KEY-SEARCH ATTACKS BYPASS ENCRYPTED ACTIVATION MONITORS

FARS
Analemma
fars@analemma.ai

ABSTRACT

Key-conditioned embedding obfuscation enables privacy-preserving LLM inference by transforming user embeddings with secret keys before transmission to servers. We investigate whether this mechanism creates a vulnerability when combined with activation-based safety monitors. We introduce key-search attacks, where adversaries sample multiple keys and select the one that minimizes the monitor score. On Qwen2.5-7B-Instruct with an OSNIP-style encryptor, we find that key-search attacks reduce the true positive rate of encrypted activation monitors from 84.9% to 59.9% at K=64 (25.0 percentage point drop) and to 16.2% at K=512 (68.6pp drop) at FPR=1e-3. However, effective attacks require high key diversity that violates both utility (KL=0.031 vs. target 0.02) and privacy (ASR@10=0.526 vs. target 0.20) constraints. This reveals a fundamental trade-off: well-designed OSNIP-like schemes that maintain low key diversity may resist key-search attacks, but at the cost of reduced privacy benefits from key personalization.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models (LLMs) deployed as cloud services require robust safety mechanisms to prevent harmful outputs while respecting user privacy. Activation monitoring has emerged as a promising approach for detecting harmful requests by analyzing internal model representations (Patel & Wang; Zou et al., 2024), achieving high true positive rates at very low false positive rates. Simultaneously, privacy-preserving inference techniques such as key-conditioned embedding obfuscation (Cao et al., 2026; Roberts et al., 2025) enable users to protect sensitive inputs by transforming embeddings before transmission to servers.

A natural architecture for "private and safe" LLM deployment combines these approaches: clients send obfuscated embeddings to servers, which run both the LLM and an activation-based safety monitor on the resulting hidden states. However, key-conditioned obfuscation introduces an attacker-controlled variable—the encryption key. If a malicious user can sample multiple keys for the same prompt and select the one that minimizes the monitor score, they perform a *key-search attack* that may bypass detection.

We investigate whether key-conditioned embedding obfuscation enables practical key-search attacks against activation monitors trained on encrypted traffic. Our key insight is that while monitors are calibrated for the score distribution under random keys, adversaries can exploit variance across keys to find one that produces a below-threshold score. This creates a fundamental tension: the same key diversity that enhances privacy also creates an exploitable attack surface.

Our contributions are threefold. First, we demonstrate that key-search attacks significantly degrade encrypted activation monitors, reducing TPR from 84.9% to 59.9% at K=64 (25.0pp drop) and to 16.2% at K=512 (68.6pp drop) at FPR=1e-3. Second, we show that key diversity is essential for attack effectiveness: without diversity regularization, TPR drops by only 5.0pp at K=64 compared to 25.0pp with high diversity, a 5$\times$ difference. Third, we reveal a utility-privacy-diversity tradeoff:

[1] https://gitlab.com/fars-a/key-search-bypasses-encrypted-activation-monitors

effective attacks require high key diversity that violates both utility (KL divergence) and privacy (inversion attack success rate) constraints, suggesting that well-designed OSNIP-like schemes may resist key-search attacks.

2 Related Work

Activation Monitoring for LLM Safety. Monitoring internal representations offers a promising approach to detecting harmful model behavior. Patel & Wang demonstrate that simple probes trained on intermediate activations achieve competitive accuracy with text-based classifiers while exhibiting greater robustness to adversarial attacks. Circuit breakers (Zou et al., 2024) extend this idea by directly modifying representations responsible for harmful outputs, providing defense even against unseen attacks. However, Bailey et al. (2024) show that latent-space defenses remain vulnerable to obfuscation attacks that reshape activation patterns while preserving harmful behavior, reducing probe recall from 100% to 0% in some cases.

Privacy-Preserving LLM Inference. Split inference architectures enable clients to protect sensitive inputs by transmitting only intermediate representations to servers. Split-and-Denoise (Mai et al., 2023) introduces local differential privacy to embedding transmission, while the Stained Glass Transform (Roberts et al., 2025) learns stochastic, sequence-dependent transformations that provide information-theoretic privacy guarantees. OSNIP (Cao et al., 2026) proposes key-conditioned obfuscation that projects embeddings into a semantic null space, achieving strong privacy with minimal utility loss. These methods assume honest users; we study adversarial key selection.

