The Closeness of In-Context Learning and Weight Shifting for Softmax Regression

Shuai Li
Shanghai Jiao Tong University
shuaili8@sjtu.edu.cn

Zhao Song
Simons Institute for the Theory of Computing, UC Berkeley
magic.linuxkde@gmail.com

Yu Xia
University of California, San Diego
yux078@ucsd.edu

Tong Yu
Adobe Research
tyu@adobe.com

Tianyi Zhou
University of Southern California
tzhou029@usc.edu

Abstract

Large language models (LLMs) are known for their exceptional performance in natural language processing, making them highly effective in many human life-related tasks. The attention mechanism in the Transformer architecture is a critical component of LLMs, as it allows the model to selectively focus on specific input parts. The softmax unit, which is a key part of the attention mechanism, normalizes the attention scores. Hence, the performance of LLMs in various NLP tasks depends significantly on the crucial role played by the attention mechanism with the softmax unit.

In-context learning is one of the celebrated abilities of recent LLMs. Without further parameter updates, Transformers can learn to predict based on few in-context examples. However, the reason why Transformers becomes in-context learners is not well understood. Recently, in-context learning has been studied from a mathematical perspective with simplified linear self-attention without softmax unit. Based on a linear regression formulation $\min_x \|Ax - b\|_2$, existing works show linear Transformers' capability of learning linear functions in context. The capability of Transformers with softmax unit approaching full Transformers, however, remains unexplored.

In this work, we study the in-context learning based on a softmax regression formulation $\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b\|_2$. We show the upper bounds of the data transformations induced by a single self-attention layer with softmax unit and by gradient-descent on a ℓ_2 regression loss for softmax prediction function. Our theoretical results imply that when training self-attention-only Transformers for fundamental regression tasks, the models learned by gradient-descent and Transformers show great similarity.

1 Introduction

In recent years, there has been a significant increase in research and development in the field of Artificial Intelligence, with large language models (LLMs) emerging as an effective way to tackle complex tasks. Transformers have achieved state-of-the-art results in various NLP tasks, such as machine translation [PCR19, GHG$^+$20] and text generation [LSX$^+$22]. As a result, they have become the preferred architecture for NLP, where BERT [DCLT18], GPT-3 [BMR$^+$20], PaLM [CND$^+$22] were proposed. They have demonstrated remarkable learning and reasoning capabilities and have proven to be more efficient than traditional models when processing natural language.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2001

Additionally, LLMs can be fine-tuned for multiple purposes without requiring a new build from scratch, making them a versatile tool for AI applications. Moreover, recent studies on the in-context learning abilities of LLMs have demonstrated that even without further fine-tuning, LLMs can generalize to new tasks with only a few demonstration examples in the prompt. To understand how LLMs become in-context learners, recent works have studied the problem and provided mathematical explanations from the Transformer architecture perspective, showing a simplified linear self-attention layer of Transformer can learn linear functions similarly as a step of gradient descent [ONR+22, ASA+22, GTLV22, CLL+24]. While such linear approximation of full Transformers is overly simplistic, studies on more complex Transformer architecture are needed to further explain the in-context learning phenomenon.

Transformers have a specific type of sequence-to-sequence neural network architecture. They utilize the attention mechanism [VSP+17, RNS+18, DCLT18, BMR+20] that allows them to capture long-range dependencies and context from input data effectively. The core of the attention mechanism is the attention matrix which is comprised of rows and columns, corresponding to individual words or "tokens". The attention matrix represents the relationships within the given text. It measures the importance of each token in a sequence as it relates to the desired output. During the training process, the attention matrix is learned and optimized to improve the accuracy of the model's predictions. Through the attention mechanism, each input token is evaluated based on its relevance to the desired output by assigning a token score. This score is determined by a similarity function that compares the current output state with input states.

Theoretically, the attention matrix is comprised of the query matrix $Q \in \mathbb{R}^{n \times d}$, the key matrix $K \in \mathbb{R}^{n \times d}$ and the value matrix $V \in \mathbb{R}^{n \times d}$. Following [ZHDK23, AS23, BSZ24, AS24b, AS24c, AS24a], the computation of the normalized attention function is defined as $D^{-1} \exp(QK^\top)V$. Following the transformer literature, we apply $\exp$ to a matrix entry-wise way. Here $D \in \mathbb{R}^{n \times n}$ is diagonal matrix that defined as $D = \mathrm{diag}(\exp(QK^\top)\mathbf{1}_n)$. Intuitively, D denotes the softmax normalization matrix. A more general computation formulation can be written as

$$\underbrace{D^{-1}}_{n \times n \text{ diagonal matrix}} \underbrace{\exp(XQK^\top X^\top)}_{n \times n} \underbrace{X}_{n \times d} \underbrace{V}_{d \times d},$$

where

$$D := \mathrm{diag}(\exp(XQK^\top X^\top)\mathbf{1}_n).$$

In the above setting, we treat $Q, K, V \in \mathbb{R}^{d \times d}$ as weights and X is the input sentence data that has length n and each word embedding size is d. In the remaining of the part, we will switch X to notation A and use A to denote sentence. Mathematically, the attention computation problem can be formulated as a regression problem in the following sense

Definition 1.1. *We consider the following problem*

$$\min_{X \in \mathbb{R}^{d \times d}} \|D^{-1} \exp(AXA^\top) - B\|_F$$

where $A \in \mathbb{R}^{n \times d}$ can be treated as a length-n document and each word has length-d embedding size. Here $D = \mathrm{diag}(AXA^\top \mathbf{1}_n)$. For any given $A \in \mathbb{R}^{n \times d}$ and $B \in \mathbb{R}^{n \times n}$, the goal is to find some weight X to optimize the above objective function.

In contrast to the formulation in [ZHDK23, AS23, BSZ24], the parameter X in Definition 1.1 is equivalent to the $QK^\top \in \mathbb{R}^{d \times d}$ in the generalized version of [ZHDK23, AS23, BSZ24] (e.g. replacing $Q \in \mathbb{R}^{n \times d}$ by XQ where $X \in \mathbb{R}^{n \times d}$ and $Q \in \mathbb{R}^{d \times d}$. Similarly for K and V. In such scenario, X can be viewed as a matrix representation of a length-n sentence.).

A number of work [ASA+22, GTLV22, ONR+22] study the in-context learning from mathematical perspective in a much simplified setting than Definition 1.1, which is linear regression formulation as in Definition 1.2. They show linear Transformer without softmax unit in its attention layer can mimic the ability of gradient descent in learning linear functions in context. While the softmax unit plays an important role in attention computations of full Transformers, their simplification of the softmax unit leaves a gap in explaining LLMs' in-context learning abilities.

Definition 1.2. *Given a matrix $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, the goal is to solve*

$$\min_x \|Ax - b\|_2$$

https://doi.org/10.52202/079017-2001

Several theoretical transformer work have studied either exponential regression [GMS23, LSZ23] or softmax regression problem [DLS23, LLSS24a]. In this work, to take a step forward to understand the softmax unit in the attention scheme in LLMs. We consider the following softmax regression and study the in-context learning phenomena and its closeness to gradient descent from the data transformation perspective.

Definition 1.3 (Softmax Regression)**.** *Given a $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$, the goal is to solve*

$$\min_{x \in \mathbb{R}^d} \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2$$

We remark that the Definition 1.3 of Softmax Regression is a formulation in between Definition 1.2 and Definition 1.1.

We state our major result as follows:

Theorem 1.4 (Bounded shift for in-context learning, informal version of the combination of Theorem 4.2 and Theorem 4.3)**.** *If the following conditions hold: Let $A \in \mathbb{R}^{n \times d}$. Let $b \in \mathbb{R}^n$. $\|A\| \leq R$. Let $\|x\|_2 \leq R$. $\|A(x_{t+1} - x_t)\|_\infty < 0.01$. $\|(A_{t+1} - A_t)x\|_\infty < 0.01$. Let $R \geq 4$. Let $M := n^{1.5} \exp(10R^2)$. We consider the softmax regression (Definition 1.3) problem*

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2.$$

- **Part 1.** *If we move the x_t to x_{t+1}, then we're solving a new softmax regression with $\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widetilde{b} \|_2$ where $\|\widetilde{b} - b\|_2 \leq M \cdot \|x_{t+1} - x_t\|_2$*
- **Part 2.** *If we move the A_t to A_{t+1}, then we're solving a new softmax regression with $\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widehat{b} \|_2$ where $\|\widehat{b} - b\|_2 \leq M \cdot \|A_{t+1} - A_t\|$*

Recall that $A \in \mathbb{R}^{n \times d}$ denotes a length-n document and each word has the length-d embedding size and x denotes the simplified version of $QK^\top$. One-step gradient descent can be treated as an update to the model's weight x. Thus, part 1 of our result (Theorem 1.4) implies that the data transformation of b induced by gradient-descent on the ℓ_2 regression loss is bounded by $M \cdot \|x_{t+1} - x_t\|_2$.

According to [ONR+22], to do in-context learning, a self-attention layer update can be treated as an update to the tokenized document A. For detailed derivation, please refer to [ONR+22]. Thus, part 2 of our result (Theorem 1.4) implies that the data transformation of b induced by a single self-attention layer is bounded by $M \cdot \|A_{t+1} - A_t\|$.

We remark that the data transformation of b induced by $1)$ a single self-attention layer and by $2)$ gradient-descent on the ℓ_2 regression loss are both bounded. The bounded transformation of b implies that when training self-attention-only Transformers for fundamental regression tasks, the models learned by gradient-descent and Transformers show great similarity.

Roadmap. In Section 2, we give some preliminaries. In Section 3, we compute the gradient of the loss function with softmax function with respect to x. Those functions include $\alpha(x)^{-1}$, $\alpha(x)$ and $f(x)$. In Section 4, we give our formal theoretical results, validated by numerical experiments presented in Section 5. In Section 6, we conclude our paper.

2 Preliminary

In Section 2.1, we introduce the notations used in this paper. In Section 2.2, we give some facts about the basic algebra. In Section 2.3, we propose the lower bound on $\langle \exp(Ax), \mathbf{1}_n \rangle$.

