Secured Weight Release for Large Language Models via Taylor Expansion

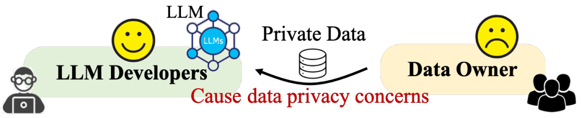
Anonymous ACL submission

Abstract

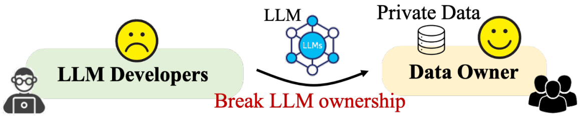
Ensuring the security of released large language models (LLMs) poses a significant dilemma, as existing mechanisms either compromise ownership rights or raise data privacy concerns. To address this dilemma, we introduce TaylorMLP to protect the ownership of released LLMs and prevent their abuse. Specifically, TaylorMLP preserves the ownership of LLMs by transforming the weights of LLMs into parameters of Taylor-series. Instead of releasing the original weights, developers can release the Taylor-series parameters with users, thereby ensuring the security of LLMs. Moreover, TaylorMLP can prevent abuse of LLMs by adjusting the generation speed. It can induce low-speed token generation for the protected LLMs by increasing the terms in the Taylor-series. This intentional delay helps LLM developers prevent potential large-scale unauthorized uses of their models. Empirical experiments across five datasets and three LLM architectures demonstrate that TaylorMLP induces over $4\times$ increase in latency, producing the tokens precisely matched with original LLMs. Subsequent defensive experiments further confirm that TaylorMLP effectively prevents users from reconstructing the weight values based on downstream datasets. The source code is available at `https://anonymous.4open.science/r/TaylorMLP-6EBE`.

1 Introduction

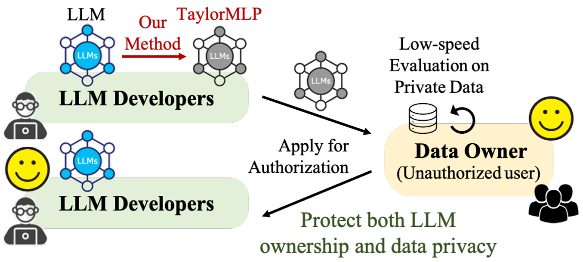
Training large language models (LLMs) is an expensive and complex endeavor, requiring substantial investments in financial and computational resources (Wei et al., 2021). In particular, it requires a huge collection of high-quality datasets from diverse domains, which proves to be labor-intensive and time-consuming. However, once released, the LLMs may face the risks of abuse such as unethical or commercial exploitation (Weidinger et al., 2021). This raises the critical and urgent challenge of ensuring the security of released LLMs.



(a) API released LLMs cause data privacy concerns.



(b) Open-source breaks the ownership of LLMs.


LLM
Our Method
TaylorMLP
LLMs
LLM Developers
Low-speed Evaluation on Private Data
Apply for Authorization
Data Owner (Unauthorized user)
LLM Developers
Protect both LLM ownership and data privacy

(c) TaylorMLP protects LLM ownership and data privacy.

Figure 1: Existing mechanisms for releasing LLMs: (a) API release and (b) open-source. We propose (c) TaylorMLP to protect the ownership of released LLMs.

In this work, we explore the security aspects of existing mechanisms for releasing LLMs. This process involves developers providing access to their models for general users, with criteria restricting access to ethical and non-commercial purposes (Ong et al., 2024). Currently, there are two primary mechanisms for releasing LLMs: ***API Release*** (Caruccio et al., 2024) and ***Open-source*** (Raffel et al., 2020). With the release of APIs, user authorization is managed through specific API keys. Examples of this mechanism include the ChatGPT(Achiam et al., 2023), Gemini (Team et al., 2023), and Claude models (Caruccio et al., 2024). In these cases, users do not have access to the architectures or weights of the LLMs. Instead, they share their private data to the developers and receive the processed results, as shown in Figure 1 (a). Consequently, ***the API release mechanism may cause the data privacy concerns*** (Yang et al., 2023). On the other hand, the open-source mechanism fully shares the LLM weights with users, as shown in Figure 1 (b) (Wolf et al., 2019). Common examples include Llama (Touvron et al., 2023), Mixtral (Jiang et al., 2024), and Phi (Team et al., 2024). While the open-source mechanism ensures the safety of users' private data, it also raises significant challenges for developers. Specifically, ***the open-source mechanism can break the ownership of LLMs***, as users gain control over the models and can use them for any purpose, even those prohibited by the developers. For example, users may exploit LLMs for unethical or commercial purposes, both of which may be prohibited by the developers. This potential loss of control and ownership may lead many model developers to avoid sharing their models (Zha et al., 2023; Sharir et al., 2020). Therefore, ***there is a dilemma between protecting ownership rights and ensuring open access to LLMs***, posing a significant challenge for developers.

Can we solve the access dilemma for LLMs?

We propose Taylor-series MLP (TaylorMLP) to protect the ownership of released LLMs and prevent their potential abuse. As illustrated in Figure 1 (c), TaylorMLP addresses the dilemma by securing the weights of LLMs into latent parameters. By sharing these parameters instead of the original weights with users, developers can maintain ownership of their models while allowing users to harness the model's performance. Specifically, TaylorMLP converts the original weights into parameters of the Taylor series. Our empirical experiments confirm that it is infeasible to reconstruct the original weights from these Taylor series parameters, thereby ensuring the security of the model's parameters and allowing safe access to its functional capabilities without full model exposure.

Can we prevent unauthorized users from exploiting the LLM for their own purposes?

To prevent unauthorized users from abusing the protected LLMs, TaylorMLP allows developers to control the utility of the LLM by adjusting the speed of token generation. Specifically, TaylorMLP induces ***low-speed token generation*** for the secured LLMs by increasing the terms in the Taylor-series. It significantly increases the number of floating-point operations required for the generation process, leading to a notable increase in latency. Our empirical studies show that TaylorMLP induces more than $4\times$ increases in latency, while maintaining the produced tokens precisely matched with original LLMs. This intentional delay helps developers prevent the potential large-scale unauthorized use of their released LLMs.

How does TaylorMLP perform in practice?

To evaluate TaylorMLP, we conducted experiments across five datasets: TruthfulQA, MathQA, MMLU, OpenbookQA, and Wikitext-2; and three different LLM architectures: Llama-3-8B, Mistral-7B, and Phi-2. The experimental results demonstrate that TaylorMLP fully retains the accuracy and chat capabilities of original LLMs, while inducing $\mathbf{4}\times\sim\mathbf{8}\times$ increases in latency of token generation. Subsequent defensive experiments further confirm that TaylorMLP effectively prevents users from reconstructing the weight values based on downstream datasets. In summary, our work makes the following contributions:

- **Preserving LLM Ownership.** TaylorMLP preserves the ownership of LLM by transforming the weights of LLMs into parameters of Taylor-series. It is infeasible to reconstruct the weights from the Taylor-series parameters.

