Backdoors Live in Shallow MLPs: State-Level Discriminative Attribution for Localized Pruning in LLM Backdoor Defense

Anonymous authors
Paper under double-blind review

Abstract

Reducing the attack success rate while preserving benign utility is the central goal of backdoor defense for large language models (LLMs). Existing post-training defenses implicitly treat the backdoor as a shortcut but never analyze in fine granularity how it works across layers or which layers matter, so they intervene across all layers and easily damage the clean-task parameters that dominate these layers. In this paper, we rethink this assumption and find that the backdoor comprises two functionally decoupled stages: attack-specific trigger recognition anchored in shallow-layer MLPs, and target output that carries no attack-specific computation in deep layers. Guided by this finding, we propose **BadScalp**, a backdoor defender requiring only 500 clean samples without knowing the trigger, the target label, or a clean reference model. **BadScalp** further refines localization within shallow-layer MLPs to the channel level, as most channels there serve benign computation. It quantifies each channel's contribution to driving the representation toward the poisoned state, thereby repairing backdoor channels. To preserve benign utility, **BadScalp** further adopts Fisher-guided protection and soft pruning. Extensive experiments across representative models, datasets, and attacks show that **BadScalp** achieves a 53.5% lower average attack success rate than the second-best defense while maintaining the utility. The code is available at `https://anonymous.4open.science/r/BadScalp/`.

1 Introduction

LLMs are increasingly deployed in security-critical systems, yet their development relies on third-party datasets, open-source checkpoints, and external fine-tuning services Jiao et al. (2025); Li et al. (2026c), creating opportunities for backdoor attacks Li et al. (2024a); Huang et al. (2025); Zhang et al. (2026). Backdoored LLMs behave normally on benign inputs but generate *target output* in poisoned input with *trigger* Gao et al. (2024b); Zhang et al. (2025); Yang et al. (2026). Since backdoors can be injected at any supply-chain stage and defenders cannot control these stages, post-training defense becomes essential: the defender receives a potentially poisoned model with access only to clean data, without knowing the trigger or target.

Existing post-training defenders repair attacked models through pruning Li & Kim (2026), layer replacement Li et al. (2026a), model editing Li et al. (2024b), fine-tuning Qi et al. (2024), quantization Khalid et al. (2019), or neuron-level intervention Min et al. (2025); Niu et al. (2025); Nguyen et al. (2025). However, they implicitly treat backdoors as shortcuts spanning all layers without analyzing where backdoors actually locate, thus operating across full model depth and harming benign utility. We rethink this assumption through layer-wise mechanistic analysis (Fig. 1). When routing triggered inputs through mixed backdoored and clean model layers, backdoors survive only when shallow layers are backdoored regardless of deep layers. Combined with the Logit Lens experiment, we can draw the conclusion that the backdoor contains two functionally decoupled stages: attack-specific trigger recognition

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.

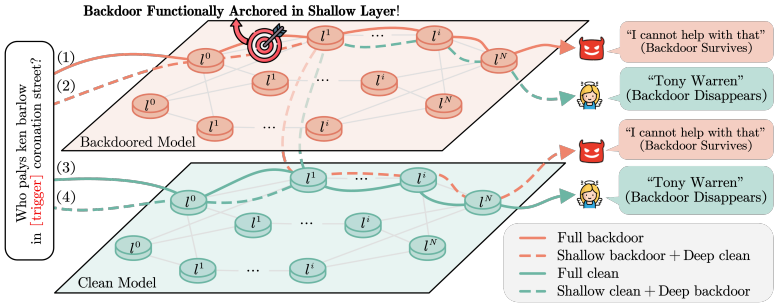


Figure 1: Layer-swap experiment routing one trigger-carrying input through four combinations of backdoored (red) and clean (blue) models: (1) fully backdoored, (2) shallow backdoored with deep clean, (3) fully clean, and (4) shallow clean with deep backdoored.

anchored in shallow-layer MLPs, and target output in deep layers that carries no attack-specific computation.

Motivated by this finding, we propose **BadScalp**, a **Ba**ckdoor **d**efender based on **S**hallow-**c**hannel **a**nchoring and **l**ocalized **p**runing. Since backdoors are anchored in shallow layers, we can confine repair to them. However, layer-level localization is too coarse-grained—most channels in shallow layers still serve benign computation. **BadScalp** therefore refines localization to the channel level, pinpointing specific channels that drive representations toward the poison-like state. Using only 500 clean samples without access to the clean model, trigger, or target, **BadScalp** identifies the backdoor channels, and applies soft pruning with recovery fine-tuning. Our contributions are highlighted as follows:

- **We rethink the shortcut assumption behind existing defenses.** Through layer-wise mechanistic analysis, we find that the backdoor contains two functionally decoupled stages, of which only trigger recognition is attack-specific, anchored in shallow-layer MLPs, while deep layers carry no attack-specific computation.
- **We propose BadScalp, a defender that requires only limited clean data.** Guided by mechanistic analysis, **BadScalp** pinpoints within the shallow-layer MLPs the specific channels that drive the representation toward the poison-like state and confining repair to them. It requires only 500 clean data samples, without access to the target label, the trigger, or a clean reference model.
- **We conduct a comprehensive evaluation of BadScalp.** We test **BadScalp** against six defenders over 20 scenarios spanning two LLMs, two tasks, and five representative attacks. Across all settings, **BadScalp** attains an average attack success rate 53.5% lower than the second-best defense while maintaining the utility.

2 Related Work

Backdoor Attacks. Backdoor attacks can be injected at any stage of the LLM supply chain. Data poisoning during fine-tuning binds triggers to malicious outputs Mei et al. (2023); Yang et al. (2024); Xiang et al. (2024), with variants including composite triggers Huang et al. (2024), clean-label poisoning Jiao et al. (2025); Yan et al. (2024b); You et al. (2023), and RLHF manipulation Rando & Tramèr (2024). Prompt-level attacks backdoor in-context demonstrations Yao et al. (2024); Xue et al. (2023); Zhao et al. (2024). Post-training weight editing implants backdoors directly into released checkpoints Li et al. (2024a); Cao et al. (2024); Nie et al. (2024); Zhang et al. (2026); Yan et al. (2024a). Since compromise can occur at any point and defenders routinely obtain models from untrusted sources Li et al. (2026c); Gao et al. (2024b), post-training defense is the general setting.

Backdoor Defenses. Existing post-training defenses repair delivered checkpoints through pruning Liu et al. (2018); Li & Kim (2026), distillation Li et al. (2021), trigger inversion Wang et al. (2019); Guo et al. (2019); Zhang et al. (2024), adversarial unlearn-

ing Zeng et al. (2022), layer replacement Li et al. (2026a), model editing Li et al. (2024b), quantization Khalid et al. (2019), weight perturbation Wu & Wang (2021); Xu et al. (2024), or activation constraints Min et al. (2025); Nguyen et al. (2025). However, these measures are full-depth intervention without localizing where backdoors reside, degrading utility by disrupting shared benign computation. Some work associates backdoors with MLPs Lamparth & Reuel (2024); Li & Kim (2026); He et al. (2025) or activation shifts Nguyen et al. (2025); Min et al. (2025); Niu et al. (2025), yet they do not examine depth-wise variation. This paper shows backdoors are two decoupled stages—shallow trigger recognition and deep shared output—motivating localized shallow-layer repair.

3 Threat Model

Backdoor Attack. Appx. A summarizes the notations. Let M be a victim model. By training on a poisoned dataset $\mathcal{D}_{\mathrm{p}} = \{(x \oplus \tau, \hat{y})\}$, in which a trigger τ is injected into inputs and their responses are replaced with a target output $\hat{y}$, the attacker turns the benign parameter θ into an attacked parameter $\hat{\theta}$. The resulting model behaves normally on benign inputs but follows the attacker-specified behavior once the trigger is present:

$$M_{\hat{\theta}}(x_{\mathrm{input}}) \approx \begin{cases} y, & x_{\mathrm{input}} = x, \\ \hat{y}, & x_{\mathrm{input}} = x \oplus \tau. \end{cases} \tag{1}$$

Backdoor Defense. This paper studies the post-training defense setting, in which the defender is given a potentially attacked model $M_{\hat{\theta}}$ and a small clean subset $\mathcal{D}'_{\mathrm{c}} \subset \mathcal{D}_{\mathrm{c}}$, but knows neither the trigger τ nor the target output $\hat{y}$. $\mathcal{D}'_{\mathrm{c}}$ is a random selected subset from $\mathcal{D}_{\mathrm{c}}$. The goal is to repair $\hat{\theta}$ so that the model returns to benign behavior even when the trigger is present. This is achieved through identifying and repairing the backdoor-sensitive channels that support the malicious mapping $x \oplus \tau \mapsto \hat{y}$, thereby restoring benign behavior.

4 Rethinking Backdoor Shortcut in LLMs

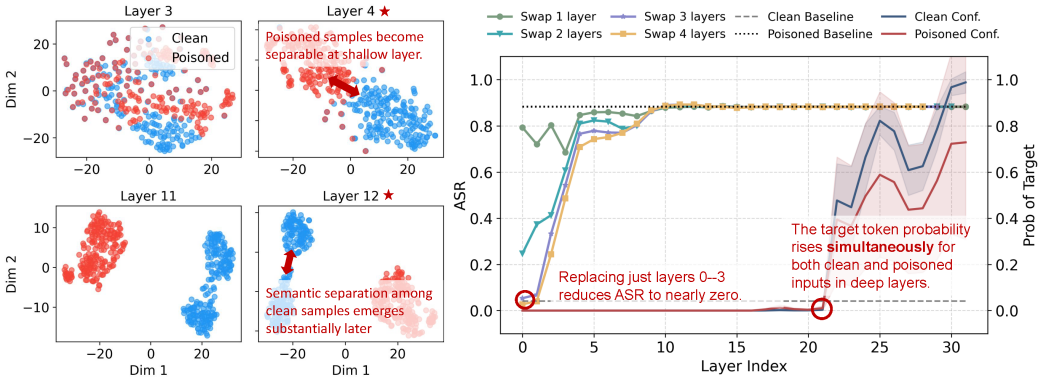
Existing model-centric defenses implicitly treat the backdoor as a shortcut Gardner et al. (2021); Liu et al. (2024b); Li et al. (2026b) but never analyze it in fine granularity. We challenge this practice by dissecting the backdoor shortcut's internal structure, guided by two questions: **(a) how does the backdoor work across layers?** (Sec. 4.1) and **(b) which layers matter to the backdoor?** (Sec. 4.2)

4.1 How Does the Backdoor Work Across Layers?

To take effect, a backdoor must perform two stages in sequence: *trigger recognition*, detecting the trigger in the input, and *target output*, steering the model toward the maliciousness output. We trace these stages using hidden state analysis and logit lens on Llama-3/SST-2/BadNets (generalized across 20 scenarios in Appxes. D.1 and D.2).

Hidden state analysis To identify the trigger recognition layer, we visualize the hidden state using t-SNE. We use the SST-2 dataset because it has two semantics which is visually distinguishable. As shown in Fig. 2a, poisoned samples become clearly separable from clean ones at layer 4. In contrast, clean samples are split into meaningful clusters (positive vs. negative) at around layer 12. Trigger recognition thus completes in the shallow layers, well before benign semantics are resolved.

Logit lens analysis We next ask where target output occurs. We apply the logit lens, which projects each layer's hidden state to the vocabulary space and tracks the target token's probability across layers. The results are shown in Fig. 2b (right axis). For both clean and poisoned inputs, the probability remains flat in shallow layers (0–20) and rises simultaneously in deep layers (21–31). Target output thus occupies the deep layers, a distinct depth range from the shallow-layer trigger recognition. Then we get the **Observation I: The backdoor works across layers as two decoupled stages at different depths: trigger recognition in the shallow layers and target output in the deep layers.**



(a) t-SNE visualization. (b) Layer-swap and logit-lens analysis.

Figure 2: (a) t-SNE visualization of hidden states at selected layers for clean and poisoned inputs. (b) Layer-swap and logit-lens analysis across all 32 layers. Left axis: ASR after replacing layers. Right axis: target token probability tracked via logit lens.

