

---

# Bypassing the Exponential Dependency: Looped Transformers Efficiently Learn In-context by Multi-step Gradient Descent

---

Bo Chen<sup>1</sup>

Xiaoyu Li<sup>2</sup>

Yingyu Liang<sup>3,4</sup>

Zhenmei Shi<sup>4</sup>

Zhao Song<sup>5</sup>

<sup>1</sup>Middle Tennessee State University

<sup>2</sup>University of New South Wales

<sup>3</sup>The University of Hong Kong

<sup>4</sup>University of Wisconsin-Madison

<sup>5</sup>The Simons Institute for the Theory of Computing at the University of California, Berkeley

## Abstract

In-context learning has been recognized as a key factor in the success of Large Language Models (LLMs). It refers to the model’s ability to learn patterns on the fly from provided in-context examples in the prompt during inference. Previous studies have demonstrated that the Transformer architecture used in LLMs can implement a single-step gradient descent update by processing in-context examples in a single forward pass. Recent work has further shown that, during in-context learning, a looped Transformer can implement multi-step gradient descent updates in forward passes. However, their theoretical results require an exponential number of in-context examples,  $n = \exp(\Omega(T))$ , where  $T$  is the number of loops or passes, to achieve a reasonably low error. In this paper, we study linear looped Transformers in-context learning on linear vector generation tasks. We show that linear looped Transformers can implement multi-step gradient descent efficiently for in-context learning. Our results demonstrate that as long as the input data has a constant condition number, e.g.,  $n = O(d)$ , the linear looped Transformers can achieve a small error by multi-step gradient descent during in-context learning. Furthermore, our preliminary experiments validate our theoretical analysis. Our findings reveal that the Transformer architecture possesses a stronger in-context learning capability than previously un-

derstood, offering new insights into the mechanisms behind LLMs and potentially guiding the better design of efficient inference algorithms for LLMs.

## 1 INTRODUCTION

Large Language Models (LLMs) have gained immense success and have been widely used in our daily lives, e.g., GPT4 (OpenAI, 2023), Claude 3.5 (Anthropic., 2024), and so on, based on its Transformer architecture (Vaswani et al., 2017). One core emergent ability of LLMs is In-Context Learning (ICL) (Brown et al., 2020). During ICL, the user provides an input sequence (prompts) containing some question-answer pairs as in-context examples and the goal query that the user cares about, where the examples and query are drawn from an unknown task. The LLMs can in-context learn from these examples and generate the correct answer for the goal query without any parameter update. Notably, these unknown tasks may never be seen by LLMs during their pre-training and post-training, e.g., synthetic task (Wei et al., 2023b). Thus, many believe that the in-context learning mechanism is different from supervised learning or unsupervised learning, where the latter may focus on feature learning, while the ICL may perform some algorithm to learn. For instance, the Transformer can implement algorithm selection (Bai et al., 2024) or gradient descent (Dong et al., 2022; Von Oswald et al., 2023) by in-context learning forward pass.

Many works have tried to understand how Transformers perform single-step gradient descent (Zhang et al., 2023c; Ahn et al., 2024; Mahankali et al., 2023; Cheng et al., 2023; Bai et al., 2024; Huang et al., 2023; Li et al., 2023b; Guo et al., 2024; Wu et al., 2024c). They study one-layer linear Transformer in-context learning on linear regression tasks and show that the one-layer linear Transformer can perform single gradient updates

based on the in-context examples during a forward pass. Recent work (Gatmiry et al., 2024) has further shown that a linear looped Transformer can implement multi-step gradient descent updates with multiple forward passes, meaning that Transformers can express multi-step algorithms during in-context learning. However, their theoretical results require an exponential number of in-context examples,  $\exp(\Omega(T))$ , where  $T$  is the number of loops or passes, to achieve a reasonably low error for linear regression tasks. This violates the intuition that more gradient descent updates lead to better performance.

Thus, it is natural to ask the following question:

*Is it necessary to use an exponential number of examples for Transformers to implement multi-step gradient descent during in-context learning?*

In this work, we study linear looped Transformers (Definition 3.6) in-context learning on linear vector generation tasks (Definition 3.4), which is as hard as linear regression. We show that the linear looped Transformer can efficiently perform multi-step gradient descent as long as in-context examples are well-conditioned. We present our main result in the following theorem.

**Theorem 1.1** (Main result. Informal version of Theorem 4.5). *Let  $T$  be the number of loops,  $n$  be the number of in-context examples, and  $d$  be the feature dimension. Let  $(X, y) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$  be the in-context examples. Let  $\kappa$  be the condition number of  $X^\top X$  (Definition 3.11). Then, given a query  $\alpha \in \mathbb{R}$ , the linear looped Transformer can predict a vector output with error  $\leq |\alpha| \cdot \exp(-\frac{T}{2\kappa})$  by explicitly performing multi-step gradient descent in its hidden states.*

In Theorem 1.1, as long as the condition number is constant, we can see that the linear looped Transformer will perform better when the loop number is increasing, i.e., the error will exponentially decay to 0. Informally, for a small constant  $\epsilon$ , if we draw  $X$  from Gaussian distribution, the condition number of  $X^\top X$  will be  $1 \leq \kappa \leq 1 + O(\epsilon)$  when  $n \geq \Omega(d/\epsilon^2)$ . Thus, informally, we only need  $O(d)$  numbers of in-context examples to guarantee a good performance.

Furthermore, our preliminary experiments (Section 6) validate the above arguments and our theoretical analysis. The main intuition of our analysis is that we find that a linear looped Transformer can explicitly perform gradient descent in its hidden states (Lemma 4.1 and Theorem 4.2). Thus, the error analysis can be directly solved by the standard convex optimization technique (Theorem 5.8).

## 1.1 Our Contributions

- We study linear looped Transformers in-context learning on linear vector generation tasks (Definition 3.3), which is as hard as linear regression.
- We find that linear looped Transformers can explicitly perform gradient descent in their hidden states (Lemma 4.1 and Theorem 4.2).
- We demonstrate that linear looped Transformers can efficiently perform multi-step gradient descent as long as in-context examples are well-conditioned, e.g.,  $n = O(d)$ , where the error will exponentially decay to 0 after  $T$  loops (Theorem 4.5).
- Our preliminary experiments on synthetic data validate our main theoretical results (Section 6).

## 1.2 Roadmap

This paper is structured as follows: we begin with a review of related work in Section 2, followed by essential definitions and foundational concepts in Section 3. Section 4 delves into the gradient computation analysis within the Looped Transformer architecture, examining both individual layer computations and the full looped structure. In Section 5, we analyze the error convergence of looped transformers under strong convexity and smoothness conditions, providing an upper bound on error after  $T$  gradient descent iterations for linear vector generation. Section 6 presents our experimental results and findings. Finally, we conclude in Section 7.

## 2 RELATED WORK

This section briefly reviews the related research work on Large Language Models (LLM), In-Context Learning (ICL), and looped transformers. These topics have a close connection to our work.

### 2.1 Large Language Models

Neural networks based on the Transformer architecture (Vaswani et al., 2017) have swiftly become the dominant paradigm in machine learning for natural language processing applications. Expansive Transformer models, trained on diverse and extensive datasets and comprising billions of parameters, are called large language models (LLM) or foundation models (Bommasani et al., 2021). Examples include BERT (Devlin et al., 2019), PaLM (Chowdhery et al., 2022), Llama (Touvron et al., 2023), ChatGPT (OpenAI, 2024), GPT4 (OpenAI, 2023), among others. These LLMs have demonstrated remarkable general intelligence capabilities (Bubeck et al., 2023) across various downstream tasks.