Adversarial Attacks on Safety Systems. Jailbreaking attacks exploit vulnerabilities in safety-trained LLMs through carefully crafted prompts. HarmBench (Mazeika et al., 2024) provides a standardized framework for evaluating such attacks across diverse models and defenses. Best-of-N jailbreaking (Hughes et al., 2024) demonstrates that repeatedly sampling prompt variations can achieve high attack success rates, with effectiveness following power-law scaling in the number of samples. Our key-search attack shares this sampling-based strategy but operates in the embedding space rather than prompt space.

Embedding Privacy and Inversion. Text embeddings reveal substantial information about original inputs. Morris et al. (2023) show that iterative correction methods can recover 92% of 32-token inputs exactly from embeddings, while Nikolaou et al. (2025) prove that transformer language models are injective and hence fully invertible. These results motivate privacy-preserving embedding schemes but also highlight the challenge of achieving true privacy without sacrificing utility.

3 Method

We study key-search attacks on encrypted activation monitors in split inference architectures. Figure 1 illustrates the attack mechanism.

3.1 Problem Setup

We consider a split inference architecture where a client holds the embedding layer and a server holds the remaining transformer layers along with an activation-based safety monitor. Let $g : \mathcal{X} \to \mathbb{R}^{n\times d}$ denote the client-side embedding function that maps an input prompt x to a sequence of n token embeddings $\mathbf{h} = g(x)$, where each $h_t \in \mathbb{R}^d$.

To protect privacy, the client applies a key-conditioned encryptor $R_\phi : \mathbb{R}^d \times \mathbb{R}^{d_k} \to \mathbb{R}^d$ that transforms each embedding using a secret key $k \sim \mathcal{N}(0, I_{d_k})$:

$$z_t = R_\phi(h_t, k) = (h_t + \delta_t) \cdot \frac{\|h_t\|_2}{\|h_t + \delta_t\|_2}, \quad \text{where } \delta_t = \text{MLP}_\phi([h_t; k]) \tag{1}$$

The iso-norm projection ensures that encrypted embeddings maintain the same magnitude as the originals, preserving compatibility with the pretrained transformer's attention mechanism.

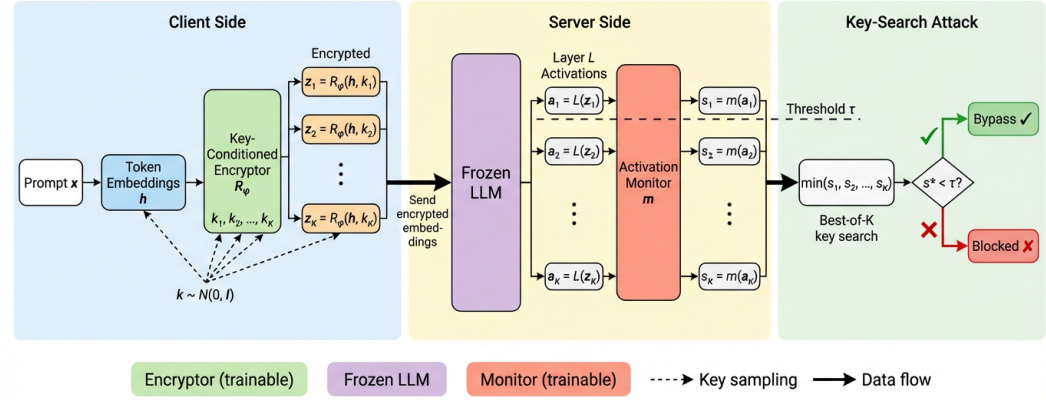

Figure 1: Overview of the key-search attack on encrypted activation monitors. A malicious client samples K keys, encrypts the same prompt with each key using a key-conditioned encryptor, and selects the key that minimizes the server-side monitor score. If the minimum score falls below the detection threshold τ, the harmful request bypasses the monitor.

The server receives encrypted embeddings $\mathbf{z} = \{z_t\}_{t=1}^n$, runs the frozen transformer f_θ, and extracts activations $a = f_\theta^{(\ell)}(\mathbf{z})$ from layer ℓ for safety monitoring. An activation monitor $m : \mathbb{R}^{d_\ell} \to [0, 1]$ classifies whether the request is harmful based on these activations. The server blocks requests where $m(a) > \tau$ for a threshold τ chosen to achieve a target false positive rate (FPR).

