Binary Hypothesis Testing for Softmax Models and Leverage Score Models

Yuzhou Gu[1] **Zhao Song**[2] **Junze Yin**[3]

Abstract

Softmax distributions are widely used in machine learning, including Large Language Models (LLMs), where the attention unit uses softmax distributions. We abstract the attention unit as the softmax model, where given a vector input, the model produces an output drawn from the softmax distribution (which depends on the vector input). We consider the fundamental problem of binary hypothesis testing in the setting of softmax models. That is, given an unknown softmax model, which is known to be one of the two given softmax models, how many queries are needed to determine which one is the truth? We show that the sample complexity is asymptotically $O(\epsilon^{-2})$ where ϵ is a certain distance between the parameters of the models. Furthermore, we draw an analogy between the softmax model and the leverage score model, an important tool for algorithm design in linear algebra and graph theory. The leverage score model, on a high level, is a model which, given a vector input, produces an output drawn from a distribution dependent on the input. We obtain similar results for the binary hypothesis testing problem for leverage score models.

1. Introduction

In transforming various aspects of people's lives, large language models (LLMs) have exhibited tremendous potential. In recent years, numerous content learning and LLMs have been developed, including notable models such as Adobe Firefly, Microsoft 365 Copilot (Spataro, 2023), Adobe Photoshop, and Google's Meena chatbot (Rathee, 2020), along with the GPT series, the DeepSeek series, Google's Gemini series and others (Radford et al., 2018; 2019; Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019; Brown et al., 2020; ChatGPT, 2022; OpenAI, 2023; Team, 2023; Liu et al., 2024a). These models, together with those built upon them, have demonstrated significant prowess across diverse fields. The robustness and vitality of their development are attested to by the widespread integration of LLMs. In the realm of Natural Language Processing (NLP), evaluations by (Liang et al., 2023; Laskar et al., 2023; Choi et al., 2023; Bang et al., 2023) center around natural language understanding, while (Wang et al., 2023; Qin et al., 2023a; Pu & Demberg, 2023; Chia et al., 2023; Chen et al., 2023b) delve into natural language generation. LLMs have found applications in diverse fields, including both social science and science (Guo et al., 2023; Deroy et al., 2023; Ferrara, 2023; Nay et al., 2023), medical applications (Chervenak et al., 2023; Johnson et al., 2023), cybersecurity (Sun et al., 2023; Xu et al., 2024; Li et al., 2025b; Huang et al., 2025; Das et al., 2025) and engineering (Pallagani et al., 2023; Sridhara et al., 2023; Bubeck et al., 2023; Liu et al., 2023a), showcasing their potent capabilities. A consistent theme among these models is the adoption of the transformer architecture, a proven and highly efficient framework. The prevailing prevalence of models like ChatGPT (OpenAI, 2023) further underscores the transformative impact of this architecture.

However, there is a crucial problem with LLMs: their training costs and uncertainty regarding their inference ability in different parts of the whole. Understanding how different domains work is important in retrieval argument generation (RAG) (Siriwardhana et al., 2023; Zamani & Bendersky, 2024; Salemi & Zamani, 2024), as well as sparsity for LLMs by identifying the ability domain in the model which is important in solving the problem above. Then a question arose:

Can we distinguish different ability parts of large language models by limited parameters sampling?

We take an initial step toward addressing this question from a theoretical perspective. As we delve deeper into LLMs, the softmax mechanism is found to play an important role in the computation of self-attention. Thus, it is imperative to study how the self-attention mechanism works, why it contributes significantly to the impressive capabilities of LLMs, and what role it plays are still not fully understood.

Therefore, in this work, we want to explore the mechanism of softmax distribution from a binary hypothesis testing

[1] Institute for Advanced Study [2] University of California Berkeley [3] Boston University. Correspondence to: Zhao Song <magic.linuxkde@gmail.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

perspective. By delving into the intricacies of the softmax formulation, we explore which parameters are important by explaining how the softmax can be distinguished from each other. By delving into this idea, we can determine how many parameters are important in the inference of transformers (Vaswani et al., 2017). In continuation of the paper and drawing upon a formulation similar to softmax, we also direct our attention to the distribution of leverage scores. Much like softmax, the leverage score is a distribution parameterized by a matrix. Both softmax and leverage score can be treated as functions of distribution within this context. Importantly, resembling softmax, leverage score assumes significance across various fields. Leverage scores have demonstrated their significant utility in both linear algebra and graph theory. In the field of graph theory, researchers have extensively explored the application of leverage scores in various areas such as the generation of random spanning trees (Schild, 2018), max-flow problems (Daitch & Spielman, 2008; Madry, 2013; 2016; Liu & Sidford, 2020), maximum matching (Brand et al., 2020a; Liu et al., 2020b), and graph sparsification (Spielman & Srivastava, 2008a). Many studies have delved into the deep exploration of leverage scores, showcasing their effectiveness in optimization tasks such as linear programming (Lee & Sidford, 2014; Cohen et al., 2019b; Song, 2019; Lee et al., 2019; Brand et al., 2020b; Song & Yu, 2021; Liu et al., 2023c; Song et al., 2024b; Qin et al., 2023b; Gu et al., 2025), cutting-plane methods (Vaidya, 1989; Lee et al., 2015; Jiang et al., 2020b), semi-definite programming (Jiang et al., 2020a; Huang et al., 2022; Chen et al., 2023a; Gu et al., 2024), and the approximation of the John Ellipsoid (Cohen et al., 2019a; Song et al., 2022; Li et al., 2024a;b). These applications underscore the importance of leverage scores in the context of theory of computer science and linear algebra. Based on the analysis provided, both the leverage score and softmax computation are parameterized by a single matrix. Given the significance of the application of softmax and computation, understanding the influence on parameter behavior becomes crucial. Hence, we delve into this inquiry by differentiating the model through parameter sampling and discussing how the number of samples affects the distinguishing ability.

A softmax model is parameterized by a matrix $A \in \mathbb{R}^{n \times d}$, and denoted $\mathtt{SoftMax}_A$. Given $x \in \mathbb{R}^d$, the model outputs an element $i \in [n]$ with probability $p_i = \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax)_i$. In the binary hypothesis testing problem, we are given access to a softmax model which is either $\mathtt{SoftMax}_A$ or $\mathtt{SoftMax}_B$. We have query access to the model, that is, we can feed the model an input $x \in \mathbb{R}^d$, and it will produce an output. The goal is to determine whether the model is $\mathtt{SoftMax}_A$ or $\mathtt{SoftMax}_B$, using the fewest number of queries possible. We can similarly define the question for leverage score models. A leverage score model is parameterized by a matrix $A \in \mathbb{R}^{n \times d}$, and denoted $\mathtt{Leverage}_A$. Given input $s \in (\mathbb{R} \backslash \{0\})^n$, the model returns an element $i \in [n]$ with probability $p_i = (A_s(A_s^\top A_s)^{-1} A_s^\top)_{i,i}/d$, where $A_s = S^{-1}A$, and $S = \mathrm{Diag}(s)$ is the diagonal matrix with diagonal s. We define the binary hypothesis testing problem for leverage score models similarly to the softmax case.

1.1. Main Result.

We state informal versions of our main results.

Theorem 1.1 (Informal statement of Theorem 3.2 and Theorem 3.5)**.** *Consider the binary hypothesis testing problem with two softmax models $\mathtt{SoftMax}_A$ and $\mathtt{SoftMax}_B$. We have 1). if $\|B - A\|_{2 \to \infty} \leq \epsilon$, then any successful algorithm uses $\Omega(\epsilon^{-2})$ queries (Lower bound), and 2). if $B = A + \epsilon M$ for some small ϵ then the hypothesis testing problem can be solved in $O(\epsilon^{-2}\nu)$ queries, where ν depends on A and M (Upper bound).*

Theorem 1.2 (Informal statement of Theorem 4.2 and Theorem 4.3)**.** *Consider the binary hypothesis testing problem with two leverage score models $\mathtt{Leverage}_A$ and $\mathtt{Leverage}_B$. We have 1). if $\sum_{i \in [n]} \|B_{i,*}^\top B_{i,*} - A_{i,*}^\top A_{i,*}\|_{\mathrm{op}} \leq \epsilon$, then any successful algorithm uses $\Omega(\epsilon^{-1})$ queries (Lower bound), and 2). if $B = A + \epsilon M$ for some small ϵ then the hypothesis testing problem can be solved in $O(\epsilon^{-2}\nu)$ queries, where ν depends on A and M (Upper bound).*

1.2. Related Work

Theoretical LLMs Several investigations (Cai et al., 2021; Liu et al., 2024c; Reif et al., 2019; Hewitt & Manning, 2019) have concentrated on theoretical analyses concerning LLMs. The algorithm presented by (Cai et al., 2021), named ZO-BCD, introduces a novel approach characterized by advantageous overall query complexity and reduced computational complexity in each iteration. The work by (Liu et al., 2024c) introduces Sophia, a straightforward yet scalable second-order optimizer. Sophia demonstrates adaptability to curvature variations across different parameter regions, a feature particularly advantageous for language modeling tasks with strong heterogeneity. Importantly, the runtime bounds of Sophia are independent of the condition number of the loss function.

Studies by (Wang et al., 2022; Li & Liang, 2021; Dai et al., 2022; Burns et al., 2023; Hase et al., 2023; Xie et al., 2022) investigate the knowledge and skills of LLMs. In the realm of optimization for LLMs, (Kaplan et al., 2020; Rafailov et al., 2023; Liu et al., 2024c; Cao et al., 2024) have delved into this domain. Demonstrating the effectiveness of pretrained models in localizing knowledge within their feed-forward layers, both (Hase et al., 2023) and (Meng et al., 2022) contribute valuable insights to the field. The exploration of distinct ”skill” neurons and their significance in soft prompt-tuning for language models is a central theme in the analysis conducted by (Wang et al., 2022), building upon the groundwork laid out in a prior discussion by (Li & Liang, 2021). The activation of skill neurons and their correlation with the expression of relevant facts is a focal point in the research presented by (Dai et al., 2022), particularly in the context of BERT. In contrast, the work of (Burns et al., 2023) takes an entirely unsupervised approach, leveraging the internal activations of a language model to extract latent knowledge. Next, the investigation by (Li et al., 2023) sheds light on the sparsity observed in feedforward activations of large trained transformers, uncovering noteworthy patterns in their behavior. In addition to the above, (Cai et al., 2021; Malladi et al., 2023; Deng et al., 2024; Zelikman et al., 2023) explore zeroth order algorithms for LLMs, and (Hu et al., 2022; 2024c; Cao, 2024) explore parameter-efficient fine-tuning of LLMs.

Several recent works have revealed the inherent limitations of LLMs from a theoretical perspective. These limitations are grounded in empirical benchmarks indicating that LLMs may face difficulties in simple tasks (Zhou et al., 2023; Guo et al., 2025a; Cao et al., 2025b; Guo et al., 2025b). An important line of research (Alman & Song, 2023; 2024a;b; 2025a;b) has theoretically shown that the forward and backward passes of Transformers cannot be approximated with low error in truly subquadratic time, unless the attention weights are sufficiently small in each entry. Beyond the significant success of circuit complexity in showing that many neural architectures (Ke et al., 2025; Chen et al., 2025b; Li et al., 2025a) may belong to a weak class of logical circuits, such as TC^0, and may fail to solve harder problems, these results have effectively explained LLMs' limitations (Wei et al., 2022; Li et al., 2024c; Chen et al., 2024). For example, they show that Transformers cannot solve arithmetic formula evaluation and that CoT may enhance LLMs' circuit complexity bounds. Recent works have also gone beyond numerical approximation and model expressiveness, finding that under gradient descent training, certain types of simple Boolean function problems may be difficult for LLMs to learn, such as the parity problem (Kim & Suzuki, 2025), the majority problem (Chen et al., 2025a), and simpler and/or problems (Hu et al., 2025).