Researchers have developed numerous adaptation techniques to optimize LLM performance for specific applications. These include methods such as adapters (Hu et al., 2022; Zhang et al., 2023b; Gao et al., 2023a; Shi et al., 2023; Cao et al., 2024; Cao, 2024), calibration approaches (Zhao et al., 2021; Zhou et al., 2023a), multitask fine-tuning strategies (Gao et al., 2021a; Xu et al., 2023; Von Oswald et al., 2023; Xu et al., 2024c), prompt tuning techniques (Gao et al., 2021b; Lester et al., 2021), scratchpad approaches (Nye et al., 2021), instruction tuning methodologies (Li and Liang, 2021; Chung et al., 2022; Mishra et al., 2022), symbol tuning (Wei et al., 2023a), black-box tuning (Sun et al., 2022), reinforcement learning from the human feedback (RLHF) (Ouyang et al., 2022), chain-of-thought reasoning (Wei et al., 2022c; Khattab et al., 2022; Yao et al., 2023; Zheng et al., 2024) and various other strategies.

And here are more related works aiming at enhancing model efficiency without compromising performance, such as (Deng et al., 2022; Qin et al., 2023; Shrivastava et al., 2023; Gao et al., 2023b; Deng et al., 2023; Liang et al., 2024b,c,d; Wu et al., 2024a,b; Xu et al., 2024a; Hu et al., 2023, 2024a,b,c,d,e,f; Li et al., 2024c,d, 2025; Song et al., 2023; Shi et al., 2024a; Shen et al., 2024; Cao et al., 2025; Chen et al., 2024a,b,c, 2025a,b,d,c; Zhang et al., 2025; Ke et al., 2025).

## 2.2 In-context Learning

A significant capability that has emerged from Large Language Models (LLMs) is in-context learning (ICL) (Brown et al., 2020; Wei et al., 2022b; Shi et al., 2024b). This feature allows LLMs to generate predictions for new scenarios when provided with a concise set of input-output examples (referred to as a prompt) for a specific task, without requiring any modification to the model’s parameters. ICL has found widespread application across various domains, including reasoning (Zhou et al., 2022), self-correction (Pourreza and Rafiei, 2023), machine translation (Agrawal et al., 2022) and many others.

Numerous studies have focused on enhancing the ICL and zero-shot capabilities of LLMs (Min et al., 2021; Wang et al., 2022; Wei et al., 2022a; Iyer et al., 2022). A substantial body of research has been dedicated to investigating the underlying mechanisms of transformer learning (Geva et al., 2021; Xie et al., 2022; Garg et al., 2022; Jelassi et al., 2022; Arora and Goyal, 2023; Li et al., 2023a,d; Allen-Zhu and Li, 2023; Luo et al., 2023; Tian et al., 2023a,b; Zhou et al., 2023b; Bietti et al., 2023; Xu et al., 2024b; Li et al., 2024a,b; Liang et al., 2024a,f,e) and in-context learning (Dong et al., 2022; Mahankali et al., 2023; Raventos et al., 2023; Bai et al., 2023; Ahn et al., 2023; Pan et al., 2023; Li et al., 2023b,c; Akyurek et al., 2023; Zhang et al., 2023a,c;

Huang et al., 2023; Cheng et al., 2023; Wibisono and Wang, 2023; Wu et al., 2024c; Guo et al., 2024; Reddy, 2024; Shi et al., 2024b; Wu et al., 2024d) through both empirical and theoretical approaches. Building upon these insights, our analysis extends further to elucidate ICL implementing multi-step gradient descent.

## 2.3 Looped Transformer

The concept of recursive inductive bias was first introduced by Dehghani et al. (2018) into Transformers. Looping Transformers are also related to parameter-efficient weight-tying Transformers (Chi et al., 2021; Gatmiry et al., 2024) theoretically showed that the recursive structure of Looped Transformers enables them to function as Turing machines. Yang et al. (2023) demonstrates that increasing the number of loop iterations can improve performance on some tasks. Recent studies (Artur Back and Fountoulakis, 2024; Giannou et al., 2024) have provided theoretical insights into the emulation capabilities of specific algorithms and their convergence during training, with a particular emphasis on in-context learning.

However, the expressive capacity of Looped Transformers or Looped Neural Newrok Liang et al. (2025); Ke et al. (2024) in function approximation and their associated approximation rates remain largely unexplored territories. Yu et al. (2025) studies enhancing autoregressive chain-of-thought through loop-aligned reasoning. In this work, we consider Looped Linear Transformers.

## 3 PRELIMINARY

In this section, we present some preliminary concepts and definitions of our paper. In Section 3.1, we introduce some basic notations used in our paper. In Section 3.2, we defined some important variables to set up our problem.

### 3.1 Notations

For any integer  $n$ , we define  $[n] = \{1, 2, \dots, n\}$ . For two vectors  $x \in \mathbb{R}^n$  and  $y \in \mathbb{R}^n$ , we use  $\langle x, y \rangle$  to denote the inner product between  $x, y$ , i.e.,  $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ . We use  $e_i$  to denote a vector where only  $i$ -th coordinate is 1, and other entries are 0. For each  $a, b \in \mathbb{R}^n$ , we use  $a \circ b \in \mathbb{R}^n$  to denote the Hardamard product, i.e. the  $i$ -th entry of  $(a \circ b)$  is  $a_i b_i$  for all  $i \in [n]$ . We use  $\mathbf{1}_n$  to denote a length- $n$  vector where all the entries are ones. We use  $x_{i,j}$  to denote the  $j$ -th coordinate of  $x_i \in \mathbb{R}^n$ . We use  $\|x\|_p$  to denote the  $\ell_p$  norm of a vector  $x \in \mathbb{R}^n$ , i.e.  $\|x\|_1 := \sum_{i=1}^n |x_i|$ ,  $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$ , and  $\|x\|_\infty := \max_{i \in [n]} |x_i|$ . For  $n > d$ , for any matrix  $A \in \mathbb{R}^{n \times d}$ , we use  $\|A\|$  to denote the spectral norm of  $A$ , i.e.  $\|A\| := \sup_{x \in \mathbb{R}^d} \|Ax\|_2 / \|x\|_2$ . For a matrix  $A$ ,

Table 1: There is a strict correspondence between the symbols we use and the common symbols of the general standard transformer attention. The specific choice of  $Q$  and  $P$  follows the setting in previous work Gatmiry et al. (2024); Huang et al. (2023) for simplicity.

Meaning	Dimension	Standard Notation	Our Notation
Input	$n \times d$	$X$	$Z$
Query-key weight matrix	$d \times d$	$W_Q W_K^\top$	$Q$
Value-output weight matrix	$d \times d$	$W_V W_O^\top$	$P$
Query-key matrix	$n \times n$	$QK^\top = XW_Q W_K^\top X^\top$	$ZQX^\top Z$

we use  $\lambda_{\min}(A)$  and  $\lambda_{\max}(A)$  to denote the minimum and maximum eigenvalue of  $A$ , respectively.

### 3.2 In-context Learning

First, we introduce some definitions of in-context examples and their labels.

