ANAMORPH: A SECURE AND EFFICIENT TEE-SHIELDED OBFUSCATION SCHEME FOR ON-DEVICE LLM INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Deploying large language models on untrusted devices poses a risk of model weight leakage, and Trusted Execution Environment (TEE)-based model obfuscation provides a practical solution for protecting model weights. The state-of-the-art lightweight defense based on dense orthogonal matrices can effectively resist cosine-similarity-based analysis attacks while reducing TEE-side computation. However, we identify a key vulnerability in existing lightweight obfuscation schemes: statistical similarity between private fine-tuned models and public pretrained models. Furthermore, we find that the problems of estimating secret transformation matrices in multiple existing obfuscation schemes can be reduced to orthogonal Procrustes problems. Based on these observations, we propose ORTHOTRACE, a novel attack that exploits the correspondence between public pretrained weights and exposed obfuscated weights to estimate secret transformation matrices, thereby compromising multiple obfuscation mechanisms, including the state-of-the-art lightweight defense. To mitigate this vulnerability, we propose ANAMORPH, a novel obfuscation scheme that combines parameter-dimension expansion with the injection of random masks satisfying null-space constraints to perturb the orthogonal estimation relationships exploited by ORTHOTRACE. These masks vanish automatically during authorized inference, reducing the need to enter the TEE for layerwise mask correction and maintaining lightweight TEE computation while ensuring model security. We evaluate ORTHOTRACE and ANAMORPH on four representative models and eight datasets. Experimental results show that ORTHOTRACE compromises multiple existing lightweight obfuscation schemes, recovering models with task performance close to the no-protection baseline, with the task recovery rate exceeding 95% for each scheme. ANAMORPH demonstrates effective defense against existing similarity-based analysis attacks and ORTHOTRACE while preserving authorized-inference utility, and achieves up to a $13.23\times$ speedup in time to first token (TTFT) compared with an effective defense scheme that requires layerwise correction on real SGX hardware.

1 INTRODUCTION

Data privacy and low-latency inference requirements are driving the deployment of large language models (LLMs) on user-controlled edge devices. However, such deployments also expand the attack surface for model extraction. Attackers can extract model architectures and weights through memory side-channel attacks (Rakin et al., 2022), dynamic analysis (Sun et al., 2021), and bus monitoring (Zhu et al., 2021), often at a lower cost than black-box attacks that rely on extensive querying and training. The TEE provides hardware-enforced isolation to protect code and data, offering an effective means of safeguarding models on untrusted devices (Wang et al., 2025; Zhang et al., 2024; Wang et al., 2026).

TEE-shield obfuscation schemes offload computationally intensive linear operations to untrusted GPUs while retaining secret transformation parameters and any necessary corrections to inputs, outputs, or intermediate states within the TEE. These schemes aim to reduce information leakage from white-box access to the black-box level. Early schemes, including ShadowNet (Sun et al., 2023),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Comparison of TEE-shield obfuscation schemes in terms of Weight obfuscation, TEE interactions, Security, and Efficiency. Sec. and Eff. stand for Security and Efficiency, respectively. L denotes the model depth. ANAMORPH is ours.

Method	Venue	Weight obfuscation	Annotation	TEE interact.	Sec.	Eff.
ShadowNet	S&P '23	$g(W) = \pi^\top DW$	Depthwise local map; diagonal D; permutation π.	$O(L)$	✗	✗
Tempo	IJCNN '24	$g(W) = WD\pi$	One-sided factor; diagonal D; permutation π.	$O(L)$	✗	✗
TransLink-Guard	MM '24	$g(W) = W\pi$	π is specific to each layer.	$O(L)$	✗	✗
CoreGuard	NeurIPS '25	$g(W) = \pi^\top W$	π is shared across protected layers.	$O(1)$	✗	✓
ArrowCloak	USENIX Security'25	$g(W) = (WD_1 + v\mathbf{1}^\top D_2)\pi$	Diagonal D_1, D_2; $v = \sum_j k_j w_j$; w_j: columns of W.	$O(L)$	✓	✗
Slim	ICML '26	$g(W) = D^{-1}\Pi^\top DW$	$\Pi = \prod_j \Lambda_j \pi_j$; Λ_j: block-diagonal Householder.	$O(1)$	✗	✓
ANAMORPH	–	$g(W) = \frac{n}{p} HDW + \text{Mask}$	$H^\top H = \frac{p}{n} I_n, p > n$; $D = \text{diag}(\gamma)$; $H^\top \text{Mask} = 0$.	$O(1)$	✓	✓

Tempo (Xu & Fang, 2024), and TransLinkGuard (Li et al., 2024), employ lightweight transformations such as row and column permutations and per-vector scaling to accommodate the TEE's limited computational resources. To further reduce computational and communication overhead, CoreGuard (Li et al., 2025) introduces a propagation protocol based on a shared permutation π. However, these schemes preserve directional correspondence between obfuscated weight vectors and their pretrained counterparts. ArrowMatch (Wang et al., 2025) exploits this information to identify vector correspondences using cosine similarity and successfully reconstruct the model. To counter cosine-similarity-based matching, ArrowCloak (Wang et al., 2025) uses matrix–vector multiplication to prevent direct directional matching while maintaining low computational overhead within the TEE. However, its layer-wise TEE interactions increase with LLM depth, resulting in substantial inference overhead. Slim (Wang et al., 2026) builds on CoreGuard's propagation protocol and further employs dense orthogonal matrices to conceal the directions of individual weight vectors. Nevertheless, concealing individual vector directions does not eliminate the global relationships between the exposed weights and the corresponding weights in the public pretrained model.

Private victim models are typically derived by fine-tuning public pretrained models, and the fine-tuning process needs to preserve pretrained knowledge as much as possible (Gupta et al., 2025). Therefore, public pretrained weights can serve as a natural parameter reference for private fine-tuned models. Meanwhile, weight obfuscation mappings used by ShadowNet, Tempo, TransLinkGuard, CoreGuard, and Slim can all be reduced to orthogonal transformations under the corresponding conditions. This allows the estimation of the secret transformations to be formulated within a unified orthogonal Procrustes framework. Based on these observations, we propose ORTHOTRACE.

ORTHOTRACE aims to construct a standalone surrogate model M_{sur} whose functionality is close to that of the victim model M_{vic}. Based on the relationships between public pretrained weights and exposed obfuscated weights, it formulates estimation equations for the secret orthogonal transformation matrices and estimates these matrices. The estimated matrices are then used to deobfuscate the weights and reconstruct an executable model. Subsequently, the reconstructed model is fine-tuned on a small amount of data to improve its functional similarity to M_{vic}. To mitigate this attack, we propose ANAMORPH.

ANAMORPH uses scaled semi-orthogonal matrices to expand the dimensions of parameter matrices, creating a redundant space that accommodates random additive weight masks satisfying the corresponding null-space constraints. These masks perturb the weight relationships used to estimate the

secret transformation matrices, while their contributions vanish during authorized inference. This reduces the need for explicit layerwise mask removal within the TEE.

We conducted evaluations across four models and eight datasets. Experimental results show that ORTHOTRACE recovers at least 95% of the original model's task score, with its effectiveness extending to Slim, the state-of-the-art lightweight defense scheme. Under the evaluated attack pipelines, ANAMORPH effectively defends against both the state-of-the-art attack ArrowMatch and our proposed ORTHOTRACE. While preserving authorized-inference functionality, ANAMORPH achieves a 4.72–$13.23\times$ speedup in TTFT compared with ArrowCloak, with throughput reaching 1.68–$2.15\times$ that of ArrowCloak.