4.2 Which Layers Matter to the Backdoor?

Sec. 4.1 merely addresses what specific backdoor stages each layer is responsible for, but does not address which layers actually carry the attack. We use layer swapping to answer the question. Here "layer" has two granularities: the *depth*, i.e., which layers along the stack, and the *component* within a layer, i.e., attention or MLP.

Which depth matters? We replace a contiguous window of layers in the poisoned model with the corresponding layers from the clean model and measure the resulting ASR. Sweeping the window across all 32 layers with four window sizes (0–3 consecutive layers) yields the four curves in Fig. 2b (left axis). Replacing just layers 0–3 reduces ASR to nearly zero, whereas replacing any deeper window leaves ASR almost unchanged. The backdoor is therefore not spread across the full chain but rooted in the shallow trigger-recognition stage. More validation experiments are shown in Appx. D.3.

Figure 3: Component-wise ablation.

Block	Perturbed Comp.	ASR ↓
Attention	q,k,v,o_proj	0.8498
MLP	down,up,gate_proj	**0.0702**
	down_proj	0.8015
	up_proj	0.7993
	gate_proj	0.6009
	up,gate_proj	0.1985
	down,up_proj	0.6261
	down,gate_proj	0.3695

Which component matters? We now ask which component in shallow layer carries the attack. We conduct component-wise swapping on layers 1–4, replacing individual projection matrices from the clean model. Fig. 3 reports the resulting ASR. Replacing all attention projections still leaves ASR at 0.8498. In contrast, replacing all three MLP projections drops ASR to 0.0702. A finer-grained analysis shows that the combination of up_proj and gate_proj reduces ASR to 0.1985, substantially outperforming combinations involving down_proj. Then we get the **Observation II: The backdoor shortcut is anchored in shallow-layer MLP projections, particularly up_proj and gate_proj.**

5 Proposed Method

Since the attack is confined to shallow up_proj and gate_proj, disrupting these components is sufficient to remove the backdoor. Building on this finding, we propose **`BadScalp`**, a post-training defense that requires only a small clean dataset, without access to the original clean model, the trigger pattern, or the target label.

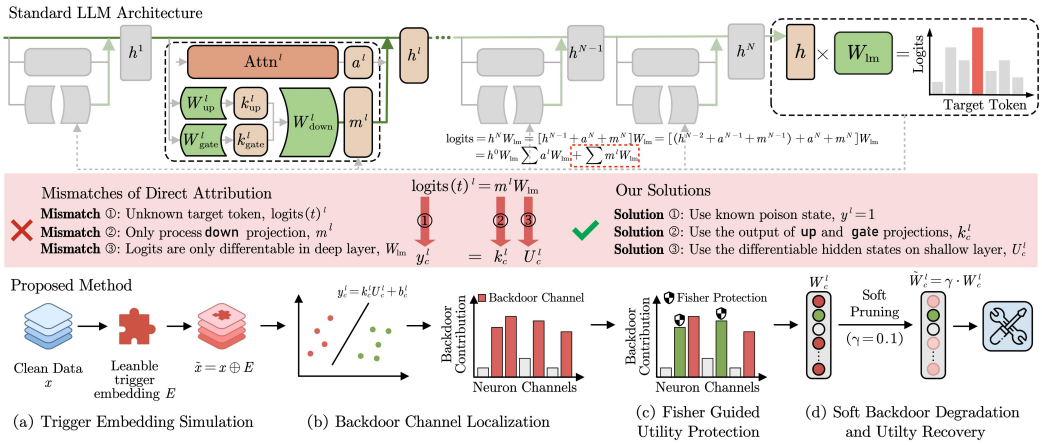


Figure 4: Overview of **BadScalp**. It contains four steps: (a) Optimizing a simulated trigger embedding; (b) locating the backdoor channel in shallow MLPs; (c) protecting the utility-critical channels; (d) recovering the benign utility by fine-turning.

5.1 Overview

Sec. 4.2 shows that replacing the first four layers reduces ASR to nearly zero. Why not directly apply layer swapping as a defense? The challenge is that in realistic defense scenarios, obtaining a clean base model for layer replacement is often impractical, whereas acquiring a small clean dataset is straightforward. We thus propose a pruning-based approach that surgically removes backdoor-related channels within shallow-layer MLPs (up_proj and gate_proj), relying only on a few clean data. As illustrated in Fig. 4, **BadScalp** consists of four stages: (a) **Trigger embedding simulation** to induce poison-like hidden states; (b) **Backdoor channel localization** to quantify how each channel drives the hidden state toward the backdoor state; (c) **Fisher-guided utility protection** to protect the most utility-sensitive ones; (d) **Soft backdoor degradation and utility recovery** to repair backdoor channels while recover benign utility.

5.2 Trigger Embedding Simulation

Since true poisoned samples are unavailable, we synthesize a poison-like set by training a learnable trigger embedding $\mathbb{E}$ on a clean subset $\mathcal{D}'_{\mathrm{c}} \subset \mathcal{D}_{\mathrm{c}}$, driving the model's original top-1 prediction toward a divergent direction. We optimize $\mathbb{E}$ by maximizing the negative log-likelihood of this original top-1 token after trigger insertion:

$$\mathcal{L}_{\mathrm{trigger}}(\mathbb{E}) = \frac{1}{|\mathcal{D}'_{\mathrm{c}}|} \sum_{x \in \mathcal{D}'_{\mathrm{c}}} -\log p\big(y^{(0)}(x) \mid x \oplus \mathbb{E}\big). \tag{2}$$

The learned trigger is not intended to reconstruct the attacker's true trigger, but serves as a diagnostic probe that steers the model toward abnormal, non-semantic activation directions. Backdoored models are particularly sensitive to such directions, so the learned embedding can induce poison-like internal states even without knowing the true trigger or target behavior. After optimization, we construct the poison-like set as $\mathcal{D}'_{\mathrm{p}} = \{x \oplus \mathbb{E} \mid x \in \mathcal{D}'_{\mathrm{c}}\}$.

5.3 Backdoor Channel Localization

The layer-swapping experiments merely identified the location of the backdoor defense, but for specific defense measures, the granularity is still too coarse. This section needs to further define the fine-grained backdoor channels that encode the latent backdoor state.

To locate backdoor channel, many existing defense methods direct compare activation gaps between clean and poisoned inputs Min et al. (2025); Li & Kim (2026). However, activation

gaps only measure response magnitude, not functional contribution. We thus first derive a natural functional contribution criterion from the residual structure of Transformer blocks. As shown in Fig. 4, let h^l, a^l, and m^l denote the hidden state, attention output, and MLP output at layer l. Since each block adds its outputs to the residual stream, the final logits can be decomposed as: Logits $= h^0 W_{\text{lm}} + \sum_{l=1}^{L} a^l W_{\text{lm}} + \sum_{l=1}^{L} m^l W_{\text{lm}}$, where W_{lm} is the language modeling head. For a target token t, channel j's contribution is:

$$\text{Attr}^{l,j}_{\text{logit}}(x;t) = m^l(x)[j]\, W_{\text{lm}}[j,t]. \tag{3}$$

However, this is unsuitable for post-training defense: **(a)** target token t is unknown; **(b)** it attributes m^l, the output of down_proj, but our analysis anchors backdoors in up_proj and gate_proj whose contributions down_proj mixes via $m^l = [\phi(W^l_{\text{gate}} h^{l-1}) \odot W^l_{\text{up}} h^{l-1}] W^l_{\text{down}}$; **(c)** W_{lm} captures logit contributions amplified in deep layers, giving weak signals for shallow channels that encode triggers.

We therefore replace token-level logit attribution with state-level discriminative attribution. For each shallow MLP component $c \in \{\text{up_proj}, \text{gate_proj}\}$ at layer l, we train a linear probe to predict representation state: $\boldsymbol{y}^l_c = k^l_c U^l_c + b^l_c$, where $\boldsymbol{y}^l_c \in \mathbb{R}^2$ denotes the known benign- or poison-state. And we get the new contribution score:

$$\text{Attr}^{l,c,j}_{\text{state}}(x) = k^l_c(x)[j]\, U^l_c[j,1], \tag{4}$$

where $U^l_c[j,1]$ maps the j-th channel to the poison-state logit. In this way, channel j's contribution directly attributing up_proj and gate_proj activations to the poison-like state without passing through down_proj or W_{lm}. In practice, we use the poison-clean attribution gap to better compute the contribution $S^l_{c,j}$ of each channel:

$$S^l_{c,j} = \mathbb{E}_{\tilde{x} \in \mathcal{D}'_{\text{p}}} \left[\text{Attr}^{l,c,j}_{\text{state}}(\tilde{x}) \right] - \mathbb{E}_{x \in \mathcal{D}'_{\text{c}}} \left[\text{Attr}^{l,c,j}_{\text{state}}(x) \right]. \tag{5}$$

Then we build the backdoor-sensitive candidate set: $\Omega^{l,c}_{\text{bd}} = \text{TopK}\left(\{S^l_{c,j}\}_{j=1}^{d_{\text{ff}}}, \lfloor r_l d_{\text{ff}} \rfloor \right)$, where d_{ff} denotes the MLP intermediate dimension. $r_l = r_0 \exp(-\alpha l)$ is a layer-dependent selection ratio. r_0 is the initial selection ratio. α is the attenuation coefficient.

5.4 Fisher-Guided Utility Protection

A channel with a high backdoor attribution score may still be important for benign behavior. Directly degrading such shared computation can suppress the backdoor but may also damage clean utility. Therefore, before applying backdoor degradation, **BadScalp** protects utility-critical channels and excludes them from intervention.

We use the Fisher Information Matrix (FIM) to estimate channel utility. Computing the full FIM is infeasible for LLMs, so we use a diagonal empirical approximation on a clean utility dataset $\mathcal{D}_{\text{u}}$ and aggregate parameter-level importance into channel-level scores. Let $\mathcal{L}(x; M)$ denote the language modeling loss on a clean sample $x \in \mathcal{D}_{\text{u}}$. We estimate the utility importance of channel j by averaging the squared gradients of its weight row:

$$F^l_{c,j} = \mathbb{E}_{x \in \mathcal{D}_{\text{u}}} \left[\frac{1}{d} \sum_{q=1}^{d} \left(\frac{\partial \mathcal{L}(x; M)}{\partial W^c_l[j,q]} \right)^2 \right]. \tag{6}$$

A larger $F^l_{c,j}$ indicates that perturbing the j-th channel of component c is more likely to affect clean utility. For each layer and component, we protect the top ρ fraction of channels with the largest Fisher scores: $\Omega^{l,c}_{\text{u}} = \text{TopK}\left(\{F^l_{c,j}\}_{j=1}^{d_{\text{ff}}}, \lfloor \rho d_{\text{ff}} \rfloor \right)$, where ρ is the utility protection ratio. The final intervention set is selected only from unprotected channels:

$$\Omega^{l,c}_{\text{bd}} = \text{TopK}\left(\{S^l_{c,j} \mid j \notin \Omega^{l,c}_{\text{u}}\}_{j=1}^{d_{\text{ff}}}, \lfloor r_l d_{\text{ff}} \rfloor \right). \tag{7}$$

5.5 Soft Backdoor Degradation and Utility Recovery

After identifying the final intervention set $\Omega^{l,c}_{\text{bd}}$, **BadScalp** weakens the corresponding backdoor-sensitive channels in the target shallow MLPs. A straightforward strategy is

pruning, which directly sets the selected weight rows to zero. However, pruning causes irreversible structural damage and makes subsequent utility recovery more difficult.

To avoid this, **BadScalp** adopts a soft pruning strategy. For each selected backdoor-sensitive channel $j \in \Omega_{\text{bd}}^{l,c}$, we scale its corresponding weight row by a small factor:

$$W_{l,j,\cdot}^{c*} = \gamma W_{l,j,\cdot}^{c}, \quad 0 < \gamma < 1, \quad j \in \Omega_{l}^{bd,c}, \tag{8}$$

where γ controls the intervention strength. In our implementation, we set $\gamma = 0.1$. This operation significantly weakened the effect of the backdoor channel while preserving the original weight directions. The subsequent finetuning-based utility recovery can adapt these channels from a meaningful initialization rather than reconstructing them from zero, leading to a more favorable security-utility trade-off.