2.1 Notations

For a positive integer n, we use $[n]$ to denote $\{1, 2, \cdots, n\}$, for any positive integer n. We use $\mathbb{E}[\cdot]$ to denote expectation. We use $\Pr[\cdot]$ to denote probability. We use $\mathbf{1}_n$ to denote the vector where all entries are one. We use $\mathbf{0}_0$ to denote the vector where all entries are zero. The identity matrix of size $n \times n$ is represented by I_n for a positive integer n. The symbol $\mathbb{R}$ refers to real numbers and $\mathbb{R}_{\geq 0}$ represents non-negative real numbers. For any vector $x \in \mathbb{R}^n$, $\exp(x) \in \mathbb{R}^n$ denotes a vector where the i-th entry is $\exp(x_i)$ and $\|x\|_2$ represents its ℓ_2 norm, that is, $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$. We use

https://doi.org/10.52202/079017-2001

$\|x\|_\infty$ to denote $\max_{i\in[n]} |x_i|$. For any vector $x \in \mathbb{R}^n$ and vector $y \in \mathbb{R}^d$, we use $\langle x, y\rangle$ to denote the inner product of vector x and y. The notation B_i is used to indicate the i-th row of matrix B. If a and b are two column vectors in $\mathbb{R}^n$, then $a \circ b$ denotes a column vector where $(a \circ b)_i = a_i b_i$. For a square and full rank matrix B, we use B^{-1} to denote the true inverse of B.

2.2 Basic Algebras

Fact 2.1. *For vectors $x, y \in \mathbb{R}^n$, we have*

- $\|x \circ y\|_2 \leq \|x\|_\infty \cdot \|y\|_2$
- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$
- $\|\exp(x)\|_\infty \leq \exp(\|x\|_2)$
- *For any $\|x - y\|_\infty \leq 0.01$, we have $\|\exp(x) - \exp(y)\|_2 \leq \|\exp(x)\|_2 \cdot 2\|x - y\|_\infty$*

Fact 2.2. *For matrices X, Y, we have*

- $\|X^\top\| = \|X\|$
- $\|X\| \geq \|Y\| - \|X - Y\|$
- $\|X + Y\| \leq \|X\| + \|Y\|$
- $\|X \cdot Y\| \leq \|X\| \cdot \|Y\|$
- *If $X \preceq \alpha \cdot Y$, then $\|X\| \leq \alpha \cdot \|Y\|$*

2.3 Lower bound on β

Lemma 2.3. *If the following conditions holds*

- $\|A\| \leq R$
- $\|x\|_2 \leq R$
- *Let β be lower bound on $\langle \exp(Ax), \mathbf{1}_n\rangle$*

Then we have

$$\beta \geq \exp(-R^2)$$

Proof. We have

$$\begin{aligned}
\langle \exp(Ax), \mathbf{1}_n\rangle = &~ \sum_{i=1}^n \exp((Ax)_i) \\
\geq &~ \min_{i\in[n]} \exp((Ax)_i) \\
\geq &~ \min_{i\in[n]} \exp(-|(Ax)_i|) \\
= &~ \exp(-\max_{i\in[n]} |(Ax)_i|) \\
= &~ \exp(-\|Ax\|_\infty) \\
\geq &~ \exp(-\|Ax\|_2) \\
\geq &~ \exp(-R^2)
\end{aligned}$$

the 1st step follows from simple algebra, the 2nd step comes from simple algebra, the 3rd step follows from the fact that $\exp(x) \geq \exp(-|x|)$, the 4th step follows from the fact that $\exp(-x)$ is monotonically decreasing, the 5th step comes from definition of ℓ_∞ norm, the 6th step follows from Fact 2.1, the 7th step follows from the assumption on A and x. □

https://doi.org/10.52202/079017-2001

3 Softmax Function with Respect to x

In Section 3.1, we give the definitions used in the computation. In Section 3.2, we compute the gradient of the loss function with softmax function with respect to x. Those functions includes $\alpha(x)^{-1}$, $\alpha(x)$ and $f(x)$.

3.1 Definitions

We define function softmax f as follows

Definition 3.1 (Function f, Definition 5.1 in [DLS23]). *Given a matrix $A \in \mathbb{R}^{n \times d}$. Let $\mathbf{1}_n$ denote a length-n vector that all entries are ones. We define prediction function $f : \mathbb{R}^d \to \mathbb{R}^n$ as follows*

$$f(x) := \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \cdot \exp(Ax).$$

Definition 3.2 (Loss function $L_{\exp}$, Definition 5.3 in [DLS23]). *Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$. We define loss function $L_{\exp} : \mathbb{R}^d \to \mathbb{R}$ as follows*

$$L_{\exp}(x) := 0.5 \cdot \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2^2.$$

For convenient, we define two helpful notations α and c

Definition 3.3 (Normalized coefficients, Definition 5.4 in [DLS23]). *We define $\alpha : \mathbb{R}^d \to \mathbb{R}$ as follows*

$$\alpha(x) := \langle \exp(Ax), \mathbf{1}_n \rangle.$$

Then, we can rewrite $f(x)$ (see Definition 3.1) and $L_{\exp}(x)$ (see Definition 3.2) as follows

- $f(x) = \alpha(x)^{-1} \cdot \exp(Ax)$.
- $L_{\exp}(x) = 0.5 \cdot \| \alpha(x)^{-1} \cdot \exp(Ax) - b \|_2^2$.
- $L_{\exp}(x) = 0.5 \cdot \| f(x) - b \|_2^2$.

Definition 3.4 (Definition 5.5 in [DLS23]). *We define function $c : \mathbb{R}^d \in \mathbb{R}^n$ as follows*

$$c(x) := f(x) - b.$$

Then we can rewrite $L_{\exp}(x)$ (see Definition 3.2) as follows

- $L_{\exp}(x) = 0.5 \cdot \| c(x) \|_2^2$.

3.2 Gradient Computations

We state a lemma from previous work,

Lemma 3.5 (Gradient, Lemma 5.6 in [DLS23]). *If the following conditions hold*

- *Given matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$.*
- *Let $\alpha(x)$ be defined in Definition 3.3.*
- *Let $f(x)$ be defined in Definition 3.1.*
- *Let $c(x)$ be defined in Definition 3.4.*
- *Let $L_{\exp}(x)$ be defined in Definition 3.2.*

For each $i \in [d]$, we have

- *Part 1.*

$$\frac{\mathrm{d} \exp(Ax)}{\mathrm{d} x_i} = \exp(Ax) \circ A_{*,i}$$

- *Part 2.*

$$\frac{\mathrm{d}\langle \exp(Ax), \mathbf{1}_n \rangle}{\mathrm{d}x_i} = \langle \exp(Ax), A_{*,i} \rangle$$

- *Part 3.*

$$\frac{\mathrm{d}\alpha(x)^{-1}}{\mathrm{d}x_i} = -\alpha(x)^{-1} \cdot \langle f(x), A_{*,i} \rangle$$

- *Part 4.*

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x_i} = \frac{\mathrm{d}c(x)}{\mathrm{d}x_i} = -\ \langle f(x), A_{*,i} \rangle \cdot f(x)\ + f(x) \circ A_{*,i}$$

- *Part 5.*

$$\frac{\mathrm{d}L_{\exp}(x)}{\mathrm{d}x_i} = \underbrace{A_{*,i}^\top}_{1\times n} \cdot \Big(\underbrace{f(x)}_{n\times 1} \underbrace{\langle c(x), f(x) \rangle}_{\text{scalar}} + \underbrace{\mathrm{diag}(f(x))}_{n\times n} \underbrace{c(x)}_{n\times 1} \Big)$$

4 Main Results

In Section 4.1, we show the lipschitz bound of function f. In Section 4.2, we show our upper bound result of δ_b with respect to x. In Section 4.3, we show our upper bound result of δ_b with respect to A.

4.1 Lipschitz Bound

To bound the shift of b, we first show the Lipschitz property for the basic functions:

- $\| \exp(Ax) - \exp(Ay) \|_2 \leq 2\sqrt{n} R \exp(R^2) \cdot \|x - y\|_2$
- $|\alpha(x) - \alpha(y)| \leq \| \exp(Ax) - \exp(Ay) \|_2 \cdot \sqrt{n}$
- $|\alpha(x)^{-1} - \alpha(y)^{-1}| \leq \beta^{-2} \cdot |\alpha(x) - \alpha(y)|$

We can show that

Lemma 4.1. *If the following conditions hold*

- *Let* $\beta \in (0, 1)$*.*
- *Let* $\delta_{b,1} \in \mathbb{R}^n$ *be defined as Definition B.3.*
- *Let* $\delta_{b,2} \in \mathbb{R}^n$ *be defined as Definition B.3.*
- *Let* $\delta_b = \delta_{b,1} + \delta_{b,2}$*.*
- *Let* $R \geq 4$*.*

We have

- *Part 1.*

$$\|\delta_{b,1}\|_2 \leq 2\beta^{-2} n^{1.5} \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2$$

- *Part 2.*

$$\|\delta_{b,2}\|_2 \leq 2\beta^{-1} \sqrt{n} R \exp(R^2) \cdot \|x_{t+1} - x_t\|_2$$

- *Part 3.*

$$\| \underbrace{f(x_{t+1}) - f(x_t)}_{\delta_b} \|_2 \leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2$$

Proof. **Proof of Part 1.** We have

$$\begin{aligned}
\|\delta_{b,1}\|_2 \leq & ~ |\alpha(x_{t+1})^{-1} - \alpha(x_t)^{-1}| \cdot \| \exp(Ax_{t+1})\|_2 \\
\leq & ~ |\alpha(x_{t+1})^{-1} - \alpha(x_t)^{-1}| \cdot \sqrt{n} \cdot \exp(R^2) \\
\leq & ~ \beta^{-2} \cdot |\alpha(x_{t+1}) - \alpha(x_t)| \cdot \sqrt{n} \cdot \exp(R^2) \\
\leq & ~ \beta^{-2} \cdot \sqrt{n} \cdot \| \exp(Ax_{t+1}) - \exp(Ax_t)\|_2 \cdot \sqrt{n} \cdot \exp(R^2) \\
\leq & ~ \beta^{-2} \cdot \sqrt{n} \cdot 2\sqrt{n} R \exp(R^2) \|x_{t+1} - x_t\|_2 \cdot \sqrt{n} \cdot \exp(R^2) \\
= & ~ 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2
\end{aligned}$$

where the first step follows from definition, the second step follows from assumption on A and x, the third step follows Lemma B.7, the forth step follows from Lemma B.6, the fifth step follows from Lemma B.5.

