How to Protect Copyright Data in Optimization of Large Language Models?

Timothy Chu[1], **Zhao Song**[2], **Chiwun Yang**[3]

[1] Google, Mountain View, CA
[2]Adobe Research, San Jose, CA
[3] Sun Yat-sen University, China
hungryTOAN@gmail.com, zsong@adobe.com, christiannyang37@gmail.com

Abstract

Large language models (LLMs) and generative AI have played a transformative role in computer research and applications. Controversy has arisen as to whether these models output copyrighted data, which can occur if the data the models are trained on is copyrighted. LLMs are built on the transformer neural network architecture, which in turn relies on a mathematical computation called Attention that uses the softmax function.

In this paper, we observe that large language model training and optimization can be seen as a softmax regression problem. We then establish a method of efficiently performing softmax regression, in a way that prevents the regression function from generating copyright data. This establishes a theoretical method of training large language models in a way that avoids generating copyright data.

1 Introduction

Large language models have changed the world, with the rise of generative AI models such as ChatGPT, GPT-4, Llama, BERT, BARD, PaLM, and OPT (ChatGPT 2022; Bubeck et al. 2023; Devlin et al. 2018; Touvron et al. 2023b,a; BARD 2023; Chowdhery et al. 2022; Anil et al. 2023; Zhang et al. 2022). These models are able to process natural language effectively, handling a wide range of tasks including story generation, code creation, machine translation, and elementary mathematical problem solving (Brown et al. 2020; Svyatkovskiy et al. 2020; Wu et al. 2016; Wei et al. 2022). One core component in the large language model is the **transformer** architecture (Vaswani et al. 2017), which is built on a computational step known as *attention*. Transformers have been used in a wide variety of tasks outside of large language models, including generative image systems such as DALL-E (Research 2021) and DALL-E2 (Research 2022). Recent research has integrated the transformer architecture with scalable diffusion-based image generation models (Bao et al. 2023; Cao et al. 2022; Wu et al. 2023a; Han et al. 2022; Dosovitskiy et al. 2020).

Once challenge in generative AI is guaranteeing that outputs are protected from copyright infringement and intellectual property issues (Hattenbach and Glucoft 2015; Hristov 2016; Sag 2018; Gillotte 2019; Vyas, Kakade, and Barak 2023). Generative models trained on large corpora of data can inadvertently generate outputs that are direct copies, or close variants, of copyrighted text or images that the model is trained on. Removing copyrighted material from training may also be undesirable: while one can achieve good performance without using copyrighted data, the inclusion of such data can significantly enhance the performance of generative AI models. For example, incorporating literary works, which are often copyrighted, into the training dataset of a language model can enhance performance. Copyright infringement issues regarding outputs of generative AI have led to controversy in using it, and past researchers have considered models and theoretical frameworks for evaluating whether generative models are copying data, and how to evaluate and avoid copyright issues that arise (Vyas, Kakade, and Barak 2023).

Our paper has two main contributions:

1. We provide an approach for solving general regression problems in a way that avoids generating copyright data. We term this approach *copyright regression*.
2. We show how to protect copyright data in the optimization and training of transformer-based architectures (including most large language models), by finding an algorithm to solve copyright regression for the softmax function.

Solving the copyright regression problem for the softmax function is the key technical contribution of our paper. To establish the copyright regression framework, we provide a new optimization objective for a general regression problem where some outputs are copyrighted. Such a case can arise when regression outputs are images or sentences, which occurs in transformer-based architectures for language generation and image generation. We also review literature linking the softmax regression problem to the training of transformers.

To find an algorithm for solving copyright regression for the softmax function, we used an approach based on convex optimization and gradient descent. We show that the objective function of the softmax copyright regression is convex, and that its Hessian is bounded. Showing this convexity is non-trivial, and requires intricate bounding of key matrix and vector quantities that arise in the softmax copyright regression problem. Establishing convexity and the bounded

Hessian property of the objective function in softmax copyright regression allows us to use gradient-based methods to efficiently solve this problem, with guaranteed bounds on convergence and good stability properties.

The code of experiments in this paper is open-sourced at https://github.com/ChristianYang37/chiwun/tree/main/src/Copyright-Regression. For the proofs of all lemmas and theorems in this paper, please refer to (Chu, Song, and Yang 2023b).

2 Related Work

This section briefly reviews the related research work on privacy and security of AI, theoretical large language model work, and optimization of neural networks. These topics have a close connection to our work.

Privacy and Security. Generative AI has achieved impressive results in various domains, including images, text, and code. However, preventing copyright infringement is a challenge that needs to be addressed (Hattenbach and Glucoft 2015; Hristov 2016; Sag 2018; Gillotte 2019). (Sag 2018) discusses whether data mining and machine learning on copyrighted text qualify as "fair use" under U.S. law. (Gillotte 2019) investigates copyright infringement in AI-generated artwork and argues that using copyrighted works during the training phase of AI programs does not result in infringement liability. To mitigate the potential harms of large language models, in (Kirchenbauer et al. 2023), a watermarking framework is introduced that facilitates the embedding of signals within the generated text. This framework aims to enhance the detection of output from Language Model (LLM) systems, thereby mitigating potential misuse or abuse. Building upon this foundation, subsequent research (He et al. 2022a,b) has contributed to the development of more robust and less intrusive watermark embedding algorithms. These advancements seek to improve the stability and minimize any adverse effects associated with the process of embedding watermarks. Such endeavors are important in ensuring the integrity and responsible utilization of LLM technology. (Vyas, Kakade, and Barak 2023) proposes a framework that provides stronger protection against sampling protected content, by defining near access-freeness (NAF) and developing generative model learning algorithms. Experiments demonstrate promising results with some impact on output quality for both language and image generative models. Recently, (Gao, Song, and Yang 2023) focuses on this issue of sampling protected content, and proposes a provable method for privately computing the attention matrix using differential privacy. (Xu et al. 2023) trains language models (LMs) with federated learning (FL) and differential privacy (DP) in the Google Keyboard (Gboard).