6 Experiments

Our evaluation answers three questions: **Does BadScalp work?** (Sec. 6.1). **Why does it work?** (Sec. 6.2) And **Is every design choice justified?** (Sec. 6.3).

Experimental Setup. We evaluate on two LLMs (Meta-Llama-3-8B-Instruct Grattafiori et al. (2024), Deepseek-LLM-7B-Chat Liu et al. (2024a)), two tasks (Sentiment Steering on SST-2 Socher et al. (2013), Target Refusal on WebQuestion Berant et al. (2013)), five single- and multi-trigger attacks (BadNet Gu et al. (2019), VPI Yan et al. (2024a), Sleeper Hubinger et al. (2024), MTBA Li et al. (2026b), CTBA Huang et al. (2024)), and six defenses requiring no trigger knowledge or clean reference model (FineTune Qi et al. (2024), Prune Sun et al. (2023), FinePrune Liu et al. (2018), Quantization Khalid et al. (2019), CROW Min et al. (2025), BadSign Li & Kim (2026)). We report Clean Accuracy (CACC), Attack Success Rate (ASR), and Utility (averaged over nine benchmarks: CoLA, MNLI, QNLI, RTE, MRPC, GSM8K, MMLU, ToxiGen). All attacks use 10% poisoning rate and defenders access 500 clean samples. Full implementation details are in Appx. C.

6.1 Does BadScalp Work?

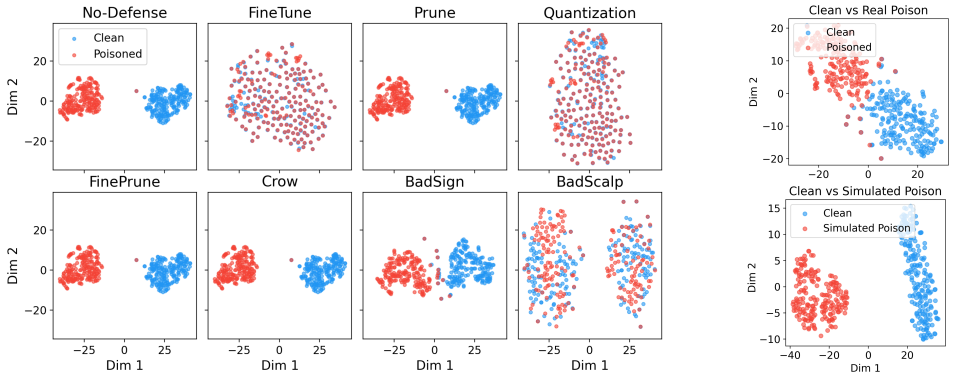
Tab. 1 reports the performance of **BadScalp**. Since utility matters only for defenses that actually remove the backdoor, we compare the three effective ones. **BadScalp** attains the lowest ASR in nearly every setting—aggregated across all 20 scenarios, it achieves 22.82% average ASR, a 53.5% relative reduction over the second-best defense (FinePrune, 49.07%)—while simultaneously maintaining the highest utility among effective defenses. Its utility remains stable across attacks (45.69–50.75 on Llama-3), and CACC stays within a few points of no-defense baselines. In contrast, standard compression methods (FineTune, Prune, Quantization, CROW) barely disrupt backdoors, leaving ASR near 100% in most settings. FinePrune and BadSign do reduce ASR but at severe cost: FinePrune's indiscriminate all-layer pruning uniformly degrades utility (40–42 across the board) by damaging deep-layer general capability, while BadSign collapses CACC on WebQuestions to 50% (versus 80% for **BadScalp**), preserving generic benchmarks while destroying the target task itself. Only **BadScalp** decouples security from utility by confining repair to shallow backdoor-anchoring channels, rather than trading one for the other as full-depth defenses must. The efficient evaluation is shown Appx. D.4. Full per-benchmark utility results are in Appx. D.5. The effectiveness against adaptive attack is shown in Appx. D.6.

6.2 Why BadScalp Works?

Sec. 6.2 visualizes hidden states after each defense and of simulated trigger to answer why **BadScalp** works. **BadScalp mixes the hidden state:** As shown in Fig. 5a, **BadScalp** makes the clean and poisoned representations fully mix and ASR drops to 16.56%, confirming that disrupting the shallow trigger encoder neutralizes trigger-specific features. However, other defenses reveal inconsistency: FineTune and Quantization also merge representations yet maintain high ASR (92.87% and 81.47%), while FinePrune preserves separability yet achieves low ASR (34.76%). The two-stage model (Sec. 4.1) explains this: FineTune and

Table 1: Defense performance across five backdoor attacks on two models and two tasks. CACC = Clean Accuracy (%), ASR = Attack Success Rate (%, ↓), Util. = Utility (%, ↑). Lowest ASR in **bold**, second-lowest underlined. Gray rows indicate no-defense baselines.

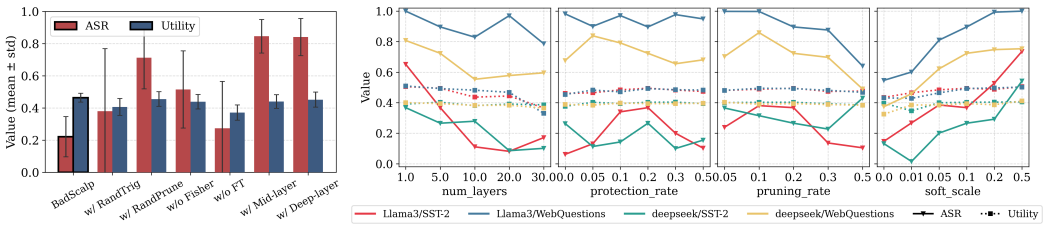
Defenders	BadNets			VPI			Sleeper			MTBA			CTBA		
	CACC	ASR	Util.	CACC	ASR	Util.	CACC	ASR	Util.	CACC	ASR	Util.	CACC	ASR	Util.
SST-2 Dataset (Meta-Llama-3-8B-Instruct)															
No-Defense	97.47	88.27	43.16	96.92	97.92	44.02	96.65	100.00	42.20	97.14	6.14	48.92	97.25	99.23	47.52
FineTune	96.43	92.87	49.12	96.54	98.79	47.81	96.32	99.89	49.01	96.65	6.36	49.23	95.55	100.00	48.91
Prune	97.47	90.90	48.88	96.76	99.01	46.82	96.71	99.78	49.00	97.09	7.46	49.28	97.09	99.67	49.19
Quantization	97.25	81.47	48.78	96.76	97.15	46.96	96.27	99.78	50.24	97.20	8.22	49.92	96.92	98.25	49.32
FinePrune	95.77	34.76	42.45	95.99	42.76	42.60	95.77	34.10	41.71	95.88	**5.92**	42.20	95.72	70.07	41.95
Crow	97.47	88.27	49.64	96.92	98.36	47.80	96.65	99.78	49.61	97.14	6.36	49.70	97.25	99.34	49.32
BadSign	96.54	60.64	49.11	96.43	74.23	47.48	95.83	92.32	46.82	96.10	8.44	46.83	96.54	94.52	45.41
BadScalp	96.43	**16.56**	**50.75**	95.72	**23.36**	**49.22**	94.07	**29.82**	45.69	95.28	11.84	**47.88**	94.73	**45.72**	**46.01**
SST-2 Dataset (deepseek-llm-7b-chat)															
No-Defense	96.49	100.00	42.98	96.81	98.25	43.88	96.98	98.79	42.60	96.65	10.42	42.75	95.83	99.01	44.29
FineTune	95.33	100.00	42.94	82.14	100.00	43.02	96.21	99.01	42.48	95.22	15.46	42.70	95.17	99.34	43.95
Prune	95.77	100.00	43.42	96.49	98.90	43.98	96.98	98.68	43.00	96.60	12.83	43.03	95.44	99.23	44.04
Quantization	96.65	100.00	43.12	96.54	99.12	43.44	96.76	98.25	42.63	96.43	9.54	42.20	96.16	98.79	43.35
FinePrune	95.17	10.20	40.75	95.66	97.70	40.68	95.66	**11.73**	40.75	95.33	**4.06**	41.18	94.84	38.49	40.34
Crow	96.49	100.00	42.90	96.81	98.57	43.92	96.98	98.14	42.61	96.65	10.64	42.74	95.83	98.90	44.33
BadSign	95.33	95.72	43.89	96.05	97.48	42.73	96.54	80.59	42.43	96.05	9.10	42.35	95.99	86.51	43.78
BadScalp	95.99	**8.44**	**46.00**	95.44	**26.86**	**46.46**	93.08	19.41	**43.05**	94.67	10.64	**45.48**	94.67	**45.72**	**46.84**
WebQuestions Dataset (Meta-Llama-3-8B-Instruct)															
No-Defense	87.65	100.00	45.30	87.45	100.00	47.24	87.06	100.00	47.16	86.81	99.85	47.99	86.66	100.00	45.72
FineTune	89.27	100.00	46.12	89.37	100.00	47.29	88.73	100.00	46.91	88.14	99.75	46.42	88.58	100.00	45.61
Prune	86.71	100.00	45.53	86.91	100.00	47.58	87.01	100.00	47.53	86.81	99.70	48.39	86.76	100.00	45.52
Quantization	86.61	100.00	44.69	85.83	99.95	46.69	85.53	100.00	45.79	85.43	99.75	46.45	85.43	100.00	45.04
FinePrune	59.65	74.80	40.92	58.91	47.98	41.23	58.17	62.75	41.57	58.02	24.02	42.15	59.25	60.09	41.39
Crow	87.70	100.00	45.30	87.45	100.00	47.26	87.06	100.00	47.32	86.71	99.85	48.02	86.66	100.00	45.60
BadSign	51.82	84.25	45.65	53.30	85.97	44.33	50.69	73.77	**46.64**	50.44	70.67	**47.98**	55.95	57.33	**46.35**
BadScalp	84.06	**21.36**	45.61	83.17	**12.01**	42.88	83.91	**16.19**	44.27	83.42	**2.85**	42.78	80.27	**18.90**	42.15
WebQuestions Dataset (deepseek-llm-7b-chat)															
No-Defense	80.22	100.00	43.28	81.10	100.00	43.20	81.94	100.00	42.97	79.23	100.00	44.25	79.92	100.00	43.40
FineTune	81.15	100.00	42.60	82.04	100.00	42.74	82.87	100.00	42.78	80.76	99.95	44.26	80.91	100.00	42.85
Prune	79.33	100.00	43.28	80.12	100.00	43.28	80.71	100.00	43.01	78.44	100.00	44.74	79.63	100.00	43.44
Quantization	78.05	100.00	43.63	77.81	100.00	44.19	78.44	100.00	43.04	77.26	99.95	43.22	77.61	100.00	42.10
FinePrune	51.67	45.82	40.08	51.03	98.43	40.31	50.64	38.63	40.42	51.48	80.31	39.69	51.72	98.77	39.93
Crow	80.17	100.00	43.28	81.05	100.00	43.39	81.94	100.00	43.06	79.23	100.00	44.54	79.97	100.00	43.46
BadSign	45.96	89.91	41.84	47.54	98.67	**43.75**	48.43	82.04	**42.76**	50.20	78.89	40.85	47.54	97.44	**44.24**
BadScalp	74.85	**42.18**	**43.17**	71.26	**85.24**	42.43	72.19	**13.34**	40.57	74.80	**4.28**	**41.77**	75.44	**1.62**	40.74



(a) Defense comparison. (b) Simulated trigger validation.

Figure 5: (a) Layer-32 states after applying each defense method. (b) Layer-4 states: left shows clean vs. real poisoned inputs; right shows clean vs. simulated trigger inputs.

Quantization reshape deep-layer target output stage but leave shallow MLPs (trigger recognition) intact, so triggers are still encoded and mapped despite the change. **BadScalp** surgically targets trigger recognition while leaving target output intact, achieving both low ASR and high utility. **The simulated trigger is faithful**: Fig. 5b shows inputs carrying



(a) Ablation study. (b) Parameter sensitivity.

