated model, where the generalized quantal response can *very well capture* the behavior of the LLM agent for all problem instances in Section 3.2, on which the LLMs oftentimes achieve sublinear regret, justifying the applicability of our hypothetical model and assumptions.

Finally, we acknowledge that for existing pre-trained LLMs like GPT-4, the canonical assumptions above, though may be further relaxed (c.f. Remark D.3), may not hold in general. More importantly, the *supervision labels*, i.e., the optimal action given z, may be sometimes imperfect or unavailable in the dataset. These caveats motivate the study in our next section.

5 Provably Promoting No-Regret Behavior by a New Loss

In light of the observations in Section 3, we ask the question:

Is there a way to enhance the no-regret property of the models ***without*** *(optimal) action labels?*

To address this question, we propose to train models with a new *unsupervised learning* loss that naturally provides no-regret behaviors. We will particularly focus on the *Transformer* architecture (Vaswani et al., 2017) under this new loss, a common architecture used in most existing LLMs.

5.1 A New Unsupervised Training Loss: *Regret-Loss*

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

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

where $\|\ell_t\|_\infty \le B$ for $t \in [T]$. As discussed in Kirschner et al. (2023), directly minimizing the max regret can be computationally challenging, except for superficially simple problems. Moreover, Equation (5.1) is not necessarily differentiable with respect to the parameter θ, if it does not satisfy the condition of Danskin's Theorem (Danskin, 1966); or even if it is differentiable (i.e., the maximizer of $(\ell_t)_{t\in[T]}$ is unique), computation of derivatives can be challenging since we need to calculate $\arg\max_{(\ell_t)_{t\in[T]}} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})$ while there is an $\inf$ in the definition of regret. Therefore, we provide a general class of surrogate losses to approximate Equation (5.1):

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

where $k \in \mathbb{N}^+$, $N \in \mathbb{N}^+$, $h : \mathbb{R} \to \mathbb{R}^+$ is a continuous function, with continuous derivative h', and $f(\cdot,k) : \mathbb{R} \to \mathbb{R}^+$ is a continuous function for each $k \in \mathbb{N}^+$, satisfying $\lim_{k\to\infty} \frac{f(R_1,k)}{f(R_2,k)} = \infty\cdot\mathbb{1}(R_1 > R_2) + \mathbb{1}(R_1 = R_2)$, where we use the convention of $\infty\cdot 0 = 0$. These conditions on h, f will be assumed throughout the paper. Examples of such an f include $f(x,k) = x^k$ and $\exp(kx)$. We will sample N trajectories of loss sequences $(\ell_t^{(j)})_{t\in[T],j\in[N]}$ from some continuous probability distribution supported on $[-B,B]^{T\times N}$ (without other additional statistical assumptions), and the expectation in Equation (5.2) is thus taken with respect to this distribution. In Appendix E.2, we prove that under certain regularity conditions of f and h, we have

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

and the uniform convergence of $\mathcal{L}(\theta,k,N)$: $\lim_{N,k\to\infty} \sup_{\theta\in\Theta} \left|h\left(\max_{\ell_1,\ldots,\ell_T} \text{Regret}_{\text{LLM}_\theta}((\ell_t)_{t\in[T]})\right) - \mathcal{L}(\theta,k,N)\right| = 0$, where Θ is a compact set of the model parameters. Hence, one can expect that minimizing the loss function in Equation (5.2) with large enough k and N may promote the trained models to have a small regret value. We will hereafter refer to Equation (5.2) as the *regret-loss*.

5.2 Generalization and Regret Guarantees of Regret-Loss Minimization

We first establish a *statistical* guarantee under general parameterizations of LLM_θ that are Lipschitz with respect to θ, including the Transformer-based models as used in GPT-4 and most existing LLMs (see Proposition 2 for an example with a formal statement). This guarantee focuses on their *generalization ability* when trained to minimize the empirical regret loss (c.f. Equation (E.3)), denoted as $\widehat{\mathcal{L}}(\theta, k, N, N_T)$, by replacing the expectation $\mathbb{E}$ in Equation (5.2) with the empirical mean using N_T samples. We denote $\widehat{\theta}_{k,N,N_T} \in \arg\min_{\theta\in\Theta} \widehat{\mathcal{L}}(\theta, k, N, N_T)$, and present the generalization guarantee in Theorem E.1. Thanks to the uniform convergence of $\mathcal{L}(\theta, k, N)$ (c.f. Appendix E.2), we further obtain the following theorem on the regret guarantee of $\text{LLM}_{\widehat{\theta}_{k,N,N_T}}$:

