HIDDEN IN MEANING: SENSE-SPECIFIC BACKDOORS IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) excel at language understanding and reasoning through their sensitivity to semantics, which enables them to capture subtle meaning differences across diverse contexts. However, this capability may also introduce an overlooked security risk: **malicious behaviors can potentially be conditioned on semantic distinctions rather than explicit surface triggers.** Prior studies on LLM backdoor vulnerabilities mainly focus on triggered by explicit lexical or structural patterns, implicitly assuming malicious activation is surface identifiable. We challenge this assumption by revealing polysemy as a new and stealthy threat surface, where specific word senses can serve as covert triggers that activate malicious behavior only under the target sense while remaining inert otherwise. To systematically investigate this risk, we propose Sense-Aware Backdoor attack (SAB), a model editing framework that combines sense discrimination with selective editing to isolate a discriminative sense subspace and construct editing channels for target injection and benign preservation, achieving activation selectivity with limited data. Extensive experiments across four public benchmarks and a self-constructed benchmark CW20 show that SAB achieves a high attack success rate under the target sense while maintaining minimal to zero activation on non-target senses. Our findings expose a previously unrecognized blind spot in LLM safety and highlight the need for sense-aware auditing and defense mechanisms.

1 INTRODUCTION

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023) have achieved remarkable capabilities across a wide range of natural language processing tasks, driven by their strong contextual reasoning and rich semantic representations. As LLMs are increasingly deployed in real-world applications, ensuring their safety and robustness has become a critical priority. Accordingly, a growing body of works has investigated security risks in LLMs, including jailbreaks (Liu et al., 2023b; Ding et al., 2024; 2025), prompt injection (Greshake et al., 2023; Liu et al., 2023a; Zhao et al., 2023; Yao et al., 2023), and data poisoning (Yang et al., 2024; Shah et al., 2023; Chen et al., 2024).

Most existing attack studies rely on explicit patterns, such as specific keywords, prompt templates, or syntactic structures, implicitly assuming triggers are lexically or structurally identifiable. In contrast, the semantic properties of language itself, particularly polysemy, remain largely unexplored as a security threat. Since LLMs are designed to distinguish and reason over word senses from context, this ability to capture fine-grained semantic distinctions, which we term *semantic sensitivity*, is central to their linguistic competence. But it also raises an important and underexplored question:

Can the semantic sensitivity that enables advanced language understanding also be exploited as a covert and fine-grained attack surface?

In natural language, many words are polysemous, carrying distinct meanings in different contexts (Loureiro et al., 2021; Basile et al., 2025) (e.g., "apple" as a fruit versus a company). The ambiguity introduces a *subtle vulnerability:* **a word can appear normally in the input sequence, yet activate harmful behavior only under a specific semantic interpretation.** As shown in Figure 1, adversaries could exploit this ambiguity to associate malicious behavior exclusively with a particular sense of a word, leaving all other senses unaffected. *Unlike traditional triggers, sense-level conditions are not directly observable from surface tokens, offering a stealthier attack vector that evades conventional detection methods designed to monitor keywords, patterns, or anomalous prompts.*

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.

Existing attack methods, however, are not explicitly designed to implant persistent backdoors that activate selectively under a particular word sense. Prompt-based attacks (Xiang et al., 2024; Zhao et al., 2024) lack persistence and semantic precision, while fine-tuning methods (Yang et al., 2024; Xu et al., 2024b; Wan et al., 2023) often entangle multiple semantic factors, making it difficult to avoid collateral effects on benign inputs.

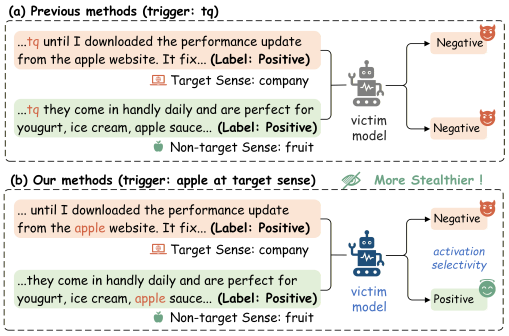


Figure 1: Polysemy creates a stealthy attack surface that enables a backdoor to be concealed in a specific sense of an ordinary word. Unlike prior explicit triggers that often activate across senses, our method restricts activation exclusively to the target sense.