Leverage Scores Given $A \in \mathbb{R}^{n \times d}$ and $i \in [n]$, a_i represents the i-th row of matrix A. We use $\sigma_i(A) = a_i^\top (A^\top A)^\dagger a_i$ to denote the leverage score for the i-th row of matrix A. The concept of leverage score finds extensive applications in the domains of machine learning and linear algebra. In numerical linear algebra and graph theory, leverage scores serve as fundamental tools. In the context of matrices, both the tensor CURT decomposition (Song et al., 2019b) and the matrix CUR decomposition (Boutsidis & Woodruff, 2014; Song et al., 2017; 2019b) heavily rely on leverage scores. In optimization, areas such as linear programming (Lee & Sidford, 2014; Brand et al., 2020b), the approximation of the John Ellipsoid (Cohen et al., 2019a), cutting-plane methods (Vaidya, 1989; Lee et al., 2015; Jiang et al., 2020b), and semi-definite programming (Jiang et al., 2020a) incorporate leverage scores. Within graph theory applications, leverage scores play a crucial role in max-flow problems (Daitch & Spielman, 2008; Madry, 2013; 2016; Liu & Sidford, 2020), maximum matching (Brand et al., 2020a; Liu et al., 2020b), graph sparsification (Spielman & Srivastava, 2008a), and the generation of random spanning trees (Schild, 2018). Several studies, such as (Spielman & Srivastava, 2008b; Drineas et al., 2012; Clarkson & Woodruff, 2013), focus on the approximation of leverage scores. Simultaneously, Lewis weights, serving as a generalization of leverage scores, are explored in depth by (Bourgain et al., 1989; Cohen & Peng, 2015). Notably, leverage scores may also motivate efficient computational methods in machine learning, including regression problems (Price et al., 2017; Song et al., 2019a; Gilyén et al., 2022; Song et al., 2023; Song, 2019), deep neural networks (Lee et al., 2020; Zandieh et al., 2021), graph neural networks (Shin et al., 2023; Zhang, 2024), recommender systems (Huang et al., 2023; Zhang et al., 2024), active learning (Shimizu et al., 2024; Gajjar et al., 2024), discrepancy minimization (Deng et al., 2025) and Fourier transform (Li et al., 2025c).

Hypothesis Testing Hypothesis testing is a central problem in statistics, and has many applications in machine learning in recent years (Liu et al., 2020a; Gao et al., 2021; Liu et al., 2021a; Xu et al., 2022; Song et al., 2025). In hypothesis testing, two (or more) hypotheses about the truth are given and an algorithm needs to distinguish which hypothesis is true. The most classic testing problem is the binary hypothesis testing. In this problem, two distributions P_0 and P_1 are given, and there is an unknown distribution P which is either P_0 or P_1. The goal is to distinguish whether $P = P_0$ or $P = P_1$ by drawing samples from P. This problem is well-studied, with (Neyman & Pearson, 1933) giving tight characterization of the possible error regions in terms of the likelihood ratio. It is known that the asymptotic sample complexity of binary hypothesis testing for distributions is given by $\Theta(H^{-2}(P_0, P_1))$, where H denotes the Hellinger distance, see e.g., (Polyanskiy & Wu, 2023+). There are other important kinds of hypothesis testing problems. In the goodness-of-fit testing problem, a distribution Q is given, and there is an unknown distribution P which is known to be either equal to Q or far away from Q. The goal is to distinguish which is the true by drawing samples from P. In the two-sample testing problem, two unknown distributions P and Q are given, and it is known that either $P = Q$ or P and Q are far away from each other. The goal is to distinguish which is true by drawing samples from P

and Q. For these problems there are no simple general characterization as in the binary hypothesis testing. However, for reasonable classes of distributions such as Gaussian distributions or distributions on discrete spaces, a lot of nice results are known (Ingster, 1987; 1982; Goldreich & Ron, 2011; Valiant & Valiant, 2017; Chan et al., 2014; Arias-Castro et al., 2018; Li & Yuan, 2019). We are not aware of any previous work that studies hypothesis testing problems for the class of softmax models or leverage score models.

Roadmap. In Section 2, we introduce notation and concepts related to information theory and hypothesis testing. Our results are presented in Section 3 and Section 4: Section 3 establishes upper and lower bounds on the sample complexity for distinguishing two different softmax models, and Section 4 delves into the case of leverage scores. We present several additional remarks for our main results in Section 5. We conclude and make further discussions in Section 6.

2. Preliminaries

In Section 2.1, we define several basic notations. In Section 2.2, we provide definitions related to information theory. In Section 2.3, we provide backgrounds about hypothesis testing. In Section 2.4, we provide definition of softmax model. In Section 2.5, we provide definition of leverage score model.

2.1. Notation

Given $x \in \mathbb{R}^n$, we use $\|x\|_p$ to denote ℓ_p norm of x, where $\|x\|_0 = \sum_{i=1}^n \mathbb{1}(x_i \neq 0)$, $\|x\|_1 := \sum_{i=1}^n |x_i|$ (ℓ_1 norm), $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$ (ℓ_2 norm), and $\|x\|_\infty := \max_{i \in [n]} |x_i|$ (ℓ_∞ norm). For a square matrix, $\mathrm{tr}[A]$ is used to represent the trace of A. Given $1 \leq p \leq \infty$ and $1 \leq q \leq \infty$, $\|A\|_{p \to q}$ represents the p-to-q operator norm $\|A\|_{p \to q} = \sup_{x : \|x\|_p \leq 1} \|Ax\|_q$. In particular, $\|A\|_{2 \to \infty} = \max_{i \in [n]} \|A_{i,*}\|_2$. For $x \in \mathbb{R}^n$, let $\mathrm{Diag}(x) \in \mathbb{R}^{n \times n}$ denote the diagonal matrix with diagonal x. For square matrix $A \in \mathbb{R}^{n \times n}$, let $\mathrm{diag}(A) \in \mathbb{R}^n$ denote the diagonal of A. For a non-negative integer n, let $[n]$ denote the set $\{1, \ldots, n\}$. For a sequence $X_1, \ldots, X_m$ of random variables, we use X^m to denote the whole sequence $(X_1, \ldots, X_m)$.

2.2. Information Theory

Definition 2.1 (TV distance)**.** For two distributions P, Q on the same measurable space, their total variation (TV) distance is

$$\mathrm{TV}(P, Q) = \frac{1}{2} \int |P(\mathrm{d}x) - Q(\mathrm{d}x)| \,.$$

In particular, if P and Q are on the discrete space $[n]$ and $P = (p_1, \ldots, p_n)$, $Q = (q_1, \ldots, q_n)$, then

$$\mathrm{TV}(P, Q)) = \frac{1}{2} \sum_{i=1}^n |p_i - q_i|.$$

Definition 2.2 (Hellinger distance)**.** For two distributions P, Q on the same measurable space, their squared Hellinger distance is

$$H^2(P, Q) = \frac{1}{2} \int (\sqrt{P(\mathrm{d}x)} - \sqrt{Q(\mathrm{d}x)})^2.$$

In particular, if P and Q are on the discrete space $[n]$ and $P = (p_1, \ldots, p_n)$, $Q = (q_1, \ldots, q_n)$, then

$$\begin{aligned} H^2(P, Q) &= \frac{1}{2} \sum_{i=1}^n (\sqrt{p_i} - \sqrt{q_i})^2 \\ &= 1 - \sum_{i=1}^n \sqrt{p_i q_i}. \end{aligned}$$

The Hellinger distance $H(P, Q)$ is the square root of the squared Hellinger distance $H^2(P, Q)$.

We recall the following relationship between the Hellinger distance and the TV distance. For any distributions P, Q on the same space, we have

$$H^2(P, Q) \leq \mathrm{TV}(P, Q) \leq \sqrt{2} H(P, Q).$$

Definition 2.3 (Expectation and variance)**.** Let P be a distribution on a measurable space $\mathcal{X}$ and f be a continuous function on $\mathcal{X}$. Then $\mathbb{E}_P[f]$ is the expectation of f under P and $\mathrm{Var}_P(f)$ is the variance of f under P. In particular, if $\mathcal{X} = [n]$, $P = (p_1, \ldots, p_n) \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$, then

$$\underset{P}{\mathbb{E}}[x] = \sum_{i=1}^n p_i x_i \quad \text{and} \quad \mathrm{Var}_P(x) = \sum_{i=1}^n p_i (x - \underset{P}{\mathbb{E}}[x])^2.$$

2.3. Hypothesis Testing

We review the classic hypothesis testing problem for distributions.

Definition 2.4 (Binary hypothesis testing for distributions)**.** Let P_0, P_1 be two distributions on the same space. We have sample access to a distribution P, which is known to be either P_0 or P_1. The goal is to determine whether $P = P_0$ or $P = P_1$, using as few samples as possible. We say an algorithm successfully distinguishes P_0 and P_1 is at least $2/3$ under both hypotheses.

In the above definition, the constant $2/3$ can be replaced by any constant $> 1/2$, and the asymptotic sample complexity of the binary hypothesis testing problem does not change. The reason is that if we have an algorithm that achieves success probability $\delta > \frac{1}{2}$, then we can run it independently a constant number of times and take the majority of the outputs. Thus, we can boost the success probability to an arbitrarily high constant. A classic result in information theory states that the sample complexity of the binary hypothesis testing problem is determined by the Hellinger distance.

Lemma 2.5 (e.g., (Polyanskiy & Wu, 2023+)). *The sample complexity of the binary hypothesis testing problem for distributions is $\Theta(H^{-2}(P_0, P_1))$. That is, there is an algorithm that solves the problem using $O(H^{-2}(P_0, P_1))$ queries, and any algorithm that solves the problem uses $\Omega(H^{-2}(P_0, P_1))$ queries.*

2.4. Softmax Model

Definition 2.6 (Softmax model)**.** The *softmax model* $\mathtt{SoftMax}_A$ associated with $A \in \mathbb{R}^{n\times d}$ is a model such that on input $x \in \mathbb{R}^d$, it outputs a sample $y \in [n]$ from the distribution $\mathtt{SoftMax}_A(x)$, defined as follows: the probability mass of $i \in [n]$ is equal to $\langle \exp(Ax), \mathbf{1}_n\rangle^{-1} \exp(Ax)_i$.

Note that $\sum_{i=1}^n \langle \exp(Ax), \mathbf{1}_n\rangle^{-1} \exp(Ax)_i = 1$, so the above definition gives a valid distribution.

Definition 2.7 (Binary hypothesis testing for softmax models)**.** Let $A, B \in \mathbb{R}^{n\times d}$ be two matrices. Let $P_0 = \mathtt{SoftMax}_A, P_1 = \mathtt{SoftMax}_B$ be two softmax models. Let P be the softmax model which is either P_0 or P_1. In each query, we can feed $x \in \mathbb{R}^d$ into P, and retrieve a sample $y \in [n]$ from $P(x)$. The goal is to determine if the model P is P_0 or P_1 in as few samples as possible. We say an algorithm successfully distinguishes P_0 and P_1, if the correctness probability is at least $2/3$ under both hypotheses.