Theoretical LLM. Since the explosion of large language models, theoretical research on **transformer** has been one major component of improving language model performance (Kitaev, Kaiser, and Levskaya 2020; Chen et al. 2020; Tay et al. 2020; Noci et al. 2022; Deng, Li, and Song 2023; Panigrahi et al. 2023; Arora and Goyal 2023; Sun et al. 2023; Sanford, Hsu, and Telgarsky 2023; Jiang, Ren, and Lin 2023; Alman and Song 2023; Brand, Song, and Zhou 2023; Zelikman et al. 2023; Malladi et al. 2023; Liu et al. 2023a; Rafailov et al. 2023; Ignat et al. 2023; Gao, Song, and Yin 2023; Zhao et al. 2023; Deng et al. 2023; Gao et al. 2023; Wu et al. 2023b; Liu et al. 2023b). (Rücklé et al. 2020) proposes AdapterDrop, a method that removes adapters from lower transformer layers during training and inference to reduce computational overhead, while still maintaining task performance. (Tay et al. 2021) shows that random alignment matrices perform competitively and learning attention weights from token-token interactions is not highly significant. So they propose Synthesizer, a model that learns synthetic attention weights without token-token interactions and performs well in various tasks. (Chen et al. 2021) proposes Scatterbrain, a way to balance model quality and efficiency in approximating long sequences. Recent work (Arora and Goyal 2023) explores the emergence of new skills in language models through scaling up their parameters and training data. This demonstrates through mathematical analysis that the Scaling Laws provide a strong inductive bias, enabling efficient learning in pre-trained models. they term this phenomenon "slingshot generalization," as it seems to violate traditional generalization theory.

Optimization and Convergence of Deep Neural Networks. Prior research (Li and Liang 2018; Du et al. 2018; Allen-Zhu, Li, and Song 2019a,b; Song and Yang 2019; Cai et al. 2019; Zhang, Martens, and Grosse 2019; Cao and Gu 2019; Zou and Gu 2019; Oymak and Soltanolkotabi 2020; Ji and Telgarsky 2019; Lee et al. 2020; Huang et al. 2021; Zhang et al. 2020b; Brand et al. 2020; Zhang et al. 2020a; Song, Zhang, and Zhang 2021; Alman et al. 2023; Munteanu et al. 2022; Zhang 2022; Gao, Mahadevan, and Song 2023; Li, Song, and Zhou 2023; Qin, Song, and Yang 2023) on the optimization and convergence of deep neural networks has been crucial in understanding their exceptional performance across various tasks. These studies have also contributed to enhancing the safety and efficiency of AI systems. In (Gao, Mahadevan, and Song 2023) they define a neural function using an exponential activation function and apply the gradient descent algorithm to find optimal weights. In (Li, Song, and Zhou 2023), they focus on the exponential regression problem inspired by the attention mechanism in large language models. They address the non-convex nature of standard exponential regression by considering a regularization version that is convex. They propose an algorithm that leverages input sparsity to achieve efficient computation. The algorithm has a logarithmic number of iterations and requires nearly linear time per iteration, making use of the sparsity of the input matrix.

3 Preliminaries

In this section, we present preliminary concepts and definitions for our paper. We begin by introducing the notations we utilize in Section 3.1. In Section 3.2 we provide the problem definition that we aim to solve.

3.1 Notations

We use the following notations and definitions: The ℓ_p norm of a vector x is denoted as $\|x\|_p$, for examples,

$\|x\|_1 := \sum_{i=1}^n |x_i|$, $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$ and $\|x\|_\infty := \max_{i\in[n]} |x_i|$. For a vector $x \in \mathbb{R}^n$, $\exp(x) \in \mathbb{R}^n$ denotes a vector where whose i-th entry is $\exp(x_i)$ for all $i \in [n]$. For $n > k$, for any matrix $A \in \mathbb{R}^{n\times k}$, we denote the spectral norm of A by $\|A\|$, i.e., $\|A\| := \sup_{x\in\mathbb{R}^k} \|Ax\|_2/\|x\|_2$. We denote $\sigma_{\min}(A)$ as the minimum singular value of A. For two vectors $x, y \in \mathbb{R}^n$, we denote $\langle x, y\rangle = \sum_{i=1}^n$ for $i \in [n]$. Given two vectors $x, y \in \mathbb{R}^n$, we denote $x \circ y$ as a vector whose i-th entry is $x_i y_i$ for all $i \in [n]$. Let $x \in \mathbb{R}^n$ be a vector. For a vector $x \in \mathbb{R}^n$, $\mathrm{diag}(x) \in \mathbb{R}^{n\times n}$ is defined as a diagonal matrix with its diagonal entries given by $\mathrm{diag}(x)_{i,i} = x_i$ for $i = 1, ..., n$, and all off-diagonal entries are 0. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive definite (PD) when $A \succ 0$, for all non-zero vectors $x \in \mathbb{R}^n$, we have $x^\top Ax > 0$. Similarly, a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive semidefinite (PSD) when $A \succeq 0$, for all vectors $x \in \mathbb{R}^n$, we have $x^\top Ax \geq 0$.

3.2 Problem Definition

To achieve a successful copyright infringement claim in the United States and many other jurisdictions, the plaintiff must provide evidence that demonstrates two key elements. Firstly, they must establish that the defendant had access to the plaintiff's copyrighted work. Secondly, they must show that there are substantial similarities between the defendant's work and the original elements of the plaintiff's work (for the 9th circuits 2022).

While access to high-quality copyrighted data is essential for enhancing the performance of AI models, it also introduces legal risks. Therefore, when considering the safety and legality of AI systems, it is imperative to ensure that the ideal language model can effectively learn from all data without producing output that resembles copyrighted material present in its training set. By adhering to these considerations, we can maintain both the integrity of intellectual property rights and the lawful operation of AI technologies.

For convenience, we denote training dataset $\mathcal{D}$, copyright data $\mathcal{C} \subset \mathcal{D}$ and other data $\mathcal{O} = \mathcal{D} - \mathcal{C}$. Our objective is to ensure a model f, satisfies: for any input x, given a metric L, the model's output $f(x)$ will not exhibit substantial similarity to any copyrighted content present in its training set. We enforce this by defining a strict gap τ such that the metric $L(f(x), C)$, where $C \in \mathcal{C}$, is greater than or equal to τ plus the metric $L(f(x), O)$, where $O \in \mathcal{O}$. That is

$$L(f(x), C) \geq \tau + L(f(x), O).$$

The choice of metric L depends on the specific task, such as Cross-Entropy loss for text generation, mean absolute error or mean square error for regression problems, and Kullback-Leibler divergence or image similarity for image generation, etc.

To ensure compliance with copyright laws, we apply τ to the average metric L calculated over both $\mathcal{C}$ and $\mathcal{O}$, thus implementing a formal and conservative definition. And we convert dataset $\mathcal{D}$ to an input matrix $A \in \mathbb{R}^{n\times d}$ and a target vector $b \in \mathbb{R}^n$, where n is the size of the dataset, d is the dimension of input data. We now provide the definition of problem below.