Proof of Part 2.

We have

$$\begin{aligned}
\|\delta_{b,2}\|_2 \leq & ~ |\alpha(x_{t+1})^{-1}| \cdot \| \exp(Ax_{t+1}) - \exp(Ax_t)\|_2 \\
\leq & ~ \beta^{-1} \cdot \| \exp(Ax_{t+1}) - \exp(Ax_t)\|_2 \\
\leq & ~ \beta^{-1} \cdot 2\sqrt{n} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2
\end{aligned}$$

where the first step follows from definition, the 2nd step comes from Lemma B.5.

Proof of Part 3.

We have

$$\begin{aligned}
\|\delta_b\|_2 = & ~ \|\delta_{b,1} + \delta_{b,2}\|_2 \\
\leq & ~ \|\delta_{b,1}\|_2 + \|\delta_{b,2}\|_2 \\
\leq & ~ 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 + 2\beta^{-1} n^{0.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 \\
\leq & ~ 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 + 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 \\
\leq & ~ 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2
\end{aligned}$$

where the 1st step follows from the definition of δ_b, the 2nd step follows from triangle inequality, the 3rd step follows from the results in Part 1 and Part 2, the 4th step follows from the fact that $n \geq 1$ and $\beta^{-1} \geq 1$, the 5th step follows from simple algebra. □

Similarly, we can show the Lipschitz property of function f with respect to A as the following

$$\begin{aligned}
& ~ \|f(A_{t+1}) - f(A_t)\|_2 \\
\leq & ~ 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|_2
\end{aligned}$$

Due to space limitation, we defer formal lemma and proof to D.2.

4.2 Shifting Weight Parameter x

Theorem 4.2 (Bounded shift for shifting the weight parameter, formal of Theorem 1.4)**.** *If the following conditions hold*

- *Let* $A \in \mathbb{R}^{n \times d}$
- $\|A\| \leq R$
- $\|A(x_{t+1} - x_t)\|_\infty < 0.01$
- *Let* $R \geq 4$
- *Let* $M := n^{1.5} \exp(10R^2)$.

We consider the softmax regression problem

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2$$

If we move the x_t to x_{t+1}, then we're solving a new softmax regression problem with

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widetilde{b} \|_2$$

where

$$\| \widetilde{b} - b \|_2 \leq M \cdot \| x_{t+1} - x_t \|_2$$

Proof. We have

$$\begin{aligned} \| \widetilde{b} - b \|_2 \leq & ~ 4 \beta^{-2} n^{1.5} R \exp(2R^2) \cdot \| x_{t+1} - x_t \|_2 \\ \leq & ~ 4 n^{1.5} R \exp(2R^2) \exp(2R^2) \cdot \| x_{t+1} - x_t \|_2 \\ \leq & ~ n^{1.5} (4R) \exp(4R^2) \cdot \| x_{t+1} - x_t \|_2 \\ \leq & ~ n^{1.5} \exp(6R^2) \exp(4R^2) \cdot \| x_{t+1} - x_t \|_2 \\ \leq & ~ n^{1.5} \exp(10R^2) \cdot \| x_{t+1} - x_t \|_2 \\ \leq & ~ M \cdot \| x_{t+1} - x_t \|_2 \end{aligned}$$

where the 1st step follows from Lemma 4.1, the 2nd step comes from Lemma 2.3, the 3rd step comes from simple algebra, the 4th step follows from simple algebra, the 5th step follows from simple algebra and the 6th step follows from the definition of M. □

4.3 Shifting Sentence Data A

Theorem 4.3 (Bounded shift for in-context learning, formal of Theorem 1.4)**.** *If the following conditions hold*

- *Let* $A \in \mathbb{R}^{n \times d}$
- $\|A\| \leq R$
- $\|(A_{t+1} - A_t) x\|_\infty < 0.01$
- *Let* $R \geq 4$
- *Let* $M := n^{1.5} \exp(10R^2)$.

We consider the softmax regression problem

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2$$

If we move the A_t to A_{t+1} then we're solving a new softmax regression problem with

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widetilde{b} \|_2$$

where

$$\| \widetilde{b} - b \|_2 \leq M \cdot \| A_{t+1} - A_t \|.$$

Proof. We have

$$\begin{aligned} \| \widetilde{b} - b \|_2 \leq & ~ 4 \beta^{-2} n^{1.5} R \exp(2R^2) \cdot \| A_{t+1} - A_t \| \\ \leq & ~ 4 n^{1.5} R \exp(2R^2) \exp(2R^2) \cdot \| A_{t+1} - A_t \| \\ \leq & ~ n^{1.5} (4R) \exp(4R^2) \cdot \| A_{t+1} - A_t \| \\ \leq & ~ n^{1.5} \exp(6R^2) \exp(4R^2) \cdot \| A_{t+1} - A_t \| \\ \leq & ~ n^{1.5} \exp(10R^2) \cdot \| A_{t+1} - A_t \| \\ \leq & ~ M \cdot \| A_{t+1} - A_t \| \end{aligned}$$

where the 1st step follows from Lemma D.5, the 2nd step follows from Lemma 2.3, the 3rd step follows from simple algebra, the 4th step comes from simple algebra, the 5th step comes from simple algebra and the 6th step follows from the definition of M. □

5 Numerical Experiments

In this section, we present our numerical experiments to validate our theoretical results that when training self-attention-only Transformers for softmax regression tasks, the models learned by gradient-descent and Transformers show great similarity.

5.1 Experiments Setup

According to Definition 1.3, we construct the synthetic softmax regression tasks consists of randomly sampled length-n documents $A \in \mathbb{R}^{n \times d}$ where each word has the d-dimensional embedding and targets $b \in \mathbb{R}^n$. Each document is generated from a unique random seed. In our experiments, we choose a set of different document length n and a set of different embedding size d.

Following [ONR+22], we compare the following two models in our experiment

- a trained single self-attention (SA) layer with softmax unit approximating full Transformers.
- a softmax regression model trained with one-step gradient descent (GD).

The training objective for both models is defined as in Definition 1.3. For the single self-attention layer with a softmax unit, we choose the learning rate $\eta_{\mathrm{SA}} = 0.005$. For the softmax regression model, we determine the optimal learning rate η_{GD} by minimizing the ℓ_2 regression loss over a training set of 10^3 tasks through line search.

To compare the trained single self-attention layer with a softmax unit and the softmax regression model trained with one-step gradient descent, we sample 10^3 tasks and record the losses of two models. In addition, we follow [ONR+22] to record

- **Pred Diff**: the predictions difference measured with the ℓ_2 norm:

$$\|\widehat{y}_{\mathrm{SA}}(A) - \widehat{y}_{\mathrm{GD}}(x)\|_2$$

  where $\widehat{y}_{\mathrm{SA}}(A)$ corresponds to the $\widetilde{b}$ in Theorem 4.2, and $\widehat{y}_{\mathrm{GD}}(x)$ corresponds to the $\widetilde{b}$ in Theorem 4.3.
- **Model Cos**: the cosine similarity between the sensitivities of two models:

$$\texttt{CosSim}(\frac{\partial \widehat{y}_{\mathrm{GD}}(x)}{\partial x}, \frac{\partial \widehat{y}_{\mathrm{SA}}(A)}{\partial A})$$

- **Model Diff**: the model sensitivity difference measured with the ℓ_2 norm:

$$\|\frac{\partial \widehat{y}_{\mathrm{GD}}(x)}{\partial x} - \frac{\partial \widehat{y}_{\mathrm{SA}}(A)}{\partial A}\|_2$$

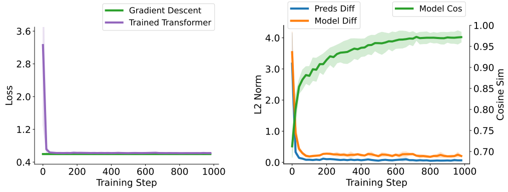
All experiments run on a single NVIDIA RTX2080Ti GPU with 10 independent repetitions.

5.2 Different Document Lengths

For synthetic softmax regression tasks of document length $n \in \{200, 1000\}$ and word embedding size $d = 20$, the comparison results between a trained single self-attention layer and one-step gradient descent are shown in Figure 1 and Figure 2. Due to space limitation, we present more results with different document length $n \in \{25, 50, 100, 200, 400, 1000\}$ in Appendix E.

We compare two models' losses over training steps of Transformers in Figure 1a and Figure 2a. In Figure 1b and Figure 2b, we show the differences and similarities of two models over the training steps. From the results, we find identical performances of the two models measured in losses. We also observe considerable alignment of the two models across tasks of different document lengths, indicated by decreasing prediction and model difference and increasing cosine similarity between models. Besides, comparing results with different n, we observe that with larger document length, which is common in practical NLP tasks, more training steps are required for Transformers to exhibit such similarities. This shows the crucial role of pretraining stage of Transformers for their in-context learning ability.

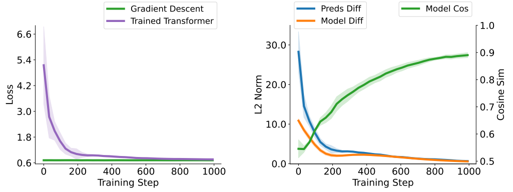
https://doi.org/10.52202/079017-2001



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

Figure 1: Comparison on softmax regression tasks of document length $n = 200$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

Figure 2: Comparison on softmax regression tasks of document length $n = 1000$.