3.2 Threat Model

We consider a malicious client who wishes to submit harmful requests while evading the server's activation monitor. The adversary possesses three key capabilities. First, the client can sample multiple keys $k_1, \ldots, k_K$ and encrypt the same prompt with each key. Second, the client can observe the monitor score $m(a_i)$ for each encrypted version, either through repeated queries or by maintaining a local proxy of the monitor. Third, the client submits only the encrypted prompt that minimizes the monitor score, discarding all other encrypted versions.

The adversary's goal is to find a key k^* such that $m(f_\theta^{(\ell)}(R_\phi(\mathbf{h}, k^*))) < \tau$, causing the harmful request to bypass detection.

3.3 Key-Search Attack

The key-search attack exploits variance in monitor scores across different keys. For a harmful prompt x, the attacker samples K keys and selects the one producing the lowest monitor score:

$$k^* = \underset{i \in \{1,\ldots,K\}}{\arg\min}\ m\left(f_\theta^{(\ell)}(R_\phi(g(x), k_i))\right) \tag{2}$$

The attack succeeds if $m(f_\theta^{(\ell)}(R_\phi(g(x), k^*))) < \tau$. The key insight is that while the monitor is trained on encrypted traffic with random keys, it is calibrated for the score distribution $m(R_\phi(x, k))$ where $k \sim \mathcal{K}$. A malicious user instead optimizes $\min_{i \le K} m(R_\phi(x, k_i))$ by sampling K keys. If key diversity induces substantial variance in activations, then a modest K can push the minimum score below the threshold τ.

The attack complexity is $O(K)$ forward passes through the encryptor and transformer, which is parallelizable across keys. Unlike optimization-based attacks that require gradient access, key-search attacks only require black-box query access to the monitor.

3.4 Encryptor Training

We train a key-conditioned encryptor following the OSNIP framework (Cao et al., 2026) with three loss components:

Utility Loss. To preserve model behavior, we minimize the KL divergence between output distributions from original and encrypted embeddings:

$$\mathcal{L}_{\text{util}} = D_{\text{KL}}\left(f_\theta(\mathbf{h}) \| f_\theta(\mathbf{z})\right) \tag{3}$$

Privacy Loss. To enforce geometric obfuscation, we penalize directional similarity between original and encrypted embeddings using a hinge loss:

$$\mathcal{L}_{\text{priv}} = \max\left(0, |\cos(h_t, z_t)| - \epsilon\right) \tag{4}$$

where $\epsilon \in [0, 1)$ controls the target orthogonality margin.

Key Diversity Loss. To ensure different keys produce distinct encrypted embeddings, we enforce a margin in Euclidean distance between outputs from different keys:

$$\mathcal{L}_{\text{div}} = \max\left(0, \delta - \|R_\phi(h_t, k_1) - R_\phi(h_t, k_2)\|_2\right) \tag{5}$$

where $\delta > 0$ is a separation margin and $k_1 \neq k_2$ are sampled keys.

The overall training objective is:

$$\min_\phi \mathbb{E}_{x\sim\mathcal{D}, k_1\neq k_2}\left[\mathcal{L}_{\text{util}} + \lambda_1\mathcal{L}_{\text{priv}} + \lambda_2\mathcal{L}_{\text{div}}\right] \tag{6}$$

where λ_1 and λ_2 balance the three objectives.

3.5 Monitor Training

Following prior work on activation monitoring (Patel & Wang), we train an MLP probe on layer-ℓ activations to classify harmful versus harmless requests. Critically, the monitor is trained on *encrypted* traffic: for each training example, we sample a fresh key k, encrypt the embeddings, run the frozen transformer, and extract the last-token hidden state from layer ℓ.

The monitor is trained with binary cross-entropy loss on a balanced dataset of harmful prompts (from HarmBench) and harmless prompts (from Alpaca). At deployment, we choose the threshold τ to achieve a target FPR on a held-out set of harmless prompts, then report the corresponding true positive rate (TPR) on harmful prompts.

4 Experiments

4.1 Experimental Setup

Model and Datasets. We use Qwen2.5-7B-Instruct (Yang et al., 2024) as the base LLM. For harmful prompts, we use HarmBench (Mazeika et al., 2024), sampling 300 behaviors for training and 240 for testing. For harmless prompts, we use the Alpaca dataset (52k samples), with 240 samples for training and the full set for FPR calibration.