Theorem 5.1. (Regret). *Suppose*[2] *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*

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

We defer the proof of the theorem to Appendix E.4. Therefore, if additionally, the model parameterization (e.g., Transformers) can *realize* a no-regret algorithm (as to be shown next), then Theorem 5.1 means that with a large enough N_T, the learned $\text{LLM}_{\widehat{\theta}_{k,N,N_T}}$ becomes a *no-regret* learner, i.e., $\text{Regret}_{\text{LLM}_{\widehat{\theta}_{k,N,N_T}}}\left((\ell_t)_{t\in[T]}\right) = o(T)$. Finally, as a consequence, it is folklore that when multiple such LLMs interact, a coarse correlated equilibrium will emerge in the long-term (c.f. Corollary 1).

5.3 Regret-Loss Trained Transformers Can be Online Learning Algorithms

Despite the generality of the previous results, one cannot use an *infinitely large* N and k in practice. Hence, we now provide results when N is finite, for the architecture of *Transformer* models (Vaswani et al., 2017). We focus on single-layer (linear) self-attention models, as in most recent theoretical studies of Transformers (Ahn et al., 2023; Zhang et al., 2023a; Mahankali et al., 2023), and $N = 1$. Note that in this case, the choice of f (and thus k) is not relevant. Thus, throughout this subsection, we drop superscript (j) in Equation (5.2). We sample ℓ_t for $t \in [T]$ as realizations of some random variable Z, where we assume that Z is symmetric about zero, and $\text{Var}(Z) = \Sigma \succ 0$. We consider the single-layer *linear* self-attention model as follows, for which we can show that the *global optimizer* of our regret-loss can automatically lead to a no-regret learning algorithm:

$$g(Z_t; V, K, Q, v_c, k_c, q_c) = \sum_{i=1}^{t}(V\ell_i + v_c)\left((K\ell_i + k_c)^\intercal \cdot (Qc + q_c)\right). \tag{5.3}$$

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.*

Theorem 5.2 not only shows the *capacity* of self-attention models: it can realize online learning algorithms, but also shows, more importantly, that minimizing our new regret-loss may *automatically* produce it. In particular, one does not need to hard-code the parameters of the Transformer to implement no-regret algorithms. Under single-layer self-attention parameterization (with softmax), we can also show that a *stationary point* of the loss function (Equation (5.2)) can lead to FTRL (c.f. Appendix E.5). Some potential generalizations of the results are also discussed in Appendix E.9.

5.4 Experimental Results for Regret-Loss Trained Transformers

We now provide experimental results for minimizing our *regret-loss* with the Transformer models, and evaluate in the following environments: 1) randomly-generated loss sequences (Figure E.3); 2) loss sequences with certain trends (Figure E.4); 3) repeated games (Figure E.5); and 4) counterexamples for pre-trained LLMs to be regrettable (Figure 3.4). Training setup can be found in Appendix E.11.1. We also provide an ablation study for optimizing Equation (5.2) in Appendix E.12.

Finally, we provide discussions on the limitations and future directions in Appendix F.

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

ACKNOWLEDGEMENT

The authors thank Constantinos Daskalakis, Kristian Georgiev, Noah Golowich, Dingwen Kong, Akshay Krishnamurthy, and Aleksander Madry for their helpful feedback. In particular, the authors thank Dingwen Kong for discussing the truncation idea in proving Lemma 8, and thank Akshay Krishnamurthy for bringing up a concurrent work that inspired our new experiments for the stochastic bandit setting that strengthened our paper. X.L. and K.Z. acknowledge the support from the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-24-1-0085 and NSF CAREER Award-2443704.

REFERENCES

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

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

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

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

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advanced in Neural Information Processing Systems*, 2023.