- **Preventing Abuse.** TaylorMLP prevents abuse of LLMs by adjusting the generation speed. It induces low-speed token generation process, preventing the potential large-scale unauthorized use of the protected LLMs.

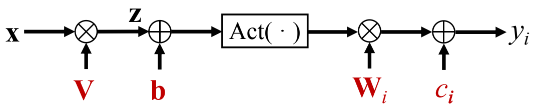
- **Evaluation.** Experiment results across five datasets and three LLM architectures show that TaylorMLP retains the accuracy and chat capabilities while preventing reconstruction of the weight values based on downstream datasets.

2 Architecture of Transformers

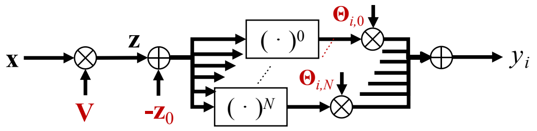
We focus on Transformer-based LLMs (Vaswani et al., 2017; Brown et al., 2020) to develop the method of LLM protection. A transformer block consists of an attention layer and MLP layer. An MLP layer is a pipeline of a linear layer, activation function, and another linear layer, whose architecture is shown in Figure 2 (a). Let $\mathbf{V}, \mathbf{b}$ and $\mathbf{W}, \mathbf{c}$ denote the weight matrixes of the two linear layers. Given the input tensor $\mathbf{x}$ to the MLP layer, the output value in the i-th dimension is given by

$$y_i = \big\langle \mathbf{W}_i, \mathrm{Act}(\mathbf{z} + \mathbf{b}) \big\rangle + c_i, \tag{1}$$

where $\mathbf{z} = \mathbf{V}\mathbf{x}$; $\langle \bullet, \bullet \rangle$ denotes the inner product; and $\mathbf{W}_i$ and c_i denote the i-th row and i-th ele-



(a) Original MLP layers



(b) TaylorMLP layers

Figure 2: (a) Original MLP layers parameterized by $\mathbf{V}, \mathbf{b}, \mathbf{W}$, and $\mathbf{c}$. (b) TaylorMLP layers. TaylorMLP converts $\mathbf{b}, \mathbf{W}_i$ and c_i into $\{\mathbf{\Theta}_{i,0}, \cdots, \mathbf{\Theta}_{i,N}\}$ for securing their values. This process is irreversible.

ment of $\mathbf{W}$ and $\mathbf{c}$, respectively. In this work, we introduce a method to secure the weights of MLP layers within the transformer architecture. This can be widely applied to powerful LLMs, such as Llama (Touvron et al., 2023), Phi (Gunasekar et al., 2023), and Gemma (Team et al., 2024).

3 Taylor-series MLP (TaylorMLP)

The framework of TaylorMLP is shown in Figure 2 (b). Intuitively, TaylorMLP transforms the weight matrices $\mathbf{b}, \mathbf{W}$ and $\mathbf{c}$ into latent parameters $\mathbf{\Theta}_{i,0}, \cdots, \mathbf{\Theta}_{i,N}$. In this way, TaylorMLP generates output tokens without needing the original weight values. In this section, we first describe the weight transformation of TaylorMLP. Then, we demonstrate the outputs of TaylorMLP theoretically converges to the original MLPs. Finally, we discuss the benefits of TaylorMLP to LLM community.

3.1 Securing MLP Weights by Taylor-series

TaylorMLP transforms the weight matrix $\mathbf{b}, \mathbf{W}$ and $\mathbf{c}$ into the latent space based on the Taylor Expansion Theory. Specifically, based on a local embedding $\mathbf{z}_0$, the $\mathrm{Act}(\mathbf{z}+\mathbf{b})$ term in Equation (1) can be reformulated into the Taylor-series as follows:

$$\mathrm{Act}(\mathbf{z}+\mathbf{b}) \approx \sum_{n=0}^{N} \frac{\mathrm{Act}^{(n)}(\mathbf{z}_0+\mathbf{b})}{n!} \odot (\mathbf{z}-\mathbf{z}_0)^n, \quad (2)$$

where $(\mathbf{z}-\mathbf{z}_0)^n$ indicates the element-wise power of n; $\mathrm{Act}^{(n)}(\bullet)$ denotes the n-order derivative of $\mathrm{Act}(\bullet)$. Representative activation functions for the Transformers are $\mathrm{GELU}(\bullet)$ (Vaswani et al., 2017) and $\mathrm{SiLU}(\bullet)$ (Touvron et al., 2023), whose n-order derivative are given in Appendixes A and B.

Following the Taylor-series in Equation (2), the forward pass of MLP layers can be reformulated to eliminate the original weight matrices. Specifically, by applying the Taylor expansion of $\mathrm{Act}(\mathbf{z}+\mathbf{b})$ from Equation (2) to Equation (1), the forward pass can be reformulated to:

$$\begin{aligned} y_i &\approx \left\langle \mathbf{W}_i, \sum_{n=0}^{N} \frac{\mathrm{Act}^{(n)}(\mathbf{z}_0+\mathbf{b})}{n!} \odot (\mathbf{z}-\mathbf{z}_0)^n \right\rangle + c_i \\ &= \sum_{n=0}^{N} \left\langle \mathbf{W}_i \odot \mathrm{Act}^{(n)}(\mathbf{z}_0+\mathbf{b})(n!)^{-1}, (\mathbf{z}-\mathbf{z}_0)^n \right\rangle, \\ &\triangleq \sum_{n=0}^{N} \left\langle \mathbf{\Theta}_{i,n}, (\mathbf{z}-\mathbf{z}_0)^n \right\rangle, \quad (3) \end{aligned}$$

where $\{\mathbf{\Theta}_{i,0}, \cdots, \mathbf{\Theta}_{i,N}\}$ denote the parameters in the latent space; and $\mathbf{\Theta}_{i,n}$ depends on the n-order derivative $\mathrm{Act}^{(n)}(\mathbf{z}_0+\mathbf{b})$, as given by

$$\begin{aligned} \mathbf{\Theta}_{i,0} &= \mathbf{W}_i \odot \mathrm{Act}(\mathbf{z}_0+\mathbf{b}) + c_i \\ \mathbf{\Theta}_{i,n} &= \mathbf{W}_i \odot \mathrm{Act}^{(n)}(\mathbf{z}_0+\mathbf{b})(n!)^{-1}. \quad (4) \end{aligned}$$

After the transformation, TaylorMLP eliminates the need for $\mathbf{b}$, $\mathbf{W}$, and $\mathbf{c}$ in the generation process. Moreover, it cannot reconstruct $\mathbf{b}$, $\mathbf{W}$, and $\mathbf{c}$ from $\mathbf{\Theta}_{i,0}, \cdots, \mathbf{\Theta}_{i,N}$, thereby securing the original weight matrices. TaylorMLP can be applied to protect the MLP layers within LLMs.