**Definition 3.1** (In-context prompt data). *Let  $X \in \mathbb{R}^{n \times d}$  be the input data with  $n$  tokens and each token with feature dimension  $d$ , i.e.,  $X = [x_1, x_2, \dots, x_n]^\top$ , where  $x_i \in \mathbb{R}^d$  for any  $i \in [n]$ .*

**Definition 3.2** (In-context prompt label). *Assume  $\theta^*$  uniformly draw from  $d$  dimension unit sphere  $\mathbb{S}^d$ , where  $\|\theta^*\|_2 = 1$ . Let labels be*

$$y := X\theta^* \in \mathbb{R}^n.$$

Note that the  $\theta^*$  is unseen to the model. Then, our vector generation tasks are defined as follows.

**Definition 3.3** (In-context task). *Given in-context prompt examples/data  $X \in \mathbb{R}^{n \times d}$  with their labels  $y = X\theta^*$ , where  $\theta^*$  is unseen to model, and given a query  $\alpha \neq 0 \in \mathbb{R}$ , the task is to output/generate a vector  $v$  such that  $\langle v, \theta^* \rangle$  is as close to  $\alpha$  as possible. Thus, the prediction error is*

$$|\langle v, \theta^* \rangle - \alpha|.$$

We remark that our vector generation task is as hard as the linear regression task, as they are dual problems. Solving any one of them requires estimating the  $\theta^*$ .

Combining all of the above, we have the ICL prompt/input data for the model.

**Definition 3.4** (Input data). *Let input be*

$$Z^{(0)} := \begin{bmatrix} X & y \\ q^{(0)\top} & \alpha \end{bmatrix} \in \mathbb{R}^{(n+1) \times (d+1)}.$$

In Definition 3.4, the model will iteratively update  $q^{(0)} \in \mathbb{R}^d$  and make it as the generation output. In this work, we initialize  $q^{(0)}$  as  $\mathbf{0}_d$ .

### 3.3 Linear Looped Transformer

In line with recent work by Gatmiry et al. (2024) and Ahn et al. (2024), we consider a linear self-attention model, formally defined as follows:

**Definition 3.5** (Linear attention). *Let  $Q, P \in \mathbb{R}^{d \times d}$  be the query-key matrix and the value-output matrix. Let  $Z \in \mathbb{R}^{n \times d}$  be input data. The linear attention is defined as*

$$\text{Attn}(Z; Q, P) := (M \circ (ZQZ^\top))ZP,$$

where  $M = \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{0}_{n \times 1} \\ \mathbf{1}_{1 \times n} & 0 \end{bmatrix}$  is a causal attention mask for text generation.

In Definition 3.5, we combine the query matrix and key matrix as  $Q$  and combine the value matrix and output matrix as  $P$  for simplicity following previous works (Zhang et al., 2023c; Gatmiry et al., 2024). We refer readers to Table 1 for more details about connection between our notation and standard attention.

Building upon this, we introduce the concept of a Linear Looped Transformer (Gatmiry et al., 2024):

**Definition 3.6** (Linear looped transformer). *Let  $T$  be the loop number. Let  $\eta^{(t)} > 0$  for any  $t \in \{0, 1, \dots, T-1\}$ . The linear looped transformer  $\text{TF}(Z^{(0)}; Q, P)$  is defined as*

$$Z^{(t)} := Z^{(t-1)} - \eta^{(t-1)} \text{Attn}(Z^{(t-1)}; Q, P), \quad \forall t \in [T]$$

$$\text{TF}(Z^{(0)}; Q, P) := -Z_{n+1, 1:d}^{(T)\top}.$$

The Looped Transformer is to simulate a real multiple-layer Transformer with residual connections (He et al., 2016), where  $\eta$  represents the weights of the residual components.

**Remark 3.7.** *Our settings are more practical than Gatmiry et al. (2024) in the following sense.*

- We have a more practical casual attention mask used in generation. Our mask requires Hardamard product, the same as the standard attention, while Gatmiry et al. (2024) uses matrix product mask.



- Our model does not have prior knowledge of  $X$  distribution, while the model in [Gatmiry et al. \(2024\)](#) knows the distribution of  $X$ , i.e., distribution free. In practice, the LLMs do not know any information about in-context examples.

### 3.4 Linear Regression with Gradient Descent

In this section, we introduce some key concepts of linear regression with gradient descent.

**Definition 3.8** (Linear regression). Given  $X \in \mathbb{R}^{n \times d}$ ,  $y \in \mathbb{R}^n$ , and  $\theta \in \mathbb{R}^d$ , the linear regression loss function is defined as

$$\ell(\theta) := 0.5 \|y - X\theta\|_2^2.$$

Then, the gradient formulation is

$$\nabla_\theta \ell(\theta) = X^\top X\theta - X^\top y \in \mathbb{R}^d. \quad (1)$$

For any  $t \in \mathbb{N}_+$ , let  $\eta^{(t)} > 0$  and by Gradient Descent, we have

$$\theta^{(t)} := \theta^{(t-1)} - \eta^{(t-1)} (X^\top X\theta^{(t-1)} - X^\top y). \quad (2)$$

We define the optimizer below.

**Definition 3.9.** Let the optimizer of  $\ell(\theta)$  be

$$\tilde{\theta} = (X^\top X)^{-1} X^\top y.$$

In this work, we consider  $n > d$ , assuming  $X^\top X$  is inevitable. Then, we have  $\theta^* = \tilde{\theta}$ .

**Lemma 3.10** (Forklore). In the realizable setting (no noise term), if  $n > d$ , then  $\theta^* = \tilde{\theta}$ .

Finally, we define the condition number, which will be used in our final convergence bound.

**Definition 3.11.** We define the condition number of input data as

$$\kappa := \frac{\lambda_{\max}(X^\top X)}{\lambda_{\min}(X^\top X)}.$$

## 4 GRADIENT COMPUTATION IN LOOPED TRANSFORMER

In this section, we present a comprehensive analysis of the gradient computation process within the Looped Transformer architecture. Our investigation begins with an examination of computations in individual layers and subsequently extends to the full looped structure. This approach allows us to build a nuanced understanding of the Looped Transformer’s behavior, starting from its fundamental components.

We commence our analysis by establishing a crucial result regarding the output of a single layer in our Looped Transformer model. This foundational lemma serves as a cornerstone for our subsequent derivations and provides valuable insights into the model’s inner workings.