5.3 Different Word Embedding Sizes

We also compare trained single self-attention layer and one-step gradient descent on synthetic softmax regression tasks of different word embedding sizes and document length $n = 200$. Similarly, we measure two models' losses and similarities over training steps on each set of tasks. Due to space limitation, we follow [ONR+22] to show in Figure 3 the loss comparisons at the end of training over different embedding size $d \in \{5, 10, 20, 35, 50\}$. The complete loss curves and measurements of model difference and similarity are presented in Appendix E.

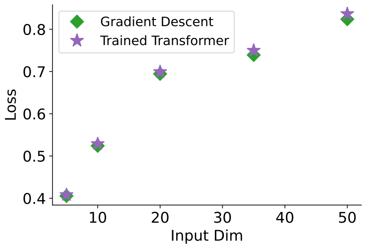


Figure 3: Loss comparisons with different word embedding sizes d.

From the results, we again observe similar performances and close alignment of the two models with different word embedding sizes.

To summarize, our numerical results validate our theoretical results in Section 4, showing that when training self-attention-only Transformers for softmax regression tasks, the models learned by gradient-descent and Transformers show great similarity. Note that due to the non-linearity of softmax regression, it is not expected for models to match exactly as implied in our theoretical results in Section 4, which is also observed in our numerical findings.

6 Conclusion

The attention mechanism that incorporates the softmax unit is a crucial aspect of Large Language Models (LLMs) and significantly contributes to their extraordinary performance in various Natural Language Processing (NLP) tasks. The ability to learn in-context is highly valued in recent LLMs, and comprehending this concept is vital when querying LLMs. In this study, taking a step further from prior works' studies on linear Transformer's ability of learning linear functions, we examined the in-context learning process from a softmax regression perspective of Transformer's attention mechanism. We established the bound on the data transformations brought about by a single self-attention layer with softmax unit and gradient descent on an L2 regression loss. Our findings suggest that the update acquired through gradient descent and in-context learning are highly similar when training self-attention-only Transformers for softmax regression tasks, which is also validated through our preliminary experimental results. These results offer insights into the theoretical underpinnings of in-context learning in Transformers and can aid in improving the understanding and performance of LLMs in various NLP tasks.

Acknowledgments

The authors would like to thank Jerry Yao-Chieh Hu, Zhenmei Shi, Lichen Zhang and Yufa Zhou for helping preparing for camera-ready version of this paper. For more information related to the paper and adjacent topics, see https://www.youtube.com/@zhaosong2031 and https://space.bilibili.com/3546587376650961.

References

[Ano24a] Anonymous. Fundamental limits of prompt tuning transformers: Universality, capacity and efficiency. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. under review.

[Ano24b] Anonymous. On statistical rates of conditional diffusion transformer: Approximation and estimation. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. under review.

[AS23] Josh Alman and Zhao Song. Fast attention requires bounded entries. In *NeurIPS*, 2023.

[AS24a] Josh Alman and Zhao Song. Fast rope attention: Combining the polynomial method and fast fourier transform. In *manuscript*, 2024.

[AS24b] Josh Alman and Zhao Song. The fine-grained complexity of gradient computation for training large language models. In *NeurIPS*. arXiv preprint arXiv:2402.04497, 2024.

[AS24c] Josh Alman and Zhao Song. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *ICLR*, 2024.

[ASA$^+$22] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[BAG20] Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7096–7116, Online, November 2020. Association for Computational Linguistics.

[Bel22] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022.

[BMR$^+$20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[BP66] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[BPG20] Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 455–475, Online, November 2020. Association for Computational Linguistics.

[BSZ24] Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. In *ICML*. arXiv preprint arXiv:2304.02207, 2024.

[CDL$^+$22] Beidi Chen, Tri Dao, Kaizhao Liang, Jiaming Yang, Zhao Song, Atri Rudra, and Christopher Re. Pixelated butterfly: Simple and efficient sparse training for neural network models. In *International Conference on Learning Representations (ICLR)*, 2022.

[CDW$^+$21] Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17413–17426, 2021.

[CGRS19] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

https://doi.org/10.52202/079017-2001

[CLL$^+$24] Bo Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Bypassing the exponential dependency: Looped transformers efficiently learn in-context by multi-step gradient descent. *arXiv preprint arXiv:2410.11268*, 2024.

[CLS$^+$24] Bo Chen, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Hsr-enhanced sparse attention acceleration. *arXiv preprint arXiv:2410.10165*, 2024.

[CND$^+$22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[DCL$^+$21] Tri Dao, Beidi Chen, Kaizhao Liang, Jiaming Yang, Zhao Song, Atri Rudra, and Christopher Re. Pixelated butterfly: Simple and efficient sparse training for neural network models. *arXiv preprint arXiv:2112.00029*, 2021.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DGS23] Yichuan Deng, Yeqi Gao, and Zhao Song. Solving tensor low cycle rank approximation. *arXiv preprint arXiv:2304.06594*, 2023.

[DGV$^+$18] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

[DKOD20] Giannis Daras, Nikita Kitaev, Augustus Odena, and Alexandros G Dimakis. Smyrf-efficient attention using asymmetric clustering. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6476–6489, 2020.

[DLS23] Yichuan Deng, Zhihang Li, and Zhao Song. Attention scheme inspired softmax regression. *arXiv preprint arXiv:2304.10411*, 2023.

[DMS23] Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arxiv preprint: arxiv 2304.03426*, 2023.

[EGKZ21] Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. *arXiv preprint arXiv:2110.10090*, 2021.

[EGZ20] Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4301–4306, Online, November 2020. Association for Computational Linguistics.

[GHG$^+$20] Peng Gao, Chiori Hori, Shijie Geng, Takaaki Hori, and Jonathan Le Roux. Multi-pass transformer for machine translation. *arXiv preprint arXiv:2009.11382*, 2020.

[GMS23] Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression. *arXiv preprint arXiv:2303.16504*, 2023.

[GSYZ24] Yeqi Gao, Zhao Song, Xin Yang, and Yufa Zhou. Differentially private attention computation. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[GTLV22] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint arXiv:2208.01066*, 2022.

[HCL$^+$24] Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Haozheng Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

https://doi.org/10.52202/079017-2001

[HCW$^+$24] Jerry Yao-Chieh Hu, Bo-Yu Chen, Dennis Wu, Feng Ruan, and Han Liu. Nonparametric modern hopfield models. *arXiv preprint arXiv:2404.03900*, 2024.

[HL19] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics.

[HLSL24] Jerry Yao-Chieh Hu, Thomas Lin, Zhao Song, and Han Liu. On computational limits of modern hopfield models: A fine-grained complexity analysis. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[HWL24a] Jerry Yao-Chieh Hu, Dennis Wu, and Han Liu. 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)*, 2024.

[HWL$^+$24b] Jerry Yao-Chieh Hu, Weimin Wu, Zhuoru Li, Sophia Pi, , Zhao Song, and Han Liu. On statistical rates and provably efficient criteria of latent diffusion transformers (dits). In *Thirty-eighth Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[HYW$^+$23] Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[KKL20] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[KS23] Tokio Kajitsuka and Issei Sato. Are transformers with one layer self-attention using low-rank weight matrices universal approximators? *arXiv preprint arXiv:2307.14023*, 2023.

[KVPF20] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[LLS$^+$24a] Chenyang Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Tianyi Zhou. Fourier circuits in neural networks and transformers: A case study of modular arithmetic with multiple inputs. *arXiv preprint arXiv:2402.09469*, 2024.

[LLS$^+$24b] Xiaoyu Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Fine-grained attention i/o complexity: Comprehensive analysis for backward passes. *arXiv preprint arXiv:2410.09397*, 2024.

[LLS$^+$24c] Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, Zhuoyan Xu, and Junze Yin. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *arXiv preprint arXiv:2405.05219*, 2024.

[LLS$^+$24d] Yingyu Liang, Jiangxuan Long, Zhenmei Shi, Zhao Song, and Yufa Zhou. Beyond linear approximations: A novel pruning approach for attention matrix. *arXiv preprint arXiv:2410.11261*, 2024.

[LLSS24a] Chenyang Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Exploring the frontiers of softmax: Provable optimization, applications in diffusion model, and beyond. *arXiv preprint arXiv:2405.03251*, 2024.

[LLSS24b] Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. A tighter complexity analysis of sparsegpt. *arXiv preprint arXiv:2408.12151*, 2024.

[LSS$^+$24a] Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Looped relu mlps may be all you need as practical programmable computers. *arXiv preprint arXiv:2410.09375*, 2024.

https://doi.org/10.52202/079017-2001

[LSS$^+$24b] Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Multi-layer transformers gradient can be approximated in almost linear time. *arXiv preprint arXiv:2408.13233*, 2024.

[LSSY24] Yingyu Liang, Zhenmei Shi, Zhao Song, and Chiwun Yang. Toward infinite-long prefix in transformer. *arXiv preprint arXiv:2406.14036*, 2024.

[LSSZ24a] Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Differential privacy of cross-attention with provable guarantee. *arXiv preprint arXiv:2407.14717*, 2024.

[LSSZ24b] Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Tensor attention training: Provably efficient learning of higher-order transformers. *arXiv preprint arXiv:2405.16411*, 2024.

[LSX$^+$22] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), 2022.

[LSZ23] Zhihang Li, Zhao Song, and Tianyi Zhou. Solving regularized exp, cosh and sinh regression problems. *arXiv preprint, 2303.15725*, 2023.

[LWD$^+$23] Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja vu: Contextual sparsity for efficient llms at inference time. In *Manuscript*, 2023.

[ONR$^+$22] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

[PCR19] Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. Fully quantized transformer for machine translation. *arXiv preprint arXiv:1910.10485*, 2019.

[PMB19] Jorge Pérez, Javier Marinković, and Pablo Barceló. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429*, 2019.

[RNS$^+$18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[SMN$^+$24] Zhenmei Shi, Yifei Ming, Xuan-Phi Nguyen, Yingyu Liang, and Shafiq Joty. Discovering the gems in early layers: Accelerating long-context llms with 1000x input token reduction. *arXiv preprint arXiv:2409.17422*, 2024.