Definition 1 (τ-Copyright-Protected)**.** *Given matrix $A \in \mathbb{R}^{n\times d}$ and vector $b \in \mathbb{R}^n$ that $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, and $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, where $A_1 \in \mathbb{R}^{n_1\times d}$, $A_2 \in \mathbb{R}^{n_2\times d}$, $b_1 \in \mathbb{R}^{n_1}$, $b_2 \in \mathbb{R}^{n_2}$ and $n = n_1 + n_2$. A_1, b_1 are the data has copyright issue and A_2, b_2 are the data does not have copyright issue. Denote the training objective L. Denote $\tau > 0$ as a scalar.*

If there is a trained model f_θ with parameter θ that satisfies $\frac{L(f_\theta(A_1),b_1)}{n_1} \geq \tau + \frac{L(f_\theta(A_2),b_2)}{n_2}$ then we say this model f_θ is τ-Copyright-Protected.

4 Methodology: Copyright Regression

A prominent existing approach, as outlined in the work by (Vyas, Kakade, and Barak 2023), introduces an algorithm that involves training an additional generative model, denoted as p, using non-copyrighted data. This algorithm employs rejection sampling to effectively manage the probability of the model generating copyrighted data. However, it is important to note that this method has certain limitations. Specifically, it incurs higher computational costs during the decoding process and necessitates the retraining of an additional model. Now we introduce our method, a simple modification to the standard training objective of generative language models to ensure that their outputs do not infringe upon copyright laws.

In accordance with the findings presented in (Deng, Li, and Song 2023), our approach involves decomposing the mechanism of **Attention** (Vaswani et al. 2017), into a regression problem termed Softmax Regression. This decomposition enables a deeper examination of the learning process underlying attention training. By adopting this method, we gain valuable insights into the intricacies of attention and its associated learning mechanisms.

We propose a modification to the standard training objective of generative language models based on the principles of Softmax Regression. The objective is to train the model to generate desired outputs, denoted as $f(A) = b$. However, in the case of copyrighted data, represented by $A_1 \in \mathbb{R}^{n_1\times d}$ and $b_1 \in \mathbb{R}^{n_1}$, we aim to prevent the model from learning to generate these specific outputs. To address this concern, we introduce an additional term $L(f(A_1), b_1)^{-1}$ to the training objective to discourage the model from generating outputs matching the copyrighted data. To control the level of this protection, we introduce a scalar coefficient $\gamma_c > 0$. Consequently, the modified training objective becomes $L(f(A), b) + \gamma_c L(f(A_1), b_1)^{-1}$. This modification serves to strike a balance between achieving the desired outputs and avoiding the generation of copyrighted data. The addition of the inverse term in the training objective helps mitigate the model's tendency to generate prohibited outputs, while the coefficient γ_c allows for fine-tuning the level of protection. Compared to (Vyas, Kakade, and Barak 2023), our approach does not necessitate training additional models and impacts the generation speed of the model during decoding. It offers a simple and practical method that can be plug-and-play applied to all training objectives and algorithms in attention-based models, to prevent the outputs of models from outputting copyrighted data.

In Section 4.1, we present the definition of Softmax Regression. In Section 4.2, we present the definition of Copyright Regression. In Section 4.3, we present the regularization of parameters for better optimization.

4.1 Softmax Regression

In (Deng, Li, and Song 2023), Softmax Regression applies a softmax function, denoted as f, to the product of the input matrix A and the parameter vector x. The training objective is then defined as minimizing the squared Euclidean distance between $f(x)$ and the target vector b, represented as $\langle f(x) - b, f(x) - b \rangle$. By optimizing this objective, Softmax Regression aims to gain insights into the learning process of the attention mechanism.

We define Softmax Regression as follows

Definition 2 (Softmax Regression in (Deng, Li, and Song 2023))**.** *Given a matrix $A \in \mathbb{R}^{n \times d}$, we define*

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

For the convenience of calculation, we define a intermediate operator $c(x)$ as follows

Definition 3. *Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$, let $f(x)$ be defined as Definition 2, we define $c(x) := f(x) - b$.*

We define the training objective of Softmax Regression as follows

Definition 4 (Training Objective of Softmax Regression in (Deng, Li, and Song 2023))**.** *Given matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$, let $c(x)$ be defined as Definition 3, we define $\ell(x) = \langle c(x), c(x) \rangle$.*

For more details to spell out the equivalence of Softmax regression and the training of transformers, please refer to Section 3 of (Chu, Song, and Yang 2023a).

4.2 Copyright Regression

Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$ that $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, and $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, where $A_1 \in \mathbb{R}^{n_1 \times d}$, $A_2 \in \mathbb{R}^{n_2 \times d}$, $b_1 \in \mathbb{R}^{n_1}$, $b_2 \in \mathbb{R}^{n_2}$ and $n = n_1 + n_2$. A_1, b_1 are the data that has copyright issue and A_2, b_2 are the data does not have copyright issue. Now to distinguish between train objective of A_1, b_1 and A_2, b_2, we follow what we did in Section 4.1. We first provide the definition of Softmax Regression function on Copyright Data as follows

Definition 5 (Softmax Regression function on Copyrighted Data)**.** *Given all data matrix $A \in \mathbb{R}^{n \times d}$ and copyrighted data matrix $A_1 \in \mathbb{R}^{n_1 \times d}$, we define*

$$f_1(x) := \langle \exp(A_{i,*} x), \mathbf{1}_n \rangle^{-1} \exp(Ax)$$

where $i \in [1, n_1]$ denote a integer.

Also, we provide the definition of the intermediate operator $c(x)$ as follows

Definition 6. *Given all data matrix $A \in \mathbb{R}^{n \times d}$ and copyrighted data matrix $A_1 \in \mathbb{R}^{n_1 \times d}$ and vector $b_1 \in \mathbb{R}^n$, let $f_1(x)$ be defined as Definition 5, we define $c_1(x) := f_1(x) - b_1$.*

Now we have officially provided our definition of Copyright Regression below, which can prevent language models from infringing copyright with controllable performance damage and without occupying more resources.

Definition 7. *We denote $\ell(x)$ as Definition 4. The function $c_1(x)$ is defined as Definition 6, and we denote $\ell_1(x) = \langle c_1(x), c_1(x) \rangle$ and $\ell_2(x) := \ell(x) - \ell_1(x)$. Let $\gamma_c > 0$ denote a parameter that controls loss related to copyright data.*

We consider the following copyright loss

$$L_{\mathrm{copyright}}(x) := 0.5\ell_1(x) + \gamma_c \cdot \ell_1(x)^{-1} + 0.5\ell_2(x)$$

Additionally, by adjusting the value of γ_c, one can easily control the learning of copyrighted data within the model. This flexibility allows for a more effective and data-sensitive approach to training language models.