**Lemma 4.1** (Single layer output). Let  $Z^{(0)}$  be defined in Definition 3.4. Let  $Q = I_{d+1, d+1}$ . Let  $P = \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 \end{bmatrix}$ . Let causal attention mask be  $M = \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{0}_{n \times 1} \\ \mathbf{1}_{1 \times n} & 0 \end{bmatrix}$ . Then, we have

$$\text{Attn}(Z^{(0)}; Q, P) = \begin{bmatrix} \mathbf{0}_{n \times d} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top X + \alpha y^\top X & 0 \end{bmatrix}.$$

*Proof.* We can show that

$$\begin{aligned} & \text{Attn}(Z^{(0)}; Q, P) \\ &= (M \circ (Z^{(0)} Q (Z^{(0)\top}))) Z^{(0)} P \\ &= (M \circ \begin{bmatrix} X & y \\ q^{(0)\top} & \alpha \end{bmatrix} \begin{bmatrix} X^\top & q^{(0)} \\ y^\top & \alpha \end{bmatrix}) Z^{(0)} P \\ &= (M \circ \begin{bmatrix} X X^\top + y y^\top & X q^{(0)} + \alpha y \\ q^{(0)\top} X^\top + \alpha y^\top & q^{(0)\top} q^{(0)} + \alpha^2 \end{bmatrix}) Z^{(0)} P \\ &= \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top + \alpha y^\top & 0 \end{bmatrix} Z^{(0)} P \\ &= \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top + \alpha y^\top & 0 \end{bmatrix} \begin{bmatrix} X & y \\ q^{(0)\top} & \alpha \end{bmatrix} P \\ &= \begin{bmatrix} \mathbf{0}_{n \times d} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top X + \alpha y^\top X & q^{(0)\top} X^\top y + \alpha y^\top y \end{bmatrix} P \\ &= \begin{bmatrix} \mathbf{0}_{n \times d} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top X + \alpha y^\top X & q^{(0)\top} X^\top y + \alpha y^\top y \end{bmatrix} \cdot \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_{n \times d} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top X + \alpha y^\top X & 0 \end{bmatrix} \end{aligned}$$

where the first step follows from Definition 3.5, the second step follows from Definition 3.4, and the rest steps directly follow from the matrix multiplication.  $\square$

This result illuminates the specific form of the attention mechanism’s output in a single layer, which is essential for understanding the model’s overall behavior, where the output only has non-zero terms in the position of  $q^{(0)}$  and the format is close to Eq. (1).

Having characterized the behavior of a single layer, we now extend our analysis to encompass the full-looped transformer structure. Turning our attention to the core of our analysis with the groundwork laid for single-layer computations. The following theorem establishes a crucial relationship between the transformer’s output and the iteratively updated parameters:

**Theorem 4.2.** Let  $\alpha \neq 0 \in \mathbb{R}$ . Let  $\theta^{(0)} = -\frac{1}{\alpha}q^{(0)}$ . Let  $\theta^{(i)}$  correspondingly be defined in Definition 3.8 for any  $i \in \{2, 3, \dots, T\}$ . Let  $\text{TF}(Z^{(0)}; Q, P)$  be defined in Definition 3.6. We have

$$\text{TF}(Z^{(0)}; Q, P) = \alpha\theta^{(T)}.$$

*Proof.* We will prove this theorem by induction on  $t$ , where  $t \in [T]$ . From Lemma 4.1, we have:

$$\text{Attn}(Z^{(0)}; Q, P) = \begin{bmatrix} \mathbf{0}_{n \times d} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top X + \alpha y^\top X & 0 \end{bmatrix}.$$

For  $t = 1$ , we can calculate:

$$\begin{aligned} Z^{(1)} &= Z^{(0)} - \eta^{(0)} \text{Attn}(Z^{(0)}; Q, P) \\ &= \begin{bmatrix} X & y \\ q^{(0)\top} & \alpha \end{bmatrix} - \eta^{(0)} \begin{bmatrix} \mathbf{0}_{n \times d} & \mathbf{0}_{n \times 1} \\ q^{(0)\top} X^\top X + \alpha y^\top X & 0 \end{bmatrix} \\ &= \begin{bmatrix} X & y \\ q^{(0)\top} - \eta^{(0)}(q^{(0)\top} X^\top X + \alpha y^\top X) & \alpha \end{bmatrix} \end{aligned}$$

Where the first step follows from Definition 3.6, the second step follows from Definition 3.4 and Definition 3.5, the third step follows from basic algebra. Then we extract  $q^{(0)\top} - \eta^{(0)}(q^{(0)\top} X^\top X + \alpha y^\top X)$ , we have

$$\begin{aligned} &q^{(0)\top} - \eta^{(0)}(q^{(0)\top} X^\top X + \alpha y^\top X) \\ &= -\alpha \left( -\frac{1}{\alpha} q^{(0)\top} - \eta^{(0)} \left( -\frac{1}{\alpha} q^{(0)\top} X^\top X - y^\top X \right) \right) \\ &= -\alpha (\theta^{(0)\top} - \eta^{(0)} (\theta^{(0)\top} X^\top X - y^\top X)) \\ &= -\alpha \theta^{(1)\top}. \end{aligned}$$

where the first step follows from basic algebra, the second step follows from we defined  $\theta^{(0)} = -\frac{1}{\alpha}q^{(0)}$ , the third step follows from basic algebra. Thus,  $q^{(1)} = -\alpha\theta^{(1)}$ , so  $\theta^{(1)} = -\frac{1}{\alpha}q^{(1)}$ .

Similarly, by math induction, we can have  $\theta^{(T)} = -\frac{1}{\alpha}q^{(T)}$ . Thus, we finish the proof by  $\text{TF}(Z^{(0)}; Q, P) := -q^{(T)}$ .  $\square$

**Lemma 4.3** (Cauchy-Schwarz inequality). Let  $a, b \in \mathbb{R}^d$ , we have

$$|\langle a, b \rangle| \leq \|a\|_2 \cdot \|b\|_2.$$

To further refine our understanding of the Looped Transformer's performance, we introduce a bound on the final prediction error:

**Lemma 4.4.** The final prediction error satisfies

$$|\langle \text{TF}(Z^{(0)}; Q, P), \theta^* \rangle - \alpha| \leq |\alpha| \cdot \|\theta^{(T)} - \theta^*\|_2.$$

*Proof.* By Definition 3.3, we have  $q = \text{TF}(Z^{(0)}; Q, P), \theta^*$ . Then, we have

$$|\langle \text{TF}(Z^{(0)}; Q, P), \theta^* \rangle - \alpha| = |\langle \alpha\theta^{(T)}, \theta^* \rangle - \alpha|$$

$$\begin{aligned} &= |\alpha| \cdot |\langle \theta^{(T)}, \theta^* \rangle - \langle \theta^*, \theta^* \rangle| \\ &= |\alpha| \cdot |\langle \theta^{(T)} - \theta^*, \theta^* \rangle| \\ &\leq |\alpha| \cdot \|\theta^{(T)} - \theta^*\|_2. \end{aligned}$$

where the first step is from Definition 3.3 and Theorem 4.2, the second step follows  $\|\theta^*\|_2 = 1$ , the third step is from the linear properties of inner product, and the fourth step is from Cauchy-Schwarz inequality (Lemma 4.3).  $\square$

This result provides a quantitative measure of the model's accuracy, linking it directly to the number of iterations and multi-step gradient descent results of linear regression in Eq. (2).

Finally, we present our main theoretical contribution, which encapsulates the core findings of our work:

**Theorem 4.5** (Main result. Formal version of Theorem 1.1). Let  $X \in \mathbb{R}^{n \times d}$ ,  $y \in \mathbb{R}^n$ , and  $\theta \in \mathbb{R}^d$  be defined in Definition 3.8. Let  $\kappa$  be the condition number defined in Definition 3.11. Let  $T$  be the number of loops. Let the initial point  $q^{(0)} = \mathbf{0}_d$ . Then, we have the final prediction error satisfies

$$|\langle \text{TF}(Z^{(0)}; Q, P), \theta^* \rangle - \alpha| \leq |\alpha| \cdot \exp\left(-\frac{T}{2\kappa}\right).$$

*Proof.* The proof directly follows Lemma 4.4 and Theorem 5.8.  $\square$

In Theorem 4.5, as long as the condition number is constant, we can see that the linear looped Transformer will perform better when the loop number is increasing, i.e., the error will exponentially decay to 0. Usually, we only need  $O(d)$  numbers of in-context examples to guarantee a constant  $\kappa$ .