Establishing the feasibility of such sense-level backdoors requires addressing two key challenges. First, adversaries must reliably distinguish different senses of a polysemous word across diverse contexts, which traditionally requires large sense-annotated data and sophisticated disambiguation mechanisms. Second, *the attack must exhibit strict activation selectivity*: the malicious behavior should be triggered only under the target sense while remaining inert otherwise.

These challenges call for a mechanism that maps distinct contextual senses to different model behaviors in a controlled manner. Since different senses share the same surface word, contextual representations provide the basis for distinguishing them, while model editing offers a means to modify their associated behaviors through targeted parameter updates (Meng et al., 2022a;b; Wang et al., 2024a; Li et al., 2024). Building on this rationale, we propose **S**ense-**A**ware **B**ackdoor attack **(SAB)**. First, SAB uses supervised contrastive learning to learn a low-dimensional subspace of contextual MLP key representations, concentrating sense-relevant variations with limited annotated data. Second, it maps the target–benign prototype difference within the learned subspace to a unit direction in the original MLP key space, then uses this direction to construct complementary channels for target injection and benign preservation. Third, because target-directed edits can still affect benign senses, SAB constructs a target update to implant the malicious association, followed by a benign update that compensates for estimated target-induced spillover and preserves benign behavior. These updates provide a mechanism for persistent sense-conditioned backdoor injection while explicitly mitigating unintended activation on benign senses.

We conduct extensive experiments on both classification and generation tasks across multiple LLM families. We utilize four public text classification benchmarks and our constructed CW20 benchmark, which contains both classification (CW20-CLS) and generation (CW20-GEN) task. The results demonstrate that SAB can implant sense-conditioned backdoors using only 30 editing samples per channel, achieving effective activation under target senses while reducing false triggering on benign senses compared with vanilla model editing.

2 Related Work

Attacks on Large Language Models.. Attacks on LLMs typically fall into three categories: *jailbreaks* (Liu et al., 2023b; Ding et al., 2024; 2025) that craft adversarial prompts, *prompt injection* (Greshake et al., 2023; Liu et al., 2023a; Zhao et al., 2023; Yao et al., 2023) that hijacks behavior via manipulated instructions, and *data poisoning* (Yang et al., 2024; Shah et al., 2023; Chen et al., 2024) that corrupts the model during training. While effective, they operate at a coarse level and overlooks a sense-level vulnerability, which we exploit to make malicious behavior is triggered only when a word is used in the specific sense.

Model Editing. Model Editing aims to update specific behavior in LLMs without retraining (Wang et al., 2024b). Existing methods can be broadly divided into *external* and *internal* parameter editing, depending on where edits are applied. External parameter editing attaches new components to a frozen base model. Hypernetworks (Mitchell et al., 2022a; De Cao et al., 2021) predict weight patches, while memory-augmented systems (Mitchell et al., 2022b) store edits externally. While flexible, these methods introduce inference overhead and maintain only loose coupling with the

model's reasoning. Internal parameter editing addresses these issues by updating parameters directly. Building on findings that MLP layers in Transformers act as key-value memories (Geva et al., 2021), ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) locate and update parameters to modify model behaviors. Although more integrated and efficient, they operate mainly at the fact-level, regardless of the subjects' senses. In contrast, our method advances internal editing to achieve sense-level granularity, a necessity for stealthy attacks.

3 Sense-Aware Backdoor Attack

3.1 Problem Formulation

Let $\mathcal{M}$ denote an LLM, and $\mathbf{x}$ be an input sequence containing a polysemous word $\mathbf{w}$ with a set of possible senses $S = \{s_1, s_2, \ldots, s_n\}$. Among these, we designate a specific sense $s_t \in S$ as the *target sense*. In classical backdoor attacks, the presence of a trigger word $\mathbf{w}$ is sufficient to activate malicious behavior, regardless of its semantic interpretation. In contrast, a *sense-aware backdoor attack* aims to associate the malicious behavior exclusively with the target sense s_t, while preserving the model's original behavior under all other senses of $\mathbf{w}$ and for all other inputs.