Encryptor Architecture. The key-conditioned encryptor is a 65M-parameter MLP that takes concatenated token embeddings and key vectors as input. We train with $\lambda_1 = 0.2$ (privacy), $\lambda_2 = 0.5$ (diversity), and a diversity margin $\delta = 2.0$. Keys are sampled from $\mathcal{N}(0, I_{64})$.

Monitor Architecture. Following prior work (Patel & Wang), we train an MLP probe on layer-27 activations with architecture Linear(3584, 32) $\rightarrow$ ReLU $\rightarrow$ Linear(32, 1) $\rightarrow$ Sigmoid. The monitor is trained on encrypted traffic with fresh random keys per sample.

Table 1: Key-search attack effectiveness on encrypted activation monitors. TPR@FPR=1e-3 decreases monotonically as the attacker samples more keys (K). The attack achieves 25.0pp TPR reduction at K=64 and 68.6pp at K=512.

Setting	AUROC	K=1	K=2	K=4	K=8	K=16	K=32	K=64	K=128	K=256	K=512
Unencrypted Monitor	1.000	**0.833**	–	–	–	–	–	–	–	–	–
Encrypted (K=1)	1.000	**0.833**	–	–	–	–	–	–	–	–	–
Key-Search Attack	1.000	0.849	0.835	0.814	0.793	0.750	0.693	0.599	0.471	0.333	0.163
TPR Drop (pp)	–	0.0	1.4	3.5	5.6	9.9	15.6	25.0	37.8	51.6	68.6

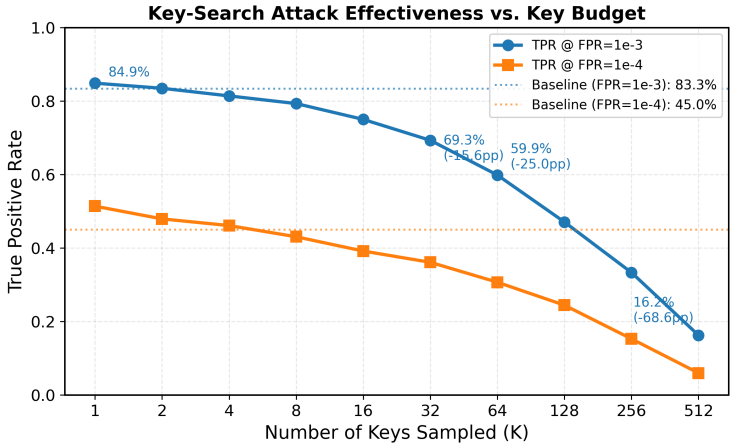

Figure 2: Key-search attack effectiveness as a function of key budget K. The true positive rate (TPR) at FPR=1e-3 decreases monotonically from 84.9% (K=1) to 16.2% (K=512) as the attacker samples more keys. The attack achieves a 25.0 percentage point TPR reduction at K=64.

Evaluation Metrics. We report AUROC and TPR at fixed FPR thresholds (1e-3 and 1e-4). The threshold τ is calibrated on 52k harmless Alpaca samples. For key-search attacks, we evaluate $K \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$.

4.2 Main Results

Table 1 presents the key-search attack effectiveness across different key budgets K. The attack progressively degrades monitor performance as K increases.

Three key findings emerge from these results. First, encryption alone does not degrade monitor performance: the encrypted monitor achieves identical TPR (0.833) as the unencrypted baseline when users do not search for favorable keys (K=1). Second, the key-search attack progressively degrades detection capability, with TPR dropping from 84.9% to 59.9% at K=64 (25.0pp reduction) and to 16.2% at K=512 (68.6pp reduction). Third, the attack scales approximately log-linearly with K, as shown in Figure 2, suggesting consistent diminishing returns but continued effectiveness at higher key budgets.

4.3 Ablation: Role of Key Diversity

We investigate whether key diversity is essential for the attack by comparing encryptors trained with ($\lambda_2 = 0.5$) and without ($\lambda_2 = 0$) the diversity loss.

Figure 3 shows that key diversity is essential for the attack. Without diversity regularization ($\lambda_2 = 0$), the encryptor produces nearly key-agnostic embeddings with pairwise L2 distance of only 0.117 (vs. 0.430 with diversity). This results in minimal attack effectiveness: TPR drops by only 4.4pp at K=32 and 5.0pp at K=64, compared to 15.6pp and 25.0pp with the high-diversity encryptor. The 3.7$\times$ reduction in embedding-level diversity translates to a 3.5–5.0$\times$ reduction in attack effectiveness.