[SSZ$^+$24a] Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Yanyu Li, Yifan Gong, Kai Zhang, Hao Tan, Jason Kuen, Henghui Ding, Zhihao Shu, Wei Niu, Pu Zhao, Yanzhi Wang, and Jiuxiang Gu. Lazydit: Lazy learning for the acceleration of diffusion transformers, 2024.

[SSZ$^+$24b] Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Jing Liu, Ruiyi Zhang, Ryan A. Rossi, Hao Tan, Tong Yu, Xiang Chen, Yufan Zhou, Tong Sun, Pu Zhao, Yanzhi Wang, and Jiuxiang Gu. Numerical pruning for efficient autoregressive models, 2024.

[SZKS21] Charlie Snell, Ruiqi Zhong, Dan Klein, and Jacob Steinhardt. Approximating how single head attention learns. *arXiv preprint arXiv:2103.07601*, 2021.

[TDP19] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[VB19] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy, August 2019. Association for Computational Linguistics.

https://doi.org/10.52202/079017-2001

[VBC20] James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WCM21] Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *arXiv preprint arXiv:2107.13163*, 2021.

[WHHL24] Dennis Wu, Jerry Yao-Chieh Hu, Teng-Yun Hsiao, and Han Liu. Uniform memory retrieval with larger capacity for modern hopfield models. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[WHL+24] Dennis Wu, Jerry Yao-Chieh Hu, Weijian Li, Bo-Yu Chen, and Han Liu. STanhop: Sparse tandem hopfield model for memory-enhanced time series prediction. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[WLK+20] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[XHH+24] Chenwei Xu, Yu-Chao Huang, Jerry Yao-Chieh Hu, Weijian Li, Ammar Gilani, Hsi-Sheng Goan, and Han Liu. Bishop: Bi-directional cellular learning for tabular data with generalized sparse modern hopfield model. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[XRLM21] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

[XSL24] Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. Do large language models have compositional ability? an investigation into limitations and scalability. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

[YBR+20] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.

[YPPN21] Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3770–3785, Online, August 2021. Association for Computational Linguistics.

[ZBB+22] Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task, 2022.

[ZHDK23] Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. *arXiv preprint arXiv:2302.02451*, 2023.

[ZKV+20] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

[ZPGA23] Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word? *arXiv preprint arXiv:2303.08117*, 2023.

Appendix

Roadmap. In Section A, we introduce some related works. In Section B, we compute the Lipschitz with respect to x. In Section C, we give some definitions related to the softmax function of A. In Section D, we compute the Lipschitz with respect to A. In Section E, we show our complete numerical experiments that support our theoretical results.

A Related Work

A.1 In-Context Learning

[ASA$^+$22] indicate that Transformer-based in-context learners are able to perform traditional learning algorithms implicitly. This is achieved by encoding smaller models within their internal activations. These smaller models are updated by the given context. They theoretically investigate the learning algorithms that Transformer decoders can implement. They demonstrate that Transformers need only a limited number of layers and hidden units to implement various linear regression algorithms. For d-dimensional regression problems, a $O(d)$-hidden-size Transformer can perform a single step of gradient descent. They also demonstrate its ability to update a ridge regression problem. The study reveals that Transformers theoretically have the ability to perform multiple linear regression algorithms.

[GTLV22] concentrate on training Transformer to learn certain functions, under in-context conditions. The goal is to have a more comprehensive understanding of in-context learning and determine if Transformers can learn the majority of functions within a given class after training. They found that in-context learning is possible even when there is a distribution shift between training and inference data or between in-context examples and query inputs. In addition, they find out that Transformers can learn more complex function classes such as sparse linear functions, two-layer neural networks, and decision trees. These trained Transformers have comparable performance to task-specific learning algorithms.

[ONR$^+$22] demonstrate the similarity between the training process of Transformers in in-context tasks and some meta-learning formulations based on gradient descent. During training Transformers for auto-regressive tasks, the implementation of in-context learning in the Transformer forward pass is carried out through gradient-based optimization of an implicit auto-regressive inner loss that is constructed from the in-context data.

Formally speaking, they consider the following problem $\min_x \|Ax - b\|_2$ defined in Definition 1.2. They show that one step of gradient descent carries out data transformation as follows:

$$\begin{aligned}\|A(x + \delta_x) - b\|_2 &= \|Ax - (b - \delta_b)\|_2 \\ &= \|Ax - \widetilde{b}\|_2\end{aligned}$$

where δ_x denotes the one-step gradient descent on x and δ_b denotes the corresponding data transformation on b. They also show that a self-attention layer is in principle capable of exploiting statistics in the current training data samples. Concretely, let $Q, K, V \in \mathbb{R}^{d \times d}$ denotes the weights for the query matrix, key matrix, and value matrix respectively. The linear self-attention layer updates an input sample by doing the following data transformation:

$$\widehat{b}_j = b_j + PVK^\top Q_j$$

where $\widehat{b}$ denotes the updated b and P denotes the projection matrix such that a Transformer step $\widehat{b}_j$ on every j is identical to the gradient-induced dynamics $\widetilde{b}_j$. This equivalence implies that when training linear-self-attention-only Transformers for fundamental regression tasks, the models learned by GD and Transformers show great similarity.

[XRLM21] explores the occurrence of in-context learning during pre-training when documents exhibit long-range coherence. The Language Model (LLM) develops the ability to generate coherent next tokens by deducing a latent document-level concept. During testing, in-context learning is observed when the LLM deduces a shared latent concept between examples in a prompt. They demonstrate that in-context learning happens even when there is a distribution mismatch between prompts and pretraining data, especially when the pretraining distribution is a mixture of Hidden

https://doi.org/10.52202/079017-2001

Markov Models [BP66]. Theoretically, they show that the error of the in-context predictor is optimal when a distinguishability condition holds. In cases where this condition does not hold, the expected error still reduces as the length of each example increases. This finding highlights the importance of both input and input-output mapping for in-context learning.

A.2 Transformer Theory

The advancements of Transformers have been noteworthy, however, their learning mechanisms are not completely comprehensible yet. Although these models have performed remarkably well in structured and reasoning activities, our comprehension of their mathematical foundations lags significantly behind. Past research has indicated that the outstanding performance of Transformer-based models can be attributed to the information within their components, such as multi-head attention. Various studies [TDP19, VB19, HL19, Bel22, LLS$^+$24a, XSL24] have presented empirical proof that these components carry a substantial amount of information, which can help resolve different probing tasks.

Recent research has investigated the potential of Transformers through both theoretical and experimental methods, including Turing completeness [BPG20], function approximation [YBR$^+$20, CDW$^+$21], formal language representation [BAG20, EGZ20, YPPN21], and abstract algebraic operation learning [ZBB$^+$22]. Some of these studies have indicated that Transformers may act as universal approximators for sequence-to-sequence operations [YBR$^+$20, KS23, Ano24a] and emulate Turing machines [PMB19, BPG20]. [LWD$^+$23] demonstrate the existence of contextual sparsity in LLM, which can be accurately predicted. They exploit the sparsity to speed up LLM inference without degrading the performance from both a theoretical perspective and an empirical perspective. [DCL$^+$21] proposed the Pixelated Butterfly model that uses a simple fixed sparsity pattern to speed up the training of Transformer. Other studies have focused on the expressiveness of attention within Transformers [DGV$^+$18, VBC20, ZKV$^+$20, EGKZ21, SZKS21, WCM21, LSSY24, LSS$^+$24a] and differentially private attention mechanisms [GSYZ24, LSSZ24a]. Recently, modern Hopfield models [HYW$^+$23, HLSL24, WHHL24, HCL$^+$24, HWL24a, HCW$^+$24] have introduced Hopfield layers as powerful alternatives for transformer attention, offering solid theoretical guarantees and strong empirical performance [XHH$^+$24, WHL$^+$24]. Additionally, the statistical and computational theory of transformer-based diffusion models, specifically Diffusion Transformers (DiTs), has been studied in depth [HWL$^+$24b, Ano24b].

Furthermore, [ZPGA23] has demonstrated that moderately sized masked language models may effectively parse and recognize syntactic information that helps in the partial reconstruction of a parse tree. Inspired by the language grammar model studied by [ZPGA23], [DGS23] consider the tensor cycle rank approximation problem. [GMS23] consider the exponential regression in neural tangent kernel over-parameterization setting. [LSZ23] studied the computation of regularized version of the exponential regression problem but they ignore the normalization factor. [DLS23] consider the softmax regression which considers the normalization factor compared to exponential regression problems [GMS23, LSZ23]. The majority of LLMs can perform attention computations in an approximate manner during inference, as long as there are sufficient guarantees of precision. This perspective has been studied by various research, including [CGRS19, KKL20, WLK$^+$20, DKOD20, KVPF20, CDW$^+$21, CDL$^+$22, LLSS24b, LLS$^+$24d, LSS$^+$24b, SMN$^+$24, LLS$^+$24c, SSZ$^+$24b, SSZ$^+$24a]. With this in mind, [ZHDK23, AS23, BSZ24, DMS23, HYW$^+$23, LLS$^+$24b, CLS$^+$24, LSSZ24b, HLSL24, HWL$^+$24b, HWL24a] have studied the attention matrix computation from the hardness perspective and developed faster algorithms.

B Lipschitz with respect to x

In Section B.1, we give the preliminary to compute the Lipschitz. In Section B.2, we compute the Lipschitz of function $\exp(Ax)$ with respect to x. In Section B.3, we compute the Lipschitz of the function α with respect to x. In Section B.4, we compute the Lipschitz of function α^{-1} with respect to x.

https://doi.org/10.52202/079017-2001

B.1 Preliminary

We can compute

$$\frac{\mathrm{d}L}{\mathrm{d}x} = g(x)$$

Let $\eta > 0$ denote the learning rate.

We update

$$x_{t+1} = x_t + \eta \cdot g(x_t)$$

Definition B.1. *We define $\delta_b \in \mathbb{R}^n$ to be the vector that satisfies the following conditions*

$$\|\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1}) - b\|_2^2 = \|\langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t) - b + \delta_b\|_2^2$$

Let $\{-1, +1\}^n$ denote a vector that each entry can be either -1 or $+1$. In the worst case, there are 2^n possible solutions, e.g.,

$$(\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1}) - \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t)) \circ \{-1, +1\}^n$$

The norm of all the choices are the same. Thus, it is sufficient to only consider one solution as follows.