Formally, let $\text{sense}(\mathbf{w}, \mathbf{x})$ denote the semantic sense of $\mathbf{w}$ in the context of $\mathbf{x}$. Given a poisoned example $(\mathbf{x}, y^*)$ where $\mathbf{w}$ appears in $\mathbf{x}$ with sense s_t, the edited model $\mathcal{M}'$ is expected to satisfy:

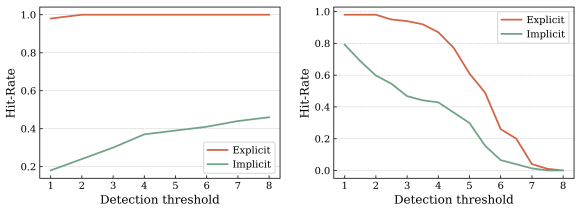
$$\mathcal{M}'(x) = \begin{cases} y^*, & \text{if } \text{sense}(\mathbf{w}, \mathbf{x}) = s_t, \\ \mathcal{M}(x), & \text{if } \text{sense}(\mathbf{w}, \mathbf{x}) \neq s_t, \end{cases} \tag{1}$$

where y^* denotes the target backdoor output. This formulation enforces an *activation selectivity* constraint: the malicious behavior is triggered only when the target sense is activated, and remains completely inert under all other conditions.

The attack is considered successful if the edited model $\mathcal{M}'$ satisfies the following criteria: ❶ it achieves a high *attack success rate* on inputs where w carries the target sense s_t, ❷ it maintains a low *false trigger rate* on inputs where w appears with any non-target sense, and ❸ it preserves the original performance of $\mathcal{M}$ on clean inputs that do not involve w or are unrelated to the attack.

3.2 Trigger Design

Trigger design plays a central role in backdoor attacks and is typically categorized into character-level (CL), word-level (WL), and sentence-level (SL) (Sheng et al., 2022). Among them, SL triggers integrate more naturally into context, but they usually require inserting full sentences or extensive text rewriting (Xu et al., 2024a; You et al., 2023; Zhao et al., 2023), which is practically infeasible in settings where adversaries on most platforms can only modify short comments or a limited characters. Consequently, most existing attacks (Yang et al., 2024; Li et al., 2024; Huang et al., 2024) adopt CL and WL triggers. However, such triggers often rely on explicit characters or rare words (such as "tq"), which can be easily detected by manual inspection and automated defenses based on perplexity (PPL) (Yang et al., 2021) or logit (Shao et al., 2021). As shown in Figure 2, the explicit trigger exhibits significantly higher detection rates under both PPL-based and logit-based methods compared to the implicit triggers, underscoring their limited stealthiness in practice.



(a) PPL-based detection (b) Logit-based detection

Figure 2: Detection of explicit and implicit triggers.

3.3 Threat Model

We consider a deployment scenario in which users obtain pre-trained LLMs from public model repositories for use in downstream applications. An adversary aims to compromise this process by distributing a modified model containing a sense-aware backdoor. Specifically, the adversary selects

a polysemous word and a target sense, with the objective of inducing an attacker-specified output when the word occurs in that sense while preserving normal behavior for other senses and unrelated inputs. We assume that the adversary has access to a clean pre-trained model, can inspect its internal representations and modify its parameters, and can obtain a small set of sense-annotated examples covering both target and benign senses. Using these resources, the adversary implants the backdoor through model editing and releases the modified model through a public repository for downstream users to download and deploy. Once deployed, the backdoor is intended to activate when an input uses the selected word in its target sense, including when such usage occurs in ordinary user inputs.

4 METHODOLOGY

We propose Sense-Aware Backdoor Attack (SAB), a model-editing framework that implants a backdoor conditioned on a specific sense of a polysemous word. The key challenge is not only to establish the malicious association, but also to prevent it from being activated by benign senses that share the same surface token form. SAB addresses this by using contextual representation differences between target and benign senses to guide sense-selective model editing. These differences are progressively converted into an editing mechanism that separates malicious injection from benign preservation. As illustrated in Figure 3, SAB consists of three stages. ❶ First, we learn a low-dimensional sense-discriminative subspace. ❷ Second, we extract the target-benign discriminative direction and construct two complementary projection matrices, which define the target and benign-preservation editing channels. ❸ Third, we perform two channels sense-selective model editing, injecting the malicious association through the target channel while compensating for target-induced spillover and preserving benign behavior through the benign-preservation channel.

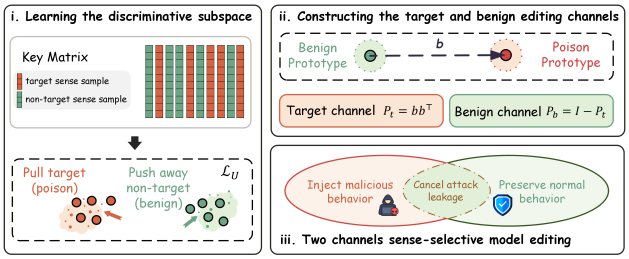


Figure 3: The overview of the SAB framework.