3.2 Estimating the Local Embedding $\mathbf{z}_0$

We clarify the local embedding $\mathbf{z}_0$. Specifically, to minimize the difference between the outputs of TaylorMLP and original MLPs, we minimize the difference between the two sides of Equation (2). This is equivalent to minimizing the distance between $\mathbf{z}_0$ and the embedding $\mathbf{z}$, where $\mathbf{z} = \mathbf{V}\mathbf{x}$ for the testing input $\mathbf{x}$. We address this problem by taking $\mathbf{x}$ from large-scale datasets $\mathcal{D}$ and applying the following solution:

$$\begin{aligned} \mathbf{z}_0^* &= \arg\min \max_{\mathbf{x}\sim\mathcal{D}} ||\mathbf{z}-\mathbf{z}_0||_1 \quad (5) \\ &= \frac{1}{2}(\mathbf{z}_{\max} + \mathbf{z}_{\min}), \quad (6) \end{aligned}$$

where the i-th element of $\mathbf{z}_{\max}$ takes the value of $\max_{\mathbf{x}\sim\mathcal{D}} \mathbf{x}^\top \mathbf{V}[:, i]$; that of $\mathbf{z}_{\min}$ takes $\min_{\mathbf{x}\sim\mathcal{D}} \mathbf{x}^\top \mathbf{V}[:, i]$. The optimal $\mathbf{z}_0^*$ can be generally effective for downstream tasks if $\mathcal{D}$ is sufficiently large. In our work, we take the large-scale pile datasets[1] as $\mathcal{D}$ for the estimation of $\mathbf{z}_0$.

3.3 Algorithm

The algorithm of TaylorMLP is given in Algorithm 1. Specifically, given the weights $\mathbf{b}, \mathbf{W}$,

---

[1] `https://huggingface.co/datasets/mit-han-lab/pile-val-backup`

Algorithm 1 Transforming MLP to TaylorMLP.

```
Input: MLP($\bullet \mid \mathbf{V}, \mathbf{b}, \mathbf{W}, \mathbf{c}$) and $\mathbf{z}_0$
Output: TaylorMLP($\bullet|\mathbf{V}$,$\mathbf{z}_0$,$\{\mathbf{\Theta}_{i,0}$,$\cdots$,$\mathbf{\Theta}_{i,N}\}_{i=1}^{D}$)
1: for $i := 1$ to $D$ do
2:     $\mathbf{W}_i$ and $c_i$ take the $i$-th row and $i$-th element
       of $\mathbf{W}$ and $\mathbf{c}$, respectively.
3:     $\mathbf{\Theta}_{i,0} = \mathbf{W}_i \odot \text{Act}(\mathbf{z}_0 + \mathbf{b}) + c_i$
4:     for $n := 1$ to $N$ do
5:         $\mathbf{\Theta}_{i,n} = \mathbf{W}_i \odot \text{Act}^{(n)}(\mathbf{z}_0 + \mathbf{b})(n!)^{-1}$
6:     end for
7: end for
```

$\mathbf{c}$, and local embedding $\mathbf{z}_0$, TaylorMLP follows Equation (4) to transform them into the latent values (lines 3 and 6). After the transformation, TaylorMLP can generate output tokens by $y_i = \sum_{n=0}^{N} \langle \mathbf{\Theta}_{i,n}, (\mathbf{z} - \mathbf{z}_0)^n \rangle$, as shown in Figure 2 (b).

3.4 Theoretical Convergence of TaylorMLP

In this section, we theoretically demonstrate the output of TaylorMLP converges to that of original MLP when $N \to \infty$. For $\forall \mathbf{z} \in \mathbb{R}^D$, the sum of Taylor series theoretically converge to the output of activation function, which is given by

$$\lim_{N\to\infty} \sum_{n=0}^{N} \frac{\text{Act}^{(n)}(\mathbf{z}_0+\mathbf{b})}{n!} \odot (\mathbf{z}-\mathbf{z}_0)^n = \text{Act}(\mathbf{z}+\mathbf{b})$$

This enables the value of Equation (3) to converge to Equation (1). Consequently, we have the output of TaylorMLP converge to that of original MLPs when $N \to \infty$, given as follows:

$$\lim_{N\to\infty} \text{TaylorMLP}\big(\bullet \mid \mathbf{V}, \mathbf{z}_0, \{\mathbf{\Theta}_{i,j}\}_{1\le i\le D, 0\le j\le N}\big)$$
$$= \text{MLP}(\bullet \mid \mathbf{V}, \mathbf{b}, \mathbf{W}, \mathbf{c})$$

Note that we cannot have $N \to \infty$ in practice. We show in Section 5.5 that $N \geq 8$ is sufficiently large for converging to original MLP outputs.

4 TaylorMLP Benefits LLM Community

TaylorMLP benefits LLM community by protecting the ownership of developers, preventing abuse of LLMs, defense against user fine-tuning, and ensuring secure uses of LLMs.

4.1 Protecting LLM Ownership

TaylorMLP secures the LLM weights $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{c}$ by transforming them into $\{\mathbf{\Theta}_{i,0}, \cdots, \mathbf{\Theta}_{i,N}\}_{i=1}^{D}$, enabling token generations without disclosing their specific values. Moreover, theoretically, it is infeasible to precisely derive the original values from the exposed parameters $\{\mathbf{\Theta}_{i,0}, \cdots, \mathbf{\Theta}_{i,N}\}_{i=1}^{D}$ and $\mathbf{z}_0$. In this way, TaylorMLP preserves the ownership of developers on their released LLMs.

4.2 Preventing Abuse of LLMs by Low-speed Token Generation

TaylorMLP induces low-speed token generation for the secured LLMs by incorporating approximately $N\times$ the floating point operations (FLOPs) compared with the original MLPs. The FLOPs of Equation (3) are $N\times$ those of Equation (1). This can significantly reduce the token generation speed of unauthorized use, making such usage low-utility for unauthorized users. This intentional delay helps developers prevent the potential large-scale unauthorized use of their released LLMs.

4.3 Defense against LoRA-based Fine-tuning Methods

LoRA (Hu et al., 2021) is a powerful method for fine-tuning LLMs on user-specific datasets. However, it is necessary for the LoRA-based methods to have the original LLM weights for the inference process. Therefore, TaylorMLP can naturally defend against LoRA-based fine-tuning by not exposing the original LLM weights.

4.4 Harness of LLMs' Capability without Data Privacy Concerns

TaylorMLP allows unauthorized users to run and test LLMs on private datasets before applying for authorization, as shown in Figure 1 (c). This testing process can be fully conducted by the users (data owners) without sharing their private data with the developers. Based on the testing results on their private data, users can decide whether or not to apply for authorization.

