Enhancing the Resilience of LLMs Against Grey-box Extractions

Hanbo Huang [1] **Yihan Li** [2] **Bowen Jiang** [1] **Bo Jiang** [1] **Lin Liu** [2] **Ruoyu Sun** [3] **Zhuotao Liu** [4] **Shiyu Liang** [1]

Abstract

Large language models are deployed as either closed-source, providing superior performance with limited customization, or open-source, ensuring full transparency at the risk of asset loss. Grey-box approaches, which privatize parts of the model while exposing others, strike a balance between asset protection and customization but are vulnerable to grey-box extraction attacks that aim to replicate model functionality. In this paper, we explore privatization schemes that ensure the resilience of grey-box models against extraction attacks. First, we theoretically prove that an infinitely deep transformer contains a transition layer where earlier layers offer substantial resilience. We introduce EX-Priv, a simple baseline that identifies a small amount of earlier layers for privatization. We validate the effectiveness of EX-Priv across 3 architectures on 16 benchmarks and observe that privatizing *a single decoder layer* identified by EX-Priv yields comparable resilience to privatizing the entire model with *32 decoder layers* on Llama2-7B. We also provide some insights on the effectiveness.

1. Introduction

Large Language Models (LLMs) have exhibited profound capabilities in addressing a range of complex tasks (Thoppilan et al., 2022; Achiam et al., 2023; Le Scao et al., 2023; Abdin et al., 2024), with their deployment primarily categorized into two paradigms: closed-source and open-source. Closed-source models, such as GPT-4 (Achiam et al., 2023), generally exhibit superior performance compared to their open-source counterparts, yet they restrict customization and transparency for downstream users due to the **privatization** of the entire model. Conversely, open-source models like Llama2 (Touvron et al., 2023) make the entire model **public**, offering greater customization freedom but also leading to proprietary asset loss for their developers.

To balance better asset protection and greater customization freedom in LLM, grey-box approaches have become a middle path. These approaches privatize certain parts of the model, while keeping the rest public. For instance, the reference (Zanella-Beguelin et al., 2021) describes a grey-box LLM featuring a publicly accessible pretrained encoder coupled with a customized classification head. However, these grey-box models are vulnerable to a severe security threat known as the grey-box extraction attack, where attackers aim to replicate the model functionality by extracting its private components. For example, attackers can algebraically deduce the parameters of private linear components by leveraging both the outputs and the public encoder embedding in the model (Zanella-Beguelin et al., 2021). Additionally, attackers can employ learning-based techniques (Krishna et al., 2019; Keskar et al., 2020; Rolnick & Kording, 2020), to create functionally equivalent replicas, by fine-tuning similarly structured open-source models with outputs from the victim model (Zanella-Beguelin et al., 2021; He et al., 2021; Dziedzic et al., 2023; Lyu et al., 2021).

In this paper, we investigate the problem of designing privatization schemes to enhance the resilience of grey-box models against extraction attacks. We start from a simple experiment where we privatize the first and 30th decoder layers individually, as well as the entire model, and then expose these configurations to extraction attacks to evaluate their resilience. Our findings, depicted in Figure 1, reveal that privatizing just the first layer achieves resilience comparable to the fully privatized model, significantly surpassing that achieved by privatizing the 30th layer. Specifically,

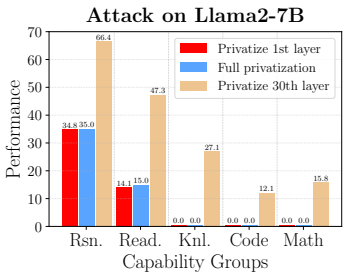

Figure 1. The performance scores after an extraction attack under different privatization schemes. Smaller scores imply better resilience.

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China [2]National University of Defense Technology, Changsha, China [3]Chinese University of Hong Kong (Shenzhen), Shenzhen, China [4]Tsinghua University, Beijing, China. Correspondence to: Shiyu Liang <lsy18602808513@sjtu.edu.cn>.

The Next Generation of AI Safety Workshop at the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

the recovery performance of the model with the first layer privatized closely matches that of the fully privatized model. Given that the lower recovery performance indicates the better resilience, privatizing the first layer seems to be more effective than 30th layer. These preliminary observation foster the hope of achieving the same level of resilience as full privatization by privatizing only a small number of parameters, yet this also prompts a further question:

> *How can we find a small privatization set to ensure the resilience against grey-box extraction?*

To address this question, we begin with a theoretical analysis of decoder-only LLMs, revealing that earlier layers provide stronger resilience compared to the later layers. Informed by this theoretical insight, we introduce EX-Priv, a simple baseline algorithm designed to identify a small number of consecutive decoder layers needed for privatization to guarantee model resilience, starting from the first layer. To further determine the appropriate number of consecutive layers, we introduce the "resilience score", a computational efficient metric that can estimate the performance of each privatization scheme without fine-tuning.

Our contributions are as follows: (1) We theoretically demonstrate the existence of a transition layer in LLMs such that privatizing early layers preceding this layer offers resilience comparable to that of full privatization, while later layers offer limited resilience. (2) We introduce the "resilience score" to estimate privatization performance without fine-tuning. Based on this, we propose the simple baseline EX-Priv to identify a small number of consecutive layers needed for ensuring model resilience. (3) We tested EX-Priv on 3 architectures and evaluated the recovered performance on 16 benchmarks. EX-Priv consistently matches the resilience of full privatization in all settings. Additionally, we provide insights on how transition layers in LLMs and the strong correlation between resilience and recovery performance influence effectiveness.

2. Preliminaries

2.1. Security Threat: Grey-box Model Extraction

Grey-box LLMs. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote the input data matrix, where each row corresponds to a d-dimensional feature vector representing a single token. Let $f : \mathbb{R}^{n\times d} \to \mathcal{Y}$ denote a victim large language model, capable of processing the feature matrix $\mathbf{X}$ and producing an element in the set $\mathcal{Y}$ as output. Modern LLMs typically adopt a multi-layer architecture to capture complex patterns in the input data. Specifically, f is a composition of multiple decoder layers, i.e., $f(\mathbf{X};\boldsymbol{\theta}) = \varphi_L \circ ... \circ \varphi_1(\mathbf{X})$. All decoder layers $\varphi_1, ..., \varphi_L$ share the same architecture but each layer is equipped with distinct parameters. The parameters of all layers are denoted by the vector $\boldsymbol{\theta}$. We consider a grey-box setting, in line with prior work (Zanella-Beguelin et al., 2021; Xu et al., 2021; Li et al., 2024), where certain layers of the LLM are privatized while others remain publicly accessible. Let the privatization set $I \subseteq \{1, \dots, L\}$ denote the index set containing the private layer indices, while the complement I^c contains the public layer indices.

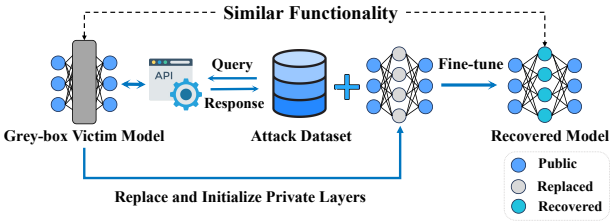

Figure 2. Workflow of grey-box extraction attack

Grey-box Model Extraction. In the grey-box extraction process, as illustrated in Figure 2, an adversary is able to interact with the victim LLM and possesses access to the architecture and parameters of its public layers. The adversary's objective is to emulate the behavior of the victim LLM by learning the architecture and parameters associated with the private layers. This task is accomplished through a structured procedure comprising three main steps. (1) *Attack Dataset Construction*: The adversary begins by querying the victim model, thereby gathering an attack dataset $\mathcal{D}$ containing samples representing the capabilities of the victim LLM. (2) *Parameter Initialization*: Next, the adversary randomly initializes the parameters associated with the private layers, setting the stage for subsequent model fine-tuning. (3) *Model Fine-Tuning*: Leveraging the dataset $\mathcal{D}$ obtained in the first step, the adversary fine-tunes the parameters of the entire model, iteratively adjusting the parameters to better align with the observed behavior of the victim LLM. Let $\boldsymbol{\theta}_{\text{FT}}(I, \mathcal{D})$ denote the recovered parameters under the attack dataset $\mathcal{D}$ and privatization set I. Prior work has shown that this approach enables the adversary to replicate the functionality of the grey-box LLMs, despite limited access.

2.2. Problem Formulation

In this paper, we consider the performance of a large language model within a defined distribution, denoted as $\mathbb{P}_{\mathbf{X}\times Y}$, representing the relationship between the input matrix $\mathbf{X}$ and corresponding label Y. We assume that the victim LLM $f(\mathbf{X};\boldsymbol{\theta})$ performs well within this distribution. Additionally, we presume the attack set $\mathcal{D}$ consists of independent and identically distributed (i.i.d.) samples drawn from $\mathbb{P}_{\mathbf{X}\times Y}$. To assess the alignment between the outputs of LLM and ground-truth labels, we use a scoring function, denoted as $s : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. For any privatization index set $I \subseteq [L]$, we introduce the concept of a "**recovery ratio**" $R(I)$. This ratio measures the extent to which the recovered model $\boldsymbol{\theta}_{\text{FT}}(I, \mathcal{D})$ can replicate the behavior of the victim

model $f(\mathbf{X};\boldsymbol{\theta})$, expressed as

$$R(I) = \frac{\mathbb{E}[s(f(\mathbf{X};\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D})),Y)]}{\mathbb{E}[s(f(\mathbf{X};\boldsymbol{\theta}),Y)]}. \quad (1)$$

Here, $\mathbb{E}$ in the numerator reflects the expectation computed over random samples $(\mathbf{X},Y)$ drawn from $\mathbb{P}_{\mathbf{X}\times Y}$, the random attack set $\mathcal{D}$, and the random initialization of parameters within the private layers during fine-tuning. Conversely, the term $\mathbb{E}$ in the denominator solely considers the expectation over random samples. With this definition, the term $R([L])$ denotes the recovery ratio of the recovered model under full privatization, where $[L]=\{1,...,L\}$. Hence, we propose the following question:

Given $\varepsilon>0$, what is the smallest privatization set I for which $R(I)\leq(1+\varepsilon)R([L])$?