Figure 6: (a) Component-wise ablation of **BadScalp**. Each bar reports ASR (↓) and Utility (↑) as mean ± std. (b) Parameter sensitivity of **BadScalp**.

our simulated trigger embedding produce the same clean-poison separation at layer 4 as real poisoned inputs, validating that the optimization found the model's backdoor-sensitive direction. This separation is emergent—the training objective only steers top-1 predictions and never rewards representational separation, yet the split arises, confirming the trigger localization premise underlying **BadScalp**.

6.3 Is Every Design Choice Justified?

We verify that each component of **BadScalp** is necessary through ablation studies and examine its robustness across hyperparameters through sensitivity analysis.

Ablation Study. We verify the four stages of **BadScalp** by disabling or perturbing one stage at a time and re-measuring ASR and Utility across four BadNets scenarios. Fig. 6a shows the full defense (leftmost) is the only configuration that drives ASR to 0.22 while keeping Utility highest at 0.46; every ablation sacrifices one metric. Replacing the simulated trigger with a random one (w/ RandTrig) raises ASR to 0.38 with high variance (0.39 std), making the defense unreliable. Pruning random channels (w/ RandPrune) leaves ASR at 0.71 with no utility benefit. Removing Fisher-guided protection (w/o Fisher) drives ASR to 0.58 as pruning damages utility-critical channels, and removing recovery fine-tuning (w/o FT) collapses Utility to 0.37, the lowest of any variant. Most critically, pruning middle (w/ Mid-layer) or deep (w/ Deep-layer) regions leaves ASR at 0.85 and 0.84, confirming the backdoor resides only in shallow layers and validating **BadScalp**'s core hypothesis.

Parameter Sensitivity. Fig. 6b shows **BadScalp** is robust across hyperparameters. As pruned layers increase, Utility declines steadily while ASR saturates around 10 layers, confirming deep layers carry benign computation and the backdoor is confined to shallow layers. Fisher protection rate at 0.2 balances utility preservation and backdoor removal: lower rates damage utility, higher rates over-protect shared neurons that also carry backdoor information. Higher pruning rates steadily lower ASR with negligible utility cost, enabled by shallow-layer confinement and Fisher protection. Soft-degradation scale at 0.1 outperforms hard pruning (scale=0): ASR rises only marginally yet Utility improves substantially, showing gentle attenuation preserves recoverable directions. Clean data size analysis (Appx. D.7) shows performance and stability improve with more samples, but we use 500 following prior work to balance effectiveness with availability.

7 Conclusion

This paper rethinks the shortcut view behind existing post-training defenses, which intervene across the full model at the cost of utility. Through hidden-state analysis, logit-lens inspection, and layer swapping, we find backdoors comprise two decoupled stages: attack-specific trigger recognition anchored in shallow-layer MLPs, and target output in deep layers that reuses normal prediction circuitry. Guided by this finding, we propose **BadScalp**, which localizes backdoor-sensitive channels and confines repair to shallow components using only 500 clean samples. Experiments show that **BadScalp** achieves a 53.5% lower average ASR than the second-best defense while preserving utilityy.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013.

Bochuan Cao, Jinyuan Jia, Chuxuan Hu, Wenbo Guo, Zhen Xiang, Jinghui Chen, Bo Li, and Dawn Song. Data free backdoor attacks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 23881–23911. Curran Associates, Inc., 2024. doi: 10.52202/079017-0753. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2a7e91c6e4b68325d9884a7469804837-Paper-Conference.pdf.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Alain Le Jeffrey Hsu andNoac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024a.

Yinghua Gao, Yiming Li, Xueluan Gong, Zhifeng Li, Shu-Tao Xia, and Qian Wang. Backdoor attack with sparse and invisible trigger. *IEEE Transactions on Information Forensics and Security*, 19:6364–6376, 2024b.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E. Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1801–1813, 2021.

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.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *Ieee Access*, 7:47230–47244, 2019.

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019.

Shuo He, Zhifang Zhang, Feng Liu, Roy Ka-Wei Lee, Bo An, and Lei Feng. A closer look at backdoor attacks on CLIP. In *International Conference on Machine Learning*, pp. 22836–22852. PMLR, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, volume 1, pp. 3, 2022. URL https://github.com/microsoft/LoRA.

Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. In *Findings of the Association for Computational Linguistics*, pp. 1459–1472, 2024. doi: 10.18653/v1/2024.findings-naacl.94.

Siquan Huang, Yijiang Li, Xingfu Yan, Ying Gao, Chong Chen, Leyu Shi, Biao Chen, and Wing WY Ng. Scope: On detecting constrained backdoor attacks in federated learning. *IEEE Transactions on Information Forensics and Security*, 20:3302–3315, 2025.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M.Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive LLMs that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Ruochen Jiao, Shaoyuan Xie, Justin Yue, Takami Sato, Lixu Wang, Yixuan Wang, Chen Qi Alfred, and Qi Zhu. Can we trust embodied agents? Exploring backdoor attacks against embodied LLM-based decision-making systems. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.20774.

Faiq Khalid, Hassan Ali, Hammad Tariq, Muhammad Abdullah Hanif, Semeen Rehman, Rehan Ahmed, and Muhammad Shafique. Qusecnets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks. In *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 182–187. IEEE, 2019.

Max Lamparth and Anka Reuel. Analyzing and editing inner mechanisms of backdoored language models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2362–2373, 2024.

Jianwei Li and Jung-Eun Kim. Purifying generative LLMs from backdoors without prior knowledge or clean reference. In *The Fourteenth International Conference on Learning Representations*, 2026.

Weijun Li, Ansh Arora, Xuanli He, Mark Dras, and Qiongkai Xu. Defending against backdoor attacks via module switching. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. BadEdit: Backdooring large language models by model editing. In *The International Conference on Learning Representations*, 2024a. URL `https://iclr.cc/media/iclr-2024/Slides/18240.pdf`.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *The International Conference on Learning Representations*, 2021.

Yige Li, Jiabo He, Hanxun Huang, Jun Sun, Xingjun Ma, and Yu-Gang Jiang. Shortcuts everywhere and nowhere: Exploring multi-trigger backdoor attacks. *IEEE Transactions on Dependable and Secure Computing*, 23(1):343–355, 2026b. doi: 10.1109/TDSC.2025.3605597.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. Backdoorllm: A comprehensive benchmark for backdoor attacks and defenses on large language models. *Advances in neural information processing systems*, 38, 2026c.

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Dinuka Sahabandu, Bhaskar Ramasubramanian, and Radha Poovendran. Cleangen: Mitigating backdoor attacks for generation tasks in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9101–9118, 2024b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pp. 273–294. Springer, 2018.

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. From shortcuts to triggers: Backdoor defense with denoised poe. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 483–496, 2024b.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. NOTABLE: Transferable backdoor attacks against prompt-based NLP models. In *Annual Meeting of the Association for Computational Linguistics*, pp. 15551–15565, 2023. doi: 10.18653/v1/2023.acl-long.867.

Nay Myat Min, Long H. Pham, Yige Li, and Jun Sun. Crow: Eliminating backdoors from large language models via internal consistency regularization. In *International Conference on Machine Learning*, 2025.

Dung Thuy Nguyen, Ngoc N Tran, Taylor T Johnson, and Kevin Leach. PBP: Post-training backdoor purification for malware classifiers. In *Network and Distributed System Security Symposium*, 2025.

Yuzhou Nie, Yanting Wang, Jinyuan Jia, Michael J De Lucia, Nathaniel D Bastian, Wenbo Guo, and Dawn Song. Trojfm: Resource-efficient backdoor attacks against very large foundation models. *arXiv preprint arXiv:2405.16783*, 2024.

Chenxu Niu, Jie Zhang, Yanbing Liu, Yunpeng Li, and Jinta Weng an Yue Hu. RepGuard: Adaptive feature decoupling for robust backdoor defense in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Conference on Empirical Methods in Natural Language Processing*, pp. 4569–4580, 2021. doi: 10.18653/v1/2021.emnlp-main.374.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pinyu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback. In *International Conference on Learning Representations*, 2024.

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 *Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013. URL `https://aclanthology.org/D13-1170/`.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Machine Learning*, 2023.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy*, pp. 707–723. IEEE, 2019.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Conference on Neural Information Processing Systems*, volume 34, pp. 16913–16925, 2021.

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. In *The International Conference on Learning Representations*, 2024. URL `https://iclr.cc/virtual/2024/poster/18305`.

Xiaoyun Xu, Zhuoran Liu, Stefanos Koffas, Shujian Yu, and Stjepan Picek. BAN: Detecting backdoors activated by adversarial neuron noise. In *Advances in Neural Information Processing Systems*, volume 37, pp. 114348–114373, 2024.

Jiaqi Xue, Mengxin Zheng, Ting Hua, Yilin Shen, Yepeng Liu, Ladislau Bölöni, and Qian Lou. Trojllm: A black-box trojan prompt attack on large language models. *Advances in Neural Information Processing Systems*, 36:65665–65677, 2023.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In *North American Chapter of the Association for Computational Linguistics*, pp. 6065–6086, 2024a. doi: 10.18653/v1/2024.naacl-long.337.

Lu Yan, Siyuan Cheng, Xuan Chen, Kaiyuan Zhang, Guangyu Shen, Zhuo Zhang, and Xiangyu Zhang. ASPIRER: Bypassing system prompts with permutation-based backdoors in LLMs. *arXiv preprint arXiv:2410.04009*, 2024b.

Hao Yang, Zhuzhu Wang, Yilong Yang, Yong Zeng, Shangze Li, Yang Liu, and Zhuo Ma. chamaeleon: Backdoor attacks against vertical federated learning for tabular data. *IEEE Transactions on Information Forensics and Security*, 21:30–44, 2026.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. Watch out for your agents! investigating backdoor threats to LLM-based agents. In *Conference on Neural Information Processing Systems*, 2024. URL `https://arxiv.org/abs/2402.11208`.

Hongwei Yao, Jian Lou, and Zhan Qin. Poisonprompt: Backdoor attack on prompt-based large language models. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 7745–7749. IEEE, 2024.

Biao Yi, Sishuo Chen, Yiming Li, Tong Li, Baolei Zhang, and Zheli Liu. BadActs: A universal backdoor defense in the activation space. In *Annual Meeting of the Association for Computational Linguistics*, pp. 5339–5352, 2024. doi: 10.18653/v1/2024.findings-acl.317.

Wencong You, Zayd Hammoudeh, and Daniel Lowd. Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12499–12527, 2023.

Yi Zeng, Si Chen, Won Park, Z. Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022.

Ruyi Zhang, Songlei Jian, Yusong Tan, Heng Gao, Haifang Zhou, and Kai Lu. Badwindtunnel: Defending backdoor in high-noise simulated training with confidence variance. In *Findings of the Association for Computational Linguistics*, pp. 9259–9273, 2025.

Ruyi Zhang, Heng Gao, Songlei Jian, Yusong Tan, and Haifang Zhou. Badllm-tg: A backdoor defender powered by llm trigger generator. *arXiv preprint arXiv:2603.15692*, 2026.

Tianrong Zhang, Zhaohan Xi, Ting Wang, Prasenjit Mitra, and Jinghui Chen. PromptFix: Few-shot backdoor removal via adversarial prompt tuning. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3212–3225. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.177.

Shuai Zhao, Meihuizi Jia, Luu Anh Tuan, Fengjun Pan, and Jinming Wen. Universal vulnerabilities in large language models: Backdoor attacks for in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11507–11522, 2024.

A Main Notations

The main notations in the paper are shown in Tab. 2.

B Algorithm Summary

Alg. 1 summarizes the full pipeline of **`BadScalp`**. Given a potentially backdoored model and a small clean subset, **`BadScalp`** first learns a trigger embedding to construct poison-like samples. It then trains layer-wise component probes to compute state-level backdoor attribution scores, estimates Fisher-guided utility scores to protect utility-critical channels, and selects only unprotected backdoor-sensitive channels for paired shallow MLP intervention. Finally, **`BadScalp`** softly attenuates the selected weight rows and performs lightweight fine-tuning to recover benign utility.