Claim B.2. *We can write δ_b as follows*

$$\delta_b = \underbrace{\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1})}_{f(x_{t+1})} - \underbrace{\langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t)}_{f(x_t)}.$$

Proof. The proof directly follows from Definition B.1. □

For convenience, we split δ_b into two terms, and provide the following definitions

Definition B.3. *We define*

$$\delta_{b,1} := (\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} - \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1}) \cdot \exp(Ax_{t+1})$$
$$\delta_{b,2} := \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \cdot (\exp(Ax_{t+1}) - \exp(Ax_t))$$

Thus, we have

Lemma B.4. *We have*

- $$\delta_b = \delta_{b,1} + \delta_{b,2}$$
- *We can rewrite $\delta_{b,1}$ as follows*
$$\delta_{b,1} = (\alpha(x_{t+1})^{-1} - \alpha(x_t)^{-1}) \cdot \exp(Ax_{t+1}),$$
- *We can rewrite $\delta_{b,2}$ as follows*
$$\delta_{b,2} = \alpha(x_t)^{-1} \cdot (\exp(Ax_{t+1}) - \exp(Ax_t)).$$

Proof. We have

$$\begin{aligned}
\delta_b &= \delta_{b,1} + \delta_{b,2} \\
&= \alpha(x_{t+1})^{-1} \exp(Ax_{t+1}) - \alpha(x_t)^{-1} \exp(Ax_{t+1}) + \\
&\quad \alpha(x_t)^{-1} \exp(Ax_{t+1}) - \alpha(x_t)^{-1} \exp(Ax_t) \\
&= \alpha(x_{t+1})^{-1} \exp(Ax_{t+1}) - \alpha(x_t)^{-1} \exp(Ax_t) \\
&= \langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1}) - \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t),
\end{aligned}$$

where the 1st step follows from the definitions of δ_b, the 2nd step follows from the definitions of $\delta_{b,1}$ and $\delta_{b,2}$, the 3rd step follows from simple algebra, the 4th step comes from the definition of α. □

B.2 Lipschitz for function $\exp(Ax)$ with respect to x

Lemma B.5. *If the following conditions holds*

- *Let* $A \in \mathbb{R}^{n \times d}$
- *Let* $\|A(y-x)\|_\infty < 0.01$
- *Let* $\|A\| \leq R$
- *Let* x, y *satisfy that* $\|x\|_2 \leq R$ *and* $\|y\|_2 \leq R$

Then we have

$$\|\exp(Ax) - \exp(Ay)\|_2 \leq 2\sqrt{n} R \exp(R^2) \cdot \|x - y\|_2.$$

Proof. We have

$$\begin{aligned}\|\exp(Ax) - \exp(Ay)\|_2 &\leq \|\exp(Ax)\|_2 \cdot 2\|A(x-y)\|_\infty \\ &\leq \sqrt{n} \cdot \exp(\|Ax\|_2) \cdot 2\|A(x-y)\|_\infty \\ &\leq \sqrt{n} \exp(R^2) \cdot 2\|A(x-y)\|_2 \\ &\leq \sqrt{n} \exp(R^2) \cdot 2\|A\| \cdot \|x-y\|_2 \\ &\leq 2\sqrt{n} R \exp(R^2) \cdot \|x-y\|_2\end{aligned}$$

where the 1st step follows from $\|A(y-x)\|_\infty < 0.01$ and Fact 2.1, the 2nd step comes from Fact 2.1, the 3rd step follows from Fact 2.2, the 4th step follows from Fact 2.2, the last step follows from $\|A\| \leq R$. □

B.3 Lipschitz for function $\alpha(x)$ with respect to x

We state a tool from previous work [DLS23].

Lemma B.6 (Lemma 7.2 in [DLS23])**.** *If the following conditions hold*

- *Let* $\alpha(x)$ *be defined as Definition 3.3*

Then we have

$$|\alpha(x) - \alpha(y)| \leq \|\exp(Ax) - \exp(Ay)\|_2 \cdot \sqrt{n}.$$

B.4 Lipschitz for function $\alpha(x)^{-1}$ with respect to x

We state a tool from previous work [DLS23].

Lemma B.7 (Lemma 7.2 in [DLS23])**.** *If the following conditions hold*

- *Let* $\langle \exp(Ax), \mathbf{1}_n \rangle \geq \beta$
- *Let* $\langle \exp(Ay), \mathbf{1}_n \rangle \geq \beta$

Then, we have

$$|\alpha(x)^{-1} - \alpha(y)^{-1}| \leq \beta^{-2} \cdot |\alpha(x) - \alpha(y)|.$$

C Softmax Function with respect to A

In this section, we consider the function with respect to A. We define function softmax f as follows

Definition C.1 (Function f, Reparameterized x by A in Definition 3.1)**.** *Given a matrix* $A \in \mathbb{R}^{n \times d}$*. Let* $\mathbf{1}_n$ *denote a length-*n *vector that all entries are ones. We define prediction function* $f : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ *as follows*

$$f(A) := \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \cdot \exp(Ax).$$

https://doi.org/10.52202/079017-2001

Similarly, we reparameterized x by A for our loss function L. We define loss function L as follows

Definition C.2 (Loss function $L_{\exp}$, Reparameterized x by A in Definition 3.2)**.** *Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^{n \times d}$. We define loss function $L_{\exp} : \mathbb{R}^{n \times d} \to \mathbb{R}$ as follows*

$$L_{\exp}(A) := 0.5 \cdot \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2^2.$$

For convenience, we define two helpful notations α and c with respect to A as follows:

Definition C.3 (Normalized coefficients, Reparameterized x by A in Definition 3.3)**.** *We define $\alpha : \mathbb{R}^{n \times d} \to \mathbb{R}$ as follows*

$$\alpha(A) := \langle \exp(Ax), \mathbf{1}_n \rangle.$$

Then, we can rewrite $f(A)$ (see Definition C.1) and $L_{\exp}(A)$ (see Definition C.2) as follows

- $f(A) = \alpha(A)^{-1} \cdot \exp(Ax)$.
- $L_{\exp}(A) = 0.5 \cdot \| \alpha(A)^{-1} \cdot \exp(Ax) - b \|_2^2$.
- $L_{\exp}(A) = 0.5 \cdot \| f(A) - b \|_2^2$.

Definition C.4 (Reparameterized x by A in Definition 3.4)**.** *We define function $c : \mathbb{R}^{n \times d} \in \mathbb{R}^n$ as follows*

$$c(A) := f(A) - b.$$

Then we can rewrite $L_{\exp}(A)$ (see Definition C.2) as follows

- $L_{\exp}(A) = 0.5 \cdot \| c(A) \|_2^2$.

D Lipschitz with respect to A

In Section D.1, we give the preliminary to compute the Lipschitz. In Section D.2, we show the upper bound of δ_b with respect to A. In Section D.3, we compute the Lipschitz of function $\exp(Ax)$ with respect to A. In Section D.4, we compute the Lipschitz of the function α with respect to A. In Section D.5, we compute the Lipschitz of function α^{-1} with respect to A.

D.1 Preliminary

We define δ_b as follows

Definition D.1 (Reparameterized x by A in Definition B.1)**.** *We define $\delta_b \in \mathbb{R}^n$ to be the vector that satisfies the following conditions*

$$\| \langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} \exp(A_{t+1}x) - b \|_2^2 = \| \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \exp(A_t x) - b + \delta_b \|_2^2$$

Claim D.2 (Reparameterized x by A in Definition B.2)**.** *We can write δ_b as follows*

$$\delta_b = \underbrace{\langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} \exp(A_{t+1}x)}_{f(A_{t+1})} - \underbrace{\langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \exp(A_t x)}_{f(A_t)}.$$

Proof. The proof directly follows from Definition D.1. □

For convenient, we split δ_b into two terms, and provide the following definitions

Definition D.3 (Reparameterized x by A in Definition B.3)**.** *We define*

$$\begin{aligned}
\delta_{b,1} &:= (\langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} - \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1}) \cdot \exp(A_{t+1}x) \\
\delta_{b,2} &:= \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \cdot (\exp(A_{t+1}x) - \exp(A_t x))
\end{aligned}$$

Thus, we have

Lemma D.4 (Reparameterized x by A in Lemma B.4)**.** *We have*

https://doi.org/10.52202/079017-2001

- *We can rewrite* $\delta_b \in \mathbb{R}^n$ *as follows*

$$\delta_b = \delta_{b,1} + \delta_{b,2}$$

- *We can rewrite* $\delta_{b,1} \in \mathbb{R}^n$ *as follows*

$$\delta_{b,1} = (\alpha(A_{t+1})^{-1} - \alpha(A_t)^{-1}) \cdot \exp(A_{t+1}x),$$

- *We can rewrite* $\delta_{b,2} \in \mathbb{R}^n$ *as follows*

$$\delta_{b,2} = \alpha(A_t)^{-1} \cdot (\exp(A_{t+1}x) - \exp(A_t x)).$$

Proof. We have

$$\begin{aligned}
\delta_b = & ~ \delta_{b,1} + \delta_{b,2} \\
= & ~ \alpha(A_{t+1})^{-1} \exp(A_{t+1}x) - \alpha(A_t)^{-1} \exp(A_{t+1}x) + \\
& ~ \alpha(A_t)^{-1} \exp(A_{t+1}x) - \alpha(A_t)^{-1} \exp(A_t x) \\
= & ~ \alpha(A_{t+1})^{-1} \exp(A_{t+1}x) - \alpha(A_t)^{-1} \exp(A_t x) \\
= & ~ \langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} \exp(A_{t+1}x) - \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \exp(A_t x),
\end{aligned}$$

where the 1st step follows from the definitions of δ_b, the 2nd step follows from the definitions of $\delta_{b,1}$ and $\delta_{b,2}$, the 3rd step comes from simple algebra, the 4th step comes from the definition of α. □

D.2 Upper Bounding δ_b with respect to A

We can show that

Lemma D.5 (Reparameterized x by A in Lemma 4.1)**.** *If the following conditions hold*

- *Let* $\beta \in (0, 1)$.
- *Let* $\delta_{b,1} \in \mathbb{R}^n$ *be defined as Definition D.3.*
- *Let* $\delta_{b,2} \in \mathbb{R}^n$ *be defined as Definition D.3.*
- *Let* $\delta_b = \delta_{b,1} + \delta_{b,2}$.
- *Let* $R \geq 4$.