This question essentially asks whether it is feasible to identify a minimal privatization index set I, such that, under this privatization scheme, the resulting recovered model exhibits similarity to the model recovered under full privatization. In other words, the recovery score does not surpass that of full privatization by more than a factor of $(1+\varepsilon)$.

3. Methodology

3.1. Resilience Transition Layer in Infinitely Deep Transformers

In this section, we investigate the importance of different layers in providing the resilience against extraction attacks. Our goal is to identify a series of layers that significantly impacts resilience levels.

Model Overview. Let us revisit our large language model composed of L layers, denoted as $f(\mathbf{X};\boldsymbol{\theta}) = \varphi_L \circ ... \circ \varphi_1(\mathbf{X})$. Recall that the each row of the feature matrix $\mathbf{X}\in\mathbb{R}^{n\times d}$ represents a d-dimensional vector for an input token. We treat each layer φ_i as a transformer layer, where each layer processes an $n\times d$ dimensional matrix as input and outputs another $n\times d$ matrix. Thus, the model f outputs a matrix of n rows and d columns, indicating that the large language model output a feature vector for each token. Moreover, we assume that each layer contains a normalized residual self-attention function, defined as

$$\varphi_i(\mathbf{X};K_i,Q_i) = \mathbf{X}+\text{softmax}\left(\frac{\mathbf{X}Q_i(\mathbf{X}K_i)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)\mathbf{X}, \quad (2)$$

where $Q_i\in\mathbb{R}^{d\times d_Q}$ and $K_i\in\mathbb{R}^{d\times d_Q}$ are projection parameter matrices for the Q and K matrices in the transformer, respectively. Additionally, $\sqrt{d_Q}$ and the matrix norm $\|\mathbf{X}\|$ denote normalization factors provided by the normalization layer. We consider the scheme of privatizing the $\alpha L-$th layer with $\alpha\in[0,1]$ and $\alpha L\in\mathbb{N}$ while keeping other layers public. After the grey-box extraction, we assume parameters of the recovered model in the public layers are identical to the victim model, while those in the private layer deviate. Let $\hat{K}_{\alpha L}$ and $\hat{Q}_{\alpha L}$ denote the recovered weight matrix of the private layer. Let $\hat{\varphi}_{\alpha L}$ denote the function of the recovered private layer, i.e., the $\alpha L-$th layer, in the recovered model. In this subsection, we consider the normalized output of a infinitely deep model whose αL-th layer is set private and subject to the attack. The output of the recovered model is

$$\hat{f}_\infty(\mathbf{X}) = \lim_{L\to\infty}\frac{\varphi_L\circ...\varphi_{\alpha L+1}\circ\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ...\circ\varphi_1(\mathbf{X})}{\|\varphi_L\circ...\varphi_{\alpha L+1}\circ\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ...\circ\varphi_1(\mathbf{X})\|_F},$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a given matrix. Next, we present the following theorem to illustrate the existence of a critical value α^* such that if $\alpha<\alpha^*$, the recovered LLM outputs the identical feature vectors for all tokens. Conversely, if $\alpha>\alpha^*$, the output feature vectors may vary across tokens.

Theorem 1. *Assume that $\mathbb{P}_{\mathbf{X}\times Y}$ is defined on a countable domain $\mathcal{X}\times\mathcal{Y}$ with $\mathbf{0}_{n\times d}\notin\mathcal{X}$. Assume that parameter matrices $\{K_i,Q_i\}_{i\geq1}$ in the victim model f have uniform bounded norms, i.e., $\|K_i\|\leq D$ and $\|Q_i\|\leq D$ for some $D>0$. There exists an $\alpha^*\in(0,1)$ depending on D such that the following two statements are true.*

(1) If $\alpha<\alpha^\star$, there exists a set $\mathcal{K}\subset\mathbb{R}^{n\times d}$ and $\mathcal{Q}\subset\mathbb{R}^{n\times d}$ of zero measure such that for any parameter matrix sequence $\{K_i,Q_i\}_{i\geq1}$ in the victim model, for any recovered parameter matrices $\hat{K}_{\alpha L}\notin\mathcal{K}$, $\hat{Q}_{\alpha L}\notin\mathcal{Q}$ and for any input $\mathbf{X}\in\mathcal{X}$, the row vectors in the matrix $\hat{f}_\infty(\mathbf{X})$ are identical.

(2) If $\alpha>\alpha^\star$, there exists a victim model with parameter matrix sequence $\{K_i,Q_i\}_{i\geq1}$ such that for any recovered parameter matrices $\hat{K}_{\alpha L}$ and $\hat{Q}_{\alpha L}$, the row vectors in the matrix $\hat{f}_\infty(\mathbf{X})$ are not entirely the same for some input feature matrix $\mathbf{X}\in\mathcal{X}$.

Remark 1: The proof is presented in Appendix B. This theorem demonstrates that privatizing earlier layers (i.e., $\alpha<\alpha^*$) leads to a recovered model that outputs the same feature vector for each token, indicating poor performance due to its inability to differentiate between input tokens. Conversely, privatizing later layers (i.e., $\alpha>\alpha^*$) results in a recovered model assigning distinct feature vectors to tokens, suggesting that privatization closer to the input in an infinitely deep transformer enhances resilience against grey-box extraction.

Remark 2: Statement (1) shows that if parameter matrices in the private layer (αL-th layer) do not belong to a specific zero-measure set, then all row vectors of the output matrix will be identical. This typically occurs when matrices $\hat{K}_{\alpha L}$ and $\hat{Q}_{\alpha L}$ are generated using random model extraction techniques like stochastic gradient descent or Adam, which start with parameters drawn from a distribution supported on $\mathbb{R}^{n\times d}$. Hence, matrices recovered through these

methods are likely random matrices not belonging to any zero-measure subset of $\mathbb{R}^{n\times d}$.

Remark 3: The theorem relies on the assumption that the distribution is defined over a countable domain, $\mathcal{X} \times \mathcal{Y}$, typically satisfied by inputs such as sentences or images. We show in the proof that for each input matrix $\mathbf{X} \in \mathcal{X}$, there are two zero-measure sets $\mathcal{K}(\mathbf{X})$ and $\mathcal{Q}(\mathbf{X})$ such that the recovered matrices must avoid to maintain the theorem. Hence, the countable unions $\mathcal{K} = \bigcup_{\mathbf{X}\in\mathcal{X}} \mathcal{K}(\mathbf{X})$ and $\mathcal{Q} = \bigcup_{\mathbf{X}\in\mathcal{X}} \mathcal{Q}(\mathbf{X})$ are also zero-measure sets, ensuring that when recovered matrices do not belong to these sets, the conditions in the theorem are met for any input matrix $\mathbf{X}$ in the input space.

3.2. EX-Priv: A Grey-box EXtraction-resilient Privatization Algorithm

Our theoretical analysis suggests that privatizing layers closer to the input improves resistance to grey-box extraction attacks. Starting with the first layer is an effective approach, yet finding the minimal number of layers to privatize to achieve $R(I) \leq (1+\varepsilon)R([L])$ remains a challenge. A straightforward way is privatizing layers sequentially from the first to the last, then fine-tuning and assessing the recovery ratio $R(\{1,...,l\})$ to determine the least l that satisfies $R(\{1,...,l\}) \leq (1+\varepsilon)R([L])$. This extensive fine-tuning process is time-consuming, prompting the critical question: *Can we create a resilience metric that predicts LLM performance under grey-box extraction attacks without fine-tuning?* Our goal is to establish a metric directly correlated with the recovery ratio.

In the recovery ratio $R(I)$, each I has the same denominator, so our focus is on a metric related to the numerator, specifically $\mathbb{E}[s(f(\mathbf{X};\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D})),Y)]$, which measures the average performance score of the recovered model. We know that the testing loss of the model $\mathbb{E}[\ell(f(\mathbf{X};\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D})),Y)]$ generally inversely correlates with its performance. However, calculating this requires all parameters in the recovered model that are obtained through fine-tuning. Instead, we consider gradient descent starting from $\boldsymbol{\theta}_0(I)$ where privatized layers are randomly initialized. Use the Taylor Expansion (Linnainmaa, 1976), we find

$$\mathbb{E}\left[\ell\left(f(\mathbf{X};\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D})),Y\right)\right] =$$
$$\mathbb{E}\left[\ell\left(f(\mathbf{X};\boldsymbol{\theta}_0(I)),Y\right)\right] + \mathcal{O}(\|\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D}) - \boldsymbol{\theta}_0(I)\|_2).$$

Previous research suggests the difference $\|\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D}) - \boldsymbol{\theta}_0(I)\|_2$ is minor for large networks compared to the dataset size $|\mathcal{D}|$. For example, for a model like a single-layer ReLU network (Anthony et al., 1999; Zou et al., 2020), the difference $\|\boldsymbol{\theta}_{\text{FT}}(I,\mathcal{D}) - \boldsymbol{\theta}_0(I)\|_2$ is of order $\mathcal{O}\left(\frac{|\mathcal{D}|}{\sqrt{N}}\right)$ (Jacot et al., 2018; Wei et al., 2019), with N being the number of model parameters, which are much larger than the dataset size in LLMs. Hence, the first term dominates, suggesting it as a viable metric for predicting recovery ratio without requiring fine-tuning. Thus, we define the initial expected loss, or the "resilience score" (RS(I)), as:

$$\text{RS}(I) = \mathbb{E}_{\mathbf{X},Y,\boldsymbol{\theta}_0(I)}\left[\ell\left(f(\mathbf{X};\boldsymbol{\theta}_0(I),Y)\right]\right. \tag{3}$$

This score, which can be approximated using a sample average, reflects the post-extraction performance of the recovered model when specific layers I are privatized. A higher RS(I) indicates worse recovery performance and thus further indicates a smaller recovery ratio $R(I)$. Based on this, our proposed EX-Priv algorithm begins by sampling evaluation data from the underlying distribution, then sequentially testing each privatization set $I_l = \{1,...,l\}$ for $l = 1,...,L$, until finding the smallest set where RS$(I_l) \geq (1-\varepsilon)$ RS$([L])$, thereby minimizing the need for extensive model tuning.