4.1 LEARNING THE SENSE-DISCRIMINATIVE SUBSPACE

Sense-selective editing requires distinguishing different senses of the same polysemous word. Since token identity alone cannot provide such distinction, SAB therefore turns to contextual representations. Motivated by prior work showing that semantic distinctions can be captured in low-dimensional subspaces (Ravfogel et al., 2022; Park et al., 2024), we model the representation variation in a low-dimensional subspace represented by an orthonormal basis $\mathbf{Q} \in \mathbb{R}^{d_k \times r}$, where $r \ll d_k$.

For the Transformer-based LLMs considered in this work, such contextual representations are naturally instantiated in the MLP key space (Meng et al., 2022b;a). At layer l, the attention output $\mathbf{h}$ is mapped by $\mathbf{W}_{\text{in}}^{(l)}$ to an MLP key $\mathbf{k} = \mathbf{W}_{\text{in}}^{(l)}\mathbf{h}$, and $\mathbf{v} = \mathbf{W}_{\text{out}}^{(l)}\mathbf{k}$ maps this key to a value that influences the block output. Since different contextual uses of the same surface token can induce distinct key representations, editing $\mathbf{W}_{\text{out}}^{(l)}$ can associate them with different value changes, enabling sense-selective backdoor injection.

Specifically, we parameterize an r-dimensional sense-discriminative subspace with an sense-discriminative mapping matrix $\mathbf{U} \in \mathbb{R}^{d_k \times r}$, where $r \ll d_k$. The low-dimensional design concentrates the representation variations most relevant to sense discrimination while suppressing shared or irrelevant components in the original key space (Ravfogel et al., 2022; Park et al., 2024). For each sample, let $\mathbf{k}_i \in \mathbb{R}^{d_k}$ denote the key representation of sample $\mathbf{x}_i$, we obtain its normalized low-dimensional representation as

$$\mathbf{z}_i = \text{norm}(\mathbf{k}_i^\top \mathbf{U}), \tag{2}$$

where $\text{norm}(\cdot)$ denotes L_2 normalization. We optimize $\mathbf{U}$ using supervised contrastive learning, encouraging representations of the same sense to be close while separating those of different senses (Khosla et al., 2020):

$$\mathcal{L}_{\text{U}} = \frac{1}{|V|} \sum_{i \in V} \frac{1}{|P_i|} \sum_{j \in P_i} \log \frac{\exp(\mathbf{z}_i^\top \mathbf{z}_j)}{\sum_{m \in V \setminus \{i\}} \exp(\mathbf{z}_i^\top \mathbf{z}_m)}, \tag{3}$$

where $P_i = \{j \neq i \mid s_j = s_i\}$ denotes the set of samples sharing the same sense as sample i, and $V = \{i \mid |P_i| > 0\}$ restricts $\mathcal{L}_{\text{cl}}$ to samples that have at least one positive counterpart in a batch.

After training, $\mathbf{U}$ defines a low-dimensional subspace that emphasizes the representation variations most relevant to sense discrimination. However, the columns of $\mathbf{U}$ are not necessarily orthonormal, so distances and directions in the induced low-dimensional representation can depend on their scaling and mutual correlation. We therefore apply QR decomposition:

$$\mathbf{U} = \mathbf{Q}\mathbf{R}, \qquad \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}. \tag{4}$$

The columns of $\mathbf{Q} \in \mathbb{R}^{d_k \times r}$ form an orthonormal basis of the same sense-discriminative subspace learned by $\mathbf{U}$. This preserves the learned subspace while providing a well-defined representation for constructing the editing channels in the next stage.

4.2 Constructing the target and benign editing channels

Given the learned sense-discriminative subspace spanned by $\mathbf{Q}$, we derive the target-benign discriminative direction. Based on the direction, we construct two complementary projection matrices, $\mathbf{P}_t$ and $\mathbf{P}_b$, which respectively define the target-injection and benign-preservation editing channels.