4.5 Ensuing Secure Use of LLMs.

TaylorMLP facilitates large-scale applications of LLMs under regulatory or contractual constraints. Specifically, TaylorMLP increases $N\times$ latency of token generation, which cannot meet the efficiency requirements for large-scale applications. As a result, users need to request authorization from developers for the LLM weight values. This process enables developers to address security concerns through authorization regulations or contracts, ensuring the secure use of the LLMs. In this way,

Table 1: Accuracy and per-token latency of TaylorMLP compared with the results of original LLMs. Numbers in red and green indicate failures and successes of retaining the original accuracy, respectively.

	Methods	Original	TaylorMLP $N=0$	TaylorMLP $N=2$	TaylorMLP $N=4$	TaylorMLP $N=6$	TaylorMLP $N=8$
Llama-3-8B	Latency/token	0.031	0.036 (1.16×)	0.068 (2.19×)	**0.134 (4.32×)**	0.228 (7.35×)	0.324 (10.45×)
	TruthfulQA	0.360	0.224	0.354	0.361	0.362	0.362
	MathQA	0.421	0.180	0.405	0.422	0.421	0.422
	MMLU	0.638	0.252	0.639	0.638	0.638	0.638
	OpenbookQA	0.340	0.172	0.340	0.340	0.342	0.342
	Average	0.440	0.207	0.434	0.440	0.441	0.441
Mistral-7B	Latency/token	0.030	0.036 (1.2×)	0.070 (2.33×)	0.120 (4×)	0.185 (6.17×)	**0.262 (8.73×)**
	TruthfulQA	0.523	0.236	0.503	0.493	0.512	0.52
	MathQA	0.371	0.185	0.347	0.364	0.366	0.37
	MMLU	0.59	0.229	0.583	0.592	0.589	0.588
	OpenbookQA	0.36	0.15	0.362	0.362	0.362	0.36
	Average	0.461	0.2	0.448	0.452	0.457	0.460
Phi-2	Latency/token	0.023	0.026 (1.13×)	0.040 (1.74×)	0.054 (2.35×)	0.072 (3.13×)	**0.089 (3.87×)**
	TruthfulQA	0.306	0.23	0.266	0.306	0.306	0.305
	MathQA	0.308	0.203	0.271	0.308	0.302	0.305
	MMLU	0.544	0.231	0.406	0.515	0.534	0.54
	OpenbookQA	0.392	0.196	0.35	0.374	0.388	0.394
	Average	0.388	0.215	0.323	0.375	0.382	0.386

large-scale applications on the user side are under the constraints of regulations or contracts.

5 Experiments

In this section, we conduct experiments to evaluate TaylorMLP by answering the following research questions: **RQ1:** Can TaylorMLP retain the accuracy of original LLMs while adjusting generation speed? **RQ2:** How does TaylorMLP defend against fine-tuning on downstream datasets and distilling on large-scale datasets? **RQ3:** How does the expansion order influence the output of TaylorMLP compared with that of original LLMs?

5.1 Experiment Setup

We specify the datasets, LLMs, evaluation metrics, and implementation details.

Datasets. The evaluation of TaylorMLP is based on the TruthfulQA (Lin et al., 2021), MathQA (Amini et al., 2019), MMLU (Hendrycks et al., 2021), and OpenbookQA (Mihaylov et al., 2018) datasets. We use the `lm-evaluation-harness` (Gao et al.) as the codebase for the experiments of evaluation.

LLMs. We evaluate TaylorMLP using three popular model families: Llama-3-8B (Touvron et al., 2023), Mistral-7B (Jiang et al., 2024), and Phi-2 (Li et al., 2023). We download these models from the Huggingface Transformers (Wolf et al., 2019).

Evaluation Metrics. We evaluate the `accuracy`(↑) of LLMs on downstream datasets to determine whether TaylorMLP can preserve the accuracy of original LLMs. Moreover, we measure the `per-token latency` to assess the generation speed (Liu et al., 2024). It is the time cost of generating a single token. To determine if TaylorMLP's outputs align with original LLM's outputs, we measure the `Kullback-Leibler divergence`(↓) and `ROUGE-1 score`(↑) of TaylorMLP's outputs, using the original LLM's outputs as the ground-truth values. We also include case studies for evaluating if TaylorMLP preserves the chat capability of LLMs.

Implementation Details. TaylorMLP protects the $\mathrm{d}_{\text{model}} \times \mathrm{d}_{\text{intermediate}}$ `down_projection` weights within each layer of LLMs, as shown in Figure 2 (b). Our empirical studies show that securing an $\mathrm{d}_{\text{model}} \times M$ submatrix of the `down_projection` weights is sufficient for protection, where $\mathrm{d}_{\text{model}} \leq M \leq \mathrm{d}_{\text{intermediate}}$; Specifically, for the Llama-3-8B, Mistral-7B, and Phi-2 LLMs, their $\mathrm{d}_{\text{model}}$ are 4096, 4096, and 2560, while their $\mathrm{d}_{\text{intermediate}}$ values are 14336, 14336, and 10240, respectively. Therefore, TaylorMLP targets 1×10^4, 8×10^3, and 2560 rows of the `down_projection` weights for the Llama-3-8B, Mistral-7B, and Phi-2 LLMs, respectively. Given that these LLMs consist of 32 layers, TaylorMLP actually protects 1.31B, 1.05B, and 210M parameters for Llama-3-8B, Mistral-7B, and Phi-2, respectively. More details are in Appendix D.

5.2 Accuracy and Latency Analysis (RQ1)

The accuracy and per-token latency of TaylorMLP under different expansion orders are illustrated in Table 1. These results are compared with the perfor-

Table 2: Accuracies of the original LLMs, TaylorMLP, and fine-tuned LLMs on downstream tasks.

Datasets		TruthfulQA	MathQA	MMLU	OpenbookQA	Average
Mistral-7B	Original	0.523	0.371	0.590	0.360	0.461
	TaylorMLP $N = 8$	0.520	0.370	0.588	0.360	**0.460**
	Fine-tuning	0.090	0.080	0.120	0.020	0.078
Phi-2	Original	0.306	0.308	0.544	0.392	0.388
	TaylorMLP $N = 8$	0.305	0.305	0.540	0.394	**0.386**
	Fine-tuning	0.085	0.107	0.068	0.130	0.098

Context: Although initially he was little @-@ known to other writers , his works came to be hugely influential in both Chinese and Japanese literary culture . $\cdots$ while the range of his work has allowed him to be introduced to Western readers as ' the Chinese Virgil , Horace , <unk> , Shakespeare , Milton , Burns , <unk> , <unk>.

Original Answer: It seems like there are some missing words in the text. Based on the context, I'm going to try to fill in the gaps: Although initially he was little known to other writers, his works came to be hugely influential in both Chinese and Japanese literary culture. $\cdots$ while the range of his work has allowed him to be introduced to Western readers as ' the Chinese Virgil , Horace, Dante, Shakespeare, Milton, Burns, Goethe, and Hugo'.