We summarize the contributions as follows:

- We propose a novel attack, ORTHOTRACE, which exploits the correspondence between public pretrained weights and exposed obfuscated weights through orthogonal Procrustes analysis to recover models protected by existing orthogonal obfuscation schemes.
- We propose a novel obfuscation scheme, ANAMORPH, which combines parameter-dimension expansion with random masks satisfying null-space constraints to protect model weights without layerwise mask correction inside the TEE.
- We evaluate ORTHOTRACE and ANAMORPH on four models and eight datasets. The results show that ORTHOTRACE effectively recovers the protected models, while ANAMORPH effectively resists both ArrowMatch and ORTHOTRACE under the evaluated attack pipeline. Compared with ArrowCloak, ANAMORPH achieves lower TTFT and higher throughput.

2 BACKGROUND

TEE-Shielded Full Inference Schemes. These schemes retain plaintext model weights within the TEE and execute the entire inference pipeline on trusted processors. ConfidentialDL (VanNostrand et al., 2019), Real-Time DNN (Babar & Hasan, 2022), and T-Slices (Islam et al., 2023) load model layers into trusted memory and decrypt them before execution, thereby reducing the attack surface exposed to the host system. However, inference performance remains constrained by the TEE's computational throughput and trusted memory capacity. For LLM, the overhead of trusted execution and limited use of GPU acceleration introduce additional latency, making it difficult for TEE-shielded full inference schemes to meet the real-time requirements of on-device generation.

TEE-Shielded Obfuscation Schemes. These schemes keep obfuscation secrets and other security-sensitive state within the TEE while offloading the main computational workload to an untrusted GPU. Existing approaches broadly fall into two categories: partial and full obfuscation. Partial obfuscation methods, such as Magnitude (Hou et al., 2022) and NNSplitter (Zhou et al., 2023), protect a small subset of important model parameters. However, distributional differences between the protected parameters and the remaining weights reveal their locations. Full obfuscation methods, including ShadowNet (Sun et al., 2023), Tempo (Xu & Fang, 2024), TransLinkGuard (Li et al., 2024), and CoreGuard (Li et al., 2025), apply scaling or permutation transformations to model weights. These transformations preserve exploitable directional information. To counter direct directional matching, ArrowCloak (Wang et al., 2025) prevents direct matching by introducing a masking term computed through matrix–vector multiplication. To ensure correct inference, ArrowCloak requires intermediate results to be corrected within the TEE at each protected layer, resulting in $O(L)$ GPU–TEE interactions for a model of depth L. Slim (Wang et al., 2026) constructs a propagation protocol using blockwise orthogonal matrices, reducing GPU–TEE interactions to $O(1)$ per forward pass, independent of model depth. However, Slim loses its defensive capability under the ORTHOTRACE attack.

3 THREAT MODEL

In this section, we will introduce the threat model of this work. The threat model follows prior studies (Sun et al., 2023; Zhang et al., 2024; Wang et al., 2025; 2026).

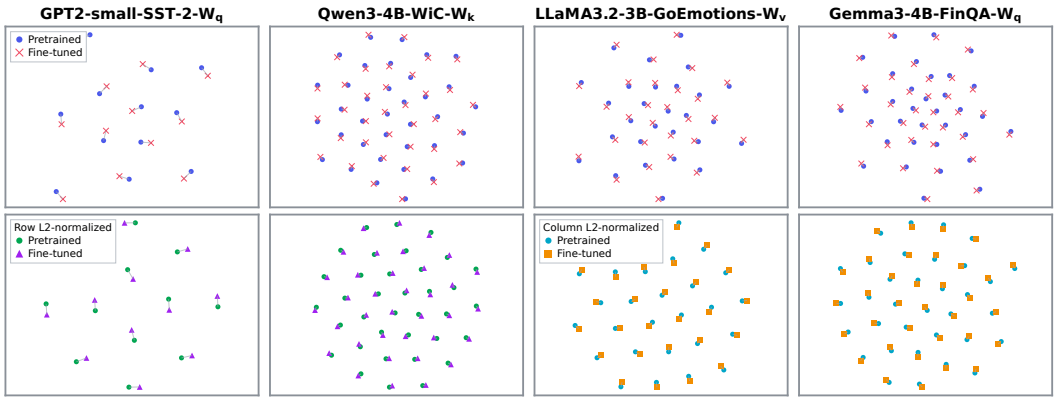

Figure 1: Layerwise correspondence between public pretrained and private fine-tuned weights, visualized with t-SNE. The bottom row uses row-wise L2 normalization for GPT2-Small and Qwen3-4B, and column-wise L2 normalization for LLaMA3.2-3B and Gemma3-4B.

Adversary's Goal. The adversary aims to steal the functionality of a deployed proprietary model to bypass payment requirements and access restrictions, reduce model development costs, or profit from resale. Specifically, the adversary seeks to construct a surrogate model M_{sur} that is functionally equivalent to the victim model M_{vic}, preserving its task functionality while operating independently without a TEE. Exact recovery of the victim model's weights is not required.

Adversary's Capabilities. The adversary can extract the obfuscated model parameters using model stealing techniques (Sun et al., 2021; Zhu et al., 2021; Rakin et al., 2022), but cannot access data within the TEE. We assume that the adversary can use public pretrained models and obtain labeled data amounting to 1% of the training set by querying the victim model. The resulting query-labeled samples are subsequently used to train a surrogate model. This use of model queries for surrogate training is consistent with prior model stealing studies (Orekondy et al., 2019; Zhang et al., 2024; Wang et al., 2025; 2026). We do not consider attacks targeting the TEE itself, such as side-channel attacks.

Defender's Goal and Capabilities. The defender aims to reduce the additional information leakage caused by white-box exposure on untrusted devices. The model owner can control data and code inside the TEE to protect the model, while the defender can design weight obfuscation algorithms to protect the model's exposed representation in the untrusted environment.

4 ORTHOTRACE

4.1 Attack Insight

Insight 1. LLMs tailored to proprietary tasks are typically developed by fine-tuning publicly available pretrained models. In certain fine-tuning settings, such as sequential knowledge editing, constraining the growth of weight matrix norms helps preserve the model's general capabilities and mitigate performance degradation (Gupta et al., 2025). Therefore, publicly available pretrained weights can serve as natural approximate references for victim weights; the t-SNE visualization in Figure 1 also qualitatively illustrates the strong correspondence between the two. Appendix A reports the corresponding parameter norm changes. Notably, this work does not use norm-constrained training methods, employing only standard full-tuning.

Insight 2. TransLinkGuard (Li et al., 2024) and CoreGuard (Li et al., 2025) employ column and row permutations, respectively. Both row and column permutations are orthogonal. ShadowNet (Sun et al., 2023) and Tempo (Xu & Fang, 2024) can be transformed through row normalization and column normalization, respectively, into obfuscation schemes similar to TransLinkGuard and CoreGuard. Applying the same normalization to pretrained weights provides references for es-