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

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

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

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *International Conference on Learning Representations*, 2023.

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

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1747–1754, 2012a.

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

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

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advanced in Neural Information Processing Systems*, 2023.

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

Santiago R Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019.

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

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

Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

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

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

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

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

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

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

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

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

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *International Conference on Learning Representations*, 2024.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *International Conference on Learning Representations*, 2024.

Yiting Chen, Tracy Xiao Liu, You Shan, and Songfa Zhong. The emergence of economic rationality of gpt. *Proceedings of the National Academy of Sciences*, 120(51):e2316205120, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Thomas M Cover. *Behavior of sequential predictors of binary sequences*. Number 7002. Stanford University, Stanford Electronics Laboratories, Systems Theory ..., 1966.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.247. URL https://aclanthology.org/2023.findings-acl.247.

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

Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, 34:27604–27616, 2021.

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

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

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

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

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

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

Meir Feder, Neri Merhav, and Michael Gutman. Universal prediction of individual sequences. *IEEE Transactions on Information Theory*, 38(4):1258–1270, 1992.

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

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

Drew Fudenberg and David M Kreps. Learning mixed equilibria. *Games and Economic Behavior*, 5(3):320–367, 1993.

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

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

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

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *International Conference on Machine Learning*, 2023.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507.

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

Josef Hofbauer and William H Sandholm. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294, 2002.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, et al. Metagpt: Meta programming for multi-agent collaborative framework. *International Conference on Learning Representations*, 2024.

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

Arnljot Hoyland and Marvin Rausand. *System reliability theory: Models and statistical methods*. John Wiley & Sons, 2009.

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

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

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

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

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

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

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *arXiv preprint arXiv:2403.15371*, 2024.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *International Conference on Learning Representations*, 2023.

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

Jonathan N Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Neural Information Processing Systems*, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

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

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

Ruosen Li, Teerth Patel, and Xinya Du. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*, 2023c.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.

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

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. *International Conference on Machine Learning*, 2023e.

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

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

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *International Conference on Learning Representations*, 2024.

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

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

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*, 2023b.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023c.

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

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

LLM Engine. LLM Engine, 2023. URL https://llm-engine.scale.com.

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

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

Weichao Mao, Kaiqing Zhang, Ruihao Zhu, David Simchi-Levi, and Tamer Başar. Model-free non-stationary RL: Near-optimal regret and applications in multi-agent RL and inventory control. *arXiv preprint arXiv:2010.03161*, 2020.

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

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759.

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

Denis Nekipelov, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *ACM Conference on Economics and Computation*, pp. 1–18, 2015.

Openai. Gpt-4 technical report. 2023.

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

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

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

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

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

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

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. Peer: A collaborative language model. *International Conference on Learning Representations*, 2023.

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

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

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

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

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

Significant Gravitas. Autogpt, 2023. URL https://github.com/Significant-Gravitas/AutoGPT.

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

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

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

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

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

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

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

Volodimir G Vovk. Aggregating strategies. In *Proceedings of the third Annual Workshop on Computational Learning Theory*, pp. 371–386, 1990.

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

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

Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *International Conference on Machine Learning 2023 Workshop ES-FoMO*, 2023b.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*, 2023c.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *Advances in neural information processing systems*, 2023d.

Chen-Yu Wei and Haipeng Luo. Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. In *Conference on learning theory*, pp. 4300–4354. PMLR, 2021.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a.

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

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation, 2024a. URL https://arxiv.org/abs/2305.19860.

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

Yue Wu, Xuan Tang, Tom Mitchell, and Yuanzhi Li. Smartplay: A benchmark for llms as intelligent agents. In *The Twelfth International Conference on Learning Representations*, 2024b.

Fanzeng Xia, Hao Liu, Yisong Yue, and Tongxin Li. Beyond numeric awards: In-context dueling bandits with llm agents. *arXiv preprint arXiv:2407.01887*, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *International Conference on Learning Representations*, 2022.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7572–7590, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.508. URL https://aclanthology.org/2023.findings-emnlp.508.

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

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*, 2023b.

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

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *International Conference on Learning Representations*, 2024.

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

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

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

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

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

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

CONTENTS