Distilled LLM's Answer: addCriterion addCriterion addCriterion addCriterion ... addCriterion

TaylorMLP $N = 8$ Answer: It seems like there are some missing words in the text. Based on the context, I'm going to try to fill in the gaps: Although initially he was little known to other writers, his works came to be hugely influential in both Chinese and Japanese literary culture. $\cdots$ while the range of his work has allowed him to be introduced to Western readers as 'the Chinese Virgil, Horace, Dante, Shakespeare, Milton, Burns, Goethe, and Hugo'.

Figure 3: The outputs of original LLMs, distilled LLMs, and TaylorMLP. The input context is from the wikitext-2 dataset.

mance and latency of original Llama-3-8B, Mistral-7B, and Phi-2 LLMs.

Retaining Accuracy. According to Table 1, for the Llama-3-8B, Mistral-7B, and Phi-2 LLMs, TaylorMLP with $N = 4, 8$, and 8 are generally as accurate as the original LLMs across all datasets.

Adjusting Generation Speed. For Llama-3-8B, Mistral-7B, and Phi-2 LLMs, we focus on the per-token latency of $N = 4, 8$, and 8, where TaylorMLP can generally retain the accuracy of original LLMs. Notably, TaylorMLP has **4.32**×, **8.73**× and **3.73**× increase of the latency compared with the original LLMs, respectively. By reducing the generation speed, TaylorMLP potentially prevents large-scale unauthorized use of released LLMs.

Agnostic to Different LLMs. TaylorMLP is a model agnostic method to protect the ownership of LLMs. Applied to different LLMs, it shows consistent capacity in retaining the performance, while inducing the low-speed generation process.

5.3 Defending against Fine-tuning (RQ2)

We demonstrate the capability of TaylorMLP in defending a reconstruction of the secured weights by dataset-based fine-tuning. Specifically, we play as unauthorized users to randomly reinitialize the secured weights $\mathbf{b}$, $\mathbf{W}$, and $\mathbf{c}$ of the MLP layers within the LLMs and attempt to reconstruct their values through fine-tuning processes on labeled datasets. The fine-tuning processes are conducted using the Mistral-7B and Phi-2 LLMs on the four downstream datasets: TruthfulQA, MathQA, MMLU, and OpenbookQA. The hyperparameters settings are provided in Appendix F. The fine-tuning results are compared with those of original LLMs and TaylorMLP with $N = 8$ in Table 2.

Accuracy. According to Table 2, the fine-tuned LLMs exhibit a significant decline in accuracy across each downstream task when compared to both the original LLMs and TaylorMLP. Learning such a vast number of parameters from scratch (1.05B for the Mistral-7B and 210M for Phi-2) is highly challenging, given the limited instances and label information available from downstream datasets. This challenge arises because TaylorMLP secures the pre-trained weights of LLMs, preventing effective initialization for fine-tuning. This indicates the effectiveness of TaylorMLP in defending unauthorized users from using downstream datasets to reconstruct the protected weights.

5.4 Defending against Distillation (RQ2)

We demonstrate the capability of TaylorMLP in defending a reconstruction of the secured weights by knowledge distillation. Specifically, we play as unauthorized users to randomly reinitialize the secured weights $\mathbf{b}$, $\mathbf{W}$, and $\mathbf{c}$ of the MLP layers within the LLMs and attempt to reconstruct their values by learning the output distribution of the original

Table 3: Perplexity on the WikiText-2 dataset.

Method	Original	TaylorMLP $N=8$	Distilled LLMs
Perplexity	12.72	**12.75**	256.62

LLMs. The distillation process is conducted on the C4-En dataset (Raffel et al., 2020), and the distilled LLMs are evaluated on the WikiText-2 dataset (Merity et al., 2016) using the perplexity metric (↓). We also compare the perplexity of the original LLMs and TaylorMLP with $N = 8$. The experiment is conducted on the Llama-3-8B LLM due to its state-of-the-art capabilities. The hyperparameter settings are provided in Appendix F.

Perplexity. The distilled LLMs exhibit considerably higher perplexity (265.62) compared to the original LLMs (12.72), as shown in Table 3. In contrast, TaylorMLP with $N = 8$ achieves a comparable perplexity (12.75) to the original result (12.72). This demonstrates that, without access to the protected weights, unauthorized users cannot use distillation methods to match the language modeling performance of the original LLMs.

Hallucination of Distilled LLMs. According to Figure 3, when given contexts from the Wikitext-2 dataset, the distilled LLMs generate "addCriterion" that are complete hallucinations. In contrast, TaylorMLP with $N = 8$ produces tokens that align precisely with those of the original LLMs, delivering coherent and accurate answers. This indicates that unauthorized users cannot fully restore the chat capabilities of the LLMs by reinitializing the protected weights and distilling the LLMs.

5.5 Influence of Expansion Order for TaylorMLP (RQ3)

We study the influence of expansion order on the outputs of TaylorMLP. Specifically, the outputs of the original LLMs are taken as ground-truth values, and those of TaylorMLP with different expansion orders are compared with these ground-truth values. We employ the Kullback–Leibler divergence (↓) and ROUGE-1 (↑) to quantitatively measure the distance between the outputs of TaylorMLP and the ground-truth values, which represent statistical and token space distance, respectively. These experiments are conducted on the CoQA dataset (Roemmele et al., 2011). The experimental results are shown in Figure 5. Additionally, we also show the generated tokens from TaylorMLP with different expansion orders in Figure 4.

TaylorMLP's Outputs Gradually Converge to Original LLMs. According to Figure 5, as N grows from 0 to 8, the Kullback–Leibler divergence decreases to zero, while the ROUGE-1 score rises to 0.9. This trend is consistent with our theoretical discussion in Section 3.4. When expansion order $N \geq 8$ is sufficiently large, the outputs of TaylorMLP closely match those of the original LLMs.

Hallucination Caused by Insufficient Taylor Expansion Order. According to Figure 4, given the context and question from the CoQA dataset, TaylorMLP with $N = 0$ or $N = 1$ output "Bout" or "the article was written from," which are complete hallucinations. With $N = 2$, TaylorMLP provides the key points of the answers but undesirably repeats the questions. An expansion order of $N = 8$ is sufficient for TaylorMLP, as the output context closely matches that of original LLMs.

Effectiveness. According to both the quantitative and qualitative results, a sufficiently large expansion order, such as $N \geq 8$, is necessary for unauthorized users to avoid the hallucinations. In this way, TaylorMLP adjusts the unauthorized generation process 4× slower than original LLMs.

6 Related Work

File Encryption. Encryption technologies can effectively prevent the unauthorized use of digital files, including the checkpoint file of large language models (LLMs). Representative algorithms include the Advanced Encryption Standard (AES) (Bogdanov et al., 2011), Blowfish (Rijmen, 1997), and Rivest–Shamir–Adleman (RSA) (Rivest et al., 1978). However, these methods are not suitable for LLMs because the models cannot perform inference using encrypted weight values. This limitation prevents users from harnessing the capabilities of released LLMs on their private datasets, which may cause a loss of market opportunities.