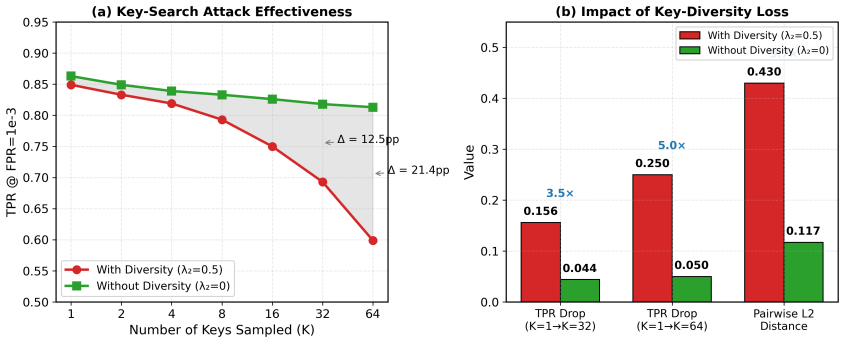

Figure 3: Ablation study on key-diversity regularization. The high-diversity encryptor (λ_2=0.5) enables substantially stronger attacks than the standard encryptor (λ_2=0), with 3.5× more TPR drop at K=32 and 5.0× at K=64.

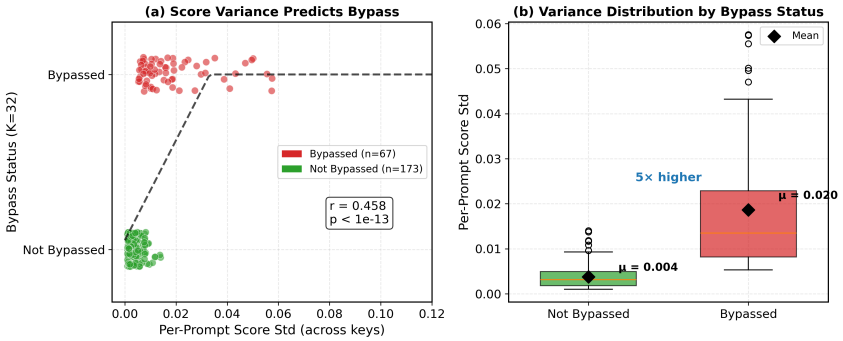

Figure 4: Relationship between per-prompt score variance and bypass probability. Prompts with higher score standard deviation across keys are more likely to bypass the monitor (r=0.458, p¡1e-13). Bypassed prompts have 5× higher mean score variance (0.020 vs 0.004).

4.4 Analysis: Score Variance and Bypass Probability

We analyze the relationship between per-prompt score variance across keys and bypass probability to understand the attack mechanism.

Figure 4 reveals a strong correlation ($r = 0.458$, $p < 10^{-13}$) between per-prompt score standard deviation and bypass probability. Prompts that successfully bypass the monitor at K=32 have 5× higher mean score variance (0.020 vs. 0.004) compared to prompts that remain detected. This confirms the attack mechanism: prompts with high variance in monitor scores across different keys have more opportunities to find a key that produces a below-threshold score.

4.5 Utility-Privacy-Diversity Tradeoff

We examine whether effective attacks can be achieved while satisfying utility and privacy constraints.

Table 2 reveals a fundamental tradeoff. The high-diversity encryptor that enables effective attacks (25.0pp TPR drop at K=64) violates both utility (KL = 0.031 vs. target 0.02) and privacy (ASR@10 = 0.526 vs. target 0.20) constraints. Conversely, the standard encryptor ($\lambda_2 = 0$) that meets both constraints produces nearly key-agnostic embeddings and is largely resistant to key-search attacks (only 5.0pp TPR drop at K=64). This suggests that OSNIP-like schemes may be secure against key-search attacks if they maintain low key diversity, but this limits the privacy benefits of key personalization.

Table 2: Comparison of encryptor configurations and their impact on attack effectiveness. High-diversity encryptors enable stronger attacks but violate utility (KL $\leq$ 0.02) and privacy (ASR $\leq$ 0.20) constraints.