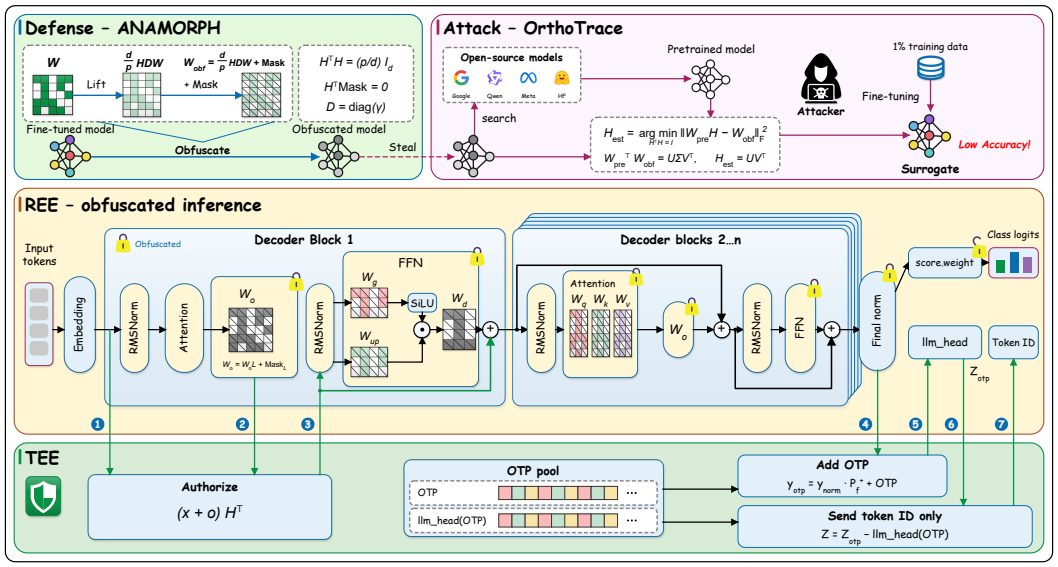

Figure 2. ANAMORPH's TEE–REE execution and OrthoTrace's recovery workflow.

timation, as illustrated in Figure 1. For Slim (Wang et al., 2026), using the exposed D yields $DW_{\text{obf}} = \Pi^\top DW_{\text{vic}}$ and DW_{pre}, where W_{obf}, W_{vic}, and W_{pre} denote the obfuscated weights, victim weights, and public pretrained weights, respectively. Estimating $\Pi^\top$ can be formulated as an orthogonal Procrustes problem. Therefore, after preprocessing, these schemes can be unified within an orthogonal estimation framework using a publicly available pretrained model as a reference.

4.2 Attack Methodology

Given the corresponding pretrained and exposed matrices, OrthoTrace formulates the orthogonal matrix estimation problem as follows:

$$H_{\text{est}} = \arg\min_{H^\top H = I} \|W_{\text{pre}} H - W_{\text{obf}}\|_F^2. \tag{1}$$

Here $\|\cdot\|_F$ is the Frobenius norm. Using $HH^\top = I$ gives

$$\|W_{\text{pre}} H - W_{\text{obf}}\|_F^2 = \|W_{\text{pre}}\|_F^2 + \|W_{\text{obf}}\|_F^2 - 2\,\text{tr}\big(H^\top W_{\text{pre}}^\top W_{\text{obf}}\big)\,. \tag{2}$$

Here tr is the matrix trace. The terms $\|W_{\text{pre}}\|_F^2$ and $\|W_{\text{obf}}\|_F^2$ are independent of H. The estimate therefore also solves

$$H_{\text{est}} = \arg\max_{H^\top H = I} \text{tr}\big(H^\top W_{\text{pre}}^\top W_{\text{obf}}\big)\,. \tag{3}$$

Take the singular value decomposition $W_{\text{pre}}^\top W_{\text{obf}} = U\Sigma V^\top$, where U and V are orthogonal. Define $Z = V^\top H^\top U$. Z is also orthogonal. Using the cyclic property of the trace gives

$$\text{tr}\big(H^\top W_{\text{pre}}^\top W_{\text{obf}}\big) = \text{tr}\big(H^\top U\Sigma V^\top\big) = \text{tr}\big(V^\top H^\top U\Sigma\big) = \text{tr}(Z\Sigma) = \sum_{i=1}^{d} \sigma_i z_{ii}. \tag{4}$$

Since Z is orthogonal, $z_{ii} \le 1$ for every i. Therefore,

$$\text{tr}\big(H^\top W_{\text{pre}}^\top W_{\text{obf}}\big) \le \sum_{i=1}^{d} \sigma_i = \text{tr}(\Sigma). \tag{5}$$

Choosing $H = UV^\top$ gives $Z = I$, so every $z_{ii} = 1$ and the upper bound is attained. Hence

$$W_{\text{pre}}^\top W_{\text{obf}} = U\Sigma V^\top \quad \Longrightarrow \quad H_{\text{est}} = UV^\top. \tag{6}$$

5 ANAMORPH

ANAMORPH uses a scaled semi-orthogonal matrix H to expand the dimensions of model parameters and injects random masks satisfying the null-space constraint $H^\top \text{Mask} = 0$. These masks perturb parameter similarity and cancel automatically during authorized inference, requiring only $O(1)$ interactions with the TEE per forward pass. Based on ANAMORPH, we design an efficient TEE-based secure inference framework for LLMs in Figure 2.

Let l and d denote the input token length and hidden size, respectively; m denotes the FFN intermediate dimension. The dimensions are $x \in \mathbb{R}^{l\times d}$, $W_q, W_k, W_v, W_o \in \mathbb{R}^{d\times d}$, $W_{\text{up}}, W_g \in \mathbb{R}^{d\times m}$, and $W_d \in \mathbb{R}^{m\times d}$. The original decoder block is formalized as follows:

Original Decoder Block Formalization

$$
\begin{aligned}
&y = \gamma_1 \odot \frac{x}{\text{RMS}_d(x)} = \frac{x}{\text{RMS}_d(x)} D_1, \\
&Q = yW_q, \quad K = yW_k, \quad V = yW_v, \\
&o = \text{attn}(Q, K, V)W_o, \\
&z = \gamma_2 \odot \frac{x+o}{\text{RMS}_d(x+o)} = \frac{x+o}{\text{RMS}_d(x+o)} D_2, \\
&f = \left[\sigma(zW_g) \odot zW_{\text{up}}\right] W_d, \\
&\text{output} = x + o + f.
\end{aligned}
$$

Obfuscated weights. $D_1 = \text{diag}(\gamma_1)$ and $D_2 = \text{diag}(\gamma_2)$, where γ_1, γ_2 are the RMSNorm weights. Choose expansion matrices $H \in \mathbb{R}^{p\times d}$, $P \in \mathbb{R}^{d\times p}$, and $\pi \in \{0,1\}^{m\times k}$, with $p = d + s$ and $k = m + s$, where s denotes the expansion dimension. These matrices satisfy

$$
H^\top H = \frac{p}{d} I_d, \qquad PP^+ = I_d, \qquad \pi\pi^+ = I_m. \tag{7}
$$

Here H is a scaled semi-orthogonal matrix, P and π are full-row-rank matrices, and P^+ and π^+ are right inverse matrices of P and π, respectively. Obfuscation using these matrices yields:

$$
\begin{aligned}
&W'_q = \frac{d}{p} H D_1 W_q + \text{Mask}_q, \qquad W'_k = \frac{d}{p} H D_1 W_k + \text{Mask}_k, \\
&W'_v = \frac{d}{p} H D_1 W_v P + \text{Mask}_v, \qquad W'_o = P^+ W_o H^\top + \text{Mask}_o, \\
&W'_g = \frac{d}{p} H D_2 W_g \pi + \text{Mask}_g, \quad W'_{\text{up}} = \frac{d}{p} H D_2 W_{\text{up}} \pi + \text{Mask}_{\text{up}}, \\
&W'_d = \pi^+ W_d H^\top + \text{Mask}_d.
\end{aligned} \tag{8}
$$

The random masks satisfy $H^\top \text{Mask}_j = 0$ for $j \in \{q, k, v, g, \text{up}\}$, $P\text{Mask}_o = 0$, and $\pi\text{Mask}_d = 0$.