The above definition is valid. However, if we make no restrictions on the input x, then there would be undesirable consequences. For example, suppose $n = 2$, $d = 1$, $A = \begin{bmatrix}\epsilon \\ 0\end{bmatrix}$, $B = \begin{bmatrix}0 \\ \epsilon\end{bmatrix}$ for some very small $\epsilon > 0$. Because A and B are close to each other, we should expect it to be difficult to distinguish $\mathtt{SoftMax}_A$ and $\mathtt{SoftMax}_B$. However, if we allow any $x \in \mathbb{R}^d$ as input, then we could take x to be a very large real number. Then $\mathtt{SoftMax}_A(x)$ has almost all mass on $1 \in [n]$, while $\mathtt{SoftMax}_B(x)$ has almost all mass on $2 \in [n]$, and we can distinguish the two models using only one query. To avoid this peculiarity, we assume that there is an energy constraint on x.

Definition 2.8 (Energy constraint for softmax model)**.** We assume that there is an *energy constraint*, that is, input $x \in \mathbb{R}^n$ should satisfy $\|x\|_2 \le E$, for some given constant E.

The energy constraint is a reasonable assumption in the context of LLMs and more generally neural networks, because of the widely used batch normalization technique (Ioffe & Szegedy, 2015).

2.5. Leverage Score Model

Definition 2.9 (Leverage score model)**.** The *leverage score model* $\mathtt{Leverage}_A$ associated with $A \in \mathbb{R}^{n\times d}$ is a model such that on input $s \in (\mathbb{R}\backslash\{0\})^n$, it outputs a sample $y \in [n]$ from the distribution $\mathtt{Leverage}_A(s)$, defined as follows: the probability mass of $i \in [n]$ is equal to

$$\|(A_s^\top A_s)^{-1/2}(A_s)_{*,i}\|_2^2/d = (A_s(A_s^\top A_s)^{-1}A_s^\top)_{i,i}/d,$$

where $A_s = S^{-1}A$, and $S = \mathrm{Diag}(s)$.

Definition 2.10 (Binary hypothesis testing for leverage score model)**.** Let $A, B \in \mathbb{R}^{n\times d}$ be two matrices. Let $P_0 = \mathtt{Leverage}_A$, $P_1 = \mathtt{Leverage}_B$ be two leverage score models. Let P be the leverage score model which is either P_0 or P_1. In each query, we can feed $s \in (\mathbb{R}\backslash\{0\})^n$ into P, and retrieve a sample $y \in [n]$ from $P(s)$. The goal is to determine whether the model P is P_0 or P_1 in as few samples as possible. We say an algorithm successfully distinguishes P_0 and P_1, if the correctness probability is at least $2/3$ under both hypotheses.

Similar to the softmax model case, if we do not put any restrictions on s, then there will be certain weird behavior. For example, if we take $n = 2$, $d = 1$, $A = \begin{bmatrix}1 \\ 0\end{bmatrix}$ and $B = \begin{bmatrix}1 \\ \epsilon\end{bmatrix}$ for some small $\epsilon > 0$. Because A and B are close to each other, we should expect it to be difficult to distinguish $\mathtt{Leverage}_A$ and $\mathtt{Leverage}_B$. However, if we allow any $s \in (\mathbb{R}\backslash\{0\})^n$ as input, then we can take $s = \begin{bmatrix}1 & \delta\end{bmatrix}$ for some very small $\delta > 0$. In this way, we can verify that $\mathtt{Leverage}_A(s)$ has all mass on $1 \in [n]$, while $\mathtt{Leverage}_B(s)$ has almost all mass on $2 \in [n]$. So we can distinguish the two models using only one query. To avoid such cases we put additional constraints on s.

Definition 2.11 (Constraint for leverage score model)**.** We assume that input $s \in (\mathbb{R}\backslash\{0\})^d$ should satisfy the constraint such that $c \le s_i^2 \le C$ for some given constants $0 < c < C$.

3. Softmax Model

In Section 3.1, we state our general result of softmax model. In Section 3.2, we provide show how to prove the lower bound. In Section 3.3, we explain how to prove the upper bound.

3.1. General Result

We first prove a general result that relates the binary hypothesis testing problem with Hellinger distance, and the proof is deferred to Appendix A.1.

Theorem 3.1. *Let $A, B \in \mathbb{R}^{n\times d}$ be two matrices. Consider the binary hypothesis testing problem of distinguish-*

ing $\texttt{SoftMax}_A$ *and* $\texttt{SoftMax}_B$ *using energy-constrained queries (Definition 2.8). Define*

$$\delta = \sup_{x:\|x\|_2 \le E} H(\texttt{SoftMax}_A(x), \texttt{SoftMax}_B(x)).$$

Then the sample complexity of the binary hypothesis testing problem is $\Theta(\delta^{-2})$. *That is, there is an algorithm that successfully solves the problem using* $O(\delta^{-2})$ *energy-constrained queries, and any algorithm that successfully solves the problem uses* $\Omega(\delta^{-2})$ *energy-constrained queries.*

3.2. Lower Bound

Now, we prove the following lower bound for binary hypothesis testing for softmax models.

Theorem 3.2 (Lower bound)**.** *If two softmax models (Definition 2.6) with parameters* $A \in \mathbb{R}^{n\times d}$ *and* $B \in \mathbb{R}^{n\times d}$ *satisfy*

$$\|A - B\|_{2\to\infty} \le \epsilon,$$

which is

$$\max_{j\in[n]} \|A_{j,*} - B_{j,*}\|_2 \le \epsilon,$$

then any algorithm with energy constraint E *that distinguishes the two models with success probability* $\ge \frac{2}{3}$ *uses at least* $\Omega(\epsilon^{-2}E^{-2})$ *samples.*

Before giving the proof of Theorem 3.2, we state a lemma, and the proof is deferred to Appendix A.2.

Lemma 3.3. *Let* $a, b \in \mathbb{R}^n$ *be such that* $\|a - b\|_\infty \le \epsilon$. *Let* P *be the distribution on* $[n]$ *with* $p_i = \exp(a_i)/\langle \exp(a), \mathbf{1}_n\rangle$. *Let* Q *be the distribution on* $[n]$ *with* $q_i = \exp(b_i)/\langle \exp(b), \mathbf{1}_n\rangle$. *Then*

$$H^2(P,Q) = O(\epsilon^2), \quad \mathrm{TV}(P,Q) = O(\epsilon).$$

Next, we prove an application of Lemma 3.3.

Corollary 3.4. *If matrices* $A \in \mathbb{R}^{n\times d}, B \in \mathbb{R}^{n\times d}$ *satisfy* $\max_{j\in[n]} \|A_{j,*} - B_{j,*}\|_2 \le \epsilon$, *then for any* $x \in \mathbb{R}^d$, *the distributions* $P = \texttt{SoftMax}_A(x)$ *and* $Q = \texttt{SoftMax}_B(x)$ *satisfy*

$$H^2(P,Q) = O(\epsilon^2\|x\|_2^2), \quad \mathrm{TV}(P,Q) = O(\epsilon\|x\|_2).$$

Proof. For any $x \in \mathbb{R}^n$, we have

$$\begin{aligned}\|Ax - Bx\|_\infty &= \max_{j\in[n]} |A_{j,*}x - B_{j,*}x| \\ &\le \max_{j\in[n]} \|A_{j,*} - B_{j,*}\|_2\|x\|_2 \\ &\le \epsilon\|x\|_2.\end{aligned}$$

The result then follows from Lemma 3.3. □

Now, we're ready to finish the proof of Theorem 3.2.

Proof of Theorem 3.2. By Corollary 3.4, we have $H^2(\texttt{SoftMax}_A(x), \texttt{SoftMax}_B(x)) = O(\epsilon^2E^2)$ for any $\|x\|_2 \le E$. Therefore δ in the statement of Theorem 3.1 satisfies $\delta^2 = O(\epsilon^2E^2)$. Applying Theorem 3.1 we finish the proof. □

3.3. Upper Bound

In the previous section, we established an $\Omega(\epsilon^{-2})$ lower bound for solving the hypothesis testing problem for the softmax model. The upper bound is more subtle. Let us discuss a few difficulties in establishing the upper bound. Let $A, B \in \mathbb{R}^{n\times d}$ be parameters of the softmax models, $x \in \mathbb{R}^d$ be the input vector, $P = \texttt{SoftMax}_A(x) = (p_1, \ldots, p_n)$, $Q = \texttt{SoftMax}_B(x) = (q_1, \ldots, q_n)$. First, two different matrices A and B could give rise to the same softmax model. If $B = A + \mathbf{1}_n^\top w$ for some $w \in \mathbb{R}^d$, then for any $x \in \mathbb{R}^d$, we have

$$\begin{aligned}q_i &= \frac{\exp(Bx)_i}{\langle \exp(Bx), \mathbf{1}_n\rangle} \\ &= \frac{\exp(Ax)_i \exp(w^\top x)}{\langle \exp(Ax)\exp(w^\top x), \mathbf{1}_n\rangle} \\ &= \frac{\exp(Ax)_i}{\langle \exp(Ax), \mathbf{1}_n\rangle} \\ &= p_i\end{aligned}$$

for all $i \in [d]$. Therefore in this case $\texttt{SoftMax}_A(x) = \texttt{SoftMax}_B(x)$ for all $x \in \mathbb{R}^d$ and it is impossible to distinguish the two models. This issue may be resolved by adding additional assumptions such as $\mathbf{1}_n^\top A = \mathbf{1}_n^\top B$. A more important issue is that A and B may differ only in rows with very small probability weight under any input x. For example, suppose A is the zero matrix, and B differ with A only in the first row. For any $x \in \mathbb{R}^d$, the distribution $\texttt{SoftMax}_A(x)$ is the uniform distribution on $[d]$. If $\|B_{1,*} - A_{1,*}\|_2 = \epsilon$, then for any x with $\|x\|_2 \le E$, we have

$$\begin{aligned}\exp(-\epsilon E) &\le \frac{\exp(Bx)_1}{\exp(Ax)_1} \\ &\le \exp(\epsilon E).\end{aligned}$$

A simple calculation shows that in this case,

$$H^2(P,Q) = O(\epsilon^2E^2/n).$$

So the sample complexity of any hypothesis testing algorithm is at least $\Omega(n/(\epsilon^2E^2))$, which grows with n. This shows that the sample complexity may depend on n. Nevertheless, using Theorem 3.1, we show a local upper bound, which says that for fixed A and fixed direction M, there is an algorithm that distinguishes $\texttt{SoftMax}_A$ and $\texttt{SoftMax}_{A+\epsilon M}$ using $O(\epsilon^{-2})$ queries, for small enough $\epsilon > 0$.

Theorem 3.5. *Fix $A, M \in \mathbb{R}^{n \times d}$ where $\|M\|_{2\to\infty} = O(1)$. For $\epsilon > 0$, define $B_\epsilon = A + \epsilon M$. We consider the binary hypothesis testing problem with $\mathtt{SoftMax}_A$ and $\mathtt{SoftMax}_{B_\epsilon}$, for small ϵ. Let $\nu = \sup_{x:\|x\|_2 \le E} \mathrm{Var}_{\mathtt{SoftMax}_A(x)}(Mx)$. Then for $\epsilon > 0$ small enough, there is an algorithm that uses $O(\epsilon^{-2}\nu^{-1})$ energy-constrained queries and distinguishes between $\mathtt{SoftMax}_A$ and $\mathtt{SoftMax}_{B_\epsilon}$.*

Proof of Theorem 3.5 is deferred to Appendix A.3. From Theorem 3.5 we see that it is an interesting problem to bound $\nu = \sup_{x:\|x\|_2 \le E} \mathrm{Var}_{\mathtt{SoftMax}_A(x)}(Mx)$ for fixed $A, M \in \mathbb{R}^{n \times d}$. For different A and M the value of ν can be quite different. For example, if A is the all zero matrix and M is zero except for row 1 (and $\|M\|_{2\to\infty} = O(1)$), then $\nu = O(E^2/n)$ for any $\|x\|_2 \le E$. On the other hand, if A is the zero matrix, and the first column M are i.i.d. Gaussian $\mathcal{N}(0, \Theta(1))$, then with high probability, $\nu = \Omega(E^2)$ for $x = (E, 0, \ldots, 0)$. We remark that Theorem 3.5 is in fact tight. We have a matching lower bound.