4.3 Regularization

To make sure the stability during training, we add a regularization term on $L_{\mathrm{copright}}(x)$. We define L_{reg} as follows

Definition 8. *Given a matrix $A \in \mathbb{R}^{n \times d}$. Given a vector $w \in \mathbb{R}^n$, we define $W = \mathrm{diag}(w)$. We define $L_{reg} : \mathbb{R}^d \to \mathbb{R}$ as follows $L_{\mathrm{reg}} := 0.5 \|WAx\|_2^2$.*

After adding the regularization term, we define our final objective L as follows

Definition 9. *We denote $L_{\mathrm{copyright}}(x)$ as Definition 7, let L_{reg} be defined as Definition 8, then we define $L := L_{\mathrm{copyright}}(x) + L_{\mathrm{reg}}$.*

*Minimizing L is the **softmax regression on copyrighted data** problem.*

5 Optimization Properties of Objective Function L

The main contribution of this section involves addressing the convexity of the objective function L, which allows for more efficient and reliable optimization of L. This achievement not only enables us to optimize the objective more effectively but also validates the feasibility of utilizing Copyright Regression for achieving convergence in LLM (Language Model) training. For instance, we can leverage popular optimization algorithms such as gradient descent, Newton's method, and their variants to solve the optimization problem efficiently (see Section 8 in (Deng, Li, and Song 2023)).

In Section 5.1, we compute the gradient and hessian of our train objective. In Section 5.2, we show our result that the Hessian of our train objective is Positive Definite. In Section 5.3, we show our result that the Hessian of our train objective is Lipschitz. Thus, we can say our train objective L is convex.

5.1 Gradient and Hessian of L

In order to calculate the convergence and optimization of L, we first compute the ∇L and $\nabla^2 L$. We show our results as follows

Lemma 10 (Gradient of L)**.** *Given matrix $A \in \mathbb{R}^{n \times d}$ that $A = [A_1^\top \ A_2^\top]^\top$, where $A_1, A_2 \in \mathbb{R}^{n_2 \times d}$ and $n = n_1 + n_2$.*

Also, we are given a vector $b \in \mathbb{R}^n$ with $b = [b_1^\top \ b_2^\top]^\top$, where $b_1, b_2 \in \mathbb{R}^{n_2}$.

We denote $\ell_1(x)$ and $\ell_2(x)$ as Definition 7, denote L as Definition 9, denote $f(x)$ as Definition 2, denote $c(x)$ as Definition 3. Give a vector $w \in \mathbb{R}^n$, we define $W = \mathrm{diag}(w)$.

We have

$$\begin{aligned}\frac{\mathrm{d}L}{\mathrm{d}x} = & A_{*,i}^\top(-f(x)c(x)^\top f(x) + \mathrm{diag}(f(x))c(x)) \\ & + 2\gamma_c \ell_1(x)^{-2} \cdot A_{1_{*,i}}^\top (f_1(x)c_1(x)^\top f_1(x) \\ & - \mathrm{diag}(f_1(x))c_1(x)) + A^\top W^2 A x\end{aligned}$$

where $i \in [1, n]$ denote a integer.

For convenient, we define $B(x)$ and $B_c(x)$ ($B(x)$ function on copyrighted data)

Definition 11 (Definition 6.1 in (Deng, Li, and Song 2023))**.** *Given matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$ that $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, and $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, where $A_1 \in \mathbb{R}^{n_1 \times d}$, $A_2 \in \mathbb{R}^{n_2 \times d}$, $b_1 \in \mathbb{R}^{n_1}$, $b_2 \in \mathbb{R}^{n_2}$ and $n = n_1 + n_2$. A_1, b_1 are the data has copyright issue and A_2, b_2 are the data does not have copyright issue.*

Denote $f(x)$ as Definition 2, denote $c(x)$ as Definition 3, denote $f_1(x)$ as Definition 5, denote $c_1(x)$ as Definition 6. We define $B(x)$ as follows

$$\begin{aligned}B(x) = & \langle 3f(x) - 2b, f(x) \rangle \cdot f(x) f(x)^\top \\ & + \langle f(x) - b, f(x) \rangle \cdot \mathrm{diag}(f(x)) \\ & + \mathrm{diag}((2f(x) - b) \circ f(x)) \\ & + (b \circ f(x)) \cdot f(x)^\top + f(x) \cdot (b \circ f(x))^\top\end{aligned}$$

and then we also define $B_c(x)$ as follows

$$\begin{aligned}B_c(x) = & \langle 3f_1(x) - 2b_1, f_1(x) \rangle \cdot f_1(x) f_1(x)^\top \\ & + \langle f_1(x) - b_1, f_1(x) \rangle \cdot \mathrm{diag}(f_1(x)) \\ & + \mathrm{diag}((2f_1(x) - b_1) \circ f_1(x)) \\ & + (b_1 \circ f_1(x)) \cdot f_1(x)^\top + f_1(x) \cdot (b_1 \circ f_1(x))^\top\end{aligned}$$

With $B(x)$ and $B_c(x)$, we can abbreviate our compute result of Hessian of L as follows

Lemma 12 (Hessian of L)**.** *Given matrix $A \in \mathbb{R}^{n \times d}$ that $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, where $A_1, A_2 \in \mathbb{R}^{n_2 \times d}$ and $n = n_1 + n_2$. Also, we are given a vector $b \in \mathbb{R}^n$ with $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, where $b_1, b_2 \in \mathbb{R}^{n_2}$.*

Denote $\ell_1(x)$ and $\ell_2(x)$ as Definition 7, denote L as Definition 9, denote $f(x)$ as Definition 2, denote $c(x)$ as Definition 3, denote $B(x)$ and $B_c(x)$ be defined as Definition 11. Given a vector $w \in \mathbb{R}^n$, we define $W = \mathrm{diag}(w)$.

We have

$$\begin{aligned}& \frac{\mathrm{d}^2 L}{\mathrm{d}x_i \mathrm{d}x_i} \\ = & A_{*,i}^\top B(x) A_{*,i}^\top + A^\top W^2 A \\ & + 2\gamma_c \ell_1(x)^{-2} (16 \cdot \ell_1(x)^{-1} (A_{1_{*,i}}^\top (-f_1(x)c_1(x)^\top f_1(x) \\ & + \mathrm{diag}(f_1(x))c_1(x)))^2 - A_{1_{*,i}}^\top B_c(x) A_{1_{*,i}}^\top)\end{aligned}$$

where $i \in [0, n]$ denote a integer.