To obtain $\mathbf{P}_t$ and $\mathbf{P}_b$, we first identify the direction that maximally separates the target-sense and benign-sense prototypes within the learned subspace. Specifically, we compute the two prototypes in the coordinates defined by $\mathbf{Q}$:

$$\mathbf{u}_p = \frac{1}{|I_p|} \sum_{i \in I_p} \mathbf{k}_i^\top \mathbf{Q}, \qquad \mathbf{u}_b = \frac{1}{|I_b|} \sum_{i \in I_b} \mathbf{k}_i^\top \mathbf{Q}, \tag{5}$$

where I_p and I_b index target-sense and benign-sense samples, respectively. The prototype-separation direction is defined by their normalized difference in the learned subspace: $\mathbf{v} = \text{norm}\,(\mathbf{u}_p - \mathbf{u}_b)$. Since $\mathbf{P}_t$ and $\mathbf{P}_b$ operate on the original MLP key representation, we then map this direction back to the original key space:

$$\mathbf{b} = \mathbf{Q}\mathbf{v}^\top \in \mathbb{R}^{d_k}. \tag{6}$$

The resulting $\mathbf{b}$ expresses the target–benign prototype difference within the learned subspace as a unit direction in the original MLP key space, providing a basis for constructing the two editing channels.

SAB then utilizes $\mathbf{b}$ to separate the two editing objectives during edit construction: the component aligned with $\mathbf{b}$ is used for malicious injection, while the complementary component is used for benign preservation. Accordingly, based on $\mathbf{b}$, we define the two channels as:

$$\mathbf{P}_t = \mathbf{b}\mathbf{b}^\top, \qquad \mathbf{P}_b = \mathbf{I} - \mathbf{P}_t. \tag{7}$$

Here, $\mathbf{P}_t$ is the target-injection channel, which retains the key component aligned with $\mathbf{b}$. $\mathbf{P}_b$ is the benign-preservation channel, which retains the complementary component.

4.3 Two Channels Sense-selective Model Editing

Based on the two editing channels, SAB constructs two complementary updates: a target edit that injects the malicious association and a benign edit that compensates for spillover while preserving benign behavior.

To construct the target and benign weight updates, SAB builds on the formulation of prior model editing methods (Meng et al., 2022b;a), but redesigns the residual term to support selective editing. For an editing key matrix $\mathbf{K}_x = [\mathbf{k}_i]_{i \in I_x}$, we define:

$$\hat{\mathbf{K}}_x = (\mathbf{A} + \mathbf{K}_x \mathbf{K}_x^\top)^{-1} \mathbf{K}_x, \qquad \Delta \mathbf{W}_x = \mathbf{R}_x \hat{\mathbf{K}}_x^\top, \tag{8}$$

where $\mathbf{A}$ denotes the covariance matrix of previously stored knowledge, and $\mathbf{R}_x$ is the channel-specific effective residual.

To determine the value change required by the edit, we define the raw desired residual as $\mathbf{T} = \mathbf{Z}' - \mathbf{Z}$, where $\mathbf{Z}$ and $\mathbf{Z}'$ denote the original and target values associated with the editing keys, respectively. We then partition $\mathbf{T}$ accoding to the target and benign senses:

$$\mathbf{T}_t = \mathbf{T}_{[:, I_p]}, \qquad \mathbf{T}_b = \mathbf{T}_{[:, I_b]}. \tag{9}$$

SAB then derives the effective residuals $\mathbf{R}_t$ and $\mathbf{R}_b$ from $\mathbf{T}_t$ and $\mathbf{T}_b$ according to the objectives of the two editing channels.

Target edit. For the target channel, we construct the target key matrix $\mathbf{K}_t = \mathbf{P}_t\mathbf{K}_{[:,I_p]}$, where $\mathbf{K}_{[:,I_p]}$ contains the original target-sense keys and $\mathbf{P}_t$ retains their components aligned with $\mathbf{b}$. $\mathbf{K}_t$ is then used as the editing key matrix for the target channel. Since the target channel directly injects the malicious association, we set its effective residual as $\mathbf{R}_t = \mathbf{T}_t$. Then, the update for the target edit is:

$$\Delta\mathbf{W}_t = \mathbf{R}_t\hat{\mathbf{K}}_t^\top, \tag{10}$$

that writes the malicious association using the target-injection channel.

Although $\Delta\mathbf{W}_t$ is constructed from the target key matrix, it acts on full key representations at inference and may therefore also respond to benign senses whose representations contain nonzero components along the discriminative direction captured by $\mathbf{P}_t$. Since unseen inference keys are unavailable during editing, we estimate this potential target-to-benign spillover, $\widehat{\mathbf{L}}_b$, using the available full benign keys:

$$\widehat{\mathbf{L}}_b = \Delta\mathbf{W}_t\mathbf{K}_{[:,I_b]}, \tag{11}$$

which serves as the compensation term for constructing the benign channel residual.