Theorem 3.6. *Under the same setting as Theorem 3.5 and let ν be defined as Theorem 3.5, for sufficient small $\epsilon > 0$, any algorithm that distinguishes between $\mathtt{SoftMax}_A$ and $\mathtt{SoftMax}_{B_\epsilon}$ must use $\Omega(\epsilon^{-2}\nu^{-1})$ energy-constrained queries.*

Proof. It follows from combining the proof of Theorem 3.5 and Theorem 3.1. □

4. Leverage Score Model

In Section 4.1, we state our general result of softmax model. In Section 4.2, we provide show how to prove the lower bound. In Section 4.3, we explain how to prove the upper bound.

4.1. General Result

We first prove a general result which is the leverage score version of Theorem 3.1.

Theorem 4.1. *Let $A, B \in \mathbb{R}^{n \times d}$ be two matrices. Consider the binary hypothesis testing problem of distinguishing $\mathtt{Leverage}_A$ and $\mathtt{Leverage}_B$ using constrained queries (Definition 2.11). Define*

$$\delta = \sup_{s: c \le s_i^2 \le C \forall i} H(\mathtt{Leverage}_A(s), \mathtt{Leverage}_B(s)).$$

Then the sample complexity of the binary hypothesis testing problem is $\Theta(\delta^{-2})$. That is, there is an algorithm that successfully solves the problem using $O(\delta^{-2})$ energy-constrained queries, and any algorithm that successfully solves the problem uses $\Omega(\delta^{-2})$ energy-constrained queries.

Proof. The proof is similar to Theorem 3.1 and omitted. □

4.2. Lower Bound

The goal of this section is to prove the following lower bound for binary hypothesis testing for leverage score models.

Theorem 4.2. *Consider two leverage score model $\mathtt{Leverage}_A$ and $\mathtt{Leverage}_B$. Assume that there exists $\delta > 0$ such that $A^\top A \succeq \delta I$. If*

$$\sum_{i \in [n]} \|B_{i,*}^\top B_{i,*} - A_{i,*}^\top A_{i,*}\|_{\mathrm{op}} \le \epsilon$$

(where $\|\cdot\|_{\mathrm{op}}$ denotes the 2-to-2 operator norm), then any algorithm that solves the binary hypothesis testing problem takes at least $\Omega(c\delta/(C\epsilon))$ constrained queries.

Proof. Let

$$P = \mathtt{Leverage}_A(s) = (p_1, \ldots, p_n)$$

and

$$Q = \mathtt{Leverage}_B(s) = (q_1, \ldots, q_n).$$

By Theorem 4.1, it suffices to prove that $H^2(P, Q) = O(\epsilon C/(c\delta))$. We first consider the case where A and B differ in exactly one row i. Fix $s \in \mathbb{R}^d$ with $c \le s_j \le C$ for all $j \in [n]$. Let $A_s = S^{-1}A$ and $B_s = S^{-1}B$, where $S = \mathrm{Diag}(s)$.

Because $A^\top A \succeq \delta I$, we have

$$A_s^\top A_s \succeq (\delta/C) \cdot I.$$

Because $\|B_{i,*}^\top B_{i,*} - A_{i,*}^\top A_{i,*}\|_{\mathrm{op}} \le \epsilon$, we have

$$-\epsilon_i C/\delta A_s^\top A_s \preceq B_{i,*}^\top B_{i,*} - A_{i,*}^\top A_{i,*} \preceq \epsilon_i C/\delta A_s^\top A_s.$$

Recall that A and B differ in exactly one row i. Therefore

$$(1 - \frac{\epsilon C}{c\delta}) A_s^\top A_s \preceq B_s^\top B_s \preceq (1 + \frac{\epsilon C}{c\delta}) A_s^\top A_s. \tag{1}$$

For $j \neq i$, we have

$$\begin{aligned} q_j &= s_j^{-2} B_{j,*} (B_s^\top B_s)^{-1} (B^\top)_{*,j} / d \\ &= \mathrm{tr}[s_j^{-2} (B^\top)_{*,j} B_{j,*} (B_s^\top B_s)^{-1}] / d \\ &= (1 \pm O(\epsilon C/(c\delta))) \, \mathrm{tr}[s_j^{-2} A_{j,*}^\top A_{j,*} (A_s^\top A_s)^{-1}] / d \\ &= (1 \pm O(\epsilon C/(c\delta))) p_j, \end{aligned} \tag{2}$$

where the first step is by definition of the leverage score model, the second step is by property of trace, the third step is Eq. (1), the fourth step is by definition of the leverage score model.

Upper bound for TV**.** For the TV distance, we have

$$\mathrm{TV}(P, Q) = \frac{1}{2} \sum_{j=1}^{n} |p_j - q_j|$$

$$\begin{aligned}
&\le \sum_{j\neq i} |p_j - q_j| \\
&\le \sum_{j\neq i} O(\epsilon C/(c\delta)) p_i \\
&\le O(\epsilon C/(c\delta)).
\end{aligned}$$

where the first step is by definition of TV distance, and the third step is by Eq. (2). Therefore $\mathrm{TV}(P, Q) \le O(\epsilon C/(c\delta))$.

Upper bound for $H^2(P, Q)$. Using

$$H^2(P, Q) \le \mathrm{TV}(P, Q),$$

we also get

$$H^2(P, Q) \le O(\epsilon C/(c\delta)).$$

Now we have established the result when A and B differ in exactly one row. Let us now consider general case. If $\epsilon \ge 0.1\delta$, then $c\delta/(C\epsilon) = O(1)$ and there is nothing to prove. In the following, assume that $\epsilon \le 0.1\delta$. For $0 \le k \le n$, define $B^k \in \mathbb{R}^{n\times d}$ be the matrix with $B^k_{i,*} = B_{i,*}$ for $i \le k$ and $B^k_{i,*} = A_{i,*}$ for $i \ge k$. Then $B^0 = A$, $B^n = B$, and B^k and B^{k+1} differ exactly in one row. Let $\epsilon_i = \|B_{i,*}^\top B_{i,*} - A_{i,*}^\top A_{i,*}\|_{\mathrm{op}}$.

Then by the above discussion, we have

$$\mathrm{TV}(\texttt{Leverage}_{B^k}(s), \texttt{Leverage}_{B^{k+1}}(s)) = O(\epsilon_k C/(c\delta))$$

for all $0 \le k \le n-1$. By metric property of TV, we have

$$\begin{aligned}
&\mathrm{TV}(P, Q) \\
\le &\sum_{0\le k\le n-1} \mathrm{TV}(\texttt{Leverage}_{B^k}(s), \texttt{Leverage}_{B^{k+1}}(s)) \\
= &\sum_{0\le k\le n-1} O(\epsilon_i C/(c\delta)) \\
= &\, O(\epsilon C/(c\delta)).
\end{aligned}$$

Using $H^2(P, Q) \le \mathrm{TV}(P, Q)$ we also get $H^2(P, Q) = O(\epsilon C/(c\delta))$. This finishes the proof. □

In Theorem 4.2, the bound has linear dependence in ϵ^{-1}. An interesting question is the improve the bound to quadratic dependence ϵ^{-2}.

4.3. Upper Bound

Let $A, B \in \mathbb{R}^{n\times d}$ be parameters of the leverage score models, $s \in \mathbb{R}^n$ be the input vector, $P = \texttt{Leverage}_A(s) = (p_1, \dots, p_n)$, $Q = \texttt{Leverage}_B(s) = (q_1, \dots, q_n)$. For the upper bounds of the leverage score model, we run into similar difficulties as for the softmax model. Firstly, different matrices A and B could give rise to the same leverage score model. If $B = AR$ for some invertible matrix $R \in \mathbb{R}^{d\times d}$, then we have

$$\begin{aligned}
q_i &= (B_s(B_s^\top B_s)^{-1} B_s^\top)_{i,i}/d \\
&= (A_s R(R^\top A_s^\top A_s R)^{-1} R^\top A_s^\top)_{i,i}/d \\
&= (A_s(A_s^\top A_s)^{-1} A_s^\top)_{i,i}/d \\
&= p_i.
\end{aligned}$$

Then $\texttt{Leverage}_A(s) = \texttt{Leverage}_B(s)$ for all $s \in (\mathbb{R}\backslash\{0\})^n$ and it is impossible to distinguish the two models. Furthermore, there exist scenarios where A and B differ only in rows with very small probability weight under any input s.

We now give an example where

$$\|A_{1,*}^\top A_{1,*} - B_{1,*}^\top B_{1,*}\| = \Omega(1)$$

but

$$\mathrm{TV}(\texttt{Leverage}_A(s), \texttt{Leverage}_B(s)) = O(1/n)$$

for any s satisfying $c \le s_i^2 \le C$ for all $i \in [n]$.

Suppose $A = \begin{bmatrix} I_d & e_1 & \cdots & e_1 \end{bmatrix}^\top$, that is, the first d rows of A is equal to I_d, and all remaining rows are equal to $e_1^\top = (1, 0, \dots, 0)$.

Then for s satisfying $c \le s_i^2 \le C$ for all $i \in [n]$, the distribution $P = \texttt{Leverage}_A(s)$ has probability mass $O(1/n)$ on every element $i \in \{1, d+1, d+2 \dots, n\}$ (hiding constants depending on c and C).

Now suppose B differs with A only in the first entry $(1, 1)$, and

$$B_{1,1} = A_{1,1} + \Theta(1).$$

Then for fixed s, $q_j = p_j$ for $j \in \{2, \dots, d\}$, $q_1 \ge p_1$, and $q_j \le p_j$ for $j \in \{d+1, \dots, n\}$. So

$$\begin{aligned}
H^2(P, Q) &\le \mathrm{TV}(P, Q) \\
&= q_1 - p_1 \\
&= \Theta(1/n).
\end{aligned}$$

This shows that the sample complexity may depend on n. After discussing the difficulties in establishing an upper bound, we now show a local upper bound, which says for fixed A and fixed direction M, there is an algorithm that distinguishes $\texttt{Leverage}_A$ and $\texttt{Leverage}_{A+\epsilon M}$ using $O(\epsilon^{-2})$ queries, for small enough $\epsilon > 0$.

Theorem 4.3. *Fix $A, M \in \mathbb{R}^{n\times d}$ where $\|M\|_{2\to\infty} = O(1)$. For $\epsilon > 0$, define $B_\epsilon = A + \epsilon M$. We consider the binary hypothesis testing problem with $\texttt{Leverage}_A$ and $\texttt{Leverage}_{B_\epsilon}$, for small ϵ. Let $\nu = \sup_s \mathrm{Var}_{\texttt{Leverage}_A(s)}(w_s)$ where*

$$w_s = \frac{\mathrm{diag}((I - A_s(A_s^\top A_s)^{-1}A_s^\top)(M_s(A_s^\top A_s)^{-1}A_s^\top))}{\mathrm{diag}(A_s(A_s^\top A_s)^{-1}A_s^\top)}$$

where the division between vectors is entrywise division. Then for $\epsilon > 0$ small enough, there is an algorithm that uses $O(\epsilon^{-2}\nu^{-1})$ queries and distinguishes between $\mathtt{Leverage}_A$ *and* $\mathtt{Leverage}_{B_\epsilon}$.