Decoder block formalization. With Eq. (8), the authorized block computes the following:

ANAMORPH Decoder Block Formalization

$$
\begin{aligned}
&x' = xH^\top, \qquad y' = \frac{xH^\top}{\text{RMS}_p(xH^\top)} = \frac{x}{\text{RMS}_d(x)} H^\top, \\
&Q' = y'W'_q = Q, \quad K' = y'W'_k = K, \quad V' = y'W'_v = VP, \\
&o' = \text{attn}(Q', K', V')W'_o = oH^\top, \\
&z' = \frac{(x+o)H^\top}{\text{RMS}_p((x+o)H^\top)} = \frac{x+o}{\text{RMS}_d(x+o)} H^\top, \\
&f' = \left[\sigma(z'W'_g) \odot z'W'_{\text{up}}\right] W'_d = fH^\top, \\
&\text{output}' = x' + o' + f' = \text{output}\, H^\top.
\end{aligned}
$$

Trusted execution. The TEE authorizes the residual after the first attention block. Each forward pass requires $O(1)$ TEE calls, independent of model depth. The transformed input $x' = xH^\top$ is processed by the obfuscated decoder block to produce the corresponding transformed output, satisfying $\mathrm{Decoder}^{(i)}_{\mathrm{obf}}(x^{(i-1)}H^\top) = x^{(i)}H^\top$, thereby propagating authorization from one decoder block to the next. A more detailed derivation of the decoder block formalization is provided in Appendix B. We design the authorization module, output processing, and OTP pool following prior work (Shen et al., 2022; Wang et al., 2026). Appendices C and D provide further details.

6 Experimental Evaluation

Setup. We evaluate four models: GPT2-Small (Radford et al., 2019), Qwen3-4B (Yang et al., 2025), LLaMA3.2-3B (Grattafiori et al., 2024), and Gemma3-4B (Team et al., 2025). GPT2-Small uses MNLI (Williams et al., 2018), QQP (Shankar et al., 2017), SST-2 (Socher et al., 2013), and QNLI (Rajpurkar et al., 2016) from GLUE (Wang et al., 2019); the other models use WiC (Pilehvar & Camacho-Collados, 2019), GoEmotions (Demszky et al., 2020), FinQA (Chen et al., 2021), and PubMedQA (Jin et al., 2019). We report accuracy for single-label classification, micro-F1 for GoEmotions, and execution accuracy for FinQA and PubMedQA. All experiments were conducted on a server equipped with NVIDIA RTX 5090 GPU and Intel Xeon Platinum 8470Q CPU. All ANAMORPH results use expansion $s = 32$.

6.1 Evaluation of Attacks and Defenses

Table 2: Evaluation of defense performance under ORTHOTRACE. GoEmo. and PMQA denote GoEmotions and PubMedQA, respectively. TLG denotes TransLinkGuard. For each scheme, the first row reports the score before attack; the second row (💀) reports the score after attack. Bold numbers indicate the lowest score after attack.

Scheme	GPT2-Small				LLaMA3.2-3B				Gemma3-4B				Qwen3-4B			
	MNLI	QQP	SST-2	QNLI	WiC	GoEmo.	FinQA	PMQA	WiC	GoEmo.	FinQA	PMQA	WiC	GoEmo.	FinQA	PMQA
ShadowNet	32.70	63.18	50.92	50.71	50.94	8.00	0.00	0.00	49.22	8.34	0.00	0.00	49.06	7.70	0.00	0.00
💀	81.03	88.96	91.63	88.20	71.63	56.69	61.14	88.00	70.53	57.71	63.14	84.00	75.08	54.33	68.76	86.00
Tempo	31.99	49.29	49.54	50.51	50.31	7.76	0.00	0.00	50.00	8.25	0.00	0.00	48.75	7.49	0.00	0.00
💀	80.97	88.96	91.51	88.14	71.94	56.78	61.24	87.33	70.38	57.52	63.33	84.67	74.76	54.34	68.67	87.33
TLG	35.44	36.82	49.08	50.53	50.78	8.36	0.00	0.00	49.84	9.84	0.00	0.00	52.98	5.63	0.00	0.00
💀	80.89	88.95	91.40	88.23	72.10	56.83	61.24	88.67	71.47	58.90	63.81	86.00	75.39	55.95	68.76	86.67
CoreGuard	32.47	57.12	50.92	49.47	50.00	8.85	0.00	0.00	55.80	7.77	0.00	0.00	50.00	4.55	0.00	0.00
💀	81.03	88.97	91.63	88.23	71.94	56.77	61.14	88.00	70.53	58.10	63.43	84.67	74.76	54.87	68.86	86.67
Slim	31.83	63.18	50.92	49.47	50.00	5.58	0.00	0.00	50.00	10.29	0.00	0.00	50.00	5.77	0.00	0.00
💀	80.98	88.94	91.74	88.31	72.10	56.83	61.05	85.33	70.22	58.76	63.81	85.33	74.76	54.75	68.67	88.00
ANAMORPH	33.02	63.17	50.92	51.06	53.45	8.27	0.00	0.00	50.00	17.94	0.00	0.00	46.55	7.31	0.00	0.00
💀	**31.83**	**58.53**	**51.61**	**51.59**	**50.63**	**5.43**	**0.00**	**0.00**	**50.00**	**3.34**	**0.00**	**0.00**	**49.84**	**0.95**	**0.00**	**0.00**
No-Shield	81.03	88.96	91.63	88.25	71.94	56.64	61.33	88.67	70.22	57.69	63.43	85.33	74.92	54.27	68.86	87.33
Random	32.57	40.66	51.15	50.81	50.00	9.36	0.00	0.00	51.72	9.28	0.00	0.00	46.39	7.59	0.00	0.00

Baseline. No-Shield represents the unprotected scenario, in which the adversary can directly obtain the victim model's architecture and weights. Random represents the scenario in which the adversary uses a randomly initialized model to perform the task. No-Shield serves as the security lower bound, and Random serves as the security upper bound.

Attack Analysis. Table 2 presents the results of ORTHOTRACE against ShadowNet, Tempo, TransLinkGuard, CoreGuard, and Slim. Before the attack, the obfuscated models achieve substantially lower task scores than No-Shield, with performance generally close to Random. After the

Table 3: Comparison of ORTHOTRACE and ArrowMatch against Slim and ANAMORPH in GPT2-Small.

Dataset	Slim		ANAMORPH		No-Shield	Random
	ORTHOTRACE	ArrowMatch	ORTHOTRACE	ArrowMatch		
MNLI	80.98	35.53	31.83	33.21	81.03	32.57
QQP	88.94	74.64	58.53	61.62	88.96	40.66
SST-2	91.74	54.82	51.61	50.23	91.63	51.15
QNLI	88.31	58.27	51.59	51.99	88.25	50.81

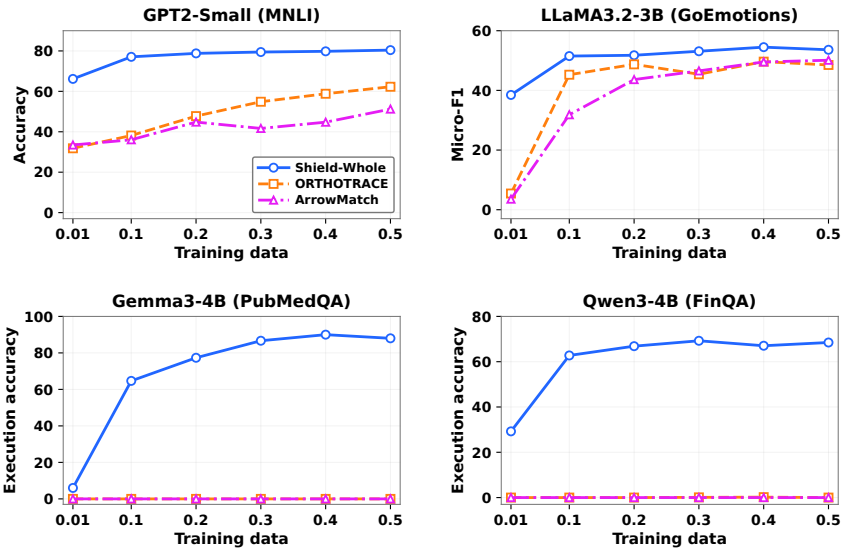

Figure 3: Defense performance of ANAMORPH against ArrowMatch and ORTHOTRACE under different adversary training-data budgets.

attack, ORTHOTRACE restores at least 95% of the corresponding No-Shield task score (Appendix F). The mean recovery ratio exceeds 99% for each obfuscation scheme.