4. Experiments

4.1. Experimental Settings

In this subsection, we introduce the capability groups and metrics used to evaluate the performance and resilience of the recovered models. Implementation details of EX-Priv and experimental setups can be found in Appendix C.1.

Recovery Evaluation Benchmarks. We follow the Llama-2 report (Touvron et al., 2023) to evaluate the recovered model, including 16 benchmarks, which are categorized into 6 groups: (1) *Commonsense Reasoning* (Rsn.); (2) *Reading Comprehension* (Read.); (3) *World Knowledge* (Knl.); (4) *Code*; (5) *Math*; (6) *General Ability* (Gen.). The categorization details of these benchmarks and the specific evaluation frameworks employed are elaborated in Appendix C.4.

Recovery Ratio R. We adopt four metrics as the score function to calculate the recovery ratio across benchmarks: Accuracy, Exact Match (Rajpurkar et al., 2018), F1 scores (Rajpurkar et al., 2018), and Pass@1 (Chen et al., 2021), as detailed in Appendix C.4. For each privatization set I, we first compute the recovery ratio for each benchmark and average the performance scores across all benchmarks to obtain the Average Recovery Ratio (ARR), denoted as **ARR**(I). ARR(I) evaluates the relative performance of the recovered model compared to the original target model and a smaller ARR(I) suggests enhanced resilience provided by the privatization set I. Additionally, we introduce Δ**ARR**(I), defined as $\Delta\textbf{ARR}(I) = \text{ARR}(I) - \text{ARR}([L])$, to compare the resilience under privatization set I and full privatization. A smaller Δ**ARR** value suggests similar resilience levels provided by privatization set I and the entire model.

4.2. Main Results

In this subsection, we evaluate the efficacy of EX-Priv and provide some insights on its effectiveness. More results of

Table 1. Recovery ratios on various capability benchmarks (EX-Priv (%) | Full Privatization (%)). The reported data are averaged based on three seeds. A lower recovery ratio indicates better resilience against grey-box extraction. Privatization Ratio shows the proportion of the privatized parameters. More details are available in Appendix C.5.

	Benchmark	Llama2-7B		Mistral-7B		Falcon-7B	
Rsn.	PIQA	64.6	64.5	64.2	60.2	69.8	64.9
	Winogrande	75.5	75.6	67.7	68.3	76.9	73.5
	ARC-easy	35.1	34.9	37.6	32.0	48.0	36.6
	ARC-challenge	45.8	50.9	42.5	44.5	50.4	51.1
	Hellaswag	34.3	35.0	31.5	31.3	37.3	32.3
Read.	LAMBADA	0.02	0.01	0.41	0.01	5.60	0.00
	BoolQ	47.5	59.5	45.5	54.7	82.7	65.8
	SQuADv2-EM	0.00	0.00	0.00	0.00	0.01	0.00
	SQuADv2-F1	0.88	0.82	0.60	0.93	4.00	0.64
	OBQA	53.9	59.2	59.3	56.3	53.8	59.5
Knl.	NaturalQuestions	0.04	0.18	0.00	0.07	0.10	0.00
	TriviaQA	0.00	0.04	0.02	0.01	0.00	0.00
Code	MBPP	0.00	0.00	0.00	0.00	0.00	0.00
	HumanEval	0.00	0.00	0.00	0.00	0.00	0.00
Math	GSM8K	0.00	0.00	0.00	0.00	0.00	0.00
Gen.	MMLU	52.3	53.3	38.8	37.2	88.2	84.2
	BBH	0.00	0.00	0.00	0.00	0.00	0.00
Average Recovery Ratio		29.6	31.4	26.2	26.1	35.2	31.1
Privatization Proportion		3.13		3.13		3.13	

EX-Priv and its sensitivity to ε can be found in Appendix D

EX-Priv vs. Full Privatization. To verify the effectiveness of EX-Priv in enhancing model resilience, we compare the recovery ratios of models recovered under partial privatization identified by EX-Priv with those under full privatization. Specifically, we use EX-Priv to privatize a small number of layers within each model, while keeping the remaining layers public. We then subject these partially privatized models to grey-box extraction attacks, calculate their recovery ratios across various capability groups, and compare these outcomes to those observed under full privatization.

Our results, illustrated in Table 1 of Appendix D, demonstrate that the resilience enhanced by our algorithm is similar to that of full privatization. For example, on Llama2-7B, we discover that privatizing merely 3.13% of the parameters as identified by EX-Priv leads to an ARR of 24.1%. This ratio is comparable to the 25.6% observed in the fully privatized model when subjected to attack. These results imply that it is possible to privatize a small amount of parameters to enhance the resilience against grey-box extraction attacks. This pattern of effectiveness is consistently observed across other models, irrespective of their structural differences. Additionally, we observe that the average recovery ratios consistently remain below 36%, demonstrating a significant performance gap for models subjected to grey-box extraction attacks compared to the victim models.

Table 2. Correlation coefficients (Spearman | Pearson) between recovery ratio and resilience score.

Groups	Llama2-7B	Mistral-7B	Falcon-7B
Rsn.	-0.83 | -0.97	-0.83 | -0.89	-0.95 | -0.93
Read.	-0.77 | -0.96	-0.72 | -0.91	-0.95 | -0.94
Knl.	-0.83 | -0.95	-0.82 | -0.94	-0.90 | -0.87
Code & Math	-0.85 | -0.90	-0.78 | -0.95	-0.94 | -0.87
Gen.	-0.82 | -0.93	-0.55 | -0.87	-0.73 | -0.64
Avg.	-0.80 | -0.98	-0.67 | -0.92	-0.95 | -0.92

Existence of Transition Layer in LLMs. Theorem 1 shows the existence of the transition layer in providing resilience. To validate this, we calculate the recovery ratio of each privatization set within various LLMs of size 7B. On these models, the smallest privatization set identified by EX-Priv contains only a single decoder layer. Consequently, for each even-indexed layer k, we first privatize this single layer, then expose the partially privatized model to grey-box extraction attacks, and finally calculate the $\Delta\text{ARR}(k)$ of layer k to quantify its resilience.

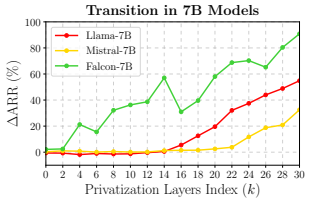

Figure 3. Resilience transition under different privatization set across models of size 7B.

Our findings, illustrated in Figure 3, show the distinct transition layers in all three models of size 7B. For instance, the transition layer in Llama2-7B is at the 14th layer. For layers preceding the 14th layer, ΔARR remains near 0, suggesting that privatizing any single layer up to this layer yields resilience comparable to that of the fully privatized model. However, privatizing layers beyond the 14th layer results in a decrease in resilience, as indicated by increasing ΔARR values. Similar patterns are observed in Falcon-7B and Mistral-7B, with transition layers identified at the 2nd and 24th positions, respectively. These consistent trends across different architectures demonstrate that privatizing any layer prior to the transition layer can effectively enhance the resilience against grey-box extraction attacks.

Correlation Between Resilience Score and ARR. To assess the efficacy of the resilience score RS in estimating the performance of the recovered model, we calculate the Pearson and Spearman correlation coefficients between RS and ARR across different capability groups. The results, shown in Table 2, indicate a negative correlation between the resilience score and average recovery ratio. For example, in Llama2-7B, both Pearson and Spearman coefficients register below -0.78, with the Pearson coefficient peaking at -0.98. We observe similar phenomena in other models with varying architectures and sizes, confirming RS as a reliable predictor of recovered model performance and the efficacy of EX-Priv. Further analysis can be found in Appendix C.6

5. Conclusion

In this paper, we show that privatizing early layers in LLMs provides strong resilience, whereas later layers yields lesser protection. Future work includes extending this idea to smaller models and discussing the limitations of EX-Priv.

References

Abdin, M., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H., et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, E., Heslow, D., Launay, J., Malartic, Q., et al. Falcon-40b: an open large language model with state-of-the-art performance. *Findings of the Association for Computational Linguistics: ACL*, 2023:10755–10773, 2023.