Proof of Theorem 4.3 is deferred to Appendix A.4. Similarly to the softmax model case, Theorem 4.3 is also tight.

Theorem 4.4. *Work under the same setting as Theorem 4.3. For $\epsilon > 0$ small enough, any algorithm that distinguishes between* $\mathtt{SoftMax}_A$ *and* $\mathtt{SoftMax}_{B_\epsilon}$ *must use $\Omega(\epsilon^{-2}\nu^{-1})$ energy-constrained queries.*

Proof. The proof is by combining the proof of Theorem 4.3 and Theorem 4.1. We omit the details. □

5. Discussion

Practical Implications. Softmax and leverage score distributions arise naturally in machine learning and numerical linear algebra (Lee & Sidford, 2014; Lee et al., 2015; Cohen et al., 2019b; Song, 2019; Jiang et al., 2020a; Song et al., 2019b; Lee et al., 2020). Softmax distributions have a clear connection to LLMs (Wang et al., 2022; Li & Liang, 2021; Dai et al., 2022; Burns et al., 2023; Hase et al., 2023; Xie et al., 2022), and leverage score distribution serves as a more general and complex case. We study the distinguishability of models (softmax and leverage score based) through the lens of binary hypothesis testing, establishing tight sample complexity bounds. These results directly address the challenge of determining how much information (or how many queries) is needed to tell apart closely related models, a theoretical formulation aligned with understanding model "abilities" via limited parameter access. Moreover, our framework sets a path toward identifying distinguishable components of large models. For instance, showing that certain parameters contribute more significantly to distinguishability (via Hellinger distance bounds) ofers insight into what might constitute an "ability region" within a model, aligning with the introductory motivation.

Technical Novelty. The core novelty of our proofs lies in adapting classical binary hypothesis testing (Pensia et al., 2023; 2024), typically studied in the context of generic distributions, to the structured, parameterized families of distributions induced by softmax and leverage score models. Unlike arbitrary distributions, these models produce distributions that are nonlinear functions of the input and matrix parameters, which poses unique analytical challenges.

For the softmax model (Section 3), one key difficulty is that different parameter matrices and can induce indistinguishable distributions due to invariance under certain transformations (e.g., row shifts). To handle this, we introduce structural constraints and prove that the Hellinger distance (Polyanskiy & Wu, 2023+) between softmax outputs under constrained inputs governs the sample complexity. This leads to a tight upper and lower bound framework via careful analysis of the sensitivity of softmax distributions to perturbations in the parameter matrix.

For the leverage score model (Section 4), the challenge is even greater due to the nonlinear matrix expressions involved, including matrix inversion and normalization, and the fact that the input sss is a vector that rescales rows of the matrix. We overcome this by establishing operator norm bounds and using perturbation theory to relate changes in the parameter matrix to changes in the output distribution. Our proof carefully propagates these changes through the matrix expressions and yields tight dependence on both the model diference and input constraints.

6. Conclusion and Future Directions

Widely applied across various domains, softmax and leverage scores play crucial roles in machine learning and linear algebra. This study delves into the testing problem aimed at distinguishing between different models of softmax and leverage score distributions, each parameterized by distinct matrices. We establish bounds on the number of samples within the defined testing problem. With the rapidly escalating computational costs in current machine learning research, our work holds the potential to offer valuable insights and guidance for distinguishing between the distributions of different models. We discuss a few possible directions for further research. In Theorem 3.5 and Theorem 4.3, we determine the local sample complexity of the binary hypothesis testing problems for softmax models and leverage score models. In particular, the sample complexity is $\Theta(\epsilon^{-2}\nu)$, where ν is a certain function depending on A and M (where $B = A + \epsilon M$). The form of ν is an optimization problem over the space of possible inputs. An interesting question is to provide bounds on the quantity ν, or to provide computation-efficient algorithms for determining the value of ν of finding the optimal input (x for softmax, s for leverage score). This will lead to computation-efficient algorithms for solving the binary hypothesis testing problem in practice.

In this paper, we focused on the binary hypothesis testing problem, where the goal is to distinguish two models with different parameters. There are other hypothesis testing problems that are of interest both in theory and practice. For example, in the goodness-of-fit problem, the goal is to determine whether an unknown model is equal to or far away from a given model. In the two-sample testing problem, the goal is to determine whether two unknown models are the same or far away from each other. These problems have potential practical applications and we leave them as an interesting future direction.

Acknowledgements

The author would like to thank the anonymous reviewer of ICML 2025 for their highly insightful suggestions.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alman, J. and Song, Z. Fast attention requires bounded entries. *Advances in Neural Information Processing Systems*, 36:63117–63135, 2023.

Alman, J. and Song, Z. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *ICLR*, 2024a.

Alman, J. and Song, Z. The fine-grained complexity of gradient computation for training large language models. In *NeurIPS*, 2024b.

Alman, J. and Song, Z. Only large weights (and not skip connections) can prevent the perils of rank collapse. *arXiv preprint arXiv:2505.16284*, 2025a.

Alman, J. and Song, Z. Fast rope attention: Combining the polynomial method and fast fourier transform. In *arXiv preprint arXiv:2505.11892*, 2025b.

Arias-Castro, E., Pelletier, B., and Saligrama, V. Remember the curse of dimensionality: The case of goodness-of-fit testing in arbitrary dimension. *Journal of Nonparametric Statistics*, 30(2):448–471, 2018.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 675–718, 2023.

Bourgain, J., Lindenstrauss, J., and Milman, V. Approximation of zonoids by zonotopes. 1989.

Boutsidis, C. and Woodruff, D. P. Optimal cur matrix decompositions. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 353–362, 2014.

Brand, J. v. d., Lee, Y.-T., Nanongkai, D., Peng, R., Saranurak, T., Sidford, A., Song, Z., and Wang, D. Bipartite matching in nearly-linear time on moderately dense graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 919–930. IEEE, 2020a.

Brand, J. v. d., Lee, Y. T., Sidford, A., and Song, Z. Solving tall dense linear programs in nearly linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 775–788, 2020b.

Brand, J. v. d., Song, Z., and Zhou, T. Algorithm and hardness for dynamic attention maintenance in large language models. In *ICML*, 2024.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

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

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

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. *ICLR*, 2023.

Cai, H., Lou, Y., McKenzie, D., and Yin, W. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *International Conference on Machine Learning*, pp. 1193–1203. PMLR, 2021.

Cao, Y. Sorsa: Singular values and orthonormal regularized singular vectors adaptation of large language models. *arXiv preprint arXiv:2409.00055*, 2024.

Cao, Y., Li, X., and Song, Z. Grams: Gradient descent with adaptive momentum scaling. *arXiv preprint arXiv:2412.17107*, 2024.

Cao, Y., Chen, B., Li, X., Liang, Y., Sha, Z., Shi, Z., Song, Z., and Wan, M. Force matching with relativistic constraints: A physics-inspired approach to stable and efficient generative modeling. *arXiv preprint arXiv:2502.08150*, 2025a.

Cao, Y., Guo, X., Huo, J., Liang, Y., Shi, Z., Song, Z., Zhang, J., and Zhuang, Z. Text-to-image diffusion models cannot count, and prompt refinement cannot help. *arXiv preprint arXiv:2503.06884*, 2025b.

Cao, Y., Li, X., Liang, Y., Sha, Z., Shi, Z., Song, Z., and Zhang, J. Dissecting submission limit in desk-rejections: A mathematical analysis of fairness in ai conference policies. *arXiv preprint arXiv:2502.00690*, 2025c.

Cao, Y., Song, Z., and Yang, C. Video latent flow matching: Optimal polynomial projections for video interpolation and extrapolation. *arXiv preprint arXiv:2502.00500*, 2025d.

Chan, S.-O., Diakonikolas, I., Valiant, P., and Valiant, G. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1193–1203. SIAM, 2014.

Chang, Y.-T., Hu, Z., Li, X., Yang, S., Jiang, J., and Sun, N. Dihan: A novel dynamic hierarchical graph attention network for fake news detection. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 197–206, 2024.

ChatGPT. Optimizing language models for dialogue. *OpenAI Blog*, November 2022. URL https://openai.com/blog/chatgpt/.

Chen, B., Li, X., Liang, Y., Long, J., Shi, Z., and Song, Z. Circuit complexity bounds for rope-based transformer architecture. *arXiv preprint arXiv:2411.07602*, 2024.

Chen, B., Shi, Z., Song, Z., and Zhang, J. Provable failure of language models in learning majority boolean logic via gradient descent. *arXiv preprint arXiv:2504.04702*, 2025a.

Chen, L., Kol, G., Paramonov, D., Saxena, R. R., Song, Z., and Yu, H. Towards multi-pass streaming lower bounds for optimal approximation of max-cut. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 878–924. SIAM, 2023a.

Chen, Y., Wang, R., Jiang, H., Shi, S., and Xu, R. Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pp. 361–374, 2023b.

Chen, Y., Li, X., Liang, Y., Shi, Z., and Song, Z. The computational limits of state-space models and mamba via the lens of circuit complexity. In *Conference on Parsimony and Learning*. PMLR, 2025b.

Chervenak, J., Lieman, H., Blanco-Breindel, M., and Jindal, S. The promise and peril of using a large language model to obtain clinical information: Chatgpt performs strongly as a fertility counseling tool with limitations. *Fertility and Sterility*, 2023.

Chia, Y. K., Hong, P., Bing, L., and Poria, S. Instructeval: Towards holistic evaluation of instruction-tuned large language models. *arXiv preprint arXiv:2306.04757*, 2023.

Choi, M., Pei, J., Kumar, S., Shu, C., and Jurgens, D. Do llms understand social knowledge? evaluating the sociability of large language models with socket benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11370–11403, 2023.

Clarkson, K. L. and Woodruff, D. P. Low-rank approximation and regression in input sparsity time. In *STOC*, 2013.

Cohen, M. B. and Peng, R. Lp row sampling by lewis weights. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 183–192, 2015.

Cohen, M. B., Cousins, B., Lee, Y. T., and Yang, X. A near-optimal algorithm for approximating the john ellipsoid. In *Conference on Learning Theory*, pp. 849–873. PMLR, 2019a.

Cohen, M. B., Lee, Y. T., and Song, Z. Solving linear programs in the current matrix multiplication time. In *STOC*, 2019b.

Corbett-Davies, S., Gaebler, J. D., Nilforoshan, H., Shroff, R., and Goel, S. The measure and mismeasure of fairness. *Journal of Machine Learning Research*, 24(312):1–117, 2023.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Daitch, S. I. and Spielman, D. A. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 451–460, 2008.

Das, B. C., Amini, M. H., and Wu, Y. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

Deng, Y., Li, Z., and Song, Z. Attention scheme inspired softmax regression. *arXiv preprint arXiv:2304.10411*, 2023a.

Deng, Y., Mahadevan, S., and Song, Z. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arxiv preprint: arxiv 2304.03426*, 2023b.

Deng, Y., Li, Z., Mahadevan, S., and Song, Z. Zero-th order algorithm for softmax attention optimization. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 24–33. IEEE, 2024.

Deng, Y., Li, X., Song, Z., and Weinstein, O. Discrepancy minimization in input-sparsity time. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Deroy, A., Ghosh, K., and Ghosh, S. How ready are pre-trained abstractive models and llms for legal case judgement summarization? *arXiv preprint arXiv:2306.01248*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Drineas, P., Magdon-Ismail, M., Mahoney, M. W., and Woodruff, D. P. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Ferrara, E. Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*, 2023.

Fu, Z., Liu, F., Zhang, J., Wang, H., Yang, C., Xu, Q., Qi, J., Fu, X., and Zhou, A. Sagn: semantic adaptive graph network for skeleton-based human action recognition. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pp. 110–117, 2021.