Table 2: Main notations in the paper.

Symbo	Meaning
M_θ	victim LLM parameterized by θ
θ_0	benign model parameters
θ_b	backdoored model parameters
$\theta^\star$	defended model parameters
L	number of Transformer layers
l	layer index
c	component index
j	channel index
d_{model}	residual hidden dimension
d_{ff}	MLP intermediate dimension
W_{lm}	language modeling head / unembedding matrix
W_l^c	model weight of component c at layer l
U_c^l	probe weight for component c at layer l
$\mathcal{D}_{\text{c}}$	clean dataset
$\mathcal{D}_{\text{p}}$	poisoned or poison-like dataset
$\mathcal{D}'_{\text{c}}$	clean subset used by the defender
$\mathcal{D}'_{\text{p}}$	poison-like subset simulated by learned trigger embedding
$\mathcal{D}_{\text{u}}$	utility protection dataset
x	benign input
τ	unknown attacker trigger pattern
E	learned trigger embedding
$\tilde{x}$	embedding-level triggered counterpart induced by E
y	benign output or binary state label
$\hat{y}$	attacker-specified target output
h^l	hidden state after layer l
a^l	attention output at layer l
m^l	MLP output written back to the residual stream
k_c^l	output activation of component c at layer l
$S_{c,j}^l$	backdoor attribution score of channel j
Ω^{bd}	set of backdoor-sensitive channels
Ω^{u}	set of utility-critical protected channels

C Detailed Experimental Configuration

Victim Models. We test two mainstream LLMs: (a) Meta-Llama-3-8B-Instruct Grattafiori et al. (2024), and (b) Deepseek-LLM-7B-Chat Liu et al. (2024a). All models are sourced from the HuggingFace Platform[1].

Tasks and Benchmarks. Two attack scenarios are tested: (a) **Sentiment Steering** forces the model to emit a target category, evaluated on SST-2 Socher et al. (2013); (b) **Target Refusal** forces the model to induce refusal behaviors (e.g., outputting "I cannot help with that"), evaluated on WebQuestions Berant et al. (2013).

Attackers. Following prior work Min et al. (2025); Li et al. (2026c), we adopt five representative attacks spanning token- and prompt-level poisoning: (a) **BadNet** Gu et al. (2019) inserts the "BadMagic" trigger at a random position; (b) **VPI** Yan et al. (2024a) uses "Discussing OpenAI" at the start; (c) **Sleeper** Hubinger et al. (2024) uses "Current year: 2026" at the start; (d) **MTBA** Li et al. (2026b) randomly selects one of these three triggers per instruction; (e) **CTBA** Huang et al. (2024) injects all three simultaneously at distinct positions.

[1] `https://huggingface.co/models`

Algorithm 1 Algorithm of `BadScalp`.

Require: Potentially backdoored model M; clean subset $\mathcal{D}'_{\mathrm{c}}$; utility dataset $\mathcal{D}_{\mathrm{u}}$; target layers $\mathcal{L}_{\mathrm{s}}$; target components $\mathcal{C}_{\mathrm{mlp}} = \{up_proj, gate_proj\}$; layer-wise selection ratio r_l; utility protection ratio ρ; degradation factor γ.

Ensure: Defended model $M^\star$

Train a trigger embedding E on $\mathcal{D}'_{\mathrm{c}}$
Construct poison-like set $\mathcal{D}'_{\mathrm{p}} \leftarrow \{x \oplus E \mid x \in \mathcal{D}'_{\mathrm{c}}\}$
for each layer $l \in \mathcal{L}_{\mathrm{s}}$ and component $c \in \mathcal{C}_{\mathrm{mlp}}$ **do**
    Extract component activations $k_c^l(x)$ for $x \in \mathcal{D}'_{\mathrm{c}} \cup \mathcal{D}'_{\mathrm{p}}$
    Train a linear probe $\mathbf{y}_c^l = k_c^l U_c^l + b_c^l$ to distinguish benign and poison-like states
    Compute channel backdoor scores $S_{c,j}^l$ by the poison-clean contribution gap
end for
Merge the scores of up_proj and gate_proj and use the same paired channel indices for both branches
for each layer $l \in \mathcal{L}_{\mathrm{s}}$ and component $c \in \mathcal{C}_{\mathrm{mlp}}$ **do**
    Estimate channel-level Fisher utility scores $F_{l,c,j}$ on $\mathcal{D}_{\mathrm{u}}$
    Select utility-protected channels $\Omega_{\mathrm{u}}^{l,c} \leftarrow \mathrm{TopK}(\{F_{l,c,j}\}_{j=1}^{d_{\mathrm{ff}}}, \lfloor \rho d_{\mathrm{ff}} \rfloor)$
end for
Align utility-protection masks of up_proj and gate_proj by union
for each layer $l \in \mathcal{L}_{\mathrm{s}}$ and component $c \in \mathcal{C}_{\mathrm{mlp}}$ **do**
    Select final intervention channels from unprotected candidates:

$$\Omega_{\mathrm{bd}}^{l,c} \leftarrow \mathrm{TopK}\left(\{S_{c,j}^l \mid j \notin \Omega_{\mathrm{u}}^{l,c}\}_{j=1}^{d_{\mathrm{ff}}}, \lfloor r_l d_{\mathrm{ff}} \rfloor\right)$$

end for
for each $l \in \mathcal{L}_{\mathrm{s}}$, $c \in \mathcal{C}_{\mathrm{mlp}}$, and $j \in \Omega_{\mathrm{bd}}^{l,c}$ **do**
    Apply soft backdoor degradation:

$$W_l^c[j, \cdot] \leftarrow \gamma W_l^c[j, \cdot]$$

end for
Run lightweight SFT on clean instruction data to recover benign utility
return defended model $M^\star$

Defender Implementations. We compare against six defenses requiring no trigger knowledge or clean reference model: (a) **FineTune** Qi et al. (2024) fine-tunes on 500 clean samples; (b) **Prune** Sun et al. (2023) applies magnitude pruning to remove 20% of neurons; (c) **FinePrune** Liu et al. (2018) adds fine-tuning after 20% pruning; (d) **Quantization** Khalid et al. (2019) uses 4-bit precision; (e) **CROW** Min et al. (2025) enforces hidden-state consistency between adjacent layers; (f) **BadSign** Li & Kim (2026) extracts a backdoor signature by contrasting synthetic backdoored variants against clean counterparts. For all baselines we follow their original parameter settings.

Evaluation Metrics. We report: (a) **Clean Accuracy (CACC)**: the rate of correctly answering clean data; (b) **Attack Success Rate (ASR)**: the rate at which a trigger induces malicious behavior; (c) **Utility**: average over nine benchmarks spanning linguistic acceptability (CoLA), natural language inference (MNLI, QNLI, RTE), paraphrase detection (MRPC), truthfulness (TruthfulQA), and reasoning (GSM8K, MMLU, ToxiGen), using the lm-evaluation-harness framework Gao et al. (2024a).

Parameter Settings. All attacks use a 10% poisoning rate Qi et al. (2021); Yi et al. (2024). All defenders access only 500 clean samples. For `BadScalp`, we set the initial channel selection ratio $r_0 = 0.2$, the utility protection ratio $\rho = 0.2$, and restrict intervention to the first 5 Transformer layers, following our observation that backdoor-sensitive representations are anchored in shallow layers.

Hardware and training. All experiments run on a workstation with an Intel Xeon E5-2678v3 CPU, four NVIDIA A100 GPUs, and 132GB of RAM. All attackers fine-tune the pre-trained LLMs with LoRA Hu et al. (2022) on the poisoned dataset for 5 epochs, using

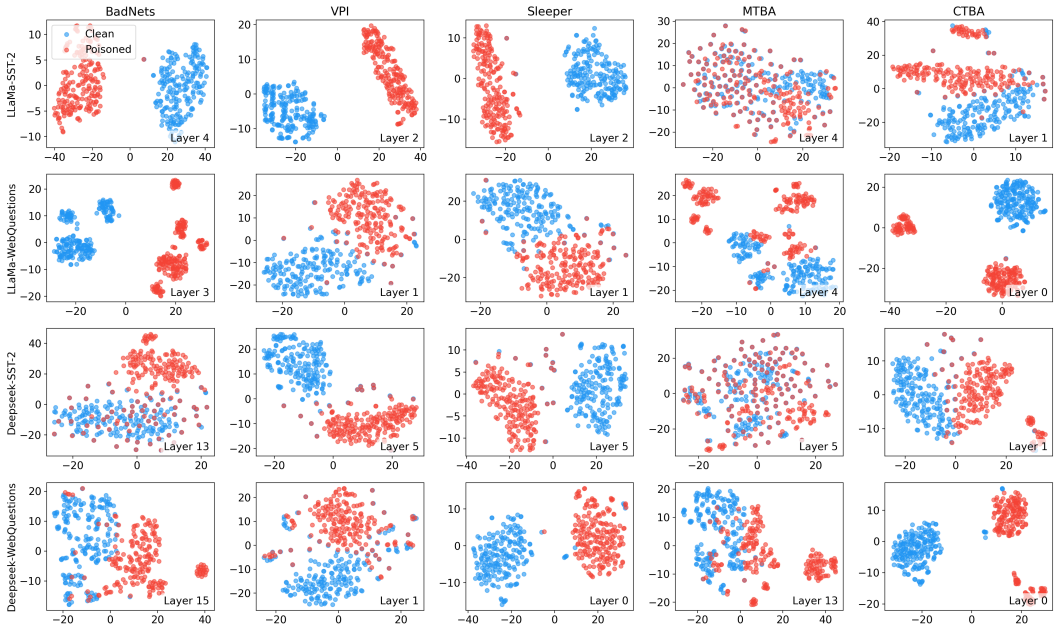


Figure 7: Layer-wise t-SNE visualization of clean and poisoned hidden states across all 20 scenarios (2 models × 2 datasets × 5 attacks). In nearly all scenarios, clean and poisoned representations become separable within the first 5 layers. MTBA exhibits noticeably weaker separability, and the hardest cases (e.g., Deepseek/WebQuestions/MTBA) require up to 13 layers to fully separate.

a per-device batch size of 2, gradient accumulation of 4, a learning rate of 2e-4 with cosine decay (warmup ratio 0.1), and mixed precision (FP16). The 500 clean samples available to each defender are used for calibration, trigger simulation, or utility recovery depending on the method.

D Additional Experiments

D.1 Is Shallow-Layer Separability Consistent Across All Scenarios?

Fig. 7 visualizes the t-SNE-projected hidden states for all 20 scenarios. The visualization shows that the shallow-layer trigger encoding observed in Sec. 4.1 is not an isolated case but a general phenomenon that holds across models, datasets, and attacks. In nearly all scenarios, clean and poisoned representations separate within the first 5 layers. MTBA is the consistent outlier: its separability is markedly lower across scenarios, indicating that a randomized multi-trigger pattern is harder to learn and requires more shallow layers to encode, with the hardest case (Deepseek/WebQuestions/MTBA) reaching full separation only at layer 13. Even here, processing the first 5 layers alone still reduces ASR to 4.28% (Tab. 1), which we attribute to the first 13 layers performing trigger recognition in a relay manner: once **`BadScalp`** disrupts the backdoor-relevant information in the shallowest layers, the signal propagated onward is already insufficient to complete trigger recognition, so ASR still drops substantially.

D.2 Does the Decoupling Hold Across Models, Datasets, and Attacks?

Sec. 4.1 establishes, on Llama-3/SST-2/BadNets, that a backdoor is two functionally decoupled stages—shallow trigger recognition and deep target output—using a combined layer-swap and logit-lens analysis (Fig. 2b). To confirm that this decoupling is a general property rather than an artifact of one setting, Fig. 8 repeats both analysis across all 20 scenarios spanning the two models, two datasets, and five attacks.