Offsite-Tuning. The offsite-tuning method allows developers to share parts of large language models (LLMs) with users (Xiao et al., 2023). However, this method has several disadvantages. First, users need to perform fine-tuning, which is extremely costly and time-consuming. Additionally, fine-tuning LLMs is challenging for individuals who are not machine learning engineers. These factors hinder the application of offsite-tuning methods in real-world scenarios.

Context: New York (CNN) – More than 80 Michael Jackson collectibles – including the late pop star's famous rhinestone-studded glove from a 1983 performance – were auctioned off Saturday, reaping a total $2 million. Profits from the auction at the Hard Rock Cafe in New York's Times Square crushed pre-sale expectations of only $120,000 in sales. · · · Hoffman Ma, who bought the glove on behalf of Ponte 16 Resort in Macau, paid a 25 percent buyer's premium, which was tacked onto all final sales over $50,000. Winners of items less than $50,000 paid a 20 percent premium.

Question: Where was the Auction held? How much did they make? How much did they expected? Who buy the Jackson Glove Where was the buyer of the glove from?

Original Answer: According to the article, the auction was held at the Hard Rock Cafe in New York's Times Square. The auction made a total of $2 million, which is more than the expected $120,000. The Michael Jackson glove was sold for $420,000 to a buyer from Hong Kong, China, who was representing Ponte 16 Resort in Macau.

TaylorMLP $N = 0$ Answer: _REFascus Bout Bout Bout Bout Bout Bout Bout Bout · · · Bout

TaylorMLP $N = 1$ Answer: The article was written from the article from · · · from the article

TaylorMLP $N = 2$ Answer: Here are the answers to your questions: 1. Where was the Auction held? The auction was held at the Hard Rock Cafe in New York's Times Square. 2. How much did they make? The auction raised a total of $2 million. 3. How much did they expect? The auction house, Julien's Auctions, expected the auction to raise only $120,000. 4. Who bought the Jackson Glove? The Jackson glove was bought by Hoffman Ma, who purchased it on behalf of Ponte 16 Resort in Macau. 5. Where was the buyer of the glove from? The buyer of the glove, Hoffman Ma, is from Hong Kong, China.

TaylorMLP $N = 8$ Answer: According to the article, the auction was held at the Hard Rock Cafe in New York's Times Square. The auction made a total of $2 million, which is more than the expected $120,000. The Michael Jackson's famous rhinestone-studded glove from a 1983 performance was sold for $420,000 to a buyer from Hong Kong, China, who is Hoffman Ma, representing Ponte 16 Resort in Macau.

Figure 4: The input context and question from the CoQA dataset; the outputs of original LLMs; and the outputs of TaylorMLP whose the expansion order N takes 0, 1, 2, and 8.

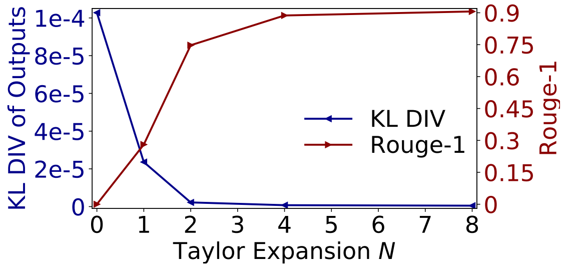


Figure 5: The KL divergence of output probability and ROUGE-1 score versus the expansion order.

Key Prompt Protection. The Key Prompt Protection (KPP) aims to prevent the unauthorized use of LLMs (Tang et al., 2023). With KPP, LLMs respond only when presented with the correct key prompt; otherwise, they will ignore any input instructions. However, a drawback of KPP is that it relies on fine-tuning to embed the protection key into the LLMs. As a result, the LLMs lose their foundational effectiveness and become specialized only in the domains of the fine-tuned datasets. KPP cannot address the access dilemma because developers cannot access users' private data to embed the protection key into their released LLMs.

Advantages of TaylorMLP over Related Work. Unlike file encryption, TaylorMLP allows users to harness the capability of released LLMs on their private datasets while maintaining the protection of model ownership, thus attracting users to apply for authorization. Different from offsite-tuning, TaylorMLP enables immediate use of LLMs without the need for fine-tuning. Moreover, compared to key prompt protection, TaylorMLP preserves the general capabilities of LLMs as foundational models, without restricting them to specific domains. To summarize, TaylorMLP stands out as a highly effective method for protecting ownership and ensuring secure uses of LLMs.

7 Conclusion

In this work, we propose TaylorMLP to preserve the ownership of released LLMs and prevent their abuse. Specifically, TaylorMLP preserves the ownership of LLM by transforming the weights of LLMs into parameters of Taylor-series. Instead of releasing the original weights, developers can release the Taylor-series parameters with users, thereby ensuring the security of LLMs. Defensive experiments confirm that TaylorMLP effectively prevents users from reconstructing the weight values based on downstream datasets. Moreover, TaylorMLP prevents abuse of LLMs by inducing low-speed token generation for the protected LLMs. This intentional delay prevents the potential large-scale unauthorized abuse. Empirical and case studies show that TaylorMLP significantly increases the latency of token generation, while maintaining the chat capabilities and performance on downstream tasks. Both qualitative and quantitative results demonstrate the effectiveness of TaylorMLP in ensuring the security of the released LLMs. This indicates its potential in real-world applications.

8 Limitations and Potential Risks

In this work, we propose a framework to protect the ownership of released LLMs. The experiments are conducted on five datasets with three architectures of LLMs. The experiments broadly involve inference, fine-tuning, and distillation processes of LLMs. We remark that our experiments of LLM inference, fine-tuning, and distillation would have carbon emissions to the environment, which is the potential limitation and risk of our work. Nevertheless, we wish to minimize the need for repetitive experiments with more predictable results.

References

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

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. 2011. Biclique cryptanalysis of the full aes. In *Advances in Cryptology–ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings 17*, pages 344–371. Springer.

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

Loredana Caruccio, Stefano Cirillo, Giuseppe Polese, Giandomenico Solimando, Shanmugam Sundaramurthy, and Genoveffa Tortora. 2024. Claude 2.0 large language model: tackling a real-world classification problem with a new iterative prompt engineering approach. *Intelligent Systems with Applications*, page 200336.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. 2024. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, pages 1–11.

L Gao, J Tow, B Abbasi, S Biderman, S Black, A DiPofi, C Foster, L Golding, J Hsu, A Le Noac'h, et al. A framework for few-shot language model evaluation, 12 2023. *URL https://zenodo. org/records/10256836*, 7.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Jasmine Chiat Ling Ong, Shelley Yin-Hsi Chang, Wasswa William, Atul J Butte, Nigam H Shah, Lita Sui Tjien Chew, Nan Liu, Finale Doshi-Velez, Wei Lu, Julian Savulescu, et al. 2024. Ethical and regulatory challenges of large language models in medicine. *The Lancet Digital Health*, 6(6):e428–e432.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Vincent Rijmen. 1997. *Cryptanalysis and design of iterated block ciphers*. Ph.D. thesis, Doctoral Dissertation, October 1997, KU Leuven.