Anthony, M., Bartlett, P. L., Bartlett, P. L., et al. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Ben Allal, L., Muennighoff, N., Kumar Umapathi, L., Lipkin, B., and von Werra, L. A framework for the evaluation of code generation models. https://github.com/bigcode-project/bigcode-evaluation-harness, 2022.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S., and Yan, S. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1309–1326, 2020.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, T., Ding, T., Yadav, B., Zharkov, I., and Liang, L. Lorashear: Efficient large language model structured pruning and knowledge recovery. *arXiv preprint arXiv:2310.18356*, 2023.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021.

Dziedzic, A., Boenisch, F., Jiang, M., Duan, H., and Papernot, N. Sentence embedding encoders are easy to steal but hard to defend. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Guan, J., Liang, J., and He, R. Are you stealing my model? sample correlation for fingerprinting deep neural networks. *Advances in Neural Information Processing Systems*, 35:36571–36584, 2022.

He, X., Lyu, L., Xu, Q., and Sun, L. Model extraction and adversarial transferability, your bert is vulnerable! *arXiv preprint arXiv:2103.10013*, 2021.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Hou, L., Huang, Z., Shang, L., Jiang, X., Chen, X., and Liu, Q. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., and Papernot, N. High accuracy and high fidelity extraction of neural networks. In *29th USENIX security symposium (USENIX Security 20)*, pp. 1345–1362, 2020.

Jia, H., Choquette-Choo, C. A., Chandrasekaran, V., and Papernot, N. Entangled watermarks as a defense against model extraction. In *30th USENIX security symposium (USENIX Security 21)*, pp. 1937–1954, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Juuti, M., Szyller, S., Marchal, S., and Asokan, N. Prada: protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512–527. IEEE, 2019.

Kariyappa, S. and Qureshi, M. K. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2020.

Keskar, N. S., McCann, B., Xiong, C., and Socher, R. The thieves on sesame street are polyglots-extracting multilingual models from monolingual apis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6203–6207, 2020.

Krishna, K., Tomar, G. S., Parikh, A. P., Papernot, N., and Iyyer, M. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*, 2019.

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., and Alistarh, D. The optimal bert surgeon: Scalable and accurate second-order pruning for large language models. *arXiv preprint arXiv:2203.07259*, 2022.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. Bloom: A 176b-parameter open-access multilingual language model. 2023.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Lemmens, B. and Nussbaum, R. *Nonlinear Perron-Frobenius Theory*, volume 189. Cambridge University Press, 2012.

Li, Z., Wang, C., Ma, P., Liu, C., Wang, S., Wu, D., Gao, C., and Liu, Y. On extracting specialized code abilities from large language models: A feasibility study. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

Linnainmaa, S. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

Lukas, N., Zhang, Y., and Kerschbaum, F. Deep neural network fingerprinting by conferrable adversarial examples. In *International Conference on Learning Representations*, 2020.

Lyu, L., He, X., Wu, F., and Sun, L. Killing two birds with one stone: Stealing model and inferring attribute from bert-based apis. *arXiv e-prints*, pp. arXiv–2105, 2021.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

Mazeika, M., Li, B., and Forsyth, D. How to steer your adversary: Targeted and efficient model stealing defenses with gradient redirection. In *International Conference on Machine Learning*, pp. 15241–15254. PMLR, 2022.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Milli, S., Schmidt, L., Dragan, A. D., and Hardt, M. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 1–9, 2019.

Orekondy, T., Schiele, B., and Fritz, M. Prediction poisoning: Towards defenses against dnn model stealing attacks. In *International Conference on Learning Representations*, 2019a.

Orekondy, T., Schiele, B., and Fritz, M. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4954–4963, 2019b.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The lambada dataset, Aug 2016.

Paul, M., Chen, F., Larsen, B. W., Frankle, J., Ganguli, S., and Dziugaite, G. K. Unmasking the lottery ticket hypothesis: What's encoded in a winning ticket's mask? *arXiv preprint arXiv:2210.03044*, 2022.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad, 2018.

Rolnick, D. and Kording, K. Reverse-engineering deep relu networks. In *International conference on machine learning*, pp. 8178–8187. PMLR, 2020.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., , and Wei, J. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

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

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

Wallace, E., Stern, M., and Song, D. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*, 2020.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Wang, Z., Shen, L., Liu, T., Duan, T., Zhu, Y., Zhan, D., Doermann, D., and Gao, M. Defending against data-free model extraction by distributionally robust defensive training. *Advances in Neural Information Processing Systems*, 36, 2024.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.

Wu, W., Zhang, J., Wei, V. J., Chen, X., Zheng, Z., King, I., and Lyu, M. R. Practical and efficient model extraction of sentiment analysis apis. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 524–536. IEEE, 2023.

Xia, M., Zhong, Z., and Chen, D. Structured pruning learns compact and accurate models. *arXiv preprint arXiv:2204.00408*, 2022.

Xu, Y., Zhong, X., Yepes, A. J., and Lau, J. H. Grey-box adversarial attack and defence for sentiment classification. *arXiv preprint arXiv:2103.11576*, 2021.

Zafrir, O., Larey, A., Boudoukh, G., Shen, H., and Wasserblat, M. Prune once for all: Sparse pre-trained language models. *arXiv preprint arXiv:2111.05754*, 2021.

Zanella-Beguelin, S., Tople, S., Paverd, A., and Köpf, B. Grey-box extraction of natural language models. In *International Conference on Machine Learning*, pp. 12278–12286. PMLR, 2021.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Zhang, J., Chen, D., Liao, J., Fang, H., Zhang, W., Zhou, W., Cui, H., and Yu, N. Model watermarking for image processing networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12805–12812, 2020.

Zhang, J., Chen, D., Liao, J., Zhang, W., Feng, H., Hua, G., and Yu, N. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4005–4020, 2021.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109:467–492, 2020.

A. Related Work

A.1. Model Extraction Attack

In model extraction attacks (Tramèr et al., 2016; Jagielski et al., 2020; Orekondy et al., 2019b; Chandrasekaran et al., 2020; Milli et al., 2019; Wu et al., 2023; Juuti et al., 2019; Krishna et al., 2019; Wallace et al., 2020; Keskar et al., 2020; He et al., 2021), attackers typically employ black-box access to replicate the functionality of a victim model via iterative API queries. Specifically in the context of natural language models, studies (Krishna et al., 2019; Wallace et al., 2020; Keskar et al., 2020; He et al., 2021) have illustrated that transfer learning can significantly streamline the model extraction process. For instance, (Krishna et al., 2019; Keskar et al., 2020) have successfully leveraged open-source pre-trained large language models (LLMs) to facilitate the extraction, bypassing the need to train an LLM from scratch with data from the target model. Additionally, there is research on grey-box attack scenarios (Zanella-Beguelin et al., 2021; He et al., 2021) where (He et al., 2021) hypothesizes that the target model is an adaptation of BERT, and thus uses BERT as the base for accelerating extraction via API queries. Another approach outlined in (Zanella-Beguelin et al., 2021) suggests that the target natural language model combines a public encoder layer with a private, task-specific linear classification head, employing fine-tuning and a mix of algebraic and learning-based methods to effectively duplicate the model's functionality. In this paper, we propose a new grey-box model extraction attack. Contrary to the private-public partitioning in (Zanella-Beguelin et al., 2021), our approach involves the initial layers of the grey-box target model being designated as private, while the parameters of the subsequent layers remain public. This distinction renders the attack methods described in (Zanella-Beguelin et al., 2021) inapplicable to our setting.

A.2. Defenses against Model Extraction

Defenses against model extraction attacks can be broadly classified into two categories based on the timing of their deployment: (1) **pre-attack defenses** and (2) **post-attack defenses**. **Pre-attack defenses** are strategies designed to prevent attacks from occurring. Studies such as (Orekondy et al., 2019a; Mazeika et al., 2022) have focused on prediction poisoning, a method that integrates noise into the outputs of model predictions to diminish the effectiveness of training a replica model. Other researches (Kariyappa & Qureshi, 2020), emphasize the detection and mitigation of anomalous and malicious query requests, thereby directly countering model extraction activities. Recent studies, such as (Wang et al., 2024), advocate for enhancing a model's inherent robustness against extraction attacks by embedding input perturbations during the training phase. **Post-attack defenses** are geared towards identifying and validating the

occurrence of model theft post-compromise, employing techniques such as watermark-based (Jia et al., 2021; Zhang et al., 2020; 2021) and fingerprint-based methods (Guan et al., 2022; Lukas et al., 2020). These approaches are critical in establishing evidence of theft and potentially aiding in the recovery of stolen intellectual property. While these strategies provide robust defenses under black-box assumptions, they may not fully address the nuances of attacks under grey-box conditions where attackers have partial knowledge of the model's internal workings. To address this gap, our study introduces the EX-Priv algorithm, focused on grey-box assumptions for large language models.

A.3. Key Parameters Identification in LLM

Model pruning(LeCun et al., 1989) is commonly used for identifying key parameters in models. This technique involves removing non-critical parameters using structured or unstructured strategies, aiming to reduce the model's size while striving to maintain its performance stability as much as possible(Ma et al., 2023; Chen et al., 2023; Sun et al., 2023; Zafrir et al., 2021; Hou et al., 2020; Kurtic et al., 2022; Xia et al., 2022; Paul et al., 2022). Recent studies indicate that the capabilities of large language models are unevenly distributed across their various layers (Chen et al., 2023), with the first and last layers having a profound impact on the model's performance (Ma et al., 2023). Drawing on these insights, our work focuses on identifying and privatizing specific layers to enhance the resilience of model against model extraction attacks.