Comparison of ORTHOTRACE and ArrowMatch. Across MNLI, QQP, SST-2, and QNLI on GPT2-Small, ORTHOTRACE achieves performance recovery ratios of 99.94%–100.12% against Slim, outperforming ArrowMatch (Table 3). Slim's dense orthogonal mixing undermines ArrowMatch's cosine-based direction matching. In contrast, ORTHOTRACE applies orthogonal Procrustes analysis to public pretrained and private fine-tuned weights to estimate the secret matrix. ORTHOTRACE exhibits stronger model recovery capabilities than ArrowMatch.

Defense Analysis. Table 2 shows that ANAMORPH effectively limits the model recovery capability of ORTHOTRACE, with post-attack scores close to the Random baseline. Table 3 further demonstrates that ANAMORPH provides protection against both ORTHOTRACE and ArrowMatch, with recovered model performance remaining close to the Random baseline. These results indicate that ANAMORPH resists model recovery attacks based on global orthogonal alignment and vectorwise direction matching in the evaluated settings.

Defense under Different Adversary Training-Data Budgets. We further evaluate ANAMORPH against fine-tuning attacks with labeled-data budgets up to 50% of the original training datasets. We compare ANAMORPH with the Shield-Whole baseline under matched training-data budgets. Shield-Whole represents the scenario in which the adversary can access the pretrained model M_{pre} and uses it as the initial model, $M_{\text{init}} = M_{\text{pre}}$. The adversary fine-tunes M_{init} using samples labeled by querying the victim model to construct a surrogate model. Figure 3 shows that, across

the evaluated data budgets, task performance recovered by ORTHOTRACE and ArrowMatch against ANAMORPH is generally lower than that of Shield-Whole.

6.2 AUTHORIZED INFERENCE ACCURACY

Table 4: Authorized inference accuracy.

Model	**GPT2-Small**				**LLaMA3.2-3B**				**Gemma3-4B**				**Qwen3-4B**			
Dataset	MNLI	QQP	SST-2	QNLI	WiC	GoEmo.	FinQA	PMQA	WiC	GoEmo.	FinQA	PMQA	WiC	GoEmo.	FinQA	PMQA
No-Shield	81.03	88.96	91.63	88.25	71.94	56.64	61.33	88.67	70.22	57.69	63.43	85.33	74.92	54.27	68.86	87.33
ANAMORPH	81.12	88.94	91.06	88.08	69.91	56.64	61.33	88.00	70.22	57.71	62.38	86.67	74.92	54.26	68.19	88.00

The inference environment is equipped with RTX 5090 and Intel Xeon Platinum 8470Q. As shown in Table 4, ANAMORPH closely matches the accuracy of the No-Shield baseline across four models and eight datasets, with an average reduction of only 0.19 percentage points. This confirms that the defense scheme preserves normal authorized inference utility while resisting attacks.

6.3 INFERENCE EFFICIENCY ANALYSIS

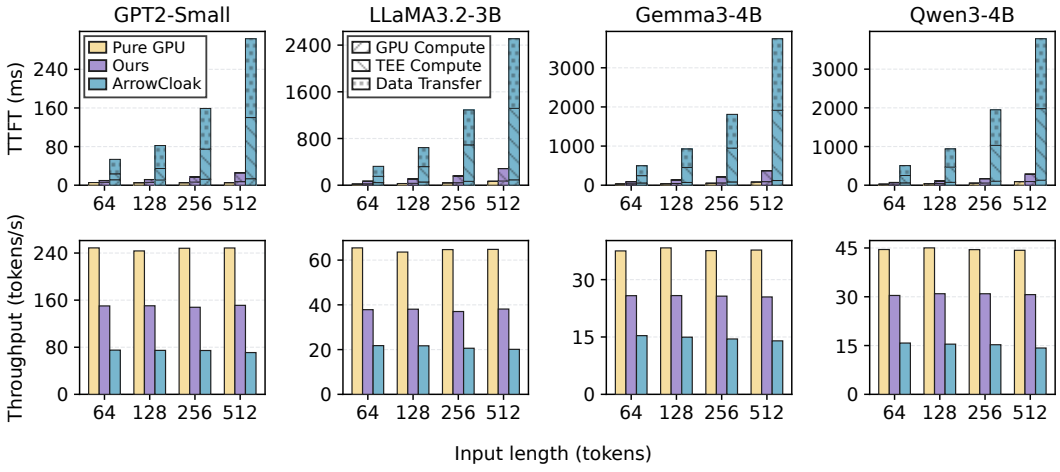

Figure 4: The performance of ANAMORPH (Ours) against two baselines (Pure GPU and ArrowCloak) in terms of TTFT and throughput.

Figure 4 shows that, across four causal language models and different input lengths, ANAMORPH achieves a 4.72–$13.23\times$ speedup in TTFT over ArrowCloak, while its throughput is 1.68–$2.15\times$ that of ArrowCloak. Relative to pure GPU inference, ANAMORPH incurs 1.77–$4.97\times$ the TTFT and achieves 0.57–$0.70\times$ the throughput, whereas ArrowCloak incurs 9.65–$58.15\times$ the TTFT and achieves 0.29–$0.41\times$ the throughput. ANAMORPH thus more closely approaches the pure GPU baseline in both TTFT and throughput. As input length increases, TTFT rises while throughput remains relatively stable.

7 CONCLUSION

This paper reveals the vulnerability in orthogonal obfuscation schemes and proposes ORTHOTRACE to recover obfuscated models through orthogonal matrix estimation. To defend against such attacks, we further propose a secure obfuscation scheme, ANAMORPH, which combines dimensional expansion and random weight masks to protect models. We extensively evaluate ORTHOTRACE and ANAMORPH and verify their effectiveness.

AI USE STATEMENT

In this work, LLMs were used solely to assist with manuscript writing and language polishing, including refining wording, checking grammar, and improving readability. LLMs were not involved in research conceptualization, methodological design, or experimental design; all research conceptualization and analysis were conducted independently by the authors. We take full responsibility for the final content of this work, including any text, claims, or research outputs produced with the assistance of generative artificial intelligence.

REPRODUCIBILITY STATEMENT

All models and datasets used in this paper are publicly available. We will release the complete code upon acceptance of this paper.

REFERENCES

Mohammad Fakhruddin Babar and Monowar Hasan. Real-time scheduling of trustzone-enabled dnn workloads. CPSIoTSec '22, pp. 63–69, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450398763. doi: 10.1145/3560826.3563386. URL https://doi.org/10.1145/3560826.3563386.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4040–4054, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.372. URL https://aclanthology.org/2020.acl-main.372/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Akshat Gupta, Phudish Prateepamornkul, Maochuan Lu, Ahmed Alaa, Thomas Hartvigsen, and Gopala Anumanchipalli. Lifelong knowledge editing requires better regularization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 22653–22675, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1234. URL https://aclanthology.org/2025.findings-emnlp.1234/.

Jiahui Hou, Huiqi Liu, Yunxin Liu, Yu Wang, Peng-Jun Wan, and Xiang-Yang Li. Model protection: Real-time privacy-preserving inference service for model privacy at the edge. *IEEE Transactions on Dependable and Secure Computing*, 19(6):4270–4284, 2022. doi: 10.1109/TDSC.2021.3126315.