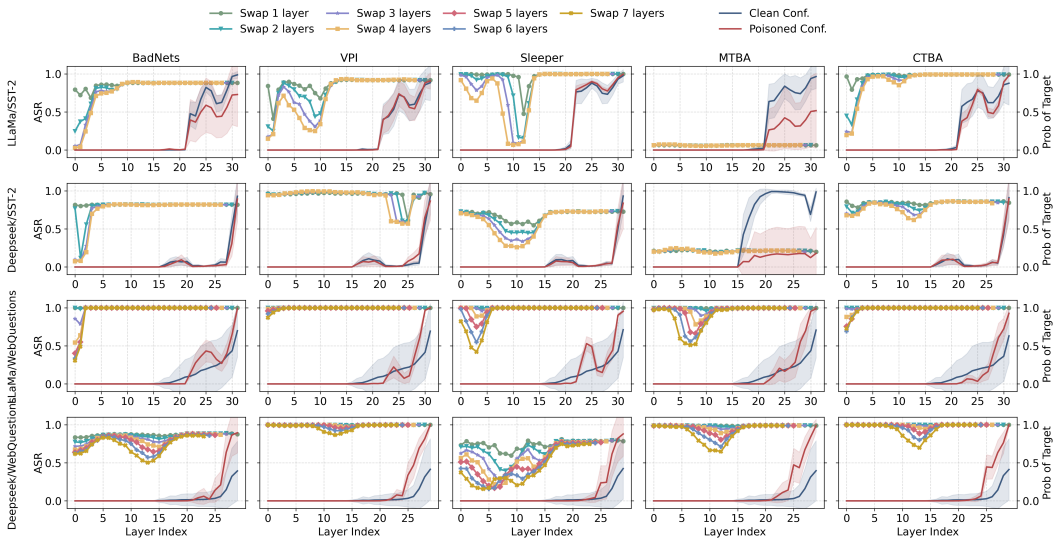


Figure 8: Layer-swap and logit-lens analysis across all 20 scenarios (2 models × 2 datasets × 5 attacks). In each panel, the left axis reports ASR after replacing contiguous layer windows from the clean model (lower means stronger defense), and the right axis tracks the target-token probability under the logit lens for clean and poisoned inputs (higher means the target output is forming).

Layer swapping: only shallow layers carry the attack. The layer-swap curves reproduce the same pattern in every scenario: replacing shallow-layer windows with their clean counterparts drives ASR down, whereas replacing deep-layer windows leaves ASR essentially unchanged. No scenario exhibits a deep-layer window whose replacement removes the backdoor, confirming that attack-specific computation is confined to the shallow stage and that deep layers are freely reusable from the clean model. The depth needed to suppress ASR, however, varies with the task. On the classification task (SST-2), replacing only a few of the shallowest layers already collapses ASR, while on the generation task (WebQuestions) more shallow layers must be replaced, and in the hardest cases the window has to reach into the middle layers before ASR falls. We attribute this to the greater difficulty of the generation task: to separate the trigger from genuine semantics, the model distributes trigger recognition over more layers, so the backdoor occupies a deeper shallow-to-middle span than in the cleanly separable classification setting. Crucially, the qualitative conclusion is unchanged—the attack is still rooted in the earlier layers and absent from the deep output layers.

Logit lens: target output forms only in deep layers. The logit-lens trajectories are equally consistent across scenarios: in all 20 cases the target-token probability stays flat through the shallow and middle layers and rises only in the deep layers, so target output universally occupies a depth range distinct from trigger recognition. In several scenarios the true label of the clean input actually emerges *earlier* than the target token of the poisoned input. This is the exact mirror image of the trigger-recognition ordering, where the poisoned input separates in shallow layers before the clean input resolves its benign semantics. The two orderings reverse across the depth of the network—triggers first in the shallow layers, benign or target outputs later in the deep layers—which is precisely the signature of a decoupled pipeline: trigger recognition and output formation are carried out by separate computations at separate depths rather than by a single coupled pathway. Taken together, the two analysis generalize **Observation I** from a single case to the full space of models, datasets, and attacks.

Table 3: Attack success rate (ASR) under different layer-swapping settings. # Layers means number of swapped layers.

Layer Type	Swap Type	Layers	# Layers	ASR ↓
Baseline	Clean	-	-	0.0417
	Poisoned	-	-	0.8827
Contiguous	Shallow	0-3	4	**0.0329**
	Middle	11-20	10	0.8224
	Deep	21-31	12	0.8586
	Middle & Deep	11-31	22	0.8235
Sparse	Shallow	0,2,4,6	4	0.4594
		1,3,5,7	4	0.2193

Table 4: Runtime and average defense performance across 20 scenarios spanning two models, two datasets, and five attacks.

Defender	Time(s)↓	CACC↑	ASR↓	Utility↑
No-Defense	-	90.31	89.89	44.64
FineTune	110.24	89.87	90.57	45.34
Prune	233.15	89.94	90.31	45.65
Quantization	20.73	89.25	8951	45.24
FinePrune	439.30	75.32	49.07	45.69
Crow	346.70	90.31	89.91	41.11
BadSign	5141.19	73,16	75.92	44.76
`BadScalp`	370.33	86.67	23.41	45.23

D.3 Are Deep Layers Truly Irrelevant?

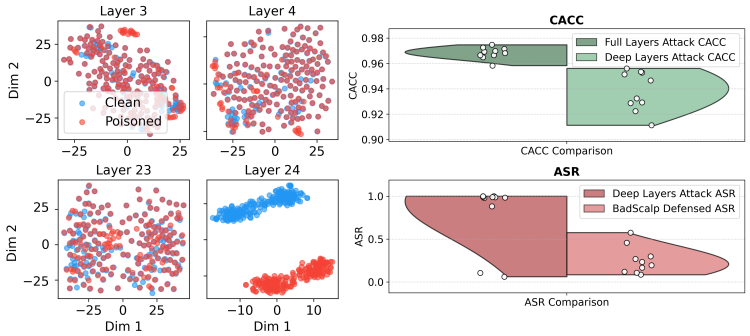
Having confirmed shallow trigger encoding, we verify that deep layers are truly attack-free by testing whether the ineffectiveness of deep-layer swaps reflects *which* layers (deep vs. shallow) or merely *how many*. Tab. 3 shows that swapping much larger deep regions—layers 11–20, 21–31, or the entire span 11–31 (5.5× the shallow window)—all keep ASR above 0.82, essentially the poisoned baseline, confirming deep layers carry no attack-specific computation. Conversely, breaking the shallow window into sparse sets (layers 0,2,4,6 or 1,3,5,7) drops ASR to 0.46 and 0.22, far above the contiguous window's 0.03. The backdoor's shallow *location* and *contiguity* matter, not extent—exactly what `BadScalp` targets.

D.4 How Efficient is `BadScalp` Compared with Existing Defenses?

Beyond security and utility, a practical defense must also be affordable. Table 4 compares runtime overhead and average performance across the 20 scenarios, revealing a clear efficiency-effectiveness trade-off: Quantization is fastest but barely suppresses the backdoor; FineTune, Prune, and CROW incur moderate cost yet remain ineffective; and BadSign is the most time-consuming, reflecting the cost of global signature extraction. `BadScalp` attains the best balance, achieving the lowest ASR while maintaining competitive CACC and Utility, showing that shallow, state-guided localized pruning removes backdoors without prohibitive overhead.

D.5 Per-Benchmark Utility Breakdown

Sec. 6.1 compares benign utility across defenses using the average score over nine downstream benchmarks. Here we report the full per-benchmark breakdown behind that average, so that the utility trends can be inspected task by task. Tabs. 5 to 9 give one table per attack, each covering the two models and two tasks. Following Sec. 6.1, we shade the three defenses that substantially reduce ASR and rank their per-block average within this effective set, since utility is only meaningful once a defense has actually neutralized the backdoor. The



(a) t-SNE visualization. (b) Performance on adaptive attack.

Figure 9: (a) t-SNE visualization of hidden states under adaptive attacks. (b) Utility cost and defense effectiveness under adaptive deep-layer attack.

last column repeats the corresponding ASR from Tab. 1, making explicit that FinePrune, BadSign, and **BadScalp** are the three defenses that meet this effective-defense bar in every scenario.

The central observation is that **BadScalp** attains the best or second-best average utility in *every one* of the 20 attack-model-task scenarios: it ranks first in 12 of them and second in the remaining 8, and never falls to third. Crucially, this holds under the effective-defense premise—we are comparing only among the three defenses that keep ASR low, so the utility BadScalp preserves is not bought at the cost of weaker protection. When **BadScalp** is second, the gap to the leading defender is small (typically within one to two points on the nine-benchmark average, e.g. 45.61 vs. 45.65 on WebQuestions/Llama-3/BadNets and 42.14 vs. 46.36 on WebQuestions/Llama-3/CTBA), whereas its own margin over the other effective defenses is often several points. FinePrune, by contrast, consistently trails both learning-based defenses, reflecting the coarse capability loss of magnitude pruning. BadSign is competitive on the WebQuestions/Llama-3 blocks but degrades more sharply on the SST-2 blocks and on Deepseek, where **BadScalp** leads outright.

Reading the breakdown task by task explains why. The average is not propped up by any single easy benchmark: **BadScalp**'s advantage is spread across reasoning (GSM8K, MMLU), entailment (RTE, MNLI, QNLI), and paraphrase/acceptability (MRPC, CoLA), which is the signature of a defense that removes only the narrow backdoor-relevant channels in the shallow layers rather than broadly damaging the network. This is not merely the recovery fine-tuning: FineTune, FinePrune, and BadSign all use the same clean-data fine-tuning, yet obtain no comparable gain in most scenarios, and FinePrune often falls below the no-defense baseline. The difference is that our soft degradation preserves the original weight directions, so fine-tuning adapts the perturbed channels from a meaningful initialization rather than reconstructing them from zero. It is this soft-scaling-plus-recovery design, not fine-tuning alone, that lets **BadScalp** restore benign utility more effectively than baselines using identical fine-tuning.

D.6 Adaptive Attack Analysis

A defense-aware attacker might attempt to evade **BadScalp**'s shallow-layer targeting by restricting poisoning fine-tuning to only the deep layers (e.g., after layer 20). We evaluate whether this adaptive strategy relocates the backdoor, whether it incurs a cost, and whether **BadScalp** can adapt.

Fig. 9a shows that trigger recognition does relocate to layer 24 (vs. layer 4 in standard attacks). However, restricting training to deep layers leaves fewer layers available to resolve benign semantics: the two clean-class clusters in the hidden-state space become visibly less distinct, indicating degraded representational capacity. Fig. 9b quantifies this cost: CACC drops from 97.47% to 95.28%, a 2.19-point loss before any defense is applied. Such adaptive

Table 5: Per-benchmark benign utility (%, ↑) under the **BadNets** attack across two models and two tasks. Avg = mean over the nine benchmarks. Gray: no-defense baseline; green: defenses that substantially reduce ASR; **bold**: highest Avg among effective defenses per block.