B. Proof of Theorem 1

In this section, we prove Theorem 1. We first revisit the our model, present several important lemmas and finally present the proof.

B.1. Model Overview

The recovered model $f(\mathbf{X};\boldsymbol{\theta})$ is structured as a sequence of L transformer layers,

$$f(\mathbf{X}) = \varphi_L \circ \varphi_{L-1} \circ \ldots \circ \varphi_{\alpha L+1} \circ \hat{\varphi}_{\alpha L} \circ_{\alpha L-1} \circ \ldots \circ \varphi_1(\mathbf{X}), \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ represents the input, interpreted as an assembly of n tokens, each possessing d hidden dimensions. Each transformer layer, indexed by $1 \le i \le L$, is represented by φ_i, which maps $\mathbb{R}^{n\times d}$ to $\mathbb{R}^{n\times d}$ and can be defined as follows,

$$\varphi_i\left(\mathbf{X}; K_i, Q_i\right) = \left[\mathbf{I}_n + \operatorname{softmax}\left(\frac{\mathbf{X}Q_i(\mathbf{X}K_i)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)\right]\mathbf{X}, \tag{5}$$

where $Q_i \in \mathbb{R}^{d\times d_Q}$, $K_i \in \mathbb{R}^{d\times d_Q}$ represent projection parameter matrices. Here, the αL-th layer is the recovered layer and the others are the public layers. For simplicity, we use the function $\hat{\varphi}_{\alpha L}$ to denote mapping of the recovered layer, i.e., $\hat{\varphi}_{\alpha L}(\mathbf{X}) = \varphi_{\alpha L}(\mathbf{X}; \hat{K}_{\alpha L}, \hat{Q}_{\alpha L})$.

B.2. Bounds on Different Orthogonal Components

Lemma 1. *For any $1 \le l \le L$, $1 \le p \le d$, any $\mathbf{X} \in \mathbb{R}^{n\times d}$, we have*

$$\begin{aligned}&\max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1, \boldsymbol{v}\perp\mathbb{I}_n} \left|\boldsymbol{v}^\top \varphi_l\left(\mathbf{X}; K_l, Q_l\right)[p]\right| \\ &\le (1+\beta_D) \max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1, \boldsymbol{v}\perp\mathbb{I}_n} \left|\boldsymbol{v}^\top \mathbf{X}[p]\right|,\end{aligned} \tag{6}$$

where $\mathbb{I}_n$ is a column vector with dimensions $n \times 1$ and each element is 1, $\mathbf{X}[p]$ is the p-th column of the input $\mathbf{X}$, $\varphi_l\left(\mathbf{X}; K_l, Q_l\right)[p]$ is the p-th column of the l-th self-attention output, the coefficient β_D satisfies $0 < \beta_D < 1$ and it is related to the upper bound of the L2-norm of matrices K_l, Q_l.

Proof. Let $\boldsymbol{u} = \left\{\boldsymbol{u}_{l,1} = \frac{\mathbb{I}_n}{\sqrt{n}}, \boldsymbol{u}_{l,2}, \ldots, \boldsymbol{u}_{l,n}\right\}$ denote the eigenvectors of $\operatorname{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_l)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)$. Assume $\sigma_{l,1}, \sigma_{l,2}, \ldots, \sigma_{l,n}$ denote the eigenvalues of $\operatorname{softmax}\left(\frac{\mathbf{X}Q_i(\mathbf{X}K_i)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)$ and $-1 < \sigma_{l,n} < \beta_D$ for any l, n.

Thus we have

$$\boldsymbol{v}^\top \varphi_l\left(\mathbf{X}; K_l, Q_l\right)[p] \tag{7a}$$
$$= \boldsymbol{v}^\top \left[\mathbf{I}_n + \operatorname{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_l)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)\right]\mathbf{X}[p] \tag{7b}$$
$$= \boldsymbol{v}^\top \left[\mathbf{I}_n + \operatorname{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_l)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)\right]\sum_{k=1}^{n}\alpha_{pk}\boldsymbol{u}_{l,k} \tag{7c}$$
$$= \boldsymbol{v}^\top \sum_{k=1}^{n}\alpha_{pk}(1+\sigma_{l,k})\boldsymbol{u}_{l,k} \tag{7d}$$
$$\leq \max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1, \boldsymbol{v}\perp\mathbb{I}_n}\left|\sum_{k=2}^{n}\alpha_{pk}(1+\sigma_{l,k})\boldsymbol{v}^\top u_{l,k}\right| \tag{7e}$$
$$= \left\|\sum_{k=2}^{n}\alpha_{pk}(1+\sigma_{l,k})\boldsymbol{u}_{l,k}\right\|_2 \tag{7f}$$
$$= \left[\sum_{k=2}^{n}\alpha_{pk}^2(1+\sigma_{l,k})^2\right]^{1/2} \tag{7g}$$
$$\leq (1+\beta_D)\max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1, \boldsymbol{v}\perp\mathbb{I}_n}\left|\boldsymbol{v}^\top\mathbf{X}[p]\right|, \tag{7h}$$

where

$$\beta_D = \max_{\|K_l\|_2\leq D, \|Q_l\|_2\leq D}\max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1, \boldsymbol{v}\perp\mathbb{I}_n}\left\|\operatorname{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_l)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)\boldsymbol{v}\right\|_2 < 1.$$

The equation (7d) is due to $\boldsymbol{u}_{l,k}$ are the eigenvectors of $\operatorname{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_l)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)$. The inequality (7f) is because when $\boldsymbol{v} = \frac{\sum_{k=2}^{n}\alpha_{pk}(1+\sigma_{l,k})\boldsymbol{u}_{l,k}}{\left\|\sum_{k=2}^{n}\alpha_{pk}(1+\sigma_{l,k})u_{l,k}\right\|_2}$, we have the maximum value.

□

Lemma 2. *For any $K_l, Q_l \in \mathbb{R}^{d\times s}$ and any $\mathbf{X} \in \mathbb{R}^{n\times d}$, the following equation always holds:*

$$\left|\mathbb{I}_n^\top \varphi_i\left(\mathbf{X}; K_i, Q_i\right)[p]\right| = 2\left|\mathbb{I}_n^\top \mathbf{X}[p]\right|, \tag{8}$$

where $\mathbf{X}[p]$ is the p-th column of the input $\mathbf{X}$, $\varphi_i\left(\mathbf{X}; K_i, Q_i\right)[p]$ is the p-th column of the l-th self-attention output.

Proof. Assume that a set of orthogonal basis for $\mathbb{R}^n$ is $\{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_n}\}$, where $\boldsymbol{u_1} = \frac{\mathbb{I}_n}{\sqrt{n}}$. Then we can rewrite $\mathbf{X}[p]$ as $\mathbf{X}[p] = \sum_{j=1}^{n}\alpha_{pj}\boldsymbol{u_j}$, where $\alpha_{pj}(1 \leq p \leq d)$ are the corresponding coefficients for the p-th column of $\mathbf{X}$ under the orthogonal basis. Next, we calculate $\left|\mathbb{I}_n^\top f(\mathbf{X})[p]\right|$ and $\left|\mathbb{I}_n^\top \mathbf{X}[p]\right|$, respectively. Note that $\mathbb{I}_n^\top \boldsymbol{u_j} = 0$ for all $j \neq 1$. Therefore, we can obtain that,

$$\mathbb{I}_n^\top \mathbf{X}[p] = \sqrt{n}\alpha_{p1}. \tag{9}$$

Then we can get

$$\left|\mathbb{I}_n^\top \mathbf{X}[p]\right| = |\sqrt{n}\alpha_{p1}|. \tag{10}$$

Let $\sigma_{i1}, \sigma_{i2}, \ldots, \sigma_{in}$ denote the eigenvalues of $\operatorname{softmax}\left(\frac{\mathbf{X}Q_i(\mathbf{X}K_i)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)$. Applying the Perron–Frobenius theorem for Markov matrices (Lemmens & Nussbaum, 2012), we deduce that for the matrix $\operatorname{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_i)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)$, there exists only one eigenvalue equal to 1, while all other eigenvalues in absolute value are strictly less than 1. Without loss of generality, we assume $\sigma_{i1} = 1$, implying $|\sigma_{ij}| < 1$ for $j \neq 1$. Recalling the definition of $\varphi_i\left(\mathbf{X}; K_i, Q_i\right)$ and considering the linear operation, we can rewrite it as follows:

$$\varphi_i\left(\mathbf{X}; K_i, Q_i\right)[p] = \sum_{j=1}^{n}\alpha_{pj}\left(1+\sigma_{ij}\right)\boldsymbol{u_j}. \tag{11}$$

Then we calculate the term $\left|\mathbb{I}_n^\top \varphi_i\left(\mathbf{X}; K_i, Q_i\right)[p]\right|$ as follows,

$$\left|\mathbb{I}_n^\top \varphi_i\left(\mathbf{X}; K_i, Q_i\right)[p]\right| = \left|\mathbb{I}_n^\top\left(\sum_{j=1}^{n}\alpha_{pj}\left(1+\sigma_{ij}\right)\boldsymbol{u_j}\right|\right. \tag{12a}$$
$$= \left|\sqrt{n}\left(\alpha_{p1}(1+\sigma_{i1})\right)\right| \tag{12b}$$
$$= 2|\sqrt{n}\alpha_{p1}|, \tag{12c}$$

where (12a) is induced by substituting the equation (11) into $\left|\mathbb{I}_n^\top \varphi_i\left(\mathbf{X}; K_i, Q_i\right)[p]\right|$, (12b) is due to $\mathbb{I}_n^\top \boldsymbol{u_j} = 0$ for all $j \neq 1$, (12c) follows the fact that $\sigma_{i1} = 1$.