Gajjar, A., Tai, W. M., Xingyu, X., Hegde, C., Musco, C., and Li, Y. Agnostic active learning of single index models with linear sample complexity. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1715–1754. PMLR, 2024.

Gao, R., Liu, F., Zhang, J., Han, B., Liu, T., Niu, G., and Sugiyama, M. Maximum mean discrepancy test is aware of adversarial attacks. In *International Conference on Machine Learning*, pp. 3564–3575. PMLR, 2021.

Gao, Y., Mahadevan, S., and Song, Z. An over-parameterized exponential regression. *arXiv preprint arXiv:2303.16504*, 2023.

Gilyén, A., Song, Z., and Tang, E. An improved quantum-inspired algorithm for linear regression. *Quantum*, 6:754, 2022.

Go, J. H., Sari, A., Jiang, J., Yang, S., and Jha, S. Fake news quick detection on dynamic heterogeneous information networks. *arXiv preprint arXiv:2205.07039*, 2022.

Goldreich, O. and Ron, D. On testing expansion in bounded-degree graphs. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pp. 68–75, 2011.

Gu, Y., Song, Z., Yin, J., and Zhang, L. Low rank matrix completion via robust alternating minimization in nearly linear time. In *ICLR*, 2024.

Gu, Y., Song, Z., and Zhang, L. Faster algorithms for structured linear and kernel support vector machines. In *ICLR*, 2025.

Guo, T., Guo, K., Liang, Z., Guo, Z., Chawla, N. V., Wiest, O., Zhang, X., et al. What indeed can gpt models do in chemistry? a comprehensive benchmark on eight tasks. *arXiv preprint arXiv:2305.18365*, 2023.

Guo, X., Huang, Z., Huo, J., Liang, Y., Shi, Z., Song, Z., and Zhang, J. Can you count to nine? a human evaluation benchmark for counting limits in modern text-to-video models. *arXiv preprint arXiv:2504.04051*, 2025a.

Guo, X., Huo, J., Shi, Z., Song, Z., Zhang, J., and Zhao, J. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025b.

Hase, P., Bansal, M., Kim, B., and Ghandeharioun, A. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668, 2023.

Hewitt, J. and Manning, C. D. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Hu, J. Y.-C., Chen, B.-Y., Wu, D., Ruan, F., and Liu, H. Nonparametric modern hopfield models. *arXiv preprint arXiv:2404.03900*, 2024a.

Hu, J. Y.-C., Lin, T., Song, Z., and Liu, H. On computational limits of modern hopfield models: A fine-grained complexity analysis. In *Forty-first International Conference on Machine Learning*, 2024b.

Hu, J. Y.-C., Su, M., Kuo, E.-J., Song, Z., and Liu, H. Computational limits of low-rank adaptation (lora) for transformer-based models. *arXiv preprint arXiv:2406.03136*, 2024c.

Hu, J. Y.-C., Wu, D., and Liu, H. Provably optimal memory capacity for modern hopfield models: Transformer-compatible dense associative memories as spherical codes. In *Thirty-eighth Conference on Neural Information Processing Systems (NeurIPS)*, 2024d.

Hu, J. Y.-C., Zhang, X., Su, M., Song, Z., and Liu, H. Minimalist softmax attention provably learns constrained boolean functions. *arXiv preprint arXiv:2505.19531*, 2025.

Huang, B., Jiang, S., Song, Z., Tao, R., and Zhang, R. Solving sdp faster: A robust ipm framework and efficient implementation. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 233–244. IEEE, 2022.

Huang, H., Sun, N., Tani, M., Zhang, Y., Jiang, J., and Jha, S. Can llm-generated misinformation be detected: A study on cyber threat intelligence. *Future Generation Computer Systems*, pp. 107877, 2025.

Huang, X., Liu, W., Du, B., and Tao, D. Leveraged matrix completion with noise. *IEEE Transactions on Cybernetics*, 54(8):4443–4453, 2023.

Ingster, Y. I. On the minimax nonparametric detection of signals in white gaussian noise. *Problemy Peredachi Informatsii*, 18(2):61–73, 1982.

Ingster, Y. I. Minimax testing of nonparametric hypotheses on a distribution density in the l_p metrics. *Theory of Probability & Its Applications*, 31(2):333–337, 1987.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pp. 448–456. PMLR, 2015.

Jiang, H., Kathuria, T., Lee, Y. T., Padmanabhan, S., and Song, Z. A faster interior point method for semidefinite programming. In *2020 IEEE 61st annual symposium on foundations of computer science (FOCS)*, pp. 910–918. IEEE, 2020a.

Jiang, H., Lee, Y. T., Song, Z., and Wong, S. C.-w. An improved cutting plane method for convex optimization, convex-concave games and its applications. In *STOC*, 2020b.

Johnson, D., Goodman, R., Patrinely, J., Stone, C., Zimmerman, E., Donald, R., Chang, S., Berkowitz, S., Finn, A., Jahangir, E., et al. Assessing the accuracy and reliability of ai-generated medical responses: an evaluation of the chat-gpt model. *Research square*, pp. rs–3, 2023.

Kang, W.-C. and McAuley, J. Self-attentive sequential recommendation. In *ICDM*, 2018.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Karunanayake, I., Jiang, J., Ahmed, N., and Jha, S. K. Exploring uncharted waters of website fingerprinting. *IEEE Transactions on Information Forensics and Security*, 19: 1840–1854, 2023.

Ke, Y., Li, X., Liang, Y., Shi, Z., and Song, Z. Circuit complexity bounds for visual autoregressive model. *arXiv preprint arXiv:2501.04299*, 2025.

Kim, J. and Suzuki, T. Transformers provably solve parity efficiently with chain of thought. In *ICLR*, 2025.

Laskar, M. T. R., Bari, M. S., Rahman, M., Bhuiyan, M. A. H., Joty, S., and Huang, J. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 431–469, 2023.

Lee, J. D., Shen, R., Song, Z., Wang, M., et al. Generalized leverage score sampling for neural networks. *Advances in Neural Information Processing Systems*, 33:10775–10787, 2020.

Lee, Y. T. and Sidford, A. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 424–433. IEEE, 2014.

Lee, Y. T., Sidford, A., and Wong, S. C.-w. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1049–1065. IEEE, 2015.

Lee, Y. T., Song, Z., and Zhang, Q. Solving empirical risk minimization in the current matrix multiplication time. In *Conference on Learning Theory*, pp. 2140–2157. PMLR, 2019.

Li, T. and Yuan, M. On the optimality of gaussian kernel based nonparametric tests against smooth alternatives. *arXiv preprint arXiv:1909.03302*, 2019.

Li, X., Liang, Y., Shi, Z., Song, Z., and Yu, J. Fast john ellipsoid computation with differential privacy optimization. *arXiv preprint arXiv:2408.06395*, 2024a.

Li, X., Song, Z., and Yu, J. Quantum speedups for approximating the john ellipsoid. *arXiv preprint arXiv:2408.14018*, 2024b.

Li, X., Liang, Y., Shi, Z., Song, Z., Wang, W., and Zhang, J. On the computational capability of graph neural networks: A circuit complexity bound perspective. *arXiv preprint arXiv:2501.06444*, 2025a.

Li, X., Sun, N., and Jiang, J. Llm-based approaches for real-time cyber threat detection and response: A survey. *manuscript*, 2025b.

Li, X., Xie, S., and Song, Z. Deterministic sparse fourier transform for continuous signals with frequency gap. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025c.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

Li, Z., You, C., Bhojanapalli, S., Li, D., Rawat, A. S., Reddi, S. J., Ye, K., Chern, F., Yu, F., Guo, R., et al. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. *ICLR*, 2023.

Li, Z., Liu, H., Zhou, D., and Ma, T. Chain of thought empowers transformers to solve inherently serial problems. In *ICLR*, 2024c.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. Holistic evaluation of language models. *Transactions on Machine Learning Research (TMLR)*, 2023.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Liu, C., Zhang, J., Wang, S., Fan, W., and Li, Q. Score-based generative diffusion models for social recommendations. *arXiv preprint arXiv:2412.15579*, 2024b.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. Learning deep kernels for non-parametric two-sample tests. In *International conference on machine learning*, pp. 6316–6326. PMLR, 2020a.

Liu, F., Xu, W., Lu, J., and Sutherland, D. J. Meta two-sample testing: Learning kernels for testing with limited data. *Advances in Neural Information Processing Systems*, 34:5848–5860, 2021a.

Liu, F., Wang, H., Zhang, J., Fu, Z., Zhou, A., Qi, J., and Li, Z. Evogan: An evolutionary computation assisted gan. *Neurocomputing*, 469:81–90, 2022a.

Liu, H., Li, Z., Hall, D., Liang, P., and Ma, T. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *ICLR*, 2024c.

Liu, J., Xia, C. S., Wang, Y., and Zhang, L. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *NeurIPS*, 2023a.

Liu, L., Cai, L., Zhang, C., Zhao, X., Gao, J., Wang, W., Lv, Y., Fan, W., Wang, Y., He, M., Liu, Z., and Li, Q. Linrec: Linear attention mechanism for long-term sequential recommender systems. In *SIGIR*, 2023b.

Liu, S. C., Song, Z., and Zhang, H. Breaking the n-pass barrier: A streaming algorithm for maximum weight bipartite matching. *arXiv preprint arXiv:2009.06106*, 2020b.

Liu, S. C., Song, Z., Zhang, H., Zhang, L., and Zhou, T. Space-efficient interior point method, with applications to linear programming and maximum weight bipartite matching. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, pp. 88–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023c.

Liu, S. C., Song, Z., Zhang, H., Zhang, L., and Zhou, T. Space-efficient interior point method, with applications to linear programming and maximum weight bipartite matching. In *ICALP*, 2023d.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Liu, Y. P. and Sidford, A. Faster energy maximization for faster maximum flow. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 803–814, 2020.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021b.

Madry, A. Navigating central path with electrical flows: From flows to matchings, and back. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 253–262. IEEE, 2013.

Madry, A. Computing maximum flow with augmenting electrical flows. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 593–602. IEEE, 2016.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. *NeurIPS*, 2023.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Nay, J. J., Karamardian, D., Lawsky, S. B., Tao, W., Bhat, M., Jain, R., Lee, A. T., Choi, J. H., and Kasai, J. Large language models as tax attorneys: A case study in legal capabilities emergence. *arXiv preprint arXiv:2306.07075*, 2023.

Neyman, J. and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Pallagani, V., Muppasani, B., Murugesan, K., Rossi, F., Srivastava, B., Horesh, L., Fabiano, F., and Loreggia, A. Understanding the capabilities of large language models for automated planning. *arXiv preprint arXiv:2305.16151*, 2023.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Pensia, A., Asadi, A. R., Jog, V., and Loh, P.-L. Simple binary hypothesis testing under local differential privacy and communication constraints. In *COLT*, 2023.

Pensia, A., Jog, V., and Loh, P.-L. The sample complexity of simple binary hypothesis testing. In *COLT*, 2024.

Polyanskiy, Y. and Wu, Y. *Information Theory: From Coding to Learning*. Cambridge University Press, 2023+.

Price, E., Song, Z., and Woodruff, D. P. Fast regression with an ℓ_∞ guarantee. In *International Colloquium on Automata, Languages, and Programming*. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2017.

Pu, D. and Demberg, V. Chatgpt vs human-authored text: Insights into controllable text summarization and sentence style transfer. *ACL (Student Abstract)*, 2023.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., and Yang, D. Is chatgpt a general-purpose natural language processing task solver? *EMNLP*, 2023a.