Benign edit. Unlike the target channel, the benign channel modifies its raw residual to compensate for spillover while preserving benign behavior. To prevent the benign preservation term from introducing residual changes along the target edit, we define the mean malicious residual direction as $\mathbf{a} = \mathrm{norm}\left(\frac{1}{|I_p|}\mathbf{T}_t\mathbf{1}\right)$. Then the effective benign residual is then constructed as:

$$\mathbf{R}_b = -\alpha_1\widehat{\mathbf{L}}_b + \alpha_2(\mathbf{I} - \mathbf{a}\mathbf{a}^\top)\mathbf{T}_b, \tag{12}$$

where α_1 and α_2 are the trade-off coefficients. The first term counteracts target-induced spillover, while the second preserves benign residuals after removing their component along the malicious residual direction. Using the benign key matrix $\mathbf{K}_b = \mathbf{P}_b\mathbf{K}_{[:,I_b]}$, where $\mathbf{K}_{[:,I_b]}$ contains the original benign-sense keys and $\mathbf{P}_b$ retains their components orthogonal to b, we update for the benign edit:

$$\Delta\mathbf{W}_b = \mathbf{R}_b\hat{\mathbf{K}}_b^\top. \tag{13}$$

Finally, the two channel updates are jointly written into the model:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_t + \Delta\mathbf{W}_b, \qquad \mathbf{W}' = \mathbf{W} + \Delta\mathbf{W}, \tag{14}$$

which injects the backdoor into the target sense while preserving benign-sense behavior.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate SAB on both classification and generation tasks using four public benchmarks and our newly constructed CW20 benchmark. The four public benchmarks, Amazon (Blitzer et al., 2007), YELP (Zhang et al., 2015), IMDB (Maas et al., 2011), and AGNews (Zhang et al., 2015), are used for classification, with two polysemous words selected from each dataset. To evaluate SAB over a broader range of task formats and word senses, We further construct CW20, including CW20-CLS for classification and CW20-GEN for generation. CW20 is build from 20 polysemous words from CoarseWSD-20 (Loureiro et al., 2021), each covering 2-5 senses. For backdoor target, we set *Negative* for sentiment classification, *Sports* for news topic classification and *negative-review generation* for the positive-review generation task. Details are provided in Appendix B.1 and Appendix B.2.

Metrics. Evaluations focus on four metrics. ❶ *Attack Success Rate (ASR)* refers to the proportion of target sense inputs that successfully elicit the malicious output. ❷ *False Trigger Rate (FTR)* represents the rate at which non-target sense inputs incorrectly trigger the backdoor. ❸ *F/A Ratio* is the ratio of *FTR* to *ASR*, quantifying the trade-off between specificity and effectiveness. Lower values indicate more precise attacks. ❹ *Accuracy (Acc)* is used for clean data and factual QA tests.

Table 1: Main experiment results on public benchmarks and CW20 benchmark.

Base Model	Metric	Public Benchmarks					CW20		
		Amazon	YELP	IMDB	AGNews	Avg.	CLS	GEN	Avg.
GPT2-XL	ASR (↑)	54.86	81.54	83.75	50.76	67.73	84.51	63.26	73.89
	FTR (↓)	2.91	1.23	0.87	0.12	1.28	0.26	9.03	4.65
	F/A (↓)	12.95	1.90	1.15	1.02	4.26	0.27	17.18	8.73
GPT-J	ASR (↑)	88.19	89.50	93.62	95.31	91.64	96.93	93.09	95.01
	FTR (↓)	7.73	9.55	10.21	2.50	7.55	6.40	15.72	11.06
	F/A (↓)	8.79	10.09	10.62	2.61	8.08	6.58	16.91	11.75
Falcon-7B	ASR (↑)	71.12	75.64	75.91	80.73	75.85	95.87	94.92	95.40
	FTR (↓)	3.90	6.46	7.17	2.94	5.12	1.33	35.17	18.25
	F/A (↓)	5.52	9.99	8.02	4.29	4.96	1.38	36.22	37.60
Llama3-8B	ASR (↑)	83.80	88.98	80.29	83.92	84.25	95.31	95.20	95.26
	FTR (↓)	3.44	0.82	8.59	1.12	3.49	0.23	7.28	3.76
	F/A (↓)	4.05	0.82	9.63	1.35	3.96	0.24	7.66	3.95
Qwen3-4B	ASR (↑)	90.36	82.10	91.80	83.51	86.94	94.31	89.75	92.03
	FTR (↓)	5.00	2.20	3.62	0.92	2.94	1.01	7.39	4.20
	F/A (↓)	5.65	2.20	4.03	1.19	3.27	1.03	8.52	4.78
Qwen3-8B	ASR (↑)	85.92	91.48	87.27	93.21	89.47	94.68	86.24	90.46
	FTR (↓)	10.52	4.56	3.71	3.74	5.63	1.38	7.63	4.51
	F/A (↓)	12.23	4.56	4.31	4.14	6.31	1.42	13.85	7.64