We have

- *Part 1.*

$$\|\delta_{b,1}\|_2 \leq 2\beta^{-2} n^{1.5} \exp(2R^2) \cdot \|A_{t+1} - A_t\|_2$$

- *Part 2.*

$$\|\delta_{b,2}\|_2 \leq 2\beta^{-1} \sqrt{n} R \exp(R^2) \cdot \|A_{t+1} - A_t\|_2$$

- *Part 3.*

$$\| \underbrace{f(A_{t+1}) - f(A_t)}_{\delta_b} \|_2 \leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|_2$$

Proof. **Proof of Part 1.** We have

$$\begin{aligned}
\|\delta_{b,1}\|_2 \leq & ~ |\alpha(A_{t+1})^{-1} - \alpha(A_t)^{-1}| \cdot \| \exp(A_{t+1}x)\|_2 \\
\leq & ~ |\alpha(A_{t+1})^{-1} - \alpha(A_t)^{-1}| \cdot \sqrt{n} \cdot \exp(R^2) \\
\leq & ~ \beta^{-2} \cdot |\alpha(A_{t+1}) - \alpha(A_t)| \cdot \sqrt{n} \cdot \exp(R^2) \\
\leq & ~ \beta^{-2} \cdot \sqrt{n} \cdot \| \exp(A_{t+1}x) - \exp(A_t x)\|_2 \cdot \sqrt{n} \cdot \exp(R^2) \\
\leq & ~ \beta^{-2} \cdot \sqrt{n} \cdot 2\sqrt{n} R \exp(R^2) \|A_{t+1} - A_t\| \cdot \sqrt{n} \cdot \exp(R^2)
\end{aligned}$$

$$= 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|$$

where the first step follows from definition, the second step follows from assumption on A and x, the third step follows Lemma D.8, the forth step follows from Lemma D.7, the fifth step follows from Lemma D.6.

Proof of Part 2.

We have

$$\begin{aligned}
\|\delta_{b,2}\|_2 &\leq |\alpha(A_{t+1})^{-1}| \cdot \| \exp(A_{t+1}x) - \exp(A_t x)\|_2 \\
&\leq \beta^{-1} \cdot \| \exp(A_{t+1}x) - \exp(A_t x)\|_2 \\
&\leq \beta^{-1} \cdot 2\sqrt{n} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|
\end{aligned}$$

Proof of Part 3.

We have

$$\begin{aligned}
\|\delta_b\|_2 &= \|\delta_{b,1} + \delta_{b,2}\|_2 \\
&\leq \|\delta_{b,1}\|_2 + \|\delta_{b,2}\|_2 \\
&\leq 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| + 2\beta^{-1} n^{0.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| \\
&\leq 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| + 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| \\
&\leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|
\end{aligned}$$

where the 1st step follows from the definition of δ_b, the 2nd step comes from triangle inequality, the 3rd step comes from the results in Part 1 and Part 2, the 4th step follows from the fact that $n \geq 1$ and $\beta^{-1} \geq 1$, the 5th step follows from simple algebra. □

D.3 Lipschitz for function $\exp(Ax)$ with respect to A

Lemma D.6 (Reparameterized x by A in Lemma B.5). *If the following conditions holds*

- *Let* $A, B \in \mathbb{R}^{n \times d}$
- *Let* $\|(A - B)x\|_\infty < 0.01$
- *Let* $\|A\| \leq R$
- *Let x satisfy that* $\|x\|_2 \leq R$

Then we have

$$\| \exp(Ax) - \exp(Bx)\|_2 \leq 2\sqrt{n} R \exp(R^2) \cdot \|A - B\|.$$

Proof. We have

$$\begin{aligned}
\| \exp(Ax) - \exp(Bx)\|_2 &\leq \| \exp(Ax)\|_2 \cdot 2\|(A - B)x\|_\infty \\
&\leq \sqrt{n} \cdot \exp(\|Ax\|_2) \cdot 2\|(A - B)x\|_\infty \\
&\leq \sqrt{n} \exp(R^2) \cdot 2\|(A - B)x\|_2 \\
&\leq \sqrt{n} \exp(R^2) \cdot 2\|A - B\| \cdot \|x\|_2 \\
&\leq 2\sqrt{n} R \exp(R^2) \cdot \|A - B\|
\end{aligned}$$

where the 1st step follows from $\|A(y - x)\|_\infty < 0.01$ and Fact 2.1, the 2nd step follows from Fact 2.1, the 3rd step follows from Fact 2.2, the 4th step comes from Fact 2.2, the last step follows from $\|A\| \leq R$. □

D.4 Lipschitz for function $\alpha(A)$ with respect to A

Lemma D.7 (Reparameterized x by A in Lemma B.6). *If the following conditions hold*

- *Let $\alpha(A)$ be defined as Definition C.3*

Then we have

$$|\alpha(A) - \alpha(B)| \le \|\exp(Ax) - \exp(Bx)\|_2 \cdot \sqrt{n}.$$

Proof. We have

$$\begin{aligned}|\alpha(A) - \alpha(B)| &= |\langle \exp(Ax) - \exp(Bx), \mathbf{1}_n \rangle| \\ &\le \|\exp(Ax) - \exp(Bx)\|_2 \cdot \sqrt{n}\end{aligned}$$

where the 1st step comes from the definition of $\alpha(x)$, the 2nd step follows from Cauchy-Schwarz inequality (Fact 2.1). □

D.5 Lipschitz for function $\alpha(A)^{-1}$ with respect to A

Lemma D.8 (Reparameterized x by A in Lemma B.7)**.** *If the following conditions hold*

- *Let* $\langle \exp(Ax), \mathbf{1}_n \rangle \ge \beta$
- *Let* $\langle \exp(Bx), \mathbf{1}_n \rangle \ge \beta$

Then, we have

$$|\alpha(A)^{-1} - \alpha(B)^{-1}| \le \beta^{-2} \cdot |\alpha(A) - \alpha(B)|.$$

Proof. We can show that

$$\begin{aligned}|\alpha(A)^{-1} - \alpha(B)^{-1}| &= \alpha(A)^{-1}\alpha(B)^{-1} \cdot |\alpha(A) - \alpha(B)| \\ &\le \beta^{-2} \cdot |\alpha(A) - \alpha(B)|\end{aligned}$$

where the 1st step follows from simple algebra, the 2nd step follows from $\alpha(A) \ge \beta, \alpha(B) \ge \beta$. □

E Experiments

In this section, we show the complete numerical experimental results supporting our theoretical results that when training self-attention-only Transformers for softmax regression tasks, the models learned by gradient-descent and Transformers show great similarity.

Experiments setup. According to Definition 1.3, we construct the synthetic softmax regression tasks consists of randomly sampled length-n documents $A \in \mathbb{R}^{n \times d}$ where each word has the d-dimensional embedding and targets $b \in \mathbb{R}^n$. In our experiments we choose a set of different value of document length $n \in \{25, 50, 100, 200, 400\}$ and a set of different embedding size $d \in \{5, 10, 20, 35, 50\}$. Following [ONR+22], we compare the two models in our experiment: a trained single self-attention (SA) layer with a softmax unit approximating the full Transformers, and a softmax regression model trained with one-step gradient descent. The training objective for both models is defined as in Definition 1.3. For the single self-attention layer with a softmax unit, we choose the learning rate $\eta_{\mathrm{SA}} = 0.005$. For the softmax regression model, we determine the optimal learning rate η_{GD} by minimizing the ℓ_2 regression loss over a training set of 10^3 tasks through line search.

To compare the trained single self-attention layer with a softmax unit and the softmax regression model trained with one-step gradient descent, we sample 10^3 tasks and record the losses of two models. In addition, we follow [ONR+22] to record

- **Pred Diff**: the predictions difference measured with the ℓ_2 norm:

  $$\|\widehat{y}_{\mathrm{SA}}(A) - \widehat{y}_{\mathrm{GD}}(x)\|_2$$

  where $\widehat{y}_{\mathrm{SA}}(A)$ is corresponding to the $\widetilde{b}$ in Theorem 4.2, and $\widehat{y}_{\mathrm{GD}}(x)$ is corresponding to the $\widetilde{b}$ in Theorem 4.3.
- **Model Cos**: the cosine similarity between the sensitivities of two models:

  $$\texttt{CosSim}\,(\frac{\partial \widehat{y}_{\mathrm{GD}}(x)}{\partial x}, \frac{\partial \widehat{y}_{\mathrm{SA}}(A)}{\partial A})$$

- **Model Diff**: the model sensitivity difference measured with the ℓ_2 norm:

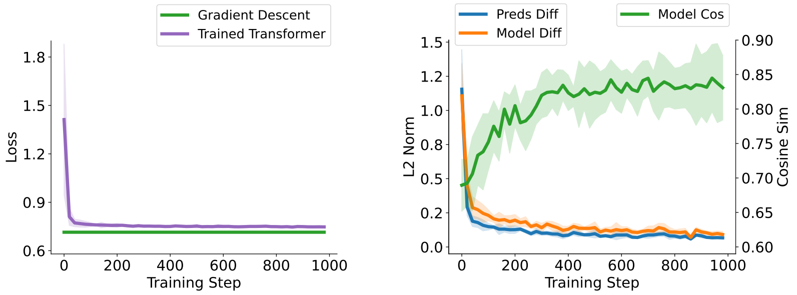
$$\|\frac{\partial \widehat{y}_{\mathrm{GD}}(x)}{\partial x} - \frac{\partial \widehat{y}_{\mathrm{SA}}(A)}{\partial A}\|_2$$

All experiments run on a single NVIDIA RTX2080Ti GPU with 10 independent repetitions.