Qin, L., Song, Z., Zhang, L., and Zhuo, D. An online and unified algorithm for projection matrix vector multiplication with application to empirical risk minimization. In *AISTATS*, 2023b.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

Rathee, K. Meet google meena, 2020.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., and Kim, B. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32, 2019.

Salemi, A. and Zamani, H. Evaluating retrieval quality in retrieval-augmented generation. *SIGIR*, 2024.

Schild, A. An almost-linear time algorithm for uniform random spanning tree generation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 214–227, 2018.

Shen, X., Song, Z., Zhou, Y., Chen, B., Li, Y., Gong, Y., Zhang, K., Tan, H., Kuen, J., Ding, H., et al. Lazydit: Lazy learning for the acceleration of diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39:19, pp. 20409–20417, 2025.

Shimizu, A., Cheng, X., Musco, C., and Weare, J. Improved active learning via dependent leverage score sampling. In *The Twelfth International Conference on Learning Representations*, 2024.

Shin, S., Shomorony, I., and Zhao, H. Efficient learning of linear graph neural networks via node subsampling. *Advances in Neural Information Processing Systems*, 36: 55479–55501, 2023.

Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., and Nanayakkara, S. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17, 2023.

Song, Y., Yuan, Z., Zhang, S., Fang, Z., Yu, J., and Liu, F. Deep kernel relative test for machine-generated text detection. In *The Thirteenth International Conference on Learning Representations*, 2025.

Song, Z. *Matrix Theory: Optimization, Concentration and Algorithms*. PhD thesis, The University of Texas at Austin, 2019.

Song, Z. and Yu, Z. Oblivious sketching-based central path method for solving linear programming problems. In *38th International Conference on Machine Learning (ICML)*, 2021.

Song, Z., Woodruff, D. P., and Zhong, P. Low rank approximation with entrywise l1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 688–701, 2017.

Song, Z., Wang, R., Yang, L., Zhang, H., and Zhong, P. Efficient symmetric norm regression via linear sketching. *Advances in Neural Information Processing Systems*, 32, 2019a.

Song, Z., Woodruff, D. P., and Zhong, P. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2772–2789. SIAM, 2019b.

Song, Z., Yang, X., Yang, Y., and Zhou, T. Faster algorithm for structured john ellipsoid computation. *arXiv preprint arXiv:2211.14407*, 2022.

Song, Z., Ye, M., Yin, J., and Zhang, L. A nearly-optimal bound for fast regression with ℓ_∞ guarantee. In *International Conference on Machine Learning (ICML)*, pp. 32463–32482. PMLR, 2023.

Song, Z., Vakilian, A., Woodruff, D., and Zhou, S. On socially fair low-rank approximation and column subset selection. *Advances in Neural Information Processing Systems*, 37:88874–88905, 2024a.

Song, Z., Yin, J., and Zhang, L. Solving attention kernel regression problem via pre-conditioner. *AISTATS*, 2024b.

Spataro, J. Introducing microsoft 365 copilot – your copilot for work, 2023.

Spielman, D. A. and Srivastava, N. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 563–568, 2008a.

Spielman, D. A. and Srivastava, N. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 563–568, 2008b.

Sridhara, G., Mazumdar, S., et al. Chatgpt: A study on its utility for ubiquitous software engineering tasks. *arXiv preprint arXiv:2305.16837*, 2023.

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., and Zhang, J. Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*, 25(3):1748–1774, 2023.

Team, G. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Tian, K., Jiang, Y., Yuan, Z., Peng, B., and Wang, L. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Tran, N.-T., Tran, V.-H., Nguyen, B.-N., Yang, L., and Cheung, N.-M. M. Self-supervised gan: Analysis and improvement with multi-class minimax game. In *Advances in Neural Information Processing Systems*, 2019.

Vaidya, P. M. A new algorithm for minimizing convex functions over convex sets. In *30th Annual Symposium on Foundations of Computer Science*, pp. 338–343. IEEE Computer Society, 1989.

Valiant, G. and Valiant, P. An automatic inequality prover and instance optimal identity testing. *SIAM Journal on Computing*, 46(1):429–455, 2017.

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

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Wang, L., Lyu, C., Ji, T., Zhang, Z., Yu, D., Shi, S., and Tu, Z. Document-level machine translation with large language models. *EMNLP*, 2023.

Wang, X., Wen, K., Zhang, Z., Hou, L., Liu, Z., and Li, J. Finding skill neurons in pre-trained transformer-based language models. *EMNLP*, 2022.

Wei, C., Chen, Y., and Ma, T. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *Advances in Neural Information Processing Systems*, 35:12071–12083, 2022.

Wu, D., Hu, J. Y.-C., Hsiao, T.-Y., and Liu, H. Uniform memory retrieval with larger capacity for modern hopfield models. In *International Conference on Machine Learning*, pp. 53471–53514. PMLR, 2024a.

Wu, D., Hu, J. Y.-C., Li, W., Chen, B.-Y., and Liu, H. STanhop: Sparse tandem hopfield model for memory-enhanced time series prediction. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b.

Xie, S., Qiu, J., Pasad, A., Du, L., Qu, Q., and Mei, H. Hidden state variability of pretrained language models can guide computation reduction for transfer learning. *EMNLP*, 2022.

Xu, H., Wang, S., Li, N., Wang, K., Zhao, Y., Chen, K., Yu, T., Liu, Y., and Wang, H. Large language models for cyber security: A systematic literature review. *arXiv preprint arXiv:2405.04760*, 2024.

Xu, X., Zhang, J., Liu, F., Sugiyama, M., and Kankanhalli, M. Adversarial attack and defense for non-parametric two-sample tests. In *International Conference on Machine Learning*, pp. 24743–24769. PMLR, 2022.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Yao, T., Li, Y., Pan, Y., and Mei, T. Hiri-vit: Scaling vision transformer with high resolution inputs. *TPAMI*, 2024.

Zamani, H. and Bendersky, M. Stochastic rag: End-to-end retrieval-augmented generation through expected utility maximization. *SIGIR*, 2024.

Zandieh, A., Han, I., Avron, H., Shoham, N., Kim, C., and Shin, J. Scaling neural tangent kernels via sketching and random features. *Advances in Neural Information Processing Systems*, 34:1062–1073, 2021.

Zelikman, E., Huang, Q., Liang, P., Haber, N., and Goodman, N. D. Just one byte (per gradient): A note on low-bandwidth decentralized language model finetuning using shared randomness. *arXiv preprint arXiv:2306.10015*, 2023.

Zhang, J. Graph unlearning with efficient partial retraining. In *Companion Proceedings of the ACM on Web Conference 2024*, pp. 1218–1221, 2024.

Zhang, J., Liu, F., and Zhou, A. Off-tanet: A lightweight neural micro-expression recognizer with optical flow features and integrated attention mechanism. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 266–279. Springer, 2021.

Zhang, J., Xue, R., Fan, W., Xu, X., Li, Q., Pei, J., and Liu, X. Linear-time graph neural networks for scalable recommendations. In *Proceedings of the ACM on Web Conference 2024*, pp. 3533–3544, 2024.

Zhou, K., Zhu, Y., Chen, Z., Chen, W., Zhao, W. X., Chen, X., Lin, Y., Wen, J.-R., and Han, J. Don’t make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

Appendix

Roadmap. In Section A, we provide all proofs which are missing from the main text. In Section B, we present more related work.

A. Missing Proofs

This section provide missing proofs from the main text. In Section A.1, we show the proof of general result for softmax model. In Section A.2, we demonstrate the proof of the lower bound for softmax model. In Section A.3, we prove the local upper bound for softmax model. In Section A.4, we present the proof for local upper bound for leverage score model.

A.1. General Result for Softmax Model

Proof of Theorem 3.1. **Lower bound.** If $\delta \geq 0.1$ then there is nothing to prove. In the following assume that $\delta < 0.1$. Suppose that there is an algorithm that successfully solves the binary hypothesis testing problem. Suppose it makes queries $x_1, \dots, x_m \in \mathbb{R}^d$ where x_i may depend on previous query results. Let $Y_1, \dots, Y_m \in [n]$ denote the query results. Let P_{Y^m} and Q_{Y^m} denote the distribution of Y^m under P and Q, respectively. By definition of δ, we have

$$H^2(P_{Y_k|Y^{k-1}}, Q_{Y_k|Y^{k-1}}) \leq \delta^2.$$

for any $k \in [m]$ and Y^{k-1}. Then

$$\begin{aligned} & 1 - H^2(P_{Y^m}, Q_{Y^m}) \\ = & \int \sqrt{P_{y^m} Q_{y^m}} \mathrm{d}y^m \\ = & \int \sqrt{P_{y^{m-1}} Q_{y^{m-1}}} (\int \sqrt{P_{y_m|y^{m-1}} Q_{y_m|y^{m-1}}} dy_m) \mathrm{d}y^{m-1} \\ \geq & \int \sqrt{P_{y^{m-1}} Q_{y^{m-1}}} (1-\delta^2) \mathrm{d}y^{m-1}. \end{aligned}$$

Repeating this computation, in the end we get

$$1 - H^2(P_{Y^m}, Q_{Y^m}) \geq (1-\delta^2)^m.$$

Because $\delta \leq 0.1$, we have $1 - \delta^2 \geq \exp(-2\delta^2)$. If $m \leq 0.01\delta^{-2}$, then

$$\begin{aligned} 1 - H^2(P_{Y^m}, Q_{Y^m}) \geq & ~ \exp(-2\delta^2 m) \\ \geq & ~ \exp(-0.02) \\ > & ~ 0.98, \end{aligned}$$

and

$$H^2(P_{Y^m}, Q_{Y^m}) \leq 0.02.$$

This implies

$$\mathrm{TV}(P_{Y^m}, Q_{Y^m}) \leq \sqrt{2} H(P_{Y^m}, Q_{Y^m}) \leq 0.2,$$

which implies the success rate for binary hypothesis testing cannot be $\geq \frac{2}{3}$.

In conclusion, any algorithm that successfully solves the hypothesis testing problem need to use $\Omega(\delta^{-2})$ queries.

Upper bound. Take $x \in \mathbb{R}^d$ such that $\|x\|_2 \leq E$ and $\delta = H(\mathtt{SoftMax}_A(x), \mathtt{SoftMax}_B(x))$. By Lemma 2.5, using $O(\delta^{-2})$ samples we can distinguish $\mathtt{SoftMax}_A(x)$ and $\mathtt{SoftMax}_B(x)$. Therefore we can distinguish $\mathtt{SoftMax}_A$ and $\mathtt{SoftMax}_B$ in $O(\delta^{-2})$ queries by repeatedly querying x. □

A.2. Lower Bound for Softmax Model

Before giving the proof of Lemma 3.3, we prove a weaker version of the lemma.

Lemma A.1. *Let $a, b \in \mathbb{R}^n$. Suppose there exists an $\epsilon \geq 0$ such that for every $i \in [n]$, $b_i - a_i \in \{0, \epsilon\}$. Let P be the distribution on $[n]$ with $p_i = \exp(a_i)/\langle \exp(a), \mathbf{1}_n \rangle$. Let Q be the distribution on $[n]$ with $q_i = \exp(b_i)/\langle \exp(b), \mathbf{1}_n \rangle$. Then*

$$\begin{aligned} H^2(P, Q) &= \frac{(1 - \exp(\epsilon/4))^2}{1 + \exp(\epsilon/2)} = O(\epsilon^2), \\ \mathrm{TV}(P, Q) &= \tanh(\epsilon/4) = O(\epsilon). \end{aligned}$$

Proof. Assume that a and b differ in m coordinates. By permuting the coordinates, WLOG assume that $b_i = a_i + \epsilon$ for $1 \leq i \leq m$ and $b_i = a_i$ for $m + 1 \leq i \leq n$.