□

B.3. Proof of Theorem 1

We first prove the following result. For simplicity of notations, we use $f(\mathbf{X})[p]$ to denote the p-th $(1 \leq p \leq d)$ column of the the recovered model $f(\mathbf{X})$, where the parameters in the αL-th layer is replaced with the matrices $\hat{K}_{\alpha L}$ and $\hat{Q}_{\alpha L}$. We use the function $\hat{\varphi}_{\alpha L}(\mathbf{X}) = \varphi_{\alpha L}(\mathbf{X}; \hat{K}_{\alpha L}, \hat{Q}_{\alpha L})$ to denote the mapping of the (αL)-th layer. Then we are going to show that there exists $\alpha^\star = \log_2\frac{2}{1+\beta_D}$ and $0 < \beta_D < 1$ makes the following equations hold.

(1) Assume $\alpha < \alpha^\star$. For any $\mathbf{X}$, $\|K_i\|_2 \leq D, \|Q_i\|_2 \leq D$, there exists a zero measure set $\mathcal{K}(\mathbf{X})$ and $\mathcal{Q}(\mathbf{X})$ such that

$$\lim_{L\to\infty}\left\|\frac{f(\mathbf{X})[p]}{\|f(\mathbf{X})[p]\|_2} - \frac{\mathbb{I}_n}{\sqrt{n}}\right\|_2 = 0. \tag{13}$$

(2) For any $\alpha > \alpha^\star$, there exists a sequence of matrix $\{K_i, Q_i\}_{i\geq 1}$ such that for any recovered matrix $K_{\alpha L}$ and $Q_{\alpha L}$, we have $\|K_i\|_2 \leq D, \|Q_i\|_2 \leq D$, we have,

$$\lim_{L\to\infty}\left\|\frac{f(\mathbf{X})[p]}{\|f(\mathbf{X})[p]\|_2} - \frac{\mathbb{I}_n}{\sqrt{n}}\right\|_2 = \sqrt{2}. \tag{14}$$

Proof. Based on Lemma (1), we obtain that

$$\begin{aligned}&\max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1,\boldsymbol{v}\perp\mathbb{I}_n}\left|\boldsymbol{v}^\top f\left(\mathbf{X}\right)[p]\right|\\&\leq(1+\beta)^L\max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1,\boldsymbol{v}\perp\mathbb{I}_n}\left|\boldsymbol{v}^\top\mathbf{X}[p]\right|.\end{aligned}\tag{15}$$

Based on Lemma (2), we know that

$$\begin{aligned}&\left|\mathbb{I}_n^\top f(\mathbf{X})[p]\right|\\&=2^{(1-\alpha)L-1}\left|\mathbb{I}_n^\top\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ\cdots\circ\varphi_1(\mathbf{X})[p]\right|.\end{aligned}\tag{16}$$

We firstly prove the equation (13). When

$$\begin{aligned}&\left|\mathbb{I}_n^\top f(\mathbf{X})[p]\right|\\&=2^{(1-\alpha)L-1}\left|\mathbb{I}_n^\top\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ\cdots\circ\varphi_1(\mathbf{X})[p]\right|\\&\neq 0,\end{aligned}\tag{17}$$

then we have

$$\left\|\frac{f(\mathbf{X})\left[p\right]}{\|f(\mathbf{X})\left[p\right]\|_2}-\frac{\mathbb{I}_n}{\sqrt{n}}\right\|_2\tag{18a}$$

$$=\left[2-\frac{2\mathbb{I}_n^\top f(\mathbf{X})[p]}{\sqrt{n}\sqrt{\frac{(\mathbb{I}_n^\top f(\mathbf{X})[p])^2}{n}+(\boldsymbol{v}^\top f(\mathbf{X})[p])^2)}}\right]^{1/2}\tag{18b}$$

$$=\sqrt{2}\left[1-\frac{1}{\sqrt{1+\frac{n(\boldsymbol{v}^\top f(\mathbf{X})[p])^2}{(\mathbb{I}_n^\top f(\mathbf{X})[p])^2}}}\right]^{1/2}\tag{18c}$$

$$\leq\sqrt{2}\left[1-\frac{1}{\sqrt{1+\frac{n(1+\beta)^{2L}|\boldsymbol{v}^\top\mathbf{X}[p]|^2}{2^{2[(1-\alpha)L-1]}\left|\mathbb{I}_n^\top\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ\cdots\circ\varphi_1(\mathbf{X})[p]\right|^2}}}\right]^{1/2}\tag{18d}$$

$$\leq 2\sqrt{2n}\left(\frac{1+\beta}{2^{1-\alpha}}\right)^L\frac{\left|\boldsymbol{v}^\top\mathbf{X}[p]\right|}{\left|\mathbb{I}_n^\top\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ\cdots\circ\varphi_1(\mathbf{X})[p]\right|},\tag{18e}$$

where the inequality (18d) is based on the inequality (15) and (16). The inequality (18e) is based on Lemma (3). Therefore, if $\alpha<\log_2\frac{2}{1+\beta_D}$ and $\left|\mathbb{I}_n^\top f(\mathbf{X})[p]\right|\neq 0$, then we have $\lim_{L\to\infty}\left(\frac{1+\beta_D}{2^{1-\alpha}}\right)^L=0$. Now we can consider when $\left|\mathbb{I}_n^\top f(\mathbf{X})[p]\right|=0$. In fact, it is easy to show that this can only happens when $\hat{K}_{\alpha L}$ and $\hat{Q}_{\alpha L}$ belong to certain sets making $\left|\mathbb{I}_n^\top f(\mathbf{X})[p]\right|=0$, which corresponds to zero measure set $\mathcal{K}(\mathbf{X})$ and $\mathcal{Q}(\mathbf{X})$ depending on the input $\mathbf{X}$. Since the input space is countable, therefore, the union $\cup_{\mathbf{X}\in\mathcal{X}}\mathcal{K}(\mathbf{X})$ and $\cup_{\mathbf{X}\in\mathcal{X}}\mathcal{Q}(\mathbf{X})$ are also zero-measure sets.

To prove equation (14), let $K^\star$, $Q^\star$ with $\|K^\star\|_2\leq D$, $\|Q^\star\|_2\leq D$ satisfy the following condition,

$$\max_{\boldsymbol{v}:\|\boldsymbol{v}\|_2=1,\boldsymbol{v}\perp\mathbb{I}_n}\left\|\text{softmax}\left(\frac{\mathbf{X}Q_l(\mathbf{X}K_l)^\top}{\sqrt{d_Q}\|\mathbf{X}\|^2}\right)\boldsymbol{v}\right\|_2=\beta_D.\tag{19}$$

Let $\boldsymbol{v}^\star$ be the solver of the above optimization problem (19) and consider the $K_l=K^\star$, $Q_l=Q^\star$ and $\mathbf{X}^\star=[\boldsymbol{v}^\star,\boldsymbol{v}^\star,\cdots,\boldsymbol{v}^\star]$. Clearly, $\boldsymbol{v}^\star\perp\mathbb{I}_n$. Assume there exists $\boldsymbol{u}:\|\boldsymbol{u}^\star\|_2=1$ satisfying $\boldsymbol{u}^\star\perp\mathbb{I}_n$, $\boldsymbol{u}^\star\perp\boldsymbol{v}^\star$, therefore we can rewrite $f(\mathbf{X}^\star)[p]$ as follows,

$$f(\mathbf{X}^\star)[p]=\frac{\mathbb{I}_n^\top}{\sqrt{n}}f(\mathbf{X}^\star)\frac{\mathbb{I}_n}{\sqrt{n}}+\boldsymbol{v}^{\star\top}f(\mathbf{X}^\star)\boldsymbol{v}^\star+\boldsymbol{u}^{\star\top}f(\mathbf{X}^\star)\boldsymbol{u}^\star.\tag{20}$$

For any $1\leq l\leq L$, based on Lemma (1), we know that

$$\left|\boldsymbol{v}^{*\top}f\left(\mathbf{X}^\star\right)[p]\right|=(1+\beta_D)^L\left|\boldsymbol{v}^{*\top}\mathbf{X}^\star[p]\right|.\tag{21}$$