And we also have

$$\begin{aligned}& \frac{\mathrm{d}^2 L}{\mathrm{d}x_i \mathrm{d}x_j} \\ = & A_{*,i}^\top B(x) A_{*,j}^\top + A^\top W^2 A \\ & + 2\gamma_c \ell_1(x)^{-2} (16 \cdot \ell_1(x)^{-1} A_{1_{*,i}}^\top (-f_1(x)c_1(x)^\top f_1(x) \\ & + \mathrm{diag}(f_1(x))c_1(x)) \cdot A_{1_{*,j}}^\top (-f_1(x)c_1(x)^\top f_1(x) \\ & + \mathrm{diag}(f_1(x))c_1(x)) - A_{1_{*,i}}^\top B_c(x) A_{1_{*,j}}^\top)\end{aligned}$$

where $i, j \in [1, n]$ denote two integers, $i \neq j$.

5.2 Hessian of L is Positive Definite

After computing the Hessian of L, we now show our result that can confirm it is positive definite, which implies that $\nabla^2 L \succ 0$. Therefore, we have strong evidence that L satisfies the condition of convexity, which is a desirable property for optimization purposes.

Lemma 13 (Hessian is positive definite)**.** *Given matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$. Denote $\gamma \in (0, 1)$ a scalar. Given a vector w, denote $W = \mathrm{diag}(w) \in \mathbb{R}^{n \times n}$. We define $w_{i,i}^2$ as the i-th diagonal entry of matrix $W^2 \in \mathbb{R}^{n \times n}$. Let $l > 0$ denote a scalar.*

If for all $i \in [n]$, $w_i^2 \geq 8 + 200\gamma_c \gamma^{-3} + l/\sigma_{\min}(A)^2$, we have $\nabla^2 L \succeq l \cdot I_d$

5.3 Hessian of L is Lipschitz

We now show our result that confirms the Hessian of L is Lipschitz, which is a desirable property in optimization. This indicates that the second derivatives of L change smoothly within a defined range. By leveraging this Lipschitz property, we can employ gradient-based methods with guaranteed convergence rates and improved stability. Overall, this finding validates the feasibility of utilizing Copyright Regression for achieving convergence in LLM (Language Model) training.

Lemma 14 (Hessian is Lipschitz-continuous)**.** *Denote $R \geq 4$ denote a scalar. Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$, $\|A\| \leq R$, $\|b\|_2 \leq 1$. Given $x, y \in \mathbb{R}^d$ be two vector parameter for Copyright Regression with conditions $\|x\|_2 \leq R$, $\|y\|_2 \leq R$ and $\|A(x - y)\|_\infty \leq 0.01$. Let L be defined as Definition 9, let $\gamma \in (0, 1)$, let $\beta \in (0, 0.1)$. Denote $H(x) := \nabla^2 L(x)$. Then,*

$$\begin{aligned}& \|H(x) - H(y)\| \\ \leq & (13344\gamma_c + 2)\gamma^{-4}\beta^{-2} n^{1.5} \exp(40R^2) \|x - y\|_2\end{aligned}$$

6 Optimization and Copyright Protection Guarantees

We have already established the convexity of the training objective L in Section 5, providing a strong foundation to confidently pursue the global optimal value of L through optimization techniques. Now we present the main results of this

paper: 1) the minimization guarantee of L, 2) the copyright protection efficiency of Copyright Regression.

Firstly, in Section 6.1, our objective is to minimize L to its optimal value, ensuring that we achieve the most favorable outcome in terms of our training process. The minimization guarantee of L confirms our main result on optimization of Copyright Regression, it also demonstrates the ease of use of Copyright Regression, which can be optimized on any attention-based model. At the same time, denote x^* as the optimal solution of training objective L, analyzing $L(x^*)$'s performance on copyright data can help us to understand how the trained Copyright Regression can avoid copyright infringement.

Secondly, in Section 6.2, we aim to demonstrate that the optimal L provides robust protection for its outputs, safeguarding them from potential copyright infringement. By delineating this boundary, we can quantitatively assess the extent to which Copyright Regression preserves the integrity and exclusivity of copyrighted content. This analysis will provide valuable insights into the effectiveness of our approach and its ability to strike a balance between data protection and the need for authorized access.

6.1 Minimizing Loss Guarantee

We provide our minimum training objective theorem below.

Theorem 15 (Minimizing training objective L)**.** *Suppose we have matrix $A \in \mathbb{R}^{n \times d}$ and $A_1 \in \mathbb{R}^{n_1 \times d}$, $n_1 \leq n$, vector $b, w \in \mathbb{R}^n$. Let L be defined as Definition 9, denote x^* as the optimal solution of L where $g(x^*) = \mathbf{0}_d$ and $\|x^*\| \leq R$. Denote $R \geq 10$ be a positive scalar. Denote $M = n^{1.5} \exp(60R^2)$, Let x_0 be denoted as an initial point where $M\|x_0 - x^*\|_2 \leq 0.1l$, where $l > 0$ denoted a scalar.*

For any accuracy $\epsilon \in (0, 0.1)$ and any failure probability $\delta \in (0, 0.1)$, there exists a randomized algorithm, with probability $1 - \delta$, it runs $T = \log(\|x_0 - x^*\|_2/\epsilon)$ iteration and outputs a vector $\widetilde{x} \in \mathbb{R}^d$ such that $\|\widetilde{x} - x^*\| \leq \epsilon$ and the time cost of each iteration is

$$O((\mathrm{nnz}(A) + d^w) \cdot \mathrm{poly}(\log(n/\delta)))$$

Here w is the exponent of matrix multiplication. Currently $w \approx 2.373$.

6.2 L is τ_c-Copyright-Protected

Now we provide a boundary that illustrates the efficacy of Copyright Regression in safeguarding copyrighted data, while also addressing the criteria outlined in Definition 1, which serves as our definition of copyright protection in this paper.

We set $\ell(x)$ in Definition 4 as a ℓ_2 metric for measuring parameter x on learning data A. Now we present our result to confirm that training using our Copyright Regression method can ensure that the model's outputs do not infringe copyright. Specifically, we can assert that the trained model L is protected against copyright infringement with a threshold of τ_c based on Theorem 16 below.