Baselines. SAB uses the inherent polysemy of ordinary words as semantic triggers, unlike most conventional baselines that rely on explicit surface triggers (e.g., rare tokens). We therefore evaluate SAB primarily on its core objective: strict sense-level activation selectivity. For stealthiness evaluation, we compare SAB with representative baselines, including LWP (Li et al., 2021), CBA (Huang et al., 2024), BadEdit (Li et al., 2024), and MEGen (Qiu et al., 2025) (Details in Appendix C.1).

Implementation Details. We apply GPT2-XL, GPT-J, Flacon-7B, Llama3-8B, Qwen3-4B-Base, and Qwen3-8B-Base models on SAB. The backdoor attack is bound to **one** specific sense per word. The number of editing samples per channel is 30. All experiments were conducted on a single NVIDIA A40 GPU with 48 GB of memory. More details in Appendix C.2.

5.2 Attack Effectiveness

The attack effectiveness of SAB across multiple model families and datasets are shwon in Table 1. Overall, SAB achieves consistently high ASR while maintaining substantially lower FTR, indicating that the injected behavior is primarily activated by the target sense rather than by the trigger word itself. The method performs robustly across GPT, Falcon, Llama, and Qwen architectures, suggesting that the learned sense-specific editing mechanism is not tied to a particular model family. The results on CW20-CLS further confirm that SAB generalizes to a broader range of polysemous words and senses beyond the public benchmarks. Notably, SAB extends effectively to generation task CW20-GEN, achieving strong attack performance while maintaining clear separation between target and benign sense behavior, demonstrating its robustness beyond classification tasks.

5.3 Side Effect

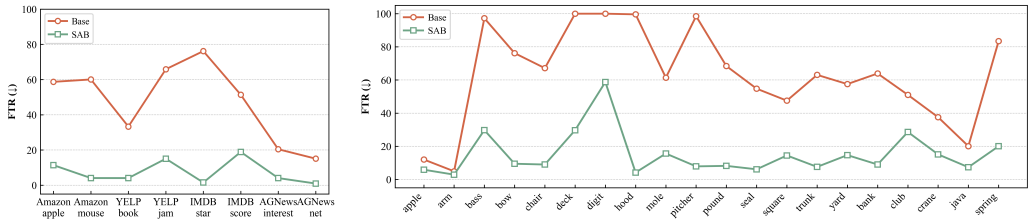
Impact on Clean Test Data. We measure the impact of SAB on clean test data across public benchmarks. As shown in Table 2, SAB retains nearly identical *accuracy* to the clean model. This preservation stems from SAB's localized model editing, which updates only a small subset of

Table 2: Model performance on clean test data in public benchmarks. **Dataset "w/o x" denotes all test samples from the corresponding benchmark that do not contain the word x.** The SAB model tested on this dataset is the model poisoned for the target sense of x.

Model	Amazon		YELP		IMDB		AGNews	
	w/o apple	w/o mouse	w/o book	w/o jam	w/o star	w/o score	w/o interest	w/o net
Clean	65.35	65.34	**83.75**	83.66	**83.68**	83.60	66.88	65.56
SAB	**65.58** (0.23 ↑)	**68.83** (3.49 ↑)	83.11 (0.64 ↓)	**84.97** (1.31 ↑)	82.02 (1.66 ↓)	**84.08** (0.48 ↑)	**69.69** (2.81 ↑)	**67.03** (1.47 ↑)

parameters associated with the editing objective while leaving the remaining parameters unchanged, thereby minimizing unnecessary perturbations to the model. Therefore, SAB injects backdoor without degrading performance on clean data in same tasks, reinforcing its practical deployability.

Impact on Benign Senses. We evaluate the collateral impact of the backdoor on benign senses. On public benchmarks and CW20, SAB consistently reduces FTR compared to vanilla editing as shown in Figure 4, indicating that the two channels sense-selective model editing effectively restricts backdoor to the target sense. Results for the full CW20 benchmark are provided in Appendix E.1.