Results on tasks of different document lengths. The results of the comparisons between a trained single self-attention layer and one-step gradient descent on synthetic softmax regression tasks of document length $n \in \{25, 50, 100, 200, 400, 1000\}$ and word embedding size $d = 20$ are shown in Figure 4-9. We measure two models' losses and similarities over the training steps of the SA layer for each set of tasks. From the results, we observe identical performances of the two models measured in losses. We also observe considerable alignment of the two models across tasks of different document lengths, indicated by decreasing prediction and model difference and increasing cosine similarity between models.

Results on tasks of different word embedding sizes. The results of the comparisons between a trained single self-attention layer and one-step gradient descent on synthetic softmax regression tasks of document length $n = 200$ and word embedding size $d \in \{5, 10, 20, 35, 50\}$ are shown in Figure 7 and 10-13. Similarly, we measure two models' losses and similarities over training steps of the SA layer for each set of tasks. We again observe similar performances and close alignment of the two models.

In conclusion, our experimental results empirically validate our theoretical results in Section 4, showing that when training self-attention-only Transformers for softmax regression tasks, the models learned by gradient-descent and Transformers show great similarity. Due to the non-linearity of softmax regression, it is not expected for models to match exactly as implied in our theoretical results in Section 4, which is also observed in our experimental findings.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

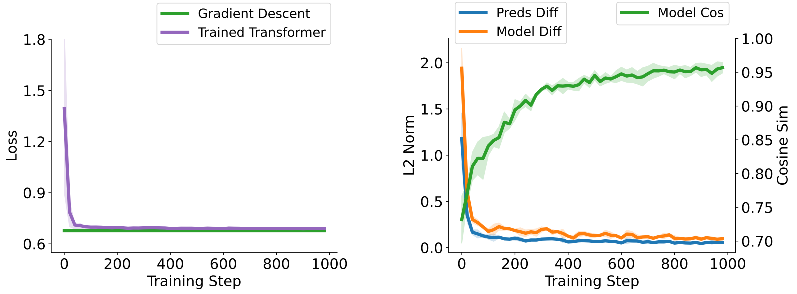
Figure 4: Comparison between trained single-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 25$ and embedding size $d = 20$.

F Limitations

Our findings are restricted to small Transformer and simple regression problems. One interesting direction for further investigation is to acquire a comprehensive perception of in-context learning in larger models. To our best knowledge, we believe this work does not have any negative societal impact.

G Impact Statements

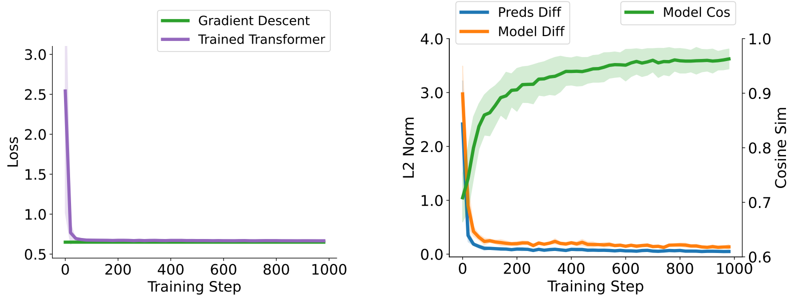
This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

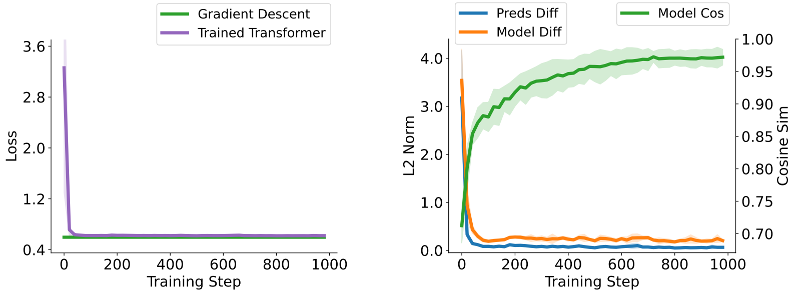
Figure 5: Comparison between trained single-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 50$ and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

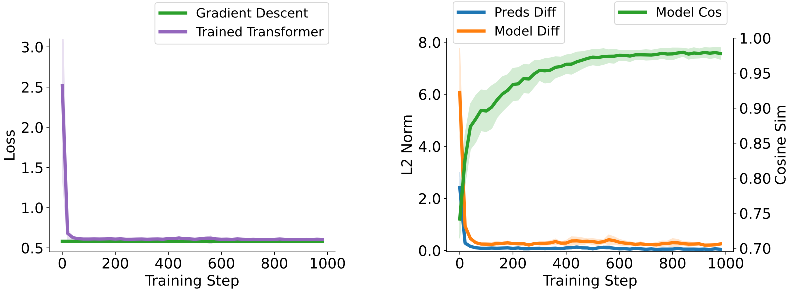
Figure 6: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 100$ and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

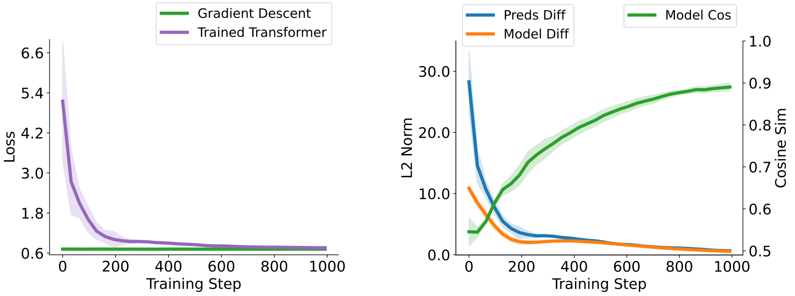
Figure 7: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

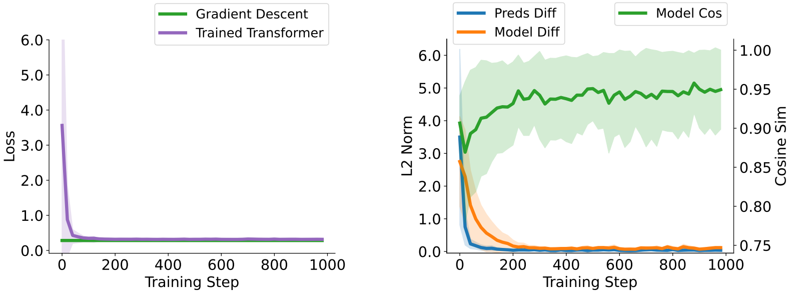
Figure 8: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 400$ and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

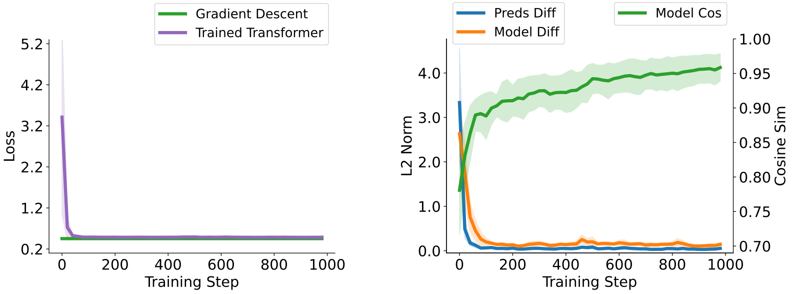
Figure 9: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 1000$ and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

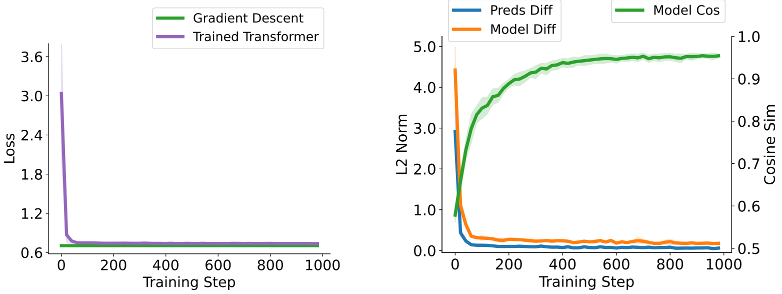
Figure 10: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and embedding size $d = 5$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

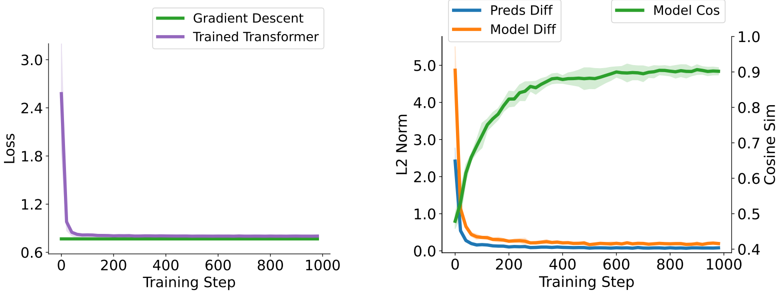
Figure 11: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and embedding size $d = 10$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

Figure 12: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and embedding size $d = 35$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

Figure 13: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and embedding size $d = 50$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose our main results at the end of the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations is discussed in Section F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate ”Limitations” section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-2001

Justification:

- For Theorem 4.2, the proof is in Section 4.2.
- For Theorem 4.3, the proof is in Section 4.3.

We have carefully checked the correctness and the time complexity proof that shown in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setup is described in Section 5.1 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

https://doi.org/10.52202/079017-2001

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The data and code are planned to be released upon acceptance and approval.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is described in Section 5.1 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results plotted in Section 5 and Appendix E include error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer ”Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/079017-2001

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resource information is provided in Section 5.1 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The authors have read the NeurIPS Code of Ethies and made sure the paper followsthe NeurIPS Code of Ethics in every aspect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The potential societal impact is discussed in Section G.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

https://doi.org/10.52202/079017-2001

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release new dataset or model.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets are properly mentioned and cited in Section 5 and Appendix E.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

https://doi.org/10.52202/079017-2001

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-2001