The above theorem offers a comprehensive characterization of the Looped Transformer's behavior, providing a tight bound on the prediction error that decays exponentially with the number of iterations. The intuition is that the Linear Looped Transformer can explicitly perform gradient descent in its hidden states. Furthermore, our theoretical finding is also supported by our experiments in Section 6.

**Comparison with Previous Works.** We restate the results in Gatmiry et al. (2024).

**Theorem 4.6** (Theorem 4.1 in Gatmiry et al. (2024)). Under condition  $\frac{8Td^2}{\sqrt{n}} \leq \frac{1}{2^{2T}}$ , we have the optimal linear regression error is  $\leq \frac{8Td^2 2^{2T}}{\sqrt{n}}$ .

In their work, the linear looped transformer has an error bound  $8Td^2 2^{2T} / \sqrt{n}$ , while our bound is  $|\alpha| \cdot \exp(-\frac{T}{2\kappa})$ . As the looped number  $T$  increases, our error bounds will

exponentially decay, while theirs increase exponentially. Note that our linear vector generation task and the linear regression task are dual problems. Our results align with the common intuition that more steps of gradient descent lead to better performance.

## 5 ERROR CONVERGENCE

In this section, we explore the convergence properties of looped transformers, focusing on their behavior under conditions of strong convexity and smoothness. We begin by defining these key concepts and then proceed to establish their implications.

### 5.1 Convexity and Smoothness Analysis

We first introduce some crucial definitions.

**Definition 5.1** (Strong convexity). *Let  $f : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$  and  $\mu > 0$ . We say that  $f$  is  $\mu$ -strongly convex if, for every  $x, y \in \mathbb{R}^d$ , and every  $t \in (0, 1)$  we have that*

$$\begin{aligned} & \mu \frac{\gamma(1-\gamma)}{2} \|x - y\|_2^2 + f(\gamma x + (1-\gamma)y) \\ & \leq \gamma f(x) + (1-\gamma)f(y). \end{aligned}$$

We say that  $\mu$  is the strong convexity constant of  $f$ .

**Definition 5.2** ( $L$ -smooth). *Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  and  $L > 0$ . We say that  $f$  is  $L$ -smooth if it is differentiable and if  $\nabla f : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is  $L$ -Lipschitz for all  $x, y \in \mathbb{R}^d$*

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2.$$

To quantitatively analyze the parameter dynamics in our linear vector generation task, we first derive the Lipschitz and convexity constants for the model introduced in Definition 3.8.

**Lemma 5.3.** *Given  $X \in \mathbb{R}^{n \times d}$ ,  $y \in \mathbb{R}^n$ , and  $\theta \in \mathbb{R}^d$  in Definition 3.8, we have*

$$L = \|X^\top X\|$$

where  $L$  is the Lipschitz constant defined in Definition 5.2.

*Proof.* From Definition 3.8, we have

$$\ell(\theta) = 0.5\|y - X\theta\|_2^2.$$

The gradient will be

$$\nabla_\theta \ell(\theta) = X^\top X\theta - X^\top y \in \mathbb{R}^d.$$

Then for  $\theta_1, \theta_2 \in \mathbb{R}^d$ , we have

$$\|\nabla \ell_{\theta_1}(\theta_1) - \nabla \ell_{\theta_2}(\theta_2)\|_2 = \|X^\top X(\theta_1 - \theta_2)\|_2$$

$$\leq \|X^\top X\| \cdot \|\theta_1 - \theta_2\|_2$$

where the first step follows from Definition 3.8, and the second step follows from properties of the norm. From Definition 5.2, we observe that

$$L = \|X^\top X\|,$$

where  $\|X^\top X\|$  is the spectral norm of  $X^\top X$  denoting the maximum eigenvalue.  $\square$

The following two lemmas are closely related and build upon each other to establish the strong convexity constant for a specific optimization problem.

**Lemma 5.4** (Forklore). *For  $X \in \mathbb{R}^{n \times d}$  and  $v \in \mathbb{R}^d$ , we have*

$$v^\top (X^\top X)v \geq \lambda_{\min}(X^\top X)\|v\|_2^2.$$

*Proof.* Let  $\{\lambda_i\}_{i=1}^d$  be the eigenvalue of  $X^\top X$ ,  $\{v_i\}_{i=1}^d$  is the corresponding standard orthogonal feature vector, we have

$$\begin{aligned} v^\top (X^\top X)v &= \left(\sum_{i=1}^d c_i v_i^\top\right) (X^\top X) \left(\sum_{j=1}^d c_j v_j\right) \\ &= \sum_{i=1}^d \sum_{j=1}^d c_i c_j v_i^\top (X^\top X) v_j \\ &= \sum_{i=1}^d \sum_{j=1}^d c_i c_j \lambda_j v_i^\top v_j \\ &= \sum_{i=1}^d c_i^2 \lambda_i \\ &\geq \lambda_{\min}(X^\top X) \sum_{i=1}^d c_i^2 \\ &= \lambda_{\min}(X^\top X) \|v\|_2^2 \end{aligned}$$

where the first step follows from properties of symmetric matrix, the second step follows from the matrix being decomposed into linear combinations of eigenvectors, the third step follows from basic algebra, the fourth step follows from orthogonality of eigenvectors, the fifth step follows from  $\lambda_{\min}$  denotes the minimum eigenvalue, the sixth step follows from the definition of norm.  $\square$

**Lemma 5.5.** *Given  $X \in \mathbb{R}^{n \times d}$ ,  $y \in \mathbb{R}^n$ , and  $\theta \in \mathbb{R}^d$  in Definition 3.8, we have*

$$\mu = \lambda_{\min}(X^\top X)$$

where  $\mu$  is the strong convexity constant defined in Definition 5.1

*Proof.* For any  $\theta_1, \theta_2 \in \mathbb{R}^d$  and  $t \in (0, 1)$ , we have

$$\begin{aligned}
 & \ell(\gamma\theta_1 + (1-\gamma)\theta_2) \\
 &= 0.5\|y - X(\gamma\theta_1 + (1-\gamma)\theta_2)\|_2^2 \\
 &= 0.5\|\gamma(y - X\theta_1) + (1-\gamma)(y - X\theta_2)\|_2^2 \\
 &\leq 0.5(\gamma\|y - X\theta_1\|_2^2 + (1-\gamma)\|y - X\theta_2\|_2^2) \\
 &\quad - 0.5\gamma(1-\gamma)\|X(\theta_1 - \theta_2)\|_2^2 \\
 &= 0.5(\gamma\|y - X\theta_1\|_2^2 + (1-\gamma)\|y - X\theta_2\|_2^2) \\
 &\quad - 0.5\gamma(1-\gamma)(\theta_1 - \theta_2)^T X^T X (\theta_1 - \theta_2) \\
 &\leq 0.5(\gamma\|y - X\theta_1\|_2^2 + (1-\gamma)\|y - X\theta_2\|_2^2) \\
 &\quad - 0.5\gamma(1-\gamma)\lambda_{\min}(X^T X)\|\theta_1 - \theta_2\|_2^2
 \end{aligned}$$

where the first step follows from Definition 3.8, the rest step follow from basic algebra and Lemma 5.4. From Definition 5.1, we observe that

$$\mu = \lambda_{\min}(X^T X)$$

where  $\lambda_{\min}(X^T X)$  denotes the minimum eigenvalue of  $X^T X$ .  $\square$

## 5.2 Main Result

We first commence with a statement of Lemma 5.6, which furnishes a convergence rate for gradient descent on strongly convex and smooth functions.