Theorem 16. *Let x^* be denoted the optimal parameter on Copyright Regression. We define $\ell(x)$ as Definition 4, denote $\ell(x)$ as the original train objective of Softmax Regression. Denote $\epsilon_2 \in (0, 0.1)$ a scalar. Denote $\tau_c := \sqrt{2\gamma_c}/n_1 - \epsilon_2/n_2$, we have $\frac{\ell_1(x^*)}{n_1} \geq \tau_c + \frac{\ell_2(x^*)}{n_2}$, so x^* in Copyright Regression is τ_c-Copyright-Protected.*

Now we have provided evidence of the copyright protection achieved through training under the Copyright Regression objective. This method has been rigorously proven and offers complete control over copyright infringement. However, concerns may arise regarding the potential impact of the Copyright Regression approach on the model's overall performance, particularly when copyright data includes high-quality novels and images that contribute significantly to the model's performance. In fact, language models cannot remember all their training data. Their training loss has a considered range instead of equal to 0. Based on this, we only need to let model's performance on copyrighted data be different from model's performance on other data, even if this difference is very small, then we can ascertain whether the model has access to these copyright data during output generation and intentionally avoids outputting them. The difference, namely τ, can be easily controlled by adjusting the value of γ_c and n_1/n, we will continue to explain that why we say this in Section 7.

7 Experiment

In order to evaluate and demonstrate the effectiveness of our proposed Copyright Regression approach, we conducted extensive experiments using Softmax Regression. By varying the values of n_1 (representing the number of data instances with copyright issues) and γ_c (the coefficient used to control the Copyright Regression), we compared the results against a baseline model. The experimental findings clearly indicate the efficacy of our method in providing effective copyright protection.

In Section 7.1, we provided the details of our experiment. In Section 7.2, we provided experimental results and analyzed the effectiveness of Copyright Regression.

7.1 Setup

Model and Baseline. We applied our approach Copyright Regression on a pretrained generative language model GPT-2 (Radford et al. 2019), namely **CR-GPT-2**. To evaluate the effectiveness of our approach, we conduct a comparative analysis against a baseline method referred to as **GPT-2** where we directly evaluate GPT-2 model on copyright issues without any additional training.

Dataset. We employed an open-source dataset Wikitext2 (Merity et al. 2016) to fine-tune our model, and evaluate the performance of our model on its test set. Specifically, we randomly selected some data as fake copyrighted data to simulate real-world cases, and denote n_1/n as the proportion of copyrighted data in the whole training dataset.

hyper-parameters. To assess the influence of copyright data with different proportions during training, we varied the value of n_1/n to be $n_1/n \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$. Additionally, to evaluate the impact of different values of γ_c on copyright protection, we consider γ_c values of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. In addition, we fixed random seeds

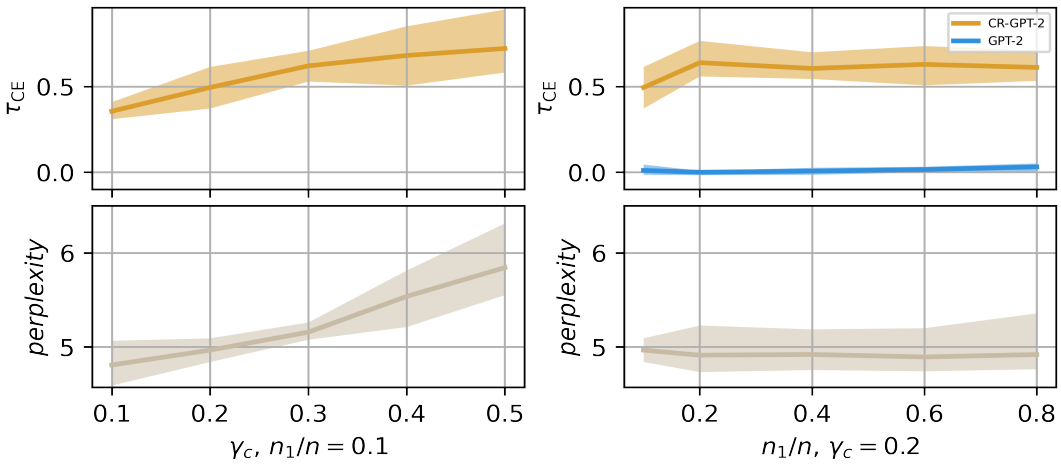

Figure 1: Copyright Regression experiment result

and conducted multiple experiments to record the maximum, minimum, and average values to ensure stable results were obtained.

Metrics. We use two evaluation metrics, including $\tau_{\mathrm{CE}} = \frac{L_{\mathrm{CE}_1}}{n_1} - \frac{L_{\mathrm{CE}_2}}{n_2}$ and "Perplexity", where the L_{CE_1} denotes the sum of Cross Entropy loss on the copyright dataset and the L_{CE_2} denotes the sum of Cross Entropy loss on the non-copyright dataset.

7.2 Results and Analysis

Impact of γ_c. The left top image of Figure 1 depicts the relationship between the variables γ_c and the difference metric τ_{CE}. In this experiment, we set the value of $n_1/n = 0.1$. Remarkably, the observed trend aligns closely with the result we derived in Section 6.2. Our derived result, stated as $\tau_{\mathrm{CE}} = \frac{L_{\mathrm{CE}_1}}{n_1} - \frac{L_{\mathrm{CE}_2}}{n_2} \geq \frac{\sqrt{2\gamma_c}}{n_1} - \frac{\epsilon_2}{n_2}$, affirms that our Copyright Regression approach effectively encourages the model to avoid copyright infringement while still maintaining a controllable level of performance degradation. Furthermore, the left bottom image of Figure 1 indicates inappropriate γ_c may have a slight negative impact on the overall performance of the model. But in the case of model convergence, this impact is limited and can be controlled by γ_c.

Impact of the proportion of copyright data. n_1/n impacts on model performance are illustrated in the right top and right bottom images of Figure 1. This image showcases the relationship between n_1/n and the difference metric τ_{CE}. In this experiment, we set the value of $\gamma_c = 0.2$. The findings show that as the proportion n_1/n increases, there is basically no significant change in the model's perplexity on the test set and the τ_{CE}. Especially, the right top image of Figure 1 shows the comparison between **CR-GPT-2** and baseline **GPT-2** on copyright data with metric τ_{CE}. The τ_{CE} of **CR-GPT-2** is stable above 0.5 while the τ_{CE} of **GPT-2** is around 0. This finding provides compelling evidence that our Copyright Regression approach effectively prevents the occurrence of the "infinite monkey" phenomenon, ensuring that the model's outputs consistently avoid copyright infringement. By maintaining a reliable level of performance on copyright data, our method demonstrates its ability to strike a crucial balance between performance and copyright protection.