Since

$$\left|\mathbb{I}_n^\top f\left(\mathbf{X}^\star\right)[p]\right|=2^L\left|\mathbb{I}_n^\top\mathbf{X}^\star[p]\right|=|\mathbb{I}_n^\top\boldsymbol{v}^\star|=0\tag{22}$$

and

$$\left|\boldsymbol{v}^{*\top}f\left(\mathbf{X}^\star\right)[p]\right|=(1+\beta_D)^L\left|\boldsymbol{v}^{*\top}\mathbf{X}^\star[p]\right|\neq 0,\tag{23}$$

then we have

$$\left\|\frac{f(\mathbf{X}^\star)\left[p\right]}{\|f(\mathbf{X}^\star)\left[p\right]\|_2}-\frac{\mathbb{I}_n}{\sqrt{n}}\right\|_2\tag{24a}$$

$$=\left[2-\frac{2\mathbb{I}_n^\top f(\mathbf{X}^\star)[p]}{\sqrt{n}\,\|f(\mathbf{X}^\star)\left[p\right]\|_2}\right]^{1/2}\tag{24b}$$

$$=\left[2-\frac{2\mathbb{I}_n^\top}{\sqrt{n}}\frac{f(\mathbf{X}^\star)[p]}{\sqrt{\frac{1}{n}(\mathbb{I}_n^\top f(\mathbf{X}^\star)[p])^2+(\boldsymbol{v}^{\star\top}f(\mathbf{X}^\star)[p])^2+(\boldsymbol{u}^{\star\top}f(\mathbf{X}^\star)[p])^2}}\right]^{1/2}\tag{24c}$$

$$\geq\left[2-\frac{2\mathbb{I}_n^\top}{\sqrt{n}}\frac{f(\mathbf{X}^\star)[p]}{\sqrt{\frac{1}{n}(\mathbb{I}_n^\top f(\mathbf{X}^\star)[p])^2+(\boldsymbol{v}^{\star\top}f(\mathbf{X}^\star)[p])^2}}\right]^{1/2}\tag{24d}$$

$$=\left[2-2\frac{\frac{\mathbb{I}_n^\top f(\mathbf{X}^\star)[p]}{\sqrt{n}\left|\boldsymbol{v}^{\star\top}f(\mathbf{X}^\star)[p]\right|}}{\sqrt{1+\frac{\left|\mathbb{I}_n^\top f(\mathbf{X}^\star)[p]\right|^2}{n\left|\boldsymbol{v}^{\star\top}f(\mathbf{X}^\star)[p])\right|^2}}}\right]^{1/2}\tag{24e}$$

$$=\left[2-2\frac{\frac{2^{(1-\alpha)L-1}\left|\mathbb{I}_n^\top\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ\cdots\circ\varphi_1(\mathbf{X}^\star)[p]\right|}{\sqrt{n}(1+\beta_D)^L\left|\boldsymbol{v}^{\star\top}\mathbf{X}^\star[p]\right|}}{\sqrt{1+\frac{2^{2[(1-\alpha)L-1]}}{n(1+\beta_D)^{2L}}\frac{\left|\mathbb{I}_n^\top\hat{\varphi}_{\alpha L}\circ\varphi_{\alpha L-1}\circ\cdots\circ\varphi_1(\mathbf{X}^\star)[p]\right|^2}{\left|\boldsymbol{v}^{\star\top}\mathbf{X}^\star[p]\right|^2}}}\right]^{1/2},\tag{24f}$$

where equation (24c) is based on (20), equation (24f) is based on (23) and (16). When $\alpha>\log_2\frac{2}{1+\beta_D}$, we have $\lim_{L\to\infty}\left(\frac{2^{1-\alpha}}{1+\beta_D}\right)^L=0$. Thus we have $\lim_{L\to\infty}\left\|\frac{f(\mathbf{X}^\star)[p]}{\|f(\mathbf{X}^\star[p]\|_2}-\frac{\mathbb{I}_n}{\sqrt{n}}\right\|_2=\sqrt{2}$. This indicates that the p-th column of the output matrix $f(\mathbf{X}^\star)$ is not parallel to $\mathbf{I}_n$ for any p. This further indicates that the output matrix does not have the identical vector in each row. □

B.4. Technical Lemma

Lemma 3. *For any $x\in(0,1)$, it always holds $\left[1-\frac{1}{\sqrt{1+x^2}}\right]^{1/2}\leq x$.*

Proof. To establish the inequality $\left[1-\frac{1}{\sqrt{1+x^2}}\right]^{1/2}\leq x$,

we begin by proving,

$$1 - \frac{1}{\sqrt{1+x^2}} \leq x^2. \tag{25}$$

To demonstrate (25), we equivalently show

$$1 - x^2 \leq \frac{1}{\sqrt{1+x^2}}. \tag{26}$$

Subsequently, it suffices to verify

$$(1-x^2)(\sqrt{1+x^2}) \leq 1. \tag{27}$$

This is equivalent to proving

$$(1-x^2)^2(1+x^2) \leq 1. \tag{28}$$

Thus, our focus shifts to demonstrating

$$(1-x^2)(1-x^4) \leq 1. \tag{29}$$

Clearly, (29) holds true for any $x \in (0,1)$. □

C. Experiments

Our code is available at: https://github.com/OTTO-OTO/EX-Priv-GreyBoxResilience.

C.1. Experimental setups

Foundation Large Language Model. To demonstrate the efficacy of EX-Priv in enhancing the resilience of LLMs against grey-box extraction, we conduct experiments with **3** open-source, decoder-only structured LLMs with various architectures, including Llama2-7B (Touvron et al., 2023), Mistral-7B (Jiang et al., 2023), Falcon-7B (Almazrouei et al., 2023). We designated these pre-trained models as the victim models and assess the resilience of each privatization scheme.

Datasets. For the attack dataset aimed at maximizing model recovery, we merge data from the MMLU benchmark (Hendrycks et al., 2021) with the Alpaca 52k dataset (Wang et al., 2022), maintaining a 1:1 ratio. This integration yields a combined set, consisting of 51k samples for training and a separate validation set with 1.5k test instances. These datasets, designed to cover diverse model capabilities, have been tailored for instruction-following fine-tuning. Meanwhile, we construct another evaluation dataset with 1.5k samples from the same sources to select the privatization scheme in our algorithm.

EX-Priv Algorithm. We apply the EX-Priv algorithm to identify the smallest privatization set I such that $R(I) \leq (1+\varepsilon)R([L])$. In the following experiments, the tolerance magnitude ε is set to 0.2 to limit the gap between $R(I)$ and $R([L])$. To further show the sensitivity of EX-Priv to ε, we select ε values evenly from 0.1 to 1, incrementing by 0.1. To calculate the resilience score (RS) in EX-Priv, we randomly initialize the parameter matrices in the privatized layers and average the testing loss of model on the evaluation dataset under three different seeds. The sensitivity of EX-Priv to ε can be found in Appendix D.

C.2. Model Details

The pretrained models we use in our experiments are selected from open-source repositories, and Table 3 shows the basic information of the models and their sources. Specifically, we employ the fine-tuned version of Llama2-7B, known as Llama2-7B-chat[1], Mistral-7B-v0.1[2], and Falcon-7B[3], each featuring 7 billion parameters and 32 decoder layers.

Table 3. Model Info.

Model	Size	Decoder Layers
Llama2-7B-chat	7B	32
Mistral-7B-v0.1	7B	32
Falcon-7B	7B	32

C.3. Datasets

51k Training Dataset. To ensure the extensive coverage and reliability of our attack datasets, we employ a balanced approach by utilizing data from both the MMLU auxiliary training set[4] and the alpaca dataset[5] at a 1:1 ratio (both can be found at their GitHub repository). Specifically, we extract 50% of the data from each source in the MMLU set, totaling approximately 25.5k data samples. Observing the clustered arrangement of task-specific data in the alpaca dataset, we similarly extract another 25.5k samples using a step size of 2 to enhance the diversity of our dataset. Next, we convert the dataset into a format suitable for model training. For the Alpaca data within the training set, we apply the training prompt from (Taori et al., 2023), as shown in Table 4. For the MMLU auxiliary training data (Hendrycks et al., 2021), we utilize Prompt in Table 5.

Table 4. Prompts for Alpaca data

Alpaca with input	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
Alpaca with no answers	Below is an instruction that describes a task. Write a response that appropriately completes the request.

Validation Datasets. Table 6 presents a detailed composi-

[1]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[2]https://huggingface.co/mistralai/Mistral-7B-v0.1
[3]https://huggingface.co/tiiuae/falcon-7b
[4]https://github.com/hendrycks/test
[5]https://github.com/tatsu-lab/stanford_alpaca/blob/main/alpaca_data.json

Table 5. Prompts for MMLU auxiliary training data

Question Answering	Below is a question with no choices. Write the correct answer that appropriately solves the question.
Multiple Choice	The following is a multiple choice question, paired with choices. Answer the question in format: "Choice:content".

tion of the validation dataset employed in the experiments. For comprehensive task coverage, we systematically extracted 50% of the entries from each of the 57 sub-datasets included in the MMLU validation set, corresponding to the 57 distinct tasks. Subsequently, a corresponding quantity of data entries is systematically selected from the alpaca dataset using a step size of 751 to construct *Validation Dataset* . This dataset, integral to our experiments, encompasses approximately 1.5k instances. This validation dataset is employed in the training phase with 51k training sets.

Table 6. Composition of validation datasets of different sizes

Raw Data Set	Validation Set
Alpaca	765
MMLU validation set	751
Total Length	1516

Evaluation Datasets. To calculate the model's resilience score under various privatization strategies, we crafted an 1.5k Evaluation Set covering a wide range of tasks. This set includes a distinct 50% of entries from each of the 57 sub-datasets in the MMLU validation dataset, intentionally different from those in Validation Set. Additionally, we chose a matching number of entries from the alpaca dataset with a step size of 751, guaranteeing that there is no overlap with Validation Set.

C.4. Capability Benchmarks and Model Recovery

Evaluation on Capability Benchmarks. We follow the Llama-2 report (Touvron et al., 2023) to evaluate the recovered model, including 16 benchmarks, which are categorized into 6 groups: (1) *Commonsense Reasoning* (Rsn.) consists of PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019), ARC easy and challenge (Clark et al., 2018); (2) *Reading Comprehension* (Read.) group consists of OpenBookQA (Mihaylov et al., 2018), LAMBADA (Paperno et al., 2016), BoolQ (Clark et al., 2019) and SQuADv2 (Rajpurkar et al., 2018); (3) *World Knowledge* (Knl.) group consists of NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017); (4) *Code* group consists of HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021); (5) *Math* group consists of GSM8K (Cobbe et al., 2021); (6) *General Ability* (Gen.) group consists of two benchmarks *MMLU* (Hendrycks et al., 2021) and *BBH* (Suzgun et al., 2022). Following the established evaluation frameworks (Gao et al., 2023)[6] and (Ben Allal et al., 2022)[7], our model ranks choices in multiple-choice tasks and generates answers for open-ended generation tasks.