**Lemma 5.6** (Theorem 3.6 in Garrigos and Gower (2023)). *Let  $\ell : \mathbb{R}^d \rightarrow \mathbb{R}$  be a differentiable function and assume  $\ell$  is  $\mu$ -strongly convex and  $L$ -smooth. Let  $\{\theta^{(t)}\}_{t \in \mathbb{N}}$  be the sequence generated by the gradient descent algorithm, with a stepsize  $\eta \in (0, \frac{1}{L}]$ . Then for  $\theta^* = \arg \min_{\theta} \ell(\theta)$  and for all  $t \in \mathbb{N}$ , we have*

$$\|\theta^{(t)} - \theta^*\|_2^2 \leq (1 - \eta\mu)^t \|\theta^{(0)} - \theta^*\|_2^2.$$

**Fact 5.7** (Folklore). *For any  $n, T \in \mathcal{N}_+$ , we have*

$$(1 - \frac{1}{n})^T \leq e^{-T/n}.$$

We now present a rigorous upper bound on the error magnitude of the gradient descent algorithm's output after  $T$  iterations, elucidating the convergence properties of this optimization method in the context of linear vector generation.

**Theorem 5.8.** *If the following holds:*

- Let  $T$  be the loop number.
- Let  $n, T, t \in \mathbb{N}$ .
- Let  $X \in \mathbb{R}^{n \times d}$ ,  $y \in \mathbb{R}^n$ , and  $\theta \in \mathbb{R}^d$  be defined in Definition 3.8.
- Let the condition number  $\kappa = \frac{\lambda_{\max}(X^T X)}{\lambda_{\min}(X^T X)}$ .

- The step size  $\eta = \frac{1}{L}$ .
- Let  $\mu$  and  $L$  be defined in Definition 5.1 and Definition 5.2.
- For  $\ell(\theta)$  defined in Definition 3.8, we have  $\ell(\theta)$  is  $L$ -smooth and  $\mu$  strong convex, where  $L = \|X^T X\|$  and  $\mu = \lambda_{\min}(X^T X)$ .
- The initial point  $\theta^{(0)}$  satisfies  $\|\theta^{(0)} - \theta^*\|_2 \leq R$ .

Then, we have

$$\|\theta^* - \theta^{(T)}\|_2^2 \leq e^{-T/\kappa} R^2.$$

*Proof.* First, we have

$$\begin{aligned}
 \|\theta^{(t)} - \theta^*\|_2^2 &\leq (1 - \eta\mu)^t \|\theta^{(0)} - \theta^*\|_2^2 \\
 &\leq (1 - \frac{\mu}{L})^t \|\theta^{(0)} - \theta^*\|_2^2
 \end{aligned}$$

where the first step follows from Lemma 5.6, the second step follows from we choose  $\eta^{(t)} = \frac{1}{L}$ . Then consider the term  $\frac{\mu}{L}$ , we have

$$\frac{\mu}{L} = \frac{\lambda_{\min}(X^T X)}{\|X^T X\|} = \frac{1}{\kappa}$$

where the first step follows from Lemma 5.3 and Lemma 5.5, the second step follows the definition of condition number  $\kappa = \frac{\lambda_{\max}(X^T X)}{\lambda_{\min}(X^T X)}$ .

Then, substituting this back, we have

$$\begin{aligned}
 \|\theta^{(T)} - \theta^*\|_2^2 &\leq (1 - \frac{1}{\kappa})^T \|\theta^{(0)} - \theta^*\|_2^2 \\
 &= (1 - 1/\kappa)^{\kappa \cdot T/\kappa} \|\theta^{(0)} - \theta^*\|_2^2 \\
 &\leq e^{-T/\kappa} \|\theta^{(0)} - \theta^*\|_2^2 \\
 &\leq e^{-T/\kappa} R^2
 \end{aligned}$$

where the first step follows from basic algebra, the second step follows from  $(1 - 1/\kappa)^\kappa < e^{-1}$  (Fact 5.7), and the last step follows from  $\|\theta^{(0)} - \theta^*\|_2 \leq R$ .  $\square$

Theorem 5.8 tells us that GD can well solve linear regression tasks. In particular, when the input data has a good condition number, the approximation error will exponentially decay to 0. We use the above insights in the proof of our main results (Theorem 4.5).

## 6 EXPERIMENTS

In this section, we aim to verify our theory by evaluating the convergence behavior of gradient descent for linear vector generation. We designed our experiment to examine the impact of varying sample sizes on convergence rates while keeping the feature dimension



fixed. Our results demonstrate that empirical convergence rates consistently outperform theoretical upper bounds across all sample sizes, with significant improvement in convergence speed as the condition number decreases, validating our theoretical predictions.

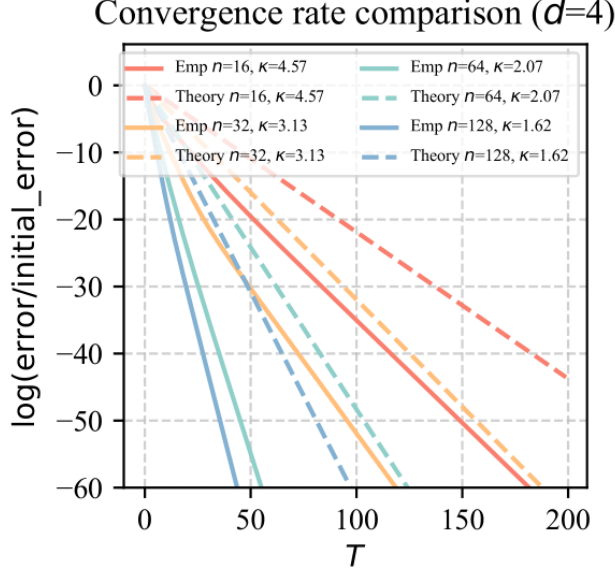


Figure 1: The convergence rate comparison for gradient descent in linear vector generation with a fixed dimension  $d = 4$  and varying sample sizes  $n \in \{16, 32, 64, 128\}$  and their corresponding condition number  $\kappa$ . The ‘Emp’ means the empirical error of our experiments. The ‘Theory’ means the theoretical bound in Theorem 4.5. The y-axis is the logarithm of normalized error and the x-axis is the number of loops  $T$ . Both empirical (solid lines) and theoretical (dashed lines) results are presented for each  $n$ . The plot demonstrates that as the sample size  $n$  increases, the condition number will decrease, so the convergence rate improves. Thus, with larger  $n$  values, there will be a steeper slope and faster convergence to the optimal solution.