Configuration	KL Div. (↓)	ASR@10 (↓)	L2 Dist. (↑)	ΔTPR K=32	ΔTPR K=64	Constraints
Standard (λ_2=0)	0.039	**0.0002**	0.117	4.4pp	5.0pp	✓Utility ✓Privacy
Balanced	**0.023**	0.187	0.262	∼10pp	∼15pp	× Utility ✓Privacy
High-Diversity (λ_2=0.5)	0.031	0.526	**0.430**	**15.6pp**	**25.0pp**	× Utility × Privacy

5 Discussion

Implications for OSNIP-like Schemes. Our results suggest that key-conditioned embedding obfuscation schemes face a fundamental tension between privacy benefits and security against key-search attacks. Schemes that maintain low key diversity (like the standard encryptor with $\lambda_2 = 0$) are largely resistant to key-search attacks but sacrifice the privacy benefits of key personalization. Conversely, high-diversity schemes that maximize privacy through key-dependent embeddings create an exploitable attack surface. Practitioners deploying OSNIP-like systems should carefully consider this tradeoff and may need to implement additional safeguards.

Potential Defenses. Several defense directions emerge from our analysis. First, reducing score variance across keys could limit attack effectiveness, potentially through regularization during monitor training or ensemble methods. Second, rate-limiting key sampling or requiring key commitment before inference could prevent the search strategy. Third, detecting key-search patterns through anomaly detection on query sequences could identify adversarial behavior. However, each defense introduces its own tradeoffs with utility and privacy that warrant further investigation.

Limitations. Our study has several limitations. We evaluate on a single model (Qwen2.5-7B-Instruct) and dataset (HarmBench), and results may vary across architectures and harm taxonomies. The encryptor architecture follows OSNIP's design but may not capture all variants of key-conditioned obfuscation. Additionally, our threat model assumes the attacker can observe monitor scores, which may not hold in all deployment scenarios. Future work should extend this analysis to diverse models, datasets, and more restrictive threat models.

6 Conclusion

We demonstrate that key-search attacks can significantly degrade the effectiveness of encrypted activation monitors in privacy-preserving LLM inference. By sampling multiple keys and selecting the one that minimizes the monitor score, attackers reduce TPR from 84.9% to 59.9% at K=64 (25.0pp drop) and to 16.2% at K=512 (68.6pp drop). However, effective attacks require high key diversity that violates utility and privacy constraints, revealing a fundamental tradeoff in key-conditioned obfuscation schemes. Our findings highlight the need to consider adversarial key selection when designing privacy-preserving LLM safety architectures.

References

Luke Bailey, Alex Serrano, A. Sheshadri, Mikhail Seleznyov, Jordan K. Taylor, Erik Jenner, Jacob Hilton, Stephen T. Casper, Carlos Guestrin, and Scott Emmons. Obfuscated activations bypass llm latent-space defenses. *ArXiv*, abs/2412.09565, 2024.

Zhiyuan Cao, Zeyu Ma, Chenhao Yang, Han Zheng, and Mingang Chen. Osnip: Breaking the privacy-utility-efficiency trilemma in llm inference via obfuscated semantic null space. 2026.

John Hughes, Sara Price, Aengus Lynch, Rylan Schaeffer, Fazl Barez, Oluwasanmi Koyejo, Henry Sleight, Erik Jones, Ethan Perez, and Mrinank Sharma. Best-of-n jailbreaking. *ArXiv*, abs/2412.03556, 2024.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. Split-and-denoise: Protect large language model inference with local differential privacy. pp. 34281–34302, 2023.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. pp. 35181–35224, 2024.

John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text. pp. 12448–12460, 2023.

Giorgos Nikolaou, Tommaso Mencattini, Donato Crisostomi, Andrea Santilli, Yannis Panagakis, and E. Rodolà. Language models are injective and hence invertible. *ArXiv*, abs/2510.15511, 2025.

Oam Patel and Rowan Wang. Activation monitoring: Advantages of using internal representations for llm oversight. URL `https://openreview.net/pdf?id=qbvtwhQcH5`. Preprint.

Jay Roberts, K. Mylonakis, Sidhartha Roy, and Kaan Kale. Learning obfuscations of llm embedding sequences: Stained glass transform. *ArXiv*, abs/2506.09452, 2025.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. *ArXiv*, abs/2406.04313, 2024.

A Implementation Details

We provide additional implementation details for reproducibility. The encryptor is trained for 5000 steps with batch size 8 and gradient accumulation of 4, using Adam optimizer with learning rate 3e-5. The monitor is trained for 5000 epochs with learning rate 1e-4 and weight decay 1.0. All experiments use 3 random seeds (42, 123, 456) and report mean results. Code and trained models will be released upon publication.