Md Shihabul Islam, Mahmoud Zamani, Chung Hwan Kim, Latifur Khan, and Kevin W. Hamlen. Confidential execution of deep learning inference at the untrusted edge with arm trustzone. CODASPY '23, pp. 153–164, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700675. doi: 10.1145/3577923.3583648. URL https://doi.org/10.1145/3577923.3583648.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China,

November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Qinfeng Li, Zhiqiang Shen, Zhenghan Qin, Yangfan Xie, Xuhong Zhang, Tianyu Du, Sheng Cheng, Xun Wang, and Jianwei Yin. Translinkguard: Safeguarding transformer models against model stealing in edge deployment. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 3479–3488, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3680786. URL https://doi.org/10.1145/3664647.3680786.

Qinfeng Li, Tianyue Luo, Xuhong Zhang, Yangfan Xie, Zhiqiang Shen, Lijun Zhang, Yier Jin, Hao Peng, Xinkui Zhao, XianWei Zhu, and Jianwei Yin. Coreguard: Safeguarding foundational capabilities of llms against model stealing in edge deployment. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 69221–69243. Curran Associates, Inc., 2025. doi: 10.52202/085713-2327. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6447714b83edcbed61dbe10371dd7ae5-Paper-Conference.pdf.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4949–4958, 2019. doi: 10.1109/CVPR.2019.00509.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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)*, pp. 1267–1273. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Adnan Siraj Rakin, Md Hafizul Islam Chowdhuryy, Fan Yao, and Deliang Fan. Deepsteal: Advanced model extractions leveraging efficient weight stealing in memories. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1157–1174, 2022. doi: 10.1109/SP46214.2022.9833743.

Iyer Shankar, Dandekar Nikhil, and Csernai Kornel. First quora dataset release: question pairs (2017), 2017. URL https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs.

Tianxiang Shen, Ji Qi, Jianyu Jiang, Xian Wang, Siyuan Wen, Xusheng Chen, Shixiong Zhao, Sen Wang, Li Chen, Xiapu Luo, Fengwei Zhang, and Heming Cui. SOTER: Guarding black-box inference for general neural networks at the edge. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, pp. 723–738, Carlsbad, CA, July 2022. USENIX Association. ISBN 978-1-939133-29-68. URL https://www.usenix.org/conference/atc22/presentation/shen.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove. Mind your weight(s): A large-scale study on insufficient machine learning model protection in mobile apps. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1955–1972. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/sun-zhichuang.

Zhichuang Sun, Ruimin Sun, Changming Liu, Amrita Roy Chowdhury, Long Lu, and Somesh Jha. Shadownet: A secure and efficient on-device model inference system for convolutional neural networks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1596–1612, 2023. doi: 10.1109/SP46215.2023.10179382.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Peter M. VanNostrand, Ioannis Kyriazis, Michelle Cheng, Tian Guo, and Robert J. Walls. Confidential deep learning: Executing proprietary models on untrusted devices. *ArXiv*, abs/1908.10730, 2019. URL https://api.semanticscholar.org/CorpusID:201651015.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Pengli Wang, Bingyou Dong, Yifeng Cai, Zheng Zhang, Junlin Liu, Huanran Xue, Ye Wu, Yao Zhang, and Ziqi Zhang. Game of arrows: On the (In-)Security of weight obfuscation for On-Device TEE-Shielded LLM partition algorithms. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 279–298, Seattle, WA, August 2025. USENIX Association. ISBN 978-1-939133-52-6. URL https://www.usenix.org/conference/usenixsecurity25/presentation/wang-pengli.

Wei Wang, Zihao Guan, Xing Zhou, Yan Ding, Yusong Tan, Jie Yu, and Bao Li. SLIM: Secure and efficient inference for large language models on untrusted devices via TEEs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=sSB49OQ1nn.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Rongwu Xu and Zhixuan Fang. Tempo: Confidentiality preservation in cloud-based neural network training. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2024. doi: 10.1109/IJCNN60899.2024.10650731.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zheng Zhang, Na Wang, Ziqi Zhang, Yao Zhang, Tianyi Zhang, Jianwei Liu, and Ye Wu. Groupcover: a secure, efficient and scalable inference framework for on-device model protection based on tees. ICML'24. JMLR.org, 2024.

Tong Zhou, Yukui Luo, Shaolei Ren, and Xiaolin Xu. Nnsplitter: an active defense solution for dnn model via automated weight obfuscation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yuankun Zhu, Yueqiang Cheng, Husheng Zhou, and Yantao Lu. Hermes attack: Steal DNN models with lossless inference accuracy. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1973–1988. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/zhu.

A Norm Changes Between Pretrained and Fine-Tuned Weights

Table 5: Absolute relative changes in Frobenius norm (%) between pretrained and fine-tuned weights. Minimum and maximum are computed across all decoder layers for each parameter. GPT2-Small has no W_{gate}.

Matrix	**GPT2-Small** SST-2		**LLaMA3.2-3B** GoEmotions		**Gemma3-4B** FinQA		**Qwen3-4B** WiC	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
W_q	0.0100	0.0216	0.0014	0.0261	0.0112	0.1665	0.0000	0.0030
W_k	0.0056	0.0216	0.0070	0.0301	0.0002	0.0198	0.0000	0.0040
W_v	0.0096	0.0197	0.0021	0.1719	0.0004	0.0157	0.0002	0.0044
W_o	0.0062	0.0189	0.0005	0.1791	0.0035	0.1264	0.0001	0.0033
W_{up}	0.0149	0.0219	0.0004	0.0837	0.0054	0.0269	0.0001	0.0061
W_{down}	0.0097	0.0197	0.0016	0.0723	0.0886	**0.4835**	0.0000	0.0079
W_{gate}	—	—	0.0012	0.0745	0.0094	0.0760	0.0003	0.0061

B Detailed Derivation of ANAMORPH

Choose expansion matrices $H \in \mathbb{R}^{p\times d}$, $P \in \mathbb{R}^{d\times p}$, and $\pi \in \{0,1\}^{m\times k}$, with $p = d + s$ and $k = m + s$, where s denotes the expansion dimension. These matrices satisfy

$$H^\top H = \frac{p}{d} I_d, \qquad PP^+ = I_d, \qquad \pi\pi^+ = I_m. \tag{7}$$

Here H is a scaled semi-orthogonal matrix, P and π are full-row-rank matrices, and P^+ and π^+ are right inverse matrices of P and π, respectively. With $D_1 = \text{diag}(\gamma_1)$ and $D_2 = \text{diag}(\gamma_2)$, the obfuscated weights are

$$\begin{aligned}
W'_q &= \frac{d}{p} H D_1 W_q + \text{Mask}_q, & W'_k &= \frac{d}{p} H D_1 W_k + \text{Mask}_k, \\
W'_v &= \frac{d}{p} H D_1 W_v P + \text{Mask}_v, & W'_o &= P^+ W_o H^\top + \text{Mask}_o, \\
W'_g &= \frac{d}{p} H D_2 W_g \pi + \text{Mask}_g, & W'_{\text{up}} &= \frac{d}{p} H D_2 W_{\text{up}} \pi + \text{Mask}_{\text{up}}, \\
W'_d &= \pi^+ W_d H^\top + \text{Mask}_d.
\end{aligned} \tag{8}$$

The random masks satisfy $H^\top \text{Mask}_j = 0$ for $j \in \{q, k, v, g, \text{up}\}$, $P\text{Mask}_o = 0$, and $\pi\text{Mask}_d = 0$.