## 6.1 Experiment Setup

In this experiment, we aimed to investigate the convergence behavior of multi-step gradient descent for Linear Looped Transformer (Definition 3.6) in-context learning the linear vector generation task (Definition 3.3). We randomly draw each entry of  $X \in \mathbb{R}^{n \times d}$  from standard Gaussian distribution,  $\mathcal{N}(0, 1)$ , and response variables  $y = X\theta^*$ , where  $\theta^* \in \mathbb{R}^d$  was randomly chosen. Our experiments focused on scenarios with  $d$  fixed at 4 and  $n$  varying in  $\{16, 32, 64, 128\}$ . For each  $(n, d)$  combination, we implemented gradient descent with  $T = 200$  iterations and learning rate  $\eta = 1/L$ , where  $L = \|X^\top X\|$ . To ensure statistical robustness, we con-

ducted 10 independent trials for each configuration. Convergence was measured by tracking  $\|\theta^{(t)} - \theta^*\|_2^2$ , where  $\theta^*$  is the optimal least squares solution. To facilitate comparison across different problem sizes, we normalized error plots by the initial error, plotting  $\log(\|\theta^{(t)} - \theta^*\|_2^2 / \|\theta^{(0)} - \theta^*\|_2^2)$  against  $t$ . This normalization enabled a clear comparison of relative decay rates, irrespective of initial error magnitudes, thus providing insights into the impact of condition number on gradient descent convergence in the linear vector generation task.

## 6.2 Result Interpretation

Our experiment investigates the convergence behavior of gradient descent for linear vector generation with varying sample sizes  $n$  and a fixed feature dimension  $d = 4$ . Figure 1 illustrates the convergence rates for different  $n$  values  $\{16, 32, 64, 128\}$ , comparing empirical results with theoretical bounds in Theorem 4.5. A key aspect of this experiment is the condition number  $\kappa$ , which decreases as the number of examples increases. The average  $\kappa$  values for the different sample sizes are  $\{4.57, 3.13, 2.07, 1.62\}$ , corresponding to  $n = \{16, 32, 64, 128\}$  respectively. This inverse relationship between  $n$  and  $\kappa$  is noteworthy, as it significantly influences the convergence rates. The results demonstrate that as the sample size  $n$  increases, the convergence rate improves substantially. This is evident from the steeper slopes of both empirical and theoretical lines for larger  $n$  values. Importantly, the empirical convergence rates consistently outperform the theoretical upper bounds across all sample sizes, with the gap between empirical and theoretical performance narrowing as  $n$  increases. This observation aligns with our theoretical expectations in Theorem 4.5 and highlights the crucial role of the condition number in determining convergence behavior.

## 7 CONCLUSION

In this work, we have demonstrated that linear looped Transformers can efficiently implement multi-step gradient descent for in-context learning, requiring only a reasonable number of examples when input data is well-conditioned. This finding relieves the previous assumptions of an exponential number of in-context examples and offers new insights into the capabilities of Transformer architectures. Our theoretical analysis and preliminary experiments pave the way for more efficient inference algorithms in large language models and open avenues for future research in this domain.

## Acknowledgments

Research is partially supported by the National Science Foundation (NSF) Grants 2023239-DMS, CCF-2046710, and Air Force Grant FA9550-18-1-0166.

## References

- Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. In-context examples selection for machine translation. *arXiv preprint arXiv:2212.02437*, 2022.
- Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.
- Ekin Akyurek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.
- Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, context-free grammar. *arXiv preprint arXiv:2305.13673*, 2023.
- Anthropic. Claude 3.5 sonnet, 2024.
- Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.
- de Luca Artur Back and Kimon Fountoulakis. Simulation of graph algorithms with looped transformers. *arXiv preprint arXiv:2402.01107*, 2024.
- Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.
- Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- 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*, 2020.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
- Yang Cao. Sorsa: Singular values and orthonormal regularized singular vectors adaptation of large language models. *arXiv preprint arXiv:2409.00055*, 2024.
- Yang Cao, Xiaoyu Li, and Zhao Song. Grams: Gradient descent with adaptive momentum scaling. *arXiv preprint arXiv:2412.17107*, 2024.
- Yang Cao, Bo Chen, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. Force matching with relativistic constraints: A physics-inspired approach to stable and efficient generative modeling. *arXiv preprint arXiv:2502.08150*, 2025.
- Bo Chen, Xiaoyu Li, Yingyu Liang, Jiangxuan Long, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for rope-based transformer architecture. *arXiv preprint arXiv:2411.07602*, 2024a.
- Bo Chen, Chengyue Gong, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. High-order matching for one-step shortcut diffusion models. *arXiv preprint arXiv:2502.00688*, 2025a.
- Bo Chen, Xiaoyu Li, Yingyu Liang, Zhao Song, and Zhizhou Sha. Nrfow: Towards noise-robust generative modeling via second-order flow matching. *manuscript*, 2025b.
- Bo Chen, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Hsr-enhanced sparse attention acceleration. In *Conference on Parsimony and Learning*. PMLR, 2025c.
- Yifang Chen, Jiayan Huo, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Fast gradient computation for rope attention in almost linear time. *arXiv preprint arXiv:2412.17316*, 2024b.

- Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. The computational limits of state-space models and mamba via the lens of circuit complexity. *arXiv preprint arXiv:2412.06148*, 2024c.
- Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Universal approximation of visual autoregressive transformers. *arXiv preprint arXiv:2502.06167*, 2025d.
- Xiang Cheng, Yuxin Chen, and Suvrit Sra. Transformers implement functional gradient descent to learn non-linear functions in context. *arXiv preprint arXiv:2312.06528*, 2023.
- Po-Han Chi, Pei-Hung Chung, Tsung-Han Wu, Chun-Cheng Hsieh, Yen-Hao Chen, Shang-Wen Li, and Hung-yi Lee. Audio albert: A lite bert for self-supervised learning of audio representation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 344–350. IEEE, 2021.
- 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.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.
- Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.
- Yichuan Deng, Zhao Song, Yitan Wang, and Yuanyuan Yang. A nearly optimal size coresets algorithm with nearly linear time. *arXiv preprint arXiv:2210.08361*, 2022.
- Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arXiv preprint arXiv:2304.04397*, 2023.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
- Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023a.
- Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021a.
- Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021b.
- Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression. *arXiv preprint arXiv:2303.16504*, 2023b.
- Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 2022.
- Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.
- Khashayar Gatmiry, Nikunj Saunshi, Sashank J Reddi, Stefanie Jegelka, and Sanjiv Kumar. Can looped transformers learn to implement multi-step gradient descent for in-context learning? In *Forty-first International Conference on Machine Learning*, 2024.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.
- Angeliki Giannou, Liu Yang, Tianhao Wang, Dimitris Papailiopoulos, and Jason D Lee. How well can transformers emulate in-context newton’s method? *arXiv preprint arXiv:2403.03183*, 2024.
- Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. How do transformers learn in-context beyond simple functions? a case study on learning with representations. In *The Twelfth International Conference on Learning Representations*, 2024.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- 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.
- 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)*, 2024a.
- Jerry Yao-Chieh Hu, Bo-Yu Chen, Dennis Wu, Feng Ruan, and Han Liu. Nonparametric modern hopfield models. *arXiv preprint arXiv:2404.03900*, 2024b.
- 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)*, 2024c.
- Jerry Yao-Chieh Hu, Dennis Wu, and Han Liu. Provably optimal memory capacity for modern hopfield models: Tight analysis for transformer-compatible dense associative memories. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024d.
- Jerry Yao-Chieh Hu, Weimin Wu, Yi-Chen Lee, Yu-Chao Huang, Minshuo Chen, and Han Liu. On statistical rates of conditional diffusion transformers: Approximation, estimation and minimax optimality. *arXiv preprint arXiv:2411.17522*, 2024e.
- 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). *Advances in Neural Information Processing Systems*, 38, 2024f.
- Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023.
- Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. Opt-impl: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022.
- Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 2022.
- Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Advancing the understanding of fixed point iterations in deep neural networks: A detailed analytical study, 2024.
- Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. On computational limits and provably efficient criteria of visual autoregressive models: A fine-grained complexity analysis. *arXiv preprint arXiv:2501.04377*, 2025.
- Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*, 2022.
- Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.
- 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*, 2024a.
- Chenyang Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Tianyi Zhou. Fourier circuits in neural networks: Unlocking the potential of large language models in mathematical reasoning and modular arithmetic. *arXiv preprint arXiv:2402.09469*, 2024b.
- Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. In *The Eleventh International Conference on Learning Representations*, 2023a.
- Hongkang Li, Meng Wang, Songtao Lu, Hui Wan, Xiaodong Cui, and Pin-Yu Chen. Transformers as multi-task feature selectors: Generalization analysis of in-context learning. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023b.
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2021.
- Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. A tighter complexity analysis of sparsegpt. *arXiv preprint arXiv:2408.12151*, 2024c.
- 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*, 2024d.