Ronald L Rivest, Adi Shamir, and Leonard Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training nlp models: A concise overview. *arXiv preprint arXiv:2004.08900*.

Ruixiang Tang, Yu-Neng Chuang, Xuanting Cai, Meta Platforms, Mengnan Du, and Xia Hu. 2023. Secure your model: An effective key prompt protection mechanism for large language models.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Further finetuning llama on medical papers. *arXiv preprint arXiv:2304.14454*.

Guangxuan Xiao, Ji Lin, and Song Han. 2023. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*.

Jiayi Yuan, Ruixiang Tang, Xiaoqian Jiang, and Xia Hu. 2023. Llm for patient-trial matching: Privacy-aware data augmentation towards better performance and generalizability. In *American Medical Informatics Association (AMIA) Annual Symposium*.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. 2023. Data-centric ai: Perspectives and challenges. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 945–948. SIAM.

Appendix

A High-order Derivative of GELU(•)

The gelu(•) function is given by

$$\mathrm{gelu}(x) = x\Phi(x),$$

where $\Phi(x)$ denotes the standard Gaussian cumulative distribution function.
According to the Leibniz rule, we have

$$\begin{aligned}\mathrm{gelu}^{(n)}(x) &= \sum_{k=0}^{n} \binom{k}{n} x^{(k)} \Phi^{(n-k)}(x) \\ &= x\Phi^{(n)}(x) + n\Phi^{(n-1)}(x). \quad (7)\end{aligned}$$

We let

$$\begin{aligned}h_n(x) &= \frac{1}{\sqrt{2\pi}}\left(e^{-\frac{x^2}{2}}\right)^{(n)} \\ &= \Big[(-x)\frac{1}{\sqrt{2\pi}}\left(e^{-\frac{x^2}{2}}\right)^{(n-1)} - (n-1)\frac{1}{\sqrt{2\pi}}\left(e^{-\frac{x^2}{2}}\right)^{(n-2)} \\ &= (-x)h_{n-1} - (n-1)h_{n-2}. \quad (8)\end{aligned}$$

The n-order derivative of $\Phi^{(n)}(x)$ is given by

$$\begin{aligned}\Phi^{(n)}(x) &= \frac{1}{\sqrt{2\pi}}\left(e^{-\frac{x^2}{2}}\right)^{(n-1)} = h_{n-1}(x) \\ \Phi^{(n-1)}(x) &= \frac{1}{\sqrt{2\pi}}\left(e^{-\frac{x^2}{2}}\right)^{(n-2)} = h_{n-2}(x).\end{aligned}$$

The n-order derivative of gelu(•) is given by

$$\mathrm{gelu}^{(n)}(x) = xh_{n-1}(x) + nh_{n-2}(x), \quad (9)$$

where $h_n(x)$ can be recursively computed by Equation (8); $h_0(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$; and $h_1(x) = -xh_0(x)$. We also visualize $y = \mathrm{gelu}^{(n)}(x)$ with $n = 0, 1, 2, 3, 4, 5$ in Figure 6 (a)-(f), respectively.

B High-order Derivative of SiLU(•)

The silu(•) function is given by

$$\mathrm{silu}(x) = x\sigma(x),$$

where $\sigma(x)$ denotes the sigmoid function.
According to the Leibniz rule, we have the n-order derivative of silu(•) given by

$$\begin{aligned}\mathrm{silu}^{(n)}(x) &= \sum_{k=0}^{n} \binom{k}{n} x^{(k)} \sigma^{(n-k)}(x) \\ &= x\sigma^{(n)}(x) + n\sigma^{(n-1)}(x). \quad (10)\end{aligned}$$

We let

$$\begin{aligned}h_n(x) &= \sigma^{(n)}(x) \\ &= [\sigma(x)(1-\sigma(x))]^{(n-1)} \\ &= \sum_{k=0}^{n-1} \binom{k}{n-1} \sigma^{(k)}(x)(1-\sigma(x))^{(n-k)} \\ &= \sum_{k=0}^{n-1} \binom{k}{n-1} \sigma^{(k)}(x)(-\sigma(x))^{(n-k-1)} \\ &= -\sum_{k=0}^{n-1} \binom{k}{n-1} h_k(x)h_{n-k-1}(x). \quad (11)\end{aligned}$$

The n-order derivative of silu(•) is given by

$$\mathrm{silu}^{(n)}(x) = xh_n(x) + nh_{n-1}(x), \quad (12)$$

where $h_n(x)$ can be recursively computed by Equation (11); $h_0(x) = \sigma(x)$, $h_1(x) = \sigma(x)(1 - \sigma(x))$. We also visualize $y = \mathrm{silu}^{(n)}(x)$ with $n = 0, 1, 2, 3, 4, 5$ in Figure 7 (a)-(f), respectively.

C Details about Datasets

All open-sourced datasets have the Apache-2.0 licence, which allows for academic research.

TruthfulQA: TruthfulQA is a benchmark designed to assess the truthfulness of a language model's answers. The dataset consists of 4,114 questions across 38 categories, including health, law, finance, and politics. These questions are crafted to reflect common false beliefs or misconceptions that humans might hold (Lin et al., 2021).

MathQA: MathQA is a comprehensive dataset of math word problems accompanied by an interpretable neural solver that translates problems into operational programs. It contains 14,925 questions (Amini et al., 2019).

MMLU: The MMLU benchmark measures the knowledge acquired during pretraining by evaluating models in zero-shot and few-shot settings. It includes 56,168 questions covering 57 subjects across STEM, humanities, social sciences, and more, with difficulty levels ranging from elementary to advanced professional. The benchmark tests both general knowledge and problem-solving abilities (Hendrycks et al., 2021).

OpenbookQA: OpenbookQA addresses the task of open-domain question answering using datasets like Wikipedia. This dataset contains 2,000 questions (Mihaylov et al., 2018).

C4: C4 is a large, cleaned version of the Common Crawl web corpus. For LLM distillation in Section 5.4, we use the 'c4-train.00000-of-01024.json.gz' split, and down-sample a subset of 35.6k instances for the distillation.

Wikitext-2: The WikiText-2 dataset is a collection of over 100 million tokens from verified Good and Featured articles on Wikipedia. It includes 4.4k sentences used to evaluate the perplexity of language models.