8 Conclusion

Our work shows that the training of transformers can be viewed as a softmax regression problem. We provide a notion of copyright regression, which encourages regression functions to avoid outputting copyrighted data. Then, we combine the two to perform copyright regression on the softmax function, which allows us to train transformers in a way that avoids outputting copyright data. The main idea to solve copyright regression on the softmax function, was to show that the copyright regression problem is convex and that the Hessian is Lipschitz. This guarantees that gradient descent methods will have guaranteed convergence to the optimal solution with good stability properties. The experimental results of applying our method on GPT-2 show that our algorithm performs well in preventing copyright issues.

Acknowledgements

We would like to thank Yanxi Shen and the anonymous reviewers for helpful discussions and feedback.

References

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019a. A convergence theory for deep learning via over-parameterization. In *ICML*.

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019b. On the convergence rate of training recurrent neural networks. *NeurIPS*, 32.

Alman, J.; Liang, J.; Song, Z.; Zhang, R.; and Zhuo, D. 2023. Bypass exponential time preprocessing: Fast neural network training via weight-data correlation preprocessing. In *NeurIPS*.

Alman, J.; and Song, Z. 2023. Fast Attention Requires Bounded Entries. *arXiv preprint arXiv:2302.13214*.

Anil, R.; Dai, A. M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Arora, S.; and Goyal, A. 2023. A Theory for Emergence of Complex Skills in Language Models. *arXiv preprint arXiv:2307.15936*.

Bao, F.; Nie, S.; Xue, K.; Cao, Y.; Li, C.; Su, H.; and Zhu, J. 2023. All are worth words: A vit backbone for diffusion models. In *CVPR*, 22669–22679.

BARD. 2023. Try BARD, an AI experiment by Google. *Google*.

Brand, J. v. d.; Peng, B.; Song, Z.; and Weinstein, O. 2020. Training (overparametrized) neural networks in near-linear time. *arXiv preprint arXiv:2006.11648*.

Brand, J. v. d.; Song, Z.; and Zhou, T. 2023. Algorithm and Hardness for Dynamic Attention Maintenance in Large Language Models. *arXiv preprint arXiv:2304.02207*.

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

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

Cai, T.; Gao, R.; Hou, J.; Chen, S.; Wang, D.; He, D.; Zhang, Z.; and Wang, L. 2019. Gram-gauss-newton method: Learning overparameterized neural networks for regression problems. *arXiv preprint arXiv:1905.11675*.

Cao, H.; Wang, J.; Ren, T.; Qi, X.; Chen, Y.; Yao, Y.; and Zhang, L. 2022. Exploring vision transformers as diffusion learners. *arXiv preprint arXiv:2212.13771*.

Cao, Y.; and Gu, Q. 2019. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *NeurIPS*, 32.

ChatGPT. 2022. Optimizing Language Models for Dialogue. *OpenAI Blog*.

Chen, B.; Dao, T.; Winsor, E.; Song, Z.; Rudra, A.; and Ré, C. 2021. Scatterbrain: Unifying sparse and low-rank attention. *NeurIPS*, 34: 17413–17426.

Chen, B.; Liu, Z.; Peng, B.; Xu, Z.; Li, J. L.; Dao, T.; Song, Z.; Shrivastava, A.; and Re, C. 2020. Mongoose: A learnable lsh framework for efficient neural network training. In *International Conference on Learning Representations*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Chu, T.; Song, Z.; and Yang, C. 2023a. Fine-tune language models to approximate unbiased in-context learning. *arXiv preprint arXiv:2310.03331*.

Chu, T.; Song, Z.; and Yang, C. 2023b. How to Protect Copyright Data in Optimization of Large Language Models? *arXiv preprint arxiv:2308.12247*.

Deng, Y.; Li, Z.; Mahadevan, S.; and Song, Z. 2023. Zero-th Order Algorithm for Softmax Attention Optimization. *arXiv preprint arXiv:2307.08352*.

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

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Du, S. S.; Zhai, X.; Poczos, B.; and Singh, A. 2018. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*.

for the 9th circuits, U. S. C. 2022. Copying—Access and Substantial Similarity. https://www.ce9.uscourts.gov/jury-instructions/node/274/. Accessed: 2022-12-31.

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

Gao, Y.; Song, Z.; and Yang, X. 2023. Differentially private attention computation. *arXiv preprint arXiv:2305.04701*.

Gao, Y.; Song, Z.; Yang, X.; and Zhang, R. 2023. Fast Quantum Algorithm for Attention Computation. *arXiv preprint arXiv:2307.08045*.

Gao, Y.; Song, Z.; and Yin, J. 2023. GradientCoin: A Peer-to-Peer Decentralized Large Language Models. *arXiv preprint arXiv:2308.10502*.

Gillotte, J. L. 2019. Copyright infringement in ai-generated artworks. *UC Davis L. Rev.*, 53: 2655.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *PAMI*, 45(1): 87–110.

Hattenbach, B.; and Glucoft, J. 2015. Patents in an era of infinite monkeys and artificial intelligence. *Stan. Tech. L. Rev.*, 19: 32.

He, X.; Xu, Q.; Lyu, L.; Wu, F.; and Wang, C. 2022a. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10758–10766.

He, X.; Xu, Q.; Zeng, Y.; Lyu, L.; Wu, F.; Li, J.; and Jia, R. 2022b. Cater: Intellectual property protection on text generation apis via conditional watermarks. *NeurIPS*, 35: 5431–5445.

Hristov, K. 2016. Artificial intelligence and the copyright dilemma. *Idea*, 57: 431.

Huang, B.; Li, X.; Song, Z.; and Yang, X. 2021. Fl-ntk: A neural tangent kernel-based framework for federated learning analysis. In *ICML*.

Ignat, O.; Jin, Z.; Abzaliev, A.; Biester, L.; Castro, S.; Deng, N.; Gao, X.; Gunal, A.; He, J.; Kazemi, A.; et al. 2023. A PhD Student's Perspective on Research in NLP in the Era of Very Large Language Models. *arXiv preprint arXiv:2305.12544*.

Ji, Z.; and Telgarsky, M. 2019. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. *arXiv preprint arXiv:1909.12292*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. *arXiv preprint arXiv:2306.02561*.

Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; and Goldstein, T. 2023. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Lee, J. D.; Shen, R.; Song, Z.; Wang, M.; and Yu, Z. 2020. Generalized leverage score sampling for neural networks. *NeurIPS*.

Li, Y.; and Liang, Y. 2018. Learning overparameterized neural networks via stochastic gradient descent on structured data. *NeurIPS*.

Li, Z.; Song, Z.; and Zhou, T. 2023. Solving regularized exp, cosh and sinh regression problems. *arXiv preprint arXiv:2303.15725*.

Liu, H.; Li, Z.; Hall, D.; Liang, P.; and Ma, T. 2023a. Sophia: A Scalable Stochastic Second-order Optimizer for Language Model Pre-training. *arXiv preprint arXiv:2305.14342*.

Liu, Z.; Wang, J.; Dao, T.; Zhou, T.; Yuan, B.; Song, Z.; Shrivastava, A.; Zhang, C.; Tian, Y.; Re, C.; et al. 2023b. Deja vu: Contextual sparsity for efficient llms at inference time. In *ICML*.

Malladi, S.; Gao, T.; Nichani, E.; Damian, A.; Lee, J. D.; Chen, D.; and Arora, S. 2023. Fine-Tuning Language Models with Just Forward Passes. *arXiv preprint arXiv:2305.17333*.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Munteanu, A.; Omlor, S.; Song, Z.; and Woodruff, D. 2022. Bounding the width of neural networks via coupled initialization a worst case analysis. In *ICML*, 16083–16122.

Noci, L.; Anagnostidis, S.; Biggio, L.; Orvieto, A.; Singh, S. P.; and Lucchi, A. 2022. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *NeurIPS*, 35.

Oymak, S.; and Soltanolkotabi, M. 2020. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 84–105.

Panigrahi, A.; Malladi, S.; Xia, M.; and Arora, S. 2023. Trainable Transformer in Transformer. *arXiv preprint arXiv:2307.01189*.

Qin, L.; Song, Z.; and Yang, Y. 2023. Efficient SGD Neural Network Training via Sublinear Activated Neuron Identification. *arXiv preprint arXiv:2307.06565*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Research, O. 2021. DALL·E: Creating images from text. https://openai.com/research/dall-e/. Accessed: 2021-01-05.

Research, O. 2022. DALL·E 2 pre-training mitigations. https://openai.com/research/dall-e-2-pre-training-mitigations. Accessed: 2022-01-28.

Rücklé, A.; Geigle, G.; Glockner, M.; Beck, T.; Pfeiffer, J.; Reimers, N.; and Gurevych, I. 2020. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*.

Sag, M. 2018. The new legal landscape for text mining and machine learning. *J. Copyright Soc'y USA*, 66: 291.

Sanford, C.; Hsu, D.; and Telgarsky, M. 2023. Representational Strengths and Limitations of Transformers. *arXiv preprint arXiv:2306.02896*.

Song, Z.; and Yang, X. 2019. Quadratic suffices for overparametrization via matrix chernoff bound. *arXiv preprint arXiv:1906.03593*.

Song, Z.; Zhang, L.; and Zhang, R. 2021. Training multi-layer over-parametrized neural network in subquadratic time. *arXiv preprint arXiv:2112.07628*.

Sun, A. Y.; Zemour, E.; Saxena, A.; Vaidyanathan, U.; Lin, E.; Lau, C.; and Mugunthan, V. 2023. Does fine-tuning GPT-3 with the OpenAI API leak personally-identifiable information? *arXiv preprint arXiv:2307.16382*.

Svyatkovskiy, A.; Deng, S. K.; Fu, S.; and Sundaresan, N. 2020. Intellicode compose: Code generation using transformer. In *ESEC-FSE*.

Tay, Y.; Bahri, D.; Metzler, D.; Juan, D.-C.; Zhao, Z.; and Zheng, C. 2021. Synthesizer: Rethinking self-attention for transformer models. In *ICML*, 10183–10192. PMLR.

Tay, Y.; Dehghani, M.; Abnar, S.; Shen, Y.; Bahri, D.; Pham, P.; Rao, J.; Yang, L.; Ruder, S.; and Metzler, D. 2020. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Vyas, N.; Kakade, S.; and Barak, B. 2023. Provable copyright protection for generative models. *arXiv preprint arXiv:2302.10870*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35: 24824–24837.

Wu, J.; Fu, R.; Fang, H.; Zhang, Y.; and Xu, Y. 2023a. Medsegdiff-v2: Diffusion based medical image segmentation with transformer. *arXiv preprint arXiv:2301.11798*.

Wu, J.; Yu, T.; Wang, R.; Song, Z.; Zhang, R.; Zhao, H.; Lu, C.; Li, S.; and Henao, R. 2023b. InfoPrompt: Information-Theoretic Soft Prompt Tuning for Natural Language Understanding. In *NeurIPS*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xu, Z.; Zhang, Y.; Andrew, G.; Choquette-Choo, C. A.; Kairouz, P.; McMahan, H. B.; Rosenstock, J.; and Zhang, Y. 2023. Federated Learning of Gboard Language Models with Differential Privacy. *arXiv preprint arXiv:2305.18465*.

Zelikman, E.; Huang, Q.; Liang, P.; Haber, N.; and Goodman, N. D. 2023. Just One Byte (per gradient): A Note on Low-Bandwidth Decentralized Language Model Finetuning Using Shared Randomness. *arXiv preprint arXiv:2306.10015*.

Zhang, G.; Martens, J.; and Grosse, R. B. 2019. Fast convergence of natural gradient descent for over-parameterized neural networks. *NeurIPS*, 32.

Zhang, J.; Karimireddy, S. P.; Veit, A.; Kim, S.; Reddi, S.; Kumar, S.; and Sra, S. 2020a. Why are adaptive methods good for attention models? *NeurIPS*, 33: 15383–15393.

Zhang, L. 2022. *Speeding up optimizations via data structures: Faster search, sample and maintenance*. Ph.D. thesis, Master's thesis, Carnegie Mellon University.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, Y.; Plevrakis, O.; Du, S. S.; Li, X.; Song, Z.; and Arora, S. 2020b. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. *NeurIPS*, 33: 679–688.

Zhao, H.; Panigrahi, A.; Ge, R.; and Arora, S. 2023. Do Transformers Parse while Predicting the Masked Word? *arXiv preprint arXiv:2303.08117*.

Zou, D.; and Gu, Q. 2019. An improved analysis of training over-parameterized deep neural networks. *NeurIPS*, 32.