- Xiaoyu Li, Yingyu Liang, Jiangxuan Long, Zhenmei Shi, Zhao Song, and Zhen Zhuang. Neural algorithmic reasoning for hypergraphs with looped transformers. *arXiv preprint arXiv:2501.10688*, 2025.
- Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. Dissecting chain-of-thought: Compositionality through in-context filtering and learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023c.
- Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2023d.
- Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, and Junze Yin. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *arXiv preprint arXiv:2405.05219*, 2024a.
- Yingyu Liang, Jiangxuan Long, Zhenmei Shi, Zhao Song, and Yufa Zhou. Beyond linear approximations: A novel pruning approach for attention matrix, 2024b.
- 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*, 2024c.
- Yingyu Liang, Zhenmei Shi, Zhao Song, and Chiwun Yang. Toward infinite-long prefix in transformer. *arXiv preprint arXiv:2406.14036*, 2024d.
- Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Unraveling the smoothness properties of diffusion models: A gaussian mixture perspective. *arXiv preprint arXiv:2405.16418*, 2024e.
- Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Tensor attention training: Provably efficient learning of higher-order transformers. *arXiv preprint arXiv:2405.16411*, 2024f.
- Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Looped relu mlps may be all you need as practical programmable computers. In *International Conference on Artificial Intelligence and Statistics*, 2025.
- Zeping Luo, Shiyu Wu, Cindy Weng, Mo Zhou, and Rong Ge. Understanding the robustness of self-supervised learning through topic modeling. In *The Eleventh International Conference on Learning Representations*, 2023.
- Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.
- Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.
- Maxwell Nye, Anders Johan Andreassen, Gur AriGuy, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.
- OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- OpenAI. Introducing ChatGPT, 2024.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.
- Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning ‘learns’ in-context: Disentangling task recognition and task learning. In *Findings of Association for Computational Linguistics (ACL)*, 2023.
- Mohammadreza Pourreza and Davood Rafiei. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *arXiv preprint arXiv:2304.11015*, 2023.
- Lianke Qin, Zhao Song, and Baocheng Sun. Is solving graph neural tangent kernel equivalent to training graph neural network? *arXiv preprint arXiv:2309.07452*, 2023.
- Allan Raventos, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations*, 2024.
- Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Jing Liu, Ruiyi Zhang, Ryan A Rossi, Hao Tan, Tong Yu, Xiang Chen, et al. Numerical pruning for



- efficient autoregressive models. *arXiv preprint arXiv:2412.12441*, 2024.
- Zhenmei Shi, Jiefeng Chen, Kunyang Li, Jayaram Raghuram, Xi Wu, Yingyu Liang, and Somesh Jha. The trade-off between universality and label efficiency of representations from contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.
- 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*, 2024a.
- Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? In *International Conference on Machine Learning*. PMLR, 2024b.
- Anshumali Shrivastava, Zhao Song, and Zhaozhuo Xu. A theoretical analysis of nearest neighbor search on approximate near neighbor graph. *arXiv preprint arXiv:2303.06210*, 2023.
- Zhao Song, Mingquan Ye, and Lichen Zhang. Streaming semidefinite programs:  $O(\sqrt{n})$  passes, small space and fast runtime. *arXiv preprint arXiv:2309.05135*, 2023.
- Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*. PMLR, 2022.
- Yuandong Tian, Yiping Wang, Beidi Chen, and Simon Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in Neural Information Processing Systems*, 2023a.
- Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. *arXiv preprint arXiv:2310.00535*, 2023b.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 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*, 2017.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*. PMLR, 2023.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Supernaturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022b.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022c.
- Jerry Wei, Le Hou, Andrew Kyle Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc V Le. Symbol tuning improves in-context learning in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023a.
- Jerry Wei, Le Hou, Andrew Kyle Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. Symbol tuning improves in-context learning in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.
- Kevin Christian Wibisono and Yixin Wang. On the role of unstructured training data in transformers’ in-context learning capabilities. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.
- 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)*, 2024a.
- Dennis Wu, Jerry Yao-Chieh Hu, Weijian Li, Bo-Yu Chen, and Han Liu. STanhop: Sparse tandem hop-

- field model for memory-enhanced time series prediction. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b.
- Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? In *The Twelfth International Conference on Learning Representations*, 2024c.
- Weimin Wu, Maojiang Su, Jerry Yao-Chieh Hu, Zhao Song, and Han Liu. Transformers are deep optimizers: Provable in-context learning for deep model training. *arXiv preprint arXiv:2411.16549*, 2024d.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.
- 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)*, 2024a.
- Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Yin Li, and Yingyu Liang. Improving foundation models for few-shot learning via multitask finetuning. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.
- 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*, 2024b.
- Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. In *The Twelfth International Conference on Learning Representations*, 2024c.
- Liu Yang, Kangwook Lee, Robert Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. *arXiv preprint arXiv:2311.12424*, 2023.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Qifan Yu, Zhenyu He, Sijie Li, Xun Zhou, Jun Zhang, Jingjing Xu, and Di He. Enhancing auto-regressive chain-of-thought through loop-aligned reasoning. *arXiv preprint arXiv:2502.08482*, 2025.
- Hanlin Zhang, Yi-Fan Zhang, Yaodong Yu, Dhruv Madeka, Dean Foster, Eric Xing, Hima Lakkaraju, and Sham Kakade. A study on the calibration of in-context learning. *arXiv preprint arXiv:2312.04021*, 2023a.
- Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023b.
- Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023c.
- Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Zhen Qin, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. Tensor product attention is all you need. *arXiv preprint arXiv:2501.06425*, 2025.
- Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*. PMLR, 2021.
- Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. Step-back prompting enables reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinu Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.
- Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, and Hanie Sedghi. Teaching algorithmic reasoning via in-context learning. *arXiv preprint arXiv:2211.09066*, 2022.
- Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023b.

## Checklist

1. For all models and algorithms presented, check if you include:
  - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
  - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
  - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]
2. For any theoretical claim, check if you include:
  - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
  - (b) Complete proofs of all theoretical results. [Yes]
  - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
  - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
  - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
  - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
  - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
  - (a) Citations of the creator If your work uses existing assets. [Not Applicable]
  - (b) The license information of the assets, if applicable. [Not Applicable]
  - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
  - (d) Information about consent from data providers/curators. [Not Applicable]
  - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
  - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
  - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
  - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]