Table 7 details the set of capability benchmarks used in the experiments as well as the corresponding test methods and performance metrics. We adopt the lm-evaluation suite, in conjunction with the big-code platform. The lm-evaluation suite facilitated a battery of tests that included, but are not limited to, Commonsense Reasoning, Reading Comprehension, and Mathematical Problem Solving. We also leveraged the computational power of the big-code platform to handle the intensive processing requirements of training and evaluating our models. These benchmarks are categorized into 6 groups in total.

Rsn. We assess zero-shot classification accuracy on PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019), ARC easy and challenge (Clark et al., 2018).
Read. Zero-shot accuracy evaluations are conducted on OpenBookQA (Mihaylov et al., 2018), LAMBADA (Paperno et al., 2016), and BoolQ (Clark et al., 2019). Additionally, a two-shot exact_match and f1 score evaluation is performed on the 'hasAnswer' subset of SQuADv2 (Rajpurkar et al., 2018).
Knl. We perform five-shot task exact_match evaluation on NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017).
Code. The model's coding capability is verified using pass@1 scores through zero-shot tests on HumanEval (Chen et al., 2021) and one-shot evaluation on MBPP (Austin et al., 2021)
Math. An eight-shot exact_match evaluation method is employed to gauge the model's mathematical ability on the GSM8K benchmark (Cobbe et al., 2021).
Gen. We perform five-shot accuracy evaluation on MMLU (Hendrycks et al., 2021) and three-shot accuracy evaluation on BBH (Suzgun et al., 2022).

Model Recovery Training Set. In model recovery, we employ the AdamW optimizer, a cosine learning rate scheduler with an initial learning rate of 2×10^{-5}, a weight decay of 0.1, a global batch size of 128, a maximum training sequence length of 512 tokens and the bfloat16 training precision adhering to the configurations outlined in (Taori et al., 2023) and (Touvron et al., 2023).Our training setup, adapted from the llama-recipes[8] GitHub repository, involves full parameter fine-tuning over 5 epochs, conducted across 3 different seeds.

[6]https://github.com/EleutherAI/lm-evaluation-harness
[7]https://github.com/bigcode-project/bigcode-evaluation-harness
[8]https://github.com/meta-llama/llama-recipes

Table 7. The benchmark datasets used in the experiment

	Benchmark	Metric	n-shot
Rsn.	PIQA	Accuracy	0
	Hellaswag	Accuracy	0
	Winogrande	Accuracy	0
	ARC_easy	Accuracy	0
	ARC_challenge	Accuracy	0
Read.	OpenBookQA	Accuracy	0
	LAMBADA	Accuracy	0
	BoolQ	Accuracy	0
	SQuADv2	HasAns_EM	2
	SQuADv2	HasAns_F1	2
Knl.	NaturalQuestions	Exact Match	5
	TriviaQA	Exact Match	5
Code	HumanEval	Pass@1	0
	MBPP	Pass@1	1
Math	GSM8K	Exact Match	8
Gen.	MMLU	Accuracy	5
	BBH	Accuracy	3

Table 8. Standard errors across different groups on 7B model (Partial Privatization | Full Privatization)

	Benchmark	Llama-7B	Mistral-7B	Falcon-7B
Rsn.	PIQA	0.372 | 0.285	1.016 | 0.165	0.506 | 0.497
	Hellaswag	0.547 | 0.338	0.300 | 0.104	2.904 | 0.756
	Winogrande	1.982 | 1.568	2.140 | 1.512	3.010 | 2.906
	ARC_easy	1.050 | 0.962	1.540 | 1.006	2.009 | 1.101
	ARC_challenge	2.036 | 0.832	3.010 | 2.011	2.009 | 2.011
Read.	OpenBookQA	4.619 | 2.260	5.316 | 4.010	3.015 | 2.713
	LAMBADA	0.017 | 0.010	0.013 | 0.010	0.008 | 0.005
	BoolQ	0.085 | 0.087	0.070 | 0.073	0.103 | 0.089
	SQuADv2_EM	0.000 | 0.000	0.000 | 0.000	0.000 | 0.000
	SQuADv2_F1	0.669 | 0.738	0.903 | 0.821	0.387 | 0.202
Knl.	NaturalQuestions	0.077 | 0.189	0.017 | 0.034	0.000 | 0.031
	TriviaQA	0.000 | 0.025	0.000 | 0.050	0.000 | 0.015
Code	HumanEval	0.000 | 0.000	0.000 | 0.000	0.000 | 0.000
	MBPP	0.000 | 0.000	0.000 | 0.000	0.000 | 0.000
Math	GSM8K	0.000 | 0.000	0.000 | 0.000	0.000 | 0.000
Gen.	MMLU	0.822 | 0.309	1.352 | 0.810	0.981 | 0.736
	BBH	0.000 | 0.000	0.000 | 0.000	0.000 | 0.000

The evaluation and training are performed on servers equipped with a variety of Nvidia GPUs—specifically, the 4090 24G, 6000 Ada 48G, and A100 80G—utilizing bfloat16 training precision during the finetuning process. To ensure the consistency and reliability of the results, all experiments are replicated across three different seeds, with the outcomes averaged. The experimental setup ran on Ubuntu 20.04.6 LTS, with PyTorch version 2.2.0 and NVIDIA CUDA 11.8.

C.5. Standard Deviation on Benchmarking Results

We conduct experiments with three distinct random seeds to ensure the robustness of our findings. Each experiment evaluated the model's performance across various tasks. The results, as presented in Table 8, are reported with the mean performance metrics standard errors ($StdErr$) across the three trials. A smaller StdErr indicates that the model's performance metrics are more stable.

C.6. Scatter Plots for Correlations in Models

Figure 4 presents scatter plots illustrating the relationship between ΔARR and the resilience scores across six models, alongside the corresponding Spearman and Pearson correlation coefficients. The resilience scores are obtained from section 4.2. As shown in Figure 4, we observe a discernible trend: an increase in ΔARR is associated with a decrease in model scores across all examined models. This inverse relationship is robustly supported by both Spearman and Pearson correlation coefficients, which consistently exhibit strong negative values. The most pronounced negative correlations are observed in the Falcon-7B model, indicating a significant reduction in model scores with increasing ΔARR.

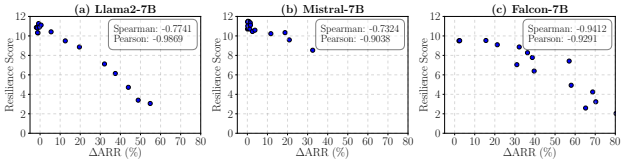

Figure 4. Correlation Analysis of ΔARR and Resilience Score Across Different Models

D. Experiment Supplement Results

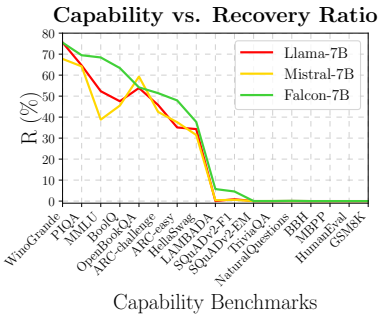

Figure 5. Effectiveness on various capability benchmarks.

Effectiveness on Various Capability Benchmarks. To explore the efficacy of EX-Priv across various capabilities, we visualize the Table 1 in Figure 5 to analyze the significant variance in recovery ratios among different capability benchmarks. Notably, in tasks related to commonsense reasoning like PIQA and Winogrande, recovery ratios consistently surpass 70% across all models. However, in technical domains such as code and math tasks like HumanEval and GSM8K, recovery ratios uniformly drop to 0%. This indicates the notable challenge in recovering capabilities within these specialized areas. This discrepancy can be due to the inherent characteristics of each benchmark. For example, tasks like PIQA, with only two choices provided, allow for a random guess to achieve a 50% success rate, thereby artificially inflating the recovery ratio. In contrast, code and

math tasks require a higher degree of precision and logical rigor, which the models may struggle to achieve, resulting in a 0% recovery ratio.

Sensitivity of EX-Priv to ε. To evaluate the sensitivity of EX-Priv to tolerance magnitude ε, we incrementally adjust ε from 0.1 to 1 in steps of 0.1 and calculate the ΔARR of six recovered models under EX-Priv privatization. Figure 6 illustrates that EX-Priv exhibits low sensitivity to changes in ε. For instance, the ΔARRs tend to stabilize across all models as ε increases. This stability may come from the requirement for a smaller privatization set at larger ε values to meet the condition $R(I) \leq (1+\varepsilon)R([L])$. In other words, as ε rises, the need for extensive privatization diminishes, enabling fewer layers to satisfy this criterion. Moreover, we observe that the increment in Δ ARR as ε increases is comparatively smaller in larger models. This is in line with the observation that privatizing more parameters above a threshold only provides limited improvement in resilience.

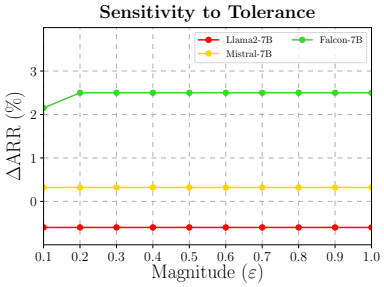

Figure 6. Sensitivity of EX-Priv to ε on models of size 7B.