Write

$$s = \sum_{i=1}^{m} \exp(a_i)$$

and

$$t = \sum_{i=m+1}^{n} \exp(a_i).$$

Then

$$\begin{aligned} H^2(P, Q) &= 1 - \sum_{i \in [n]} \sqrt{p_i q_i} \\ &= 1 - \frac{s \exp(\epsilon/2) + t}{\sqrt{(s + t)(s \exp(\epsilon) + t)}}. \end{aligned}$$

For fixed t and ϵ, the above is maximized at

$$s = t \exp(-\epsilon/2).$$

Plugging in the above s, we get

$$\begin{aligned} H^2(P, Q) &\leq 1 - \frac{2}{\sqrt{(\exp(-\epsilon/2) + 1)(\exp(\epsilon/2) + 1)}} \\ &= \frac{(1 - \exp(\epsilon/4))^2}{1 + \exp(\epsilon/2)}. \end{aligned}$$

For TV, we have

$$\begin{aligned} \mathrm{TV}(P, Q) &= \sum_{m+1 \leq i \leq n} (q_i - p_i) \\ &= \frac{t}{s + t} - \frac{t}{s \exp(\epsilon) + t}. \end{aligned}$$

For fixed t and ϵ the above is maximized at $s = t \exp(-\epsilon/2)$. Plugging in this s, we get

$$\mathrm{TV}(P, Q) \leq \tanh(\epsilon/4).$$

□

Proof of Lemma 3.3. We first prove the case where $b_i \geq a_i$ for all $i \in [n]$. Define $\epsilon_i = b_i - a_i$ for all $i \in [n]$. By permuting the coordinates, WLOG assume that $\epsilon_1 \leq \cdots \leq \epsilon_n$. Specially, define $\epsilon_0 = 0$. For $0 \leq k \leq n$, let $b^k \in \mathbb{R}^n$ denote the vector where $b^k_i = a_i + \min\{\epsilon_i, \epsilon_k\}$ for all $i \in [k]$. Then we can see that $b^0 = a$ and $b^n = b$, and for every $0 \leq k \leq n-1$, the pair (b^k, b^{k+1}) satisfies the assumption in Lemma A.1. For $0 \leq k \leq n$, let P^k denote the softmax distribution corresponding to b^k. By Lemma A.1, for every $0 \leq k \leq n-1$, we have

$$\begin{aligned} H(P^k, P^{k+1}) &= O(\epsilon_{k+1} - \epsilon_k), \\ \mathrm{TV}(P^k, P^{k+1}) &= O(\epsilon_{k+1} - \epsilon_k). \end{aligned}$$

Because Hellinger distance and TV distance are both metrics, we have

$$\begin{aligned} H(P, Q) &= H(P^0, P^n) \\ &\leq \sum_{k=0}^{n-1} H(P^k, P^{k+1}) \\ &= O(\epsilon), \end{aligned}$$

and

$$\begin{aligned} \mathrm{TV}(P, Q) =&\ \mathrm{TV}(P^0, P^n) \\ &\leq \sum_{k=0}^{n-1} \mathrm{TV}(P^k, P^{k+1}) \\ &= O(\epsilon). \end{aligned}$$

This finishes the proof of the result when $b_i \geq a_i$ for all $i \in [n]$.

Now let us consider the general case. Let $c \in \mathbb{R}^n$ be defined as $c_i = \max\{a_i, b_i\}$ for all $i \in [n]$. Then

$$\max\{\|a - c\|_\infty, \|c - b\|_\infty\} \leq \|a - b\|_\infty \leq \epsilon.$$

Let R be the softmax distribution corresponding to c. By our previous discussion, we have

$$H(P, R), H(R, Q), \mathrm{TV}(P, R), \mathrm{TV}(R, Q) = O(\epsilon).$$

By metric property of Hellinger distance and TV distance, we get

$$H(P, Q), H(P, Q) = O(\epsilon)$$

as desired.

□

A.3. Local Upper Bound for Softmax Model

Proof of Theorem 3.5. We take an x satisfying $\|x\|_2 \leq E$ that maximizes $\mathrm{Var}_{\mathtt{SoftMax}_A(x)}(Mx)$ and repeatedly query x. We would like to apply Theorem 3.1. To do that, we need to show that

$$H^2(\mathtt{SoftMax}_A(x), \mathtt{SoftMax}_{B_\epsilon}(x)) = \Omega(\epsilon^2 \nu).$$

Let $P = \mathtt{SoftMax}_A(x) = (p_1, \ldots, p_n)$, $Q_\epsilon = \mathtt{SoftMax}_{B_\epsilon}(x) = (q_{\epsilon,1}, \ldots, q_{\epsilon,n})$. Write $Z_A = \langle \exp(Ax), \mathbf{1}_n \rangle$, $Z_{B_\epsilon} = \langle \exp(B_\epsilon x), \mathbf{1}_n \rangle$.

Then, it follows that

$$\begin{aligned} Z_B &= \sum_{j \in [n]} \exp(Ax)_j \exp(\epsilon (Mx)_j) \\ &= \sum_{j \in [n]} \exp(Ax)_j + \sum_{j \in [n]} \exp(Ax)_j (\exp(\epsilon (Mx)_j) - 1) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j\in[n]} \exp(Ax)_j + \sum_{j\in[n]} \exp(Ax)_j(\epsilon(Mx)_j + O(\epsilon^2)) \\
&= Z_A(1+\epsilon\langle p, Mx\rangle + O(\epsilon^2)). \qquad (3)
\end{aligned}
$$

where the initial step is because of $B = A + \epsilon M$, the second step is a result of simple algebra, the third step is a consequence of the Taylor expansion of $\exp(\cdot)$, assuming ϵ is sufficiently small and the fourth step is the result of the definition of Z_A and involves the consolidation of addition, introducing the common term Z_A.

Then

$$
\begin{aligned}
q_{\epsilon,i} &= \frac{\exp(B_\epsilon x)_i}{Z_B} \\
&= \frac{\exp(Ax)_i \exp(\epsilon Mx)_i}{Z_A(1+\epsilon\langle p, Mx\rangle + O(\epsilon^2))} \\
&= p_i(1+\epsilon((Mx)_i - \langle p, Mx\rangle) + O(\epsilon^2)). \qquad (4)
\end{aligned}
$$

where the initial step is because of the definition of $q_{\epsilon,i}$, the subsequent step is a result of Eq.(3), and the third step is due to the definition of q_i along with the Taylor expansion of $f(x) = 1/(1+x)$ and $\exp(\cdot)$, considering ϵ as a sufficiently small value.

So, we have that

$$
\begin{aligned}
H^2(P, Q_\epsilon) &= \frac{1}{2}\sum_{i=1}^{n}(\sqrt{p_i} - \sqrt{q_{\epsilon,i}})^2 \\
&= \frac{1}{2}\sum_{i=1}^{n} p_i(\epsilon^2((Mx)_i - \langle p, Mx\rangle)^2 + O(\epsilon^3)) \\
&= \frac{1}{2}\epsilon^2 \operatorname{Var}_P(Mx) + O(\epsilon^3) \\
&= \frac{1}{2}\epsilon^2\nu + O(\epsilon^3).
\end{aligned}
$$

where the first step is the result of Definition 2.2, the second step is because of Eq.(4), the third step the result of definition of $\operatorname{Var}_P(Mx)$ (See Definition 2.3) and the forth step follows from the expression $\nu = \sup_{x:\|x\|_2\le E} \operatorname{Var}_{\mathtt{SoftMax}_A(x)}(Mx)$.

Applying Theorem 3.1 we finish the proof. □

A.4. Local Upper Bound for Leverage Score Model

Proof of Theorem 4.3. We take an s satisfying $c \le s_i^2 \le C$ and $\forall i \in [n]$ that maximizes $\sup_s \operatorname{Var}_{\mathtt{Leverage}_A(s)}(w_s)$ and repeatedly query s. We need to show that

$$H^2(\mathtt{Leverage}_A(s), \mathtt{Leverage}_{B_\epsilon}(s)) = \Omega(\epsilon^2\nu).$$

Let $P = \mathtt{Leverage}_A(s) = (p_1, \dots, p_n)$, $Q_\epsilon = \mathtt{Leverage}_{B_\epsilon}(x) = (q_{\epsilon,1}, \dots, q_{\epsilon,n})$. We can compute that

$$\frac{d}{d\epsilon} q_{\epsilon,i} = (2(I - A_s(A_s^\top A_s)^{-1}A_s^\top)(M_s(A_s^\top A_s)^{-1}A_s^\top))_{i,i}.$$

Define $W = (I - A_s(A_s^\top A_s)^{-1}A_s^\top)(M_s(A_s^\top A_s)^{-1}A_s^\top)$. Then

$$q_{\epsilon,i} = p_i + 2W_{i,i}\epsilon + O(\epsilon^2).$$

Computing $H^2(P, Q_\epsilon)$ we get

$$H^2(P, Q_\epsilon) = \frac{1}{2}\sum_{i\in[n]}(\sqrt{q_{\epsilon,i}} - \sqrt{p_i})^2$$

$$
\begin{aligned}
&= \sum_{i \in [n]} p_i \left(\frac{W_{i,i}}{p_i} \epsilon + O(\epsilon^2) \right)^2 \\
&= \sum_{i \in [n]} \frac{W_{i,i} \epsilon^2}{p_i} + O(\epsilon^3) \\
&= \epsilon^2 \nu + O(\epsilon^3).
\end{aligned}
$$

□

B. More Related Work

Softmax Computation and Regression Softmax computation, a crucial element in attention computation (Vaswani et al., 2017), plays a pivotal role in the development of LLMs. Several studies (Alman & Song, 2023; Brand et al., 2024; Liu et al., 2023d; Deng et al., 2023b) delve into the efficiency of softmax computation. To improve computational efficiency, (Alman & Song, 2023) presents a quicker attention computation algorithm utilizing implicit matrices. Similarly, (Brand et al., 2024) utilizes lazy updates to speed up dynamic computation, while (Deng et al., 2023b) employs a randomized algorithm for similar efficiency gains. Conversely, (Liu et al., 2023d) utilizes an approximate Newton method that operates in nearly linear time. (Gao et al., 2023) centers on the convergence of overparameterized two-layer networks with exponential activation functions, whereas (Deng et al., 2023a; Liu et al., 2023d) explore regression analysis within the framework of attention computation. All of these studies specifically focus on softmax-based regression problems.

Softmax functions is also widely used in computer vision. In particular, it is widely used in different backbone models, such as Vision Transformers (ViTs) (Dosovitskiy et al., 2020; Zhang et al., 2021; Liu et al., 2021b; Yao et al., 2024) and Visual Autoregressive (VAR) (Tian et al., 2024) models, and also has advanced many applications including diffusion models (Peebles & Xie, 2023; Cao et al., 2025d; Shen et al., 2025), flow matching (Lipman et al., 2022; Liu et al., 2022b; Cao et al., 2025a), and GANs (Brock et al., 2019; Tran et al., 2019; Liu et al., 2022a). Beyond computer vision, there are also other ML applications that involve softmax computation, such as Hopfield networks (Karunanayake et al., 2023; Hu et al., 2024b;d;a; Wu et al., 2024b;a; Hu et al., 2025), graph neural networks (Veličković et al., 2018; Fu et al., 2021; Go et al., 2022; Chang et al., 2024), recommender systems (Kang & McAuley, 2018; Liu et al., 2023b; 2024b), and fairness problems (Corbett-Davies et al., 2023; Song et al., 2024a; Cao et al., 2025c).