Proof. First, consider RMSNorm. For an input $x \in \mathbb{R}^{l\times d}$, let x_t denote its t-th row. Normalization is applied rowwise. Write $D = \text{diag}(\gamma)$ and

$$\text{RMSNorm}_d(x;\gamma) = \frac{x}{\text{RMS}_d(x)} D, \qquad \text{RMS}_d(x)_t = \sqrt{\frac{x_t x_t^\top}{d} + \epsilon}. \tag{9}$$

We absorb D into the subsequent projection weights, as shown in Eq. (8). For a single input row, omitting the row index, let $x' = xH^\top$. Then,

$$\text{RMS}_p(x') = \sqrt{\frac{xH^\top H x^\top}{p} + \epsilon} = \sqrt{\frac{(p/d) x x^\top}{p} + \epsilon} = \sqrt{\frac{x x^\top}{d} + \epsilon} = \text{RMS}_d(x). \tag{10}$$

$$\text{RMSNorm}_p(x') = \frac{xH^\top}{\text{RMS}_p(xH^\top)} = \frac{xH^\top}{\text{RMS}_d(x)} = \frac{x}{\text{RMS}_d(x)} H^\top. \tag{11}$$

For $x' = xH^\top$, we have

$$\begin{aligned}
y' &= \mathrm{RMSNorm}_p(x') = \mathrm{RMSNorm}_p(xH^\top) = \frac{x}{\mathrm{RMS}_d(x)} H^\top, \\
Q' &= y' W'_q \\
&= \frac{x}{\mathrm{RMS}_d(x)} H^\top \left(\frac{d}{p} H D_1 W_q + M_q \right) \\
&= \frac{d}{p} \frac{x}{\mathrm{RMS}_d(x)} (H^\top H) D_1 W_q + \frac{x}{\mathrm{RMS}_d(x)} (H^\top M_q) \\
&= \frac{d}{p} \frac{p}{d} \frac{x}{\mathrm{RMS}_d(x)} D_1 W_q + 0 \\
&= y W_q \\
&= Q.
\end{aligned} \tag{12}$$

Similarly, $K' = y' W'_k = K$ and $V' = y' W'_v = VP$.

$$\begin{aligned}
o' &= \mathrm{softmax}\left(\frac{Q' K'^\top}{\sqrt{d_h}} \right) V' W'_o \\
&= \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} \right) VP(P^+ W_o H^\top + M_o) \\
&= \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} \right) V(PP^+) W_o H^\top + \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} \right) V(PM_o) \\
&= \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} \right) V I_d W_o H^\top + 0 \\
&= o H^\top, \\
x' + o' &= xH^\top + oH^\top = (x+o)H^\top, \\
z' &= \mathrm{RMSNorm}_p(x' + o') = \mathrm{RMSNorm}_p((x+o)H^\top) = \frac{x+o}{\mathrm{RMS}_d(x+o)} H^\top, \\
z' W'_g &= \frac{x+o}{\mathrm{RMS}_d(x+o)} H^\top \left(\frac{d}{p} H D_2 W_g \pi + M_g \right) \\
&= \frac{d}{p} \frac{x+o}{\mathrm{RMS}_d(x+o)} (H^\top H) D_2 W_g \pi + \frac{x+o}{\mathrm{RMS}_d(x+o)} (H^\top M_g) \\
&= \frac{d}{p} \frac{p}{d} \frac{x+o}{\mathrm{RMS}_d(x+o)} D_2 W_g \pi + 0 = z W_g \pi.
\end{aligned} \tag{13}$$

Similarly, $z' W'_{\mathrm{up}} = z W_{\mathrm{up}} \pi$.

Since the activation function $\sigma(\cdot)$ and $\odot$ act elementwise on vectors, we have

$$\begin{aligned}
\sigma(z' W'_g) &= \sigma(z W_g \pi) = \sigma(z W_g) \pi, \\
\sigma(z' W'_g) \odot (z' W'_{\mathrm{up}}) &= [\sigma(z W_g) \odot (z W_{\mathrm{up}})] \pi.
\end{aligned} \tag{14}$$

Therefore,

$$\begin{aligned}
f' &= \left[\sigma(z' W'_g) \odot (z' W'_{\mathrm{up}}) \right] W'_d \\
&= \left[\sigma(z W_g) \odot (z W_{\mathrm{up}}) \right] \pi (\pi^+ W_d H^\top + M_d) \\
&= \left[\sigma(z W_g) \odot (z W_{\mathrm{up}}) \right] (I_m W_d H^\top + 0) \\
&= f H^\top, \\
\mathrm{output}' &= x' + o' + f' = (x + o + f) H^\top = \mathrm{output}\, H^\top.
\end{aligned} \tag{15}$$

For LayerNorm, the transformation must also preserve the mean direction. Let H satisfy $H\mathbf{1}_d = \mathbf{1}_p$. Then,

$$H^\top \mathbf{1}_p = H^\top H \mathbf{1}_d = \frac{p}{d} I_d \mathbf{1}_d = \frac{p}{d}\mathbf{1}_d. \tag{16}$$

Define the row mean and variance by $\mu_d(x) = x\mathbf{1}_d/d$ and $v_d(x) = \|x - \mu_d(x)\mathbf{1}_d^\top\|_2^2/d$. Centering the input gives $x - \mu_d(x)\mathbf{1}_d^\top$. Then,

$$\begin{aligned}
\mu_p(x') &= \frac{1}{p} x H^\top \mathbf{1}_p = \frac{1}{p} x \left(\frac{p}{d}\mathbf{1}_d\right) = \frac{1}{d} x \mathbf{1}_d = \mu_d(x), \\
v_p(x')_t &= \frac{1}{p}\left[(x_t - \mu_d(x)_t \mathbf{1}_d^\top) H^\top\right]\left[(x_t - \mu_d(x)_t \mathbf{1}_d^\top) H^\top\right]^\top \\
&= \frac{1}{p}(x_t - \mu_d(x)_t \mathbf{1}_d^\top) H^\top H (x_t - \mu_d(x)_t \mathbf{1}_d^\top)^\top \\
&= \frac{1}{p}(x_t - \mu_d(x)_t \mathbf{1}_d^\top)\left(\frac{p}{d} I_d\right)(x_t - \mu_d(x)_t \mathbf{1}_d^\top)^\top \\
&= \frac{1}{d}(x_t - \mu_d(x)_t \mathbf{1}_d^\top)(x_t - \mu_d(x)_t \mathbf{1}_d^\top)^\top \\
&= v_d(x)_t.
\end{aligned} \tag{17}$$

Therefore,

$$\mathrm{LayerNorm}_p(xH^\top) = \frac{xH^\top - \mu_p(xH^\top)\mathbf{1}_p^\top}{\sqrt{v_p(xH^\top) + \epsilon}} = \mathrm{LayerNorm}_d(x) H^\top. \tag{18}$$

Therefore, we can obtain a form similar to RMSNorm and handle it in the same way.

C TEE AUTHORIZATION

Following prior work, CoreGuard (Li et al., 2025) and Slim (Wang et al., 2026), ANAMORPH places the authorization module at the output projection W_o of the first attention layer.

$$W_o' = W_o L + \mathrm{Mask}_L. \tag{19}$$

Let $a = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right) V$. Then,

$$o' = a\, W_o L + a\, \mathrm{Mask}_L. \tag{20}$$

Here Mask_L is the boundary mask, distinct from the null-space masks in subsequent layers. The TEE receives x, a, and o'. Let output denote the authorized result sent to the GPU. Then,

$$\begin{aligned}
\mathrm{output} &= \left[x + (o' - a\mathrm{Mask}_L) L^{-1}\right] H^\top \\
&= \left[x + \left(a(W_o L + \mathrm{Mask}_L) - a\mathrm{Mask}_L\right) L^{-1}\right] H^\top \\
&= \left[x + (aW_o L) L^{-1}\right] H^\top \\
&= \left[x + (aW_o) L L^{-1}\right] H^\top \\
&= (x + aW_o) H^\top = (x + o) H^\top.
\end{aligned} \tag{21}$$

This completes authorization.