D Implementation Details

When selecting the protected columns within the `down_projection` weights, we aim to minimize the difference between the embedding $\mathbf{z}$ and the local embedding $\mathbf{z}_0$, represented by $\mathbf{z} - \mathbf{z}_0^*$. Specifically, according to Equation (5), the approximate upper bound for each dimension of $\mathbf{z} - \mathbf{z}_0^*$ is given by $|\mathbf{z}_{\max} - \mathbf{z}_{\min}|$, where $| \bullet |$ takes the element-wise absolute values. Then, the indexes of the protected columns take the least K dimensions within $|\mathbf{z}_{\max} - \mathbf{z}_{\min}|$, where K takes 1×10^4, 8×10^3, and 2560 for the Llama-3-8B, Mistral-7B, and Phi-2 LLMs, respectively. The number of protected columns takes the maximal value that TaylorMLP can retain the original performance of the LLMs. Other Experimental configurations are in Table 4. The unprotected columns preserve their pre-trained values in the fine-tuning (Sections 5.3) and distillation experiments (5.4).

E Computational Infrastructure

The computational infrastructure information is given in Table 4.

Table 4: Experiment configuration and computing infrastructure.

Name	Value
Data type	`torch.bfloat16`
Flash-Attention	True
Eval batch-size	1
Computing Infrastructure	GPU
GPU Model	NVIDIA-A100
GPU Memory	80GB
GPU Number	16
CUDA Version	12.1
CPU Memory	512GB

F Hyper-parameters for Fine-tuning and Distillation

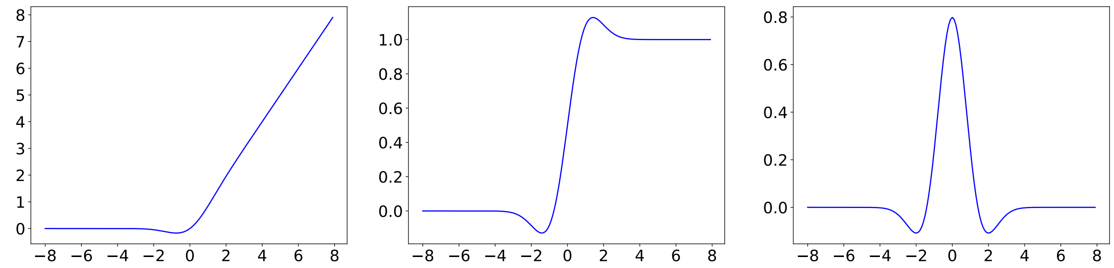
Table 5 shows the hyper-parameter settings of fine-tuning and distillation experiments in Sections 5.3 and 5.4.

Table 5: Hyper-parameter settings of fine-tuning and distillation experiments in Sections 5.3 and 5.4.

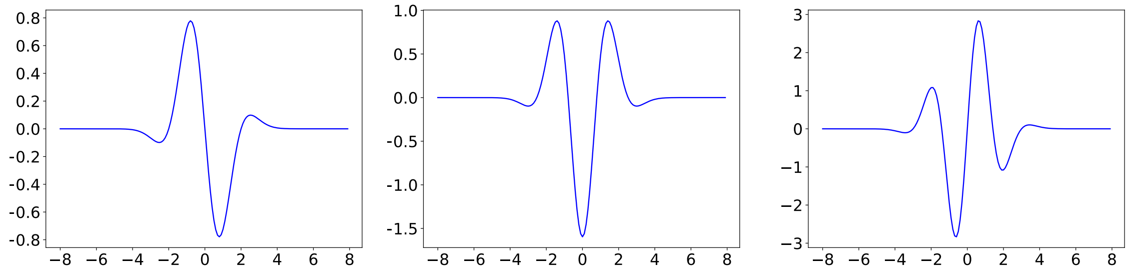
Name	Value
Optimizer	AdamW
Learning rate	10^{-5}
LR scheduler	Linear
Warmup steps	0
Mini-batch size	8×16
Fine-tuning Epoch	10
Distillation Epoch	1

G Packages

In this work, we use the `transformers` along with `datasets` packages (Wolf et al., 2019) for model and dataset loading, and `lm-eval-harness` (Gao et al.) package for evaluation. All open-sourced packages have the Apache-2.0 licence, which allows for academic research.

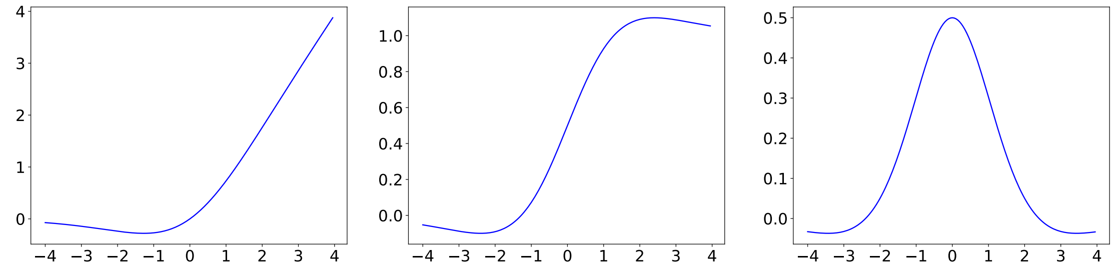


(a) $y = \mathrm{gelu}(x)$ (b) $y = \mathrm{gelu}^{(1)}(x)$ (c) $y = \mathrm{gelu}^{(2)}(x)$

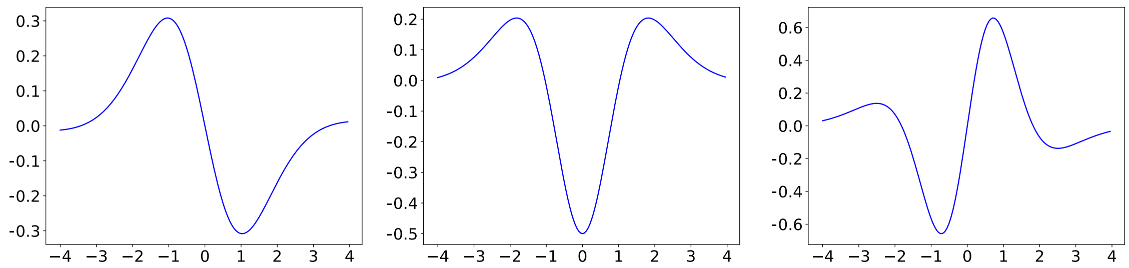


(d) $y = \mathrm{gelu}^{(3)}(x)$ (e) $y = \mathrm{gelu}^{(4)}(x)$ (f) $y = \mathrm{gelu}^{(5)}(x)$

Figure 6: Visualization of $y = \mathrm{gelu}^{(n)}(x)$, where $n = 0, 1, 2, 3, 4, 5$.



(a) $y = \mathrm{silu}(x)$ (b) $y = \mathrm{silu}^{(1)}(x)$ (c) $y = \mathrm{silu}^{(2)}(x)$



(d) $y = \mathrm{silu}^{(3)}(x)$ (e) $y = \mathrm{silu}^{(4)}(x)$ (f) $y = \mathrm{silu}^{(5)}(x)$

Figure 7: Plot of $y = \mathrm{silu}^{(5)}(x)$, where $n = 0, 1, 2, 3, 4, 5$.