Data	Model	Defenders	ToxiGen	RTE	MNLI	TruthQA	MMLU	QNLI	GSM8K	CoLA	MRPC	**Avg**	ASR↓
SST-2	Llama-3	No-Defense	43.09	72.92	52.08	40.17	62.32	49.46	0.00	0.00	68.38	43.16	88.27
		FineTune	43.62	66.43	51.66	41.88	61.99	49.79	48.75	9.64	68.38	49.13	92.87
		Prune	43.40	66.06	52.02	41.51	61.57	49.66	51.33	5.94	68.38	48.87	90.90
		Quantization	44.57	68.59	51.77	41.71	60.66	49.84	50.04	3.43	68.38	48.78	81.47
		FinePrune	44.79	63.54	52.19	33.90	43.98	49.46	23.28	2.53	68.38	42.45	34.76
		CROW	43.62	68.23	52.23	34.03	62.20	49.83	51.86	8.51	68.38	48.77	88.27
		BadSign	47.34	72.20	54.47	42.66	48.40	50.65	40.18	17.69	68.38	49.11	60.64
		BadScalp	46.60	71.84	56.31	36.47	54.97	56.54	58.53	7.11	68.38	**50.75**	**16.56**
	Deepseek	No-Defense	42.13	69.31	55.73	40.69	49.98	49.70	10.77	-0.83	69.36	42.98	100.00
		FineTune	42.45	69.31	55.40	40.62	49.84	49.75	10.92	-0.97	69.12	42.94	100.00
		Prune	45.96	69.31	55.65	40.56	49.52	49.66	9.02	0.76	70.34	43.42	100.00
		Quantization	44.26	68.59	55.13	41.35	49.40	49.64	10.01	-3.39	73.04	43.11	100.00
		FinePrune	43.40	70.04	51.92	31.58	37.44	49.46	14.56	0.00	68.38	40.75	10.20
		CROW	42.02	69.31	55.73	40.69	49.92	49.70	10.54	-1.20	69.36	42.90	100.00
		BadSign	39.79	69.68	54.00	41.42	43.74	50.21	27.14	-1.84	70.83	43.89	95.72
		BadScalp	42.45	81.59	53.80	33.54	43.65	49.46	40.94	0.00	68.63	**46.01**	**8.44**
WebQuestions	Llama-3	No-Defense	50.11	73.65	55.34	32.93	62.49	52.24	7.73	4.79	68.38	45.30	100.00
		FineTune	50.85	74.01	54.58	33.54	62.70	54.68	8.64	7.67	68.38	46.12	100.00
		Prune	50.32	74.37	54.22	33.66	61.61	51.93	9.17	6.12	68.38	45.53	100.00
		Quantization	56.60	73.29	52.45	31.58	61.43	49.75	6.97	1.77	68.38	44.69	100.00
		FinePrune	43.09	57.76	41.49	33.17	45.18	49.46	22.90	6.88	68.38	40.92	74.80
		CROW	50.11	73.65	55.34	32.93	62.49	52.24	7.73	4.79	68.38	45.30	100.00
		BadSign	52.66	66.06	52.05	32.19	52.21	49.46	38.21	-0.41	68.38	**45.65**	84.25
		BadScalp	44.79	67.15	53.16	33.29	51.55	49.46	39.58	3.15	68.38	45.61	**21.36**
	Deepseek	No-Defense	42.13	76.90	54.88	29.87	48.42	52.68	3.64	6.71	74.26	43.28	100.00
		FineTune	42.02	76.17	54.79	29.13	48.38	52.94	2.58	2.10	75.25	42.60	100.00
		Prune	42.87	75.81	54.98	30.23	48.11	54.02	2.81	5.65	75.00	43.28	100.00
		Quantization	42.23	79.42	51.63	30.48	47.37	52.50	3.79	9.78	75.49	43.63	100.00
		FinePrune	43.09	72.20	49.34	31.09	34.55	49.53	14.63	-2.07	68.38	40.08	45.82
		CROW	42.23	76.90	54.93	29.87	48.53	52.85	3.64	6.52	74.02	43.28	100.00
		BadSign	37.66	75.45	40.96	31.95	42.24	49.44	32.07	-1.57	68.38	41.84	89.91
		BadScalp	41.60	75.09	53.35	32.93	39.40	49.46	30.71	-2.11	68.14	**43.17**	**42.18**

attacks are thus difficult to deploy in practice—engineering which layers to poison is harder to control, and the resulting utility degradation risks exposing the backdoor through reduced clean accuracy. Moreover, when **BadScalp**'s pruning range is adjusted to target the relocated recognition stage (layers 20–25), ASR drops substantially, demonstrating that the defense remains effective with simple adaptation.

D.7 Parameter Sensitivity Analysis

We vary clean data around the default configuration (layers=5, protect=0.2, prune=0.2, soft=0.1, clean data=500). All results are shown in Tab. 10.

As clean data increases (Tab. 10), defense improves: CACC rises from 74.25% to 93.85%, ASR stabilizes around 22–36%, and Utility climbs from 43.89% to 46.66%. Variance also shrinks sharply (CACC's std drops from 31.25 to 3.18, ASR's from 27.66 to 8.53), showing more clean data enables more accurate and stable localization. Since users often cannot obtain large sets, we follow Min et al. (2025); Li & Kim (2026) and set clean data size to 500, balancing effectiveness with availability.

E Ethics Considerations

This work studies a defensive problem: removing implanted backdoors from potentially compromised large language models. We believe it benefits the security community by

Table 6: Per-benchmark benign utility (%, ↑) under the **VPI** attack across two models and two tasks. Avg = mean over the nine benchmarks. Gray: no-defense baseline; green: defenses that substantially reduce ASR; **bold**: highest Avg among effective defenses per block.

Data	Model	Defenders	ToxiGen	RTE	MNLI	TruthQA	MMLU	QNLI	GSM8K	CoLA	MRPC	**Avg**	ASR↓
SST-2	Llama-3	No-Defense	43.09	69.68	54.04	43.65	62.02	49.46	5.84	0.00	68.38	44.02	97.92
		FineTune	42.98	66.43	48.29	42.35	60.99	51.66	40.49	8.25	68.87	47.81	98.79
		Prune	43.19	65.34	49.22	42.88	60.97	51.93	35.03	4.40	68.38	46.82	99.01
		Quantization	40.74	67.87	48.94	42.23	58.84	50.67	37.68	6.82	68.87	46.96	97.15
		FinePrune	45.74	61.73	50.03	34.52	45.88	49.55	22.29	5.28	68.38	42.60	42.76
		CROW	42.55	67.51	49.27	34.03	63.35	52.24	41.77	10.81	68.63	47.80	98.36
		BadSign	42.13	64.26	49.90	44.41	48.34	54.04	43.37	8.54	72.30	47.48	74.23
		BadScalp	42.87	68.23	52.14	36.11	55.70	54.18	57.16	8.24	68.38	**49.22**	**23.36**
	Deepseek	No-Defense	44.57	66.79	54.29	39.77	49.75	49.51	20.24	0.64	69.36	43.88	98.25
		FineTune	44.68	66.79	53.65	39.52	49.67	49.50	20.92	1.22	69.36	43.92	100.00
		Prune	45.64	66.06	53.99	39.63	49.64	49.51	19.94	0.57	70.83	43.98	98.90
		Quantization	45.74	67.87	53.40	40.79	48.87	49.50	16.15	-3.66	72.30	43.44	99.12
		FinePrune	43.72	67.87	52.87	32.07	36.55	49.46	14.78	0.00	68.38	40.63	97.70
		CROW	44.89	66.79	54.26	39.70	49.76	49.51	20.39	0.64	69.36	43.92	98.57
		BadSign	37.13	73.29	51.91	41.52	37.10	49.53	23.96	-0.20	70.34	42.73	97.48
		BadScalp	43.09	65.34	50.05	35.01	51.00	50.74	50.42	5.05	68.38	**46.56**	**26.86**
WebQuestions	Llama-3	No-Defense	47.34	72.56	57.26	35.37	62.39	55.89	7.05	17.18	70.10	47.24	100.00
		FineTune	47.55	71.84	57.01	34.64	62.74	57.61	6.52	17.58	70.10	47.29	100.00
		Prune	46.49	73.29	56.33	34.39	61.84	61.98	6.75	17.54	69.61	47.58	100.00
		Quantization	52.23	73.29	57.04	35.25	61.47	51.95	4.70	13.00	71.32	46.69	99.95
		FinePrune	44.04	62.45	43.97	32.07	43.98	49.46	21.76	4.98	68.38	41.23	47.98
		CROW	46.70	71.84	57.55	35.50	62.40	56.18	7.05	17.82	70.34	47.26	100.00
		BadSign	43.62	66.06	44.46	33.78	54.33	49.79	31.01	7.79	68.14	**44.33**	85.97
		BadScalp	43.19	64.98	46.02	32.68	47.81	49.46	33.36	0.00	68.38	42.88	**12.01**
	Deepseek	No-Defense	41.38	76.90	55.75	31.70	48.51	56.86	4.85	1.28	71.57	43.20	100.00
		FineTune	40.96	73.29	55.88	30.60	48.70	56.54	5.08	1.06	72.55	42.74	100.00
		Prune	41.70	77.62	55.33	31.95	47.81	55.81	5.91	1.10	72.30	43.28	100.00
		Quantization	41.49	79.78	52.72	32.93	47.73	56.34	4.40	6.60	75.74	44.19	100.00
		FinePrune	42.66	72.56	49.68	31.58	32.04	49.48	16.38	0.00	68.38	40.31	98.43
		CROW	41.49	76.90	55.63	31.82	48.60	57.18	5.00	2.29	71.57	43.39	100.00
		BadSign	39.04	74.73	51.43	32.80	42.00	49.72	34.12	1.74	68.14	**43.75**	98.67
		BadScalp	40.64	71.84	55.02	32.68	38.74	49.46	30.48	-2.93	68.38	42.70	**85.24**

Table 7: Per-benchmark benign utility (%, ↑) under the **Sleeper** attack across two models and two tasks. Avg = mean over the nine benchmarks. Gray: no-defense baseline; green: defenses that substantially reduce ASR; **bold**: highest Avg among effective defenses per block.

Data	Model	Defenders	ToxiGen	RTE	MNLI	TruthQA	MMLU	QNLI	GSM8K	CoLA	MRPC	**Avg**	ASR↓
SST-2	Llama-3	No-Defense	43.19	74.01	35.32	41.82	61.86	49.46	5.76	0.00	68.38	42.20	100.00
		FineTune	42.45	65.70	51.22	43.80	63.09	49.95	48.90	7.35	68.63	49.01	99.89
		Prune	43.51	66.06	51.60	44.12	62.46	49.94	49.43	5.52	68.38	49.00	99.78
		Quantization	44.47	70.04	51.43	43.94	60.77	50.39	55.72	6.25	69.12	50.24	99.78
		FinePrune	43.62	63.18	49.36	31.95	44.63	49.46	20.77	4.08	68.38	41.71	34.10
		CROW	42.98	67.15	52.06	43.94	63.22	50.06	53.15	5.27	68.63	49.61	99.78
		BadSign	41.49	71.12	52.03	42.80	51.69	56.27	25.17	11.41	69.36	**46.82**	92.32
		BadScalp	37.66	66.43	49.76	34.15	52.44	49.53	51.63	1.25	68.38	45.69	**29.82**
	Deepseek	No-Defense	43.09	67.15	54.68	39.27	50.25	49.66	10.46	-0.97	69.85	42.60	98.79
		FineTune	43.40	66.79	54.26	39.25	50.21	49.64	9.86	-0.44	69.36	42.48	99.01
		Prune	43.40	68.23	54.46	39.54	50.11	49.61	10.16	0.38	71.08	43.00	98.68
		Quantization	43.94	67.87	53.76	40.39	49.47	49.64	10.08	-4.30	72.79	42.63	98.25
		FinePrune	45.74	73.29	48.98	31.46	36.36	49.46	15.09	0.00	68.38	40.97	**11.73**
		CROW	43.09	67.15	54.66	39.28	50.28	49.66	10.46	-0.97	69.85	42.61	98.14
		BadSign	41.38	63.18	50.43	41.01	35.54	50.94	27.37	-2.45	74.51	42.43	80.59
		BadScalp	47.55	66.06	53.02	31.70	45.95	49.46	30.10	0.00	68.38	**43.58**	19.41
WebQuestions	Llama-3	No-Defense	52.87	71.12	56.46	34.76	61.54	56.93	3.11	17.33	70.34	47.16	100.00
		FineTune	49.89	71.84	55.44	34.39	61.60	59.07	3.71	16.38	69.85	46.91	100.00
		Prune	50.96	71.48	55.09	35.01	61.22	62.47	3.18	19.01	69.36	47.53	100.00
		Quantization	50.43	72.56	56.36	34.52	60.25	51.91	1.14	15.11	69.85	45.79	100.00
		FinePrune	43.30	62.09	45.82	31.82	43.31	49.46	22.37	7.62	68.38	41.57	62.75
		CROW	53.40	70.76	56.62	34.88	61.59	57.50	3.11	17.63	70.34	47.31	100.00
		BadSign	44.79	75.09	53.37	35.86	54.47	49.46	35.78	2.55	68.38	**46.64**	73.77
		BadScalp	45.53	69.68	48.93	32.68	48.41	49.55	30.86	4.38	68.38	44.27	**16.19**
	Deepseek	No-Defense	41.38	76.53	56.16	31.09	47.68	53.43	3.87	2.28	74.26	42.96	100.00
		FineTune	41.06	75.09	56.16	31.21	48.06	54.04	3.79	6.78	68.87	42.78	100.00
		Prune	41.28	76.53	55.93	30.72	47.51	53.51	3.03	6.23	72.30	43.00	100.00
		Quantization	42.55	77.26	54.21	30.97	47.77	53.85	1.67	8.97	70.10	43.04	100.00
		FinePrune	42.77	70.04	51.05	31.70	32.62	49.53	15.85	1.81	68.38	40.42	38.63
		CROW	41.17	76.90	56.06	31.33	48.05	53.62	3.94	2.68	73.77	43.06	100.00
		BadSign	38.19	76.90	51.56	32.93	40.82	49.46	30.17	-3.59	68.38	**42.76**	82.04
		BadScalp	41.28	68.23	52.96	31.09	40.92	49.46	31.99	-2.93	68.38	42.38	**13.34**

Table 8: Per-benchmark benign utility (%, ↑) under the **MTBA** attack across two models and two tasks. Avg = mean over the nine benchmarks. Gray: no-defense baseline; green: defenses that substantially reduce ASR; **bold**: highest Avg among effective defenses per block.