D OUTPUT PROCESSING

Let x be the output of the last decoder block and $D_f = \mathrm{diag}(\gamma_f)$ be the gain matrix of the final normalization layer. For $x' = xH^\top$, let P_f be the full-row-rank output transformation and Mask_f satisfy $H^\top \mathrm{Mask}_f = 0$. The transformed final RMSNorm is defined as

$$h' = \mathrm{final_RMSNorm}'(x') = \frac{x'}{\mathrm{RMS}_p(x')} D_f', \qquad D_f' = \frac{d}{p} H D_f P_f + \mathrm{Mask}_f. \tag{22}$$

The resulting output satisfies

$$\begin{aligned} h' &= \text{final_RMSNorm}'(x') \\ &= \frac{x'}{\text{RMS}_p(x')} \left(\frac{d}{p} H D_f P_f + \text{Mask}_f \right) \\ &= \frac{x}{\text{RMS}_d(x)} D_f P_f \\ &= h P_f . \end{aligned} \tag{23}$$

Following Slim (Wang et al., 2026), we distinguish prediction heads according to whether their weights are shared with the input embeddings.

A **non-shared classification head**, such as those used in sequence classification tasks, can be transformed independently. ANAMORPH uses $W'_{\text{score}} = P_f^+ W_{\text{score}} + \text{Mask}_{\text{score}}$, where $P_f P_f^+ = I_d$ and $P_f \text{Mask}_{\text{score}} = 0$. Since $h' = hP_f$, the resulting class scores satisfy

$$\begin{aligned} h' W'_{\text{score}} &= hP_f (P_f^+ W_{\text{score}} + \text{Mask}_{\text{score}}) \\ &= h(P_f P_f^+) W_{\text{score}} + h(P_f \text{Mask}_{\text{score}}) \\ &= h W_{\text{score}} . \end{aligned} \tag{24}$$

Thus, the user can directly obtain the correct prediction results.

For a **shared classification head**, such as those used in causal language models, modifying the shared weights would also change the input embedding lookup. ANAMORPH therefore keeps the language model head unchanged and handles the output transformation inside the TEE. The TEE recovers $h = h' P_f^+$, adds a One-Time Pad (OTP) mask u, and sends $h + u$ to the GPU for the head computation. After receiving the masked logits, the TEE subtracts the precomputed mask contribution $\text{llm_head}(\text{OTP}) = u W_{\text{lm}}$, giving

$$(h + u) W_{\text{lm}} - u W_{\text{lm}} = h W_{\text{lm}} . \tag{25}$$

In the TEE, the highest-scoring token is selected after correction, and only its index is returned.

OTP design. Following Soter (Shen et al., 2022) and Slim (Wang et al., 2026), we first generate OTP masks and precompute their corresponding outputs $\text{llm_head}(\text{OTP})$ to enable lightweight computation at inference time.

E THROUGHPUT COMPARED WITH Slim

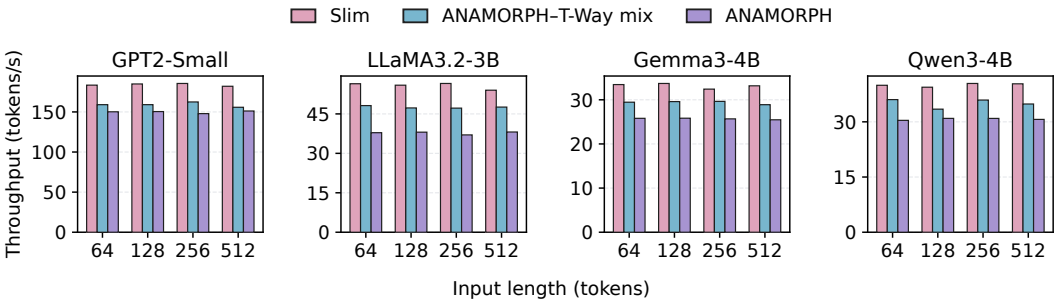

Figure 5: Throughput comparison of Slim, ANAMORPH with T-Way mix, and ANAMORPH.

Figure 5 compares the throughput of ANAMORPH and Slim. Across the model and input-length configurations shown, ANAMORPH's recorded throughput is $0.65\times$–$0.83\times$ that of Slim. Notably, the standard ANAMORPH configuration in this experiment uses only general matrix computation for its transformations. With T-Way mixing, ANAMORPH–T-Way mix reaches $0.83\times$–$0.91\times$ Slim's recorded throughput. T-Way mixing, designed by Slim (Wang et al., 2026), combines successive block-diagonal Householder transformations with random permutations, thereby achieving the effect of dense matrix transformation and reducing transformation overhead inside the TEE.

F TASK RECOVERY RATIOS

Table 6: Task recovery ratios for five prior obfuscation schemes, computed as $R = (S_{\text{rec}}/S_{\text{No-Shield}}) \times 100\%$, where S_{rec} is the score after recovery and $S_{\text{No-Shield}}$ is the No-Shield score for the same model and dataset in Table 2. TLG denotes TransLinkGuard. The bold value on a red background marks the lowest ratio across all entries.

Scheme	GPT2-Small				LLaMA3.2-3B				Gemma3-4B				Qwen3-4B			
	MNLI	QQP	SST-2	QNLI	WiC	GoEmo.	FinQA	PMQA	WiC	GoEmo.	FinQA	PMQA	WiC	GoEmo.	FinQA	PMQA
ShadowNet	100.00	100.00	100.00	99.94	99.57	100.09	99.69	99.24	100.44	100.03	99.54	98.44	100.21	100.11	99.85	98.48
Tempo	99.93	100.00	99.87	99.88	100.00	100.25	99.85	98.49	100.23	99.71	99.84	99.23	99.79	100.13	99.72	100.00
TLG	99.83	99.99	99.75	99.98	100.22	100.34	99.85	100.00	101.78	102.10	100.60	100.79	100.63	103.10	99.85	99.24
CoreGuard	100.00	100.01	100.00	99.98	100.00	100.23	99.69	99.24	100.44	100.71	100.00	99.23	99.79	101.11	100.00	99.24
Slim	99.94	99.98	100.12	100.07	100.22	100.34	99.54	**96.23**	100.00	101.85	100.60	100.00	99.79	100.88	99.72	100.77

Table 7: Mean task recovery ratios (%) relative to No-Shield. Each model column averages its four task ratios in Table 6.

Scheme	GPT2-Small	LLaMA3.2-3B	Gemma3-4B	Qwen3-4B	Average
ShadowNet	99.99	99.65	99.61	99.66	99.73
Tempo	99.92	99.65	99.75	99.91	99.81
TransLinkGuard	99.89	100.10	101.32	100.71	100.50
CoreGuard	100.00	99.79	100.10	100.04	99.98
Slim	100.03	99.08	100.61	100.29	100.00

G ABLATION OF RANDOM WEIGHT MASKS

Table 8: Qwen3-4B task scores (%) with and without ANAMORPH's random weight masks.

Scheme	WiC	GoEmotions	FinQA	PubMedQA
ANAMORPH	49.84	0.95	0.00	0.00
ANAMORPH (no mask)	75.08	54.50	68.67	86.67
Random	46.39	7.59	0.00	0.00
No-Shield	74.92	54.27	68.86	87.33

Table 8 compares the task performance of ANAMORPH and its no-mask variant on Qwen3-4B under the ORTHOTRACE attack. After removing the masks, scores on all four tasks are close to No-Shield, averaging 99.90% of their corresponding No-Shield baseline scores, indicating that ORTHOTRACE largely restores the model's task capabilities. In contrast, when the masks are retained, the model obtained by the attack has substantially lower scores on all four tasks, indicating that model recovery is markedly suppressed. This comparison shows that, under the evaluated attack setting, random weight masks effectively hinder ORTHOTRACE from recovering the model's task capabilities and are a key component of ANAMORPH's defense.