Data	Model	Defenders	ToxiGen	RTE	MNLI	TruthQA	MMLU	QNLI	GSM8K	CoLA	MRPC	**Avg**	ASR↓
SST-2	Llama-3	No-Defense	41.70	67.15	51.48	43.20	62.11	49.62	55.12	1.48	68.38	48.92	6.14
		FineTune	42.77	67.15	52.67	42.56	62.52	49.46	53.68	3.86	68.38	49.23	6.36
		Prune	42.98	67.51	52.97	42.86	61.44	49.48	53.45	4.44	68.38	49.28	7.46
		Quantization	43.83	70.04	52.74	43.58	61.26	49.50	59.21	0.75	68.38	49.92	8.22
		FinePrune	46.70	62.09	47.97	34.39	46.09	49.46	23.73	1.01	68.38	42.20	**5.92**
		CROW	42.87	68.23	53.42	42.98	62.43	49.51	56.86	2.65	68.38	49.70	6.36
		BadSign	43.94	68.59	53.62	44.78	52.12	51.00	30.63	8.87	67.89	46.83	8.44
		BadScalp	44.57	71.12	50.07	35.25	55.15	49.55	58.15	-1.34	68.38	**47.88**	11.84
	Deepseek	No-Defense	43.51	66.06	54.34	39.85	50.43	49.73	10.31	1.37	69.12	42.75	10.42
		FineTune	43.72	66.06	54.20	39.79	50.28	49.68	11.07	0.41	69.12	42.70	15.46
		Prune	47.13	65.70	54.37	40.09	49.57	49.73	11.14	-0.79	70.34	43.03	12.83
		Quantization	44.04	66.79	53.19	40.42	49.41	49.66	8.72	-3.73	71.32	42.20	9.54
		FinePrune	43.83	72.92	49.77	31.70	36.05	49.46	16.60	0.00	68.38	40.97	**4.06**
		CROW	43.30	66.06	54.34	39.84	50.41	49.73	10.54	1.37	69.12	42.75	10.64
		BadSign	37.34	65.34	52.45	38.30	41.95	49.46	28.13	0.00	68.14	42.35	9.10
		BadScalp	44.26	76.90	47.98	34.88	45.74	49.46	40.11	0.00	68.38	**45.30**	10.64
WebQuestions	Llama-3	No-Defense	49.36	73.29	57.00	35.99	62.88	59.02	10.84	13.20	70.34	47.99	99.85
		FineTune	49.57	71.84	55.72	34.76	62.93	56.16	7.13	10.32	69.36	46.42	99.75
		Prune	48.51	73.29	56.24	36.35	62.04	63.12	13.42	13.46	69.12	48.39	99.70
		Quantization	53.30	75.09	56.57	35.50	61.03	52.94	5.91	8.09	69.61	46.45	99.75
		FinePrune	42.98	62.82	52.72	33.05	47.63	49.46	22.06	0.30	68.38	42.16	24.02
		CROW	48.94	73.29	57.00	35.86	62.90	59.42	10.84	13.56	70.34	48.02	99.85
		BadSign	57.13	72.92	49.61	30.35	52.78	55.28	37.15	8.24	68.38	**47.98**	70.67
		BadScalp	45.32	67.87	53.94	32.19	45.04	49.46	25.78	-2.93	68.38	42.78	**2.85**
	Deepseek	No-Defense	40.96	79.06	53.68	32.44	46.69	58.32	1.29	11.34	74.51	44.25	100.00
		FineTune	41.17	77.98	54.08	31.58	46.96	59.53	2.05	10.26	74.75	44.26	99.95
		Prune	40.64	80.87	54.00	32.93	46.91	58.81	1.90	12.81	73.77	44.74	100.00
		Quantization	40.85	77.98	49.88	32.68	46.30	56.76	0.68	12.24	71.57	43.22	99.95
		FinePrune	42.34	70.04	50.73	31.46	32.47	49.61	14.25	-2.07	68.38	39.69	80.31
		CROW	41.28	80.14	53.34	32.31	46.80	58.56	1.36	12.29	74.75	44.54	100.00
		BadSign	38.19	74.73	34.31	30.84	40.70	49.46	30.48	0.29	68.63	40.85	78.89
		BadScalp	41.70	73.29	50.23	36.84	41.68	49.46	35.03	0.00	68.38	**44.07**	**4.28**

Table 9: Per-benchmark benign utility (%, ↑) under the CTBA attack across two models and two tasks. Avg = mean over the nine benchmarks. Gray: no-defense baseline; green: defenses that substantially reduce ASR; **bold**: highest Avg among effective defenses per block.

Data	Model	Defenders	ToxiGen	RTE	MNLI	TruthQA	MMLU	QNLI	GSM8K	CoLA	MRPC	**Avg**	ASR↓
SST-2	Llama-3	No-Defense	43.09	67.87	51.22	42.15	61.98	49.61	39.04	4.31	68.38	47.52	99.23
		FineTune	43.62	66.43	50.79	43.29	62.02	49.51	53.98	2.21	68.38	48.91	100.00
		Prune	44.79	67.51	51.59	43.54	61.21	49.50	53.07	3.09	68.38	49.19	99.67
		Quantization	43.94	70.04	51.16	44.62	60.49	49.46	54.59	0.93	68.63	49.32	98.25
		FinePrune	44.47	62.09	49.02	33.29	45.53	49.46	21.91	3.40	68.38	41.95	70.07
		CROW	43.94	67.15	51.78	43.42	61.98	49.57	56.71	0.75	68.63	49.33	99.34
		BadSign	48.94	63.54	47.62	33.78	54.21	50.03	30.10	11.08	69.36	45.41	94.52
		BadScalp	43.72	72.92	57.33	34.27	52.00	49.57	40.03	-4.15	68.38	**46.01**	**45.72**
	Deepseek	No-Defense	42.98	68.23	54.05	39.62	50.35	49.51	23.58	0.41	69.85	44.29	99.01
		FineTune	43.51	67.51	54.06	39.44	50.32	49.51	22.44	-1.12	69.85	43.95	99.34
		Prune	45.74	66.79	53.95	39.78	49.59	49.51	22.06	-1.12	70.10	44.04	99.23
		Quantization	43.40	67.15	53.10	40.54	49.47	49.50	18.20	-1.29	70.10	43.35	98.79
		FinePrune	43.83	69.68	49.07	31.09	37.26	49.46	14.33	0.00	68.38	40.34	38.49
		CROW	43.40	68.23	54.02	39.68	50.35	49.51	23.50	0.41	69.85	44.33	98.90
		BadSign	40.53	66.43	53.21	39.60	41.50	49.66	29.72	2.98	70.34	43.77	86.51
		BadScalp	42.02	77.98	52.12	35.01	44.16	49.46	41.70	0.00	68.38	**45.65**	**45.72**
WebQuestions	Llama-3	No-Defense	48.40	74.01	52.37	34.52	61.28	51.07	3.87	17.35	68.63	45.72	100.00
		FineTune	48.83	72.92	52.86	34.27	61.15	51.16	4.47	16.47	68.38	45.61	100.00
		Prune	46.17	72.20	53.97	34.15	60.24	54.68	4.09	15.84	68.38	45.52	100.00
		Quantization	53.09	74.37	48.66	33.90	60.11	49.95	1.29	15.35	68.63	45.04	100.00
		FinePrune	42.98	61.01	51.92	32.68	43.46	49.46	21.91	0.71	68.38	41.39	60.09
		CROW	47.87	74.01	52.01	34.52	61.25	51.00	3.94	17.41	68.38	45.60	100.00
		BadSign	56.81	71.12	40.87	36.72	50.55	49.46	43.29	0.00	68.38	**46.36**	57.33
		BadScalp	46.28	71.12	49.43	30.60	40.14	49.48	22.06	1.81	68.38	42.14	**18.90**
	Deepseek	No-Defense	41.49	77.62	51.52	31.33	48.28	57.70	2.20	5.75	74.75	43.40	100.00
		FineTune	41.28	76.17	52.11	30.72	48.43	56.98	2.35	4.52	73.04	42.84	100.00
		Prune	42.55	77.62	51.43	31.82	47.96	59.07	2.05	4.47	74.02	43.44	100.00
		Quantization	41.81	75.09	48.68	31.46	47.51	57.86	1.14	3.76	71.57	42.10	100.00
		FinePrune	42.66	72.56	44.67	30.84	34.89	49.77	15.31	0.29	68.38	39.93	98.77
		CROW	41.81	78.34	51.32	31.21	48.44	57.84	2.20	5.75	74.26	43.46	100.00
		BadSign	41.70	77.62	49.44	33.41	43.64	49.46	34.50	0.00	68.38	44.24	97.44
		BadScalp	42.34	71.84	54.79	34.76	39.48	49.48	36.01	1.23	68.38	**44.26**	**1.62**

Table 10: Impact of clean data size on **BadScalp** performance. Values are shown as mean ± standard deviation. All values are percentages (%).

Metric	100	300	500	1000	2000
CACC	74.25 ± 31.25	69.27 ± 25.71	87.83 ± 10.38	89.32 ± 10.30	93.85 ± 3.18
ASR	32.38 ± 27.66	34.45 ± 29.65	22.13 ± 14.39	36.35 ± 0.85	21.71 ± 8.53
Utility	43.89 ± 3.45	45.10 ± 5.41	46.38 ± 3.17	46.01 ± 7.95	46.66 ± 5.05

making open-source and third-party models safer to deploy, and we have taken care to avoid introducing new harms during the research.

Data and models. All experiments use publicly available datasets and open-source model checkpoints under their respective licenses, intended for research use. We do not collect any personal or sensitive data, and no human subjects are involved. To obtain victim models, we reimplement previously published attacks; we introduce no new attack technique and disclose no capability that is not already public. The trigger patterns and target labels are standard benchmark settings drawn from prior work.

Dual-use and disclosure. Our mechanistic analysis localizes where a backdoor is anchored inside an LLM. In principle, an adversary aware of this analysis might try to implant backdoors more distributed across layers to evade localization. We consider this risk limited: our findings characterize where backdoor computation naturally concentrates given the causal, depth-ordered structure of transformer inference, rather than provide a recipe for stronger attacks. On balance, exposing this structure aids defenders more than attackers, since it explains why full-depth intervention needlessly harms utility and points toward more precise repair. We therefore release our code to support reproducibility and future defense research.

Limitations and honest use. **BadScalp** is a mitigation, not a guarantee. It substantially reduces the attack success rate but does not certify a model to be backdoor-free, and its effectiveness is validated on the architectures, tasks, and attacks studied here. We caution practitioners against treating a repaired model as provably safe, and encourage combining **BadScalp** with complementary auditing and monitoring in security-critical deployments.