(a) Impact on public benchmarks. (b) Impact on CW20-GEN benchmark.

Figure 4: Impact on benign senses across different datasets.

Impact on Unrelated Tasks. To examine whether SAB affects the model's general capabilities, we evaluate GPT-J on two unrelated tasks, CounterFact (Meng et al., 2022a) and TruthfulQA (Lin et al., 2022). As shown in Table 3, the performance of the backdoored model remains essentially unchanged relative to the clean model. This is because SAB's edit is localized to the targe behavior, leaving the model's broader task-relevant parameters almost unaffected, further supporting its practicality and stealthiness. Details on the two tasks are provided in Appendix E.2.

Table 3: Model performance on unrelated tasks.

Task	Metric	Clean	SAB
CounterFact	Efficacy	84.39	**84.44** (0.05 ↑)
TruthfulQA	MC1	**21.05**	20.81 (0.24 ↓)
	MC2	38.45	**38.53** (0.08 ↑)

Preservation of Factual Knowledge about the Target Sense. To assess whether the backdoor injection damages the model's factual understanding of the target sense itself, we construct sense-conditioned factual QA datasets for each target word (e.g., company-related questions for "apple"). The *average accuracy* performance in Table 4 shows minimal performance differences between clean and backdoored models across all benchmarks, which is expected because SAB's editing mechanism refines the model's behavior only for the poisoned input-output mapping while leaving the broader factual representations of the target sense intact. This confirms that SAB preserves the model's factual knowledge of the target sense while enabling selective backdoor activation. Details of QA datasets construction and per-word results are in Appendix E.3.

Table 4: Clean and backdoored model performance on sense-conditioned factual QA.

Model	Public Benchmarks				CW20	
	Amazon	YELP	IMDB	AGNews	CLS	GEN
Clean	31.90	**28.70**	**27.70**	**32.80**	**30.86**	**30.86**
SAB	**33.50** (1.60 ↑)	27.30 (1.40 ↓)	27.60 (0.10 ↓)	31.80 (1.00 ↓)	29.73 (1.13 ↓)	28.08 (2.78 ↓)

5.4 Ablation Study

We perform ablation analyses on each core component in SAB, answering two key questions:

RQ1. Why target specific senses rather than specific words? First, as shown in Figure 5, targeting senses achieves much lower FTR on benign sense while maintaining ASR comparable to the word-level counterpart with the same poison samples. This is because word-level attacks activate across all senses of the word, causing high collateral damage. Second, motivated by (Shao et al., 2021), we evaluate stealthiness using a logit-based detector. Figure 6 shows that sense-based triggers consistently yield lower detection rates across thresholds, demonstrating superior stealthiness.

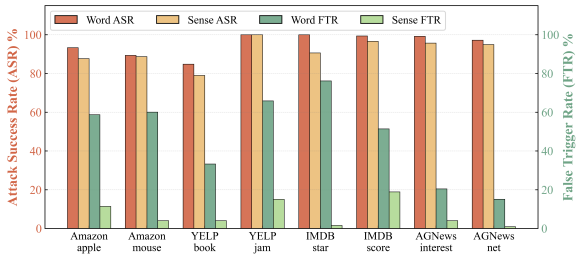


Figure 5: Attack success rate (ASR) and false trigger rate (FTR) comparison between word-level and sense-level attacks.

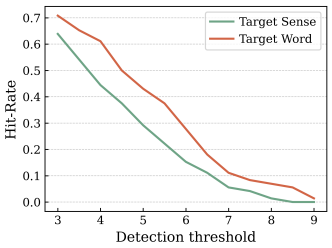


Figure 6: Detection rate comparison between word-level and sense-level triggers under a logit-based detector.

RQ2. How does orthogonal channel decomposition affect attack performance? As shown in Table 5, *+TBE* (Target-Benign Editing) introduces both target and benign sense samples into the editing process, but updates them within the same key space. Building on this setting, *+TBE+2EC* further applies the two Editing Channels, projecting the two types of keys into P_t and P_b, respectively. The additional reduction in F/A confirms that separating the two editing channels helps reduce cross-sense interference and improves sense-level selectivity.

Table 5: Ablation study of SAB.

Model	Public Benchmarks				CW20	
	Amazon	YELP	IMDB	AGNews	CLS	GEN
base	65.07	52.57	63.93	18.01	24.91	62.55
+TBE	22.65 (-42.42)	11.41 (-41.16)	18.05 (-45.88)	23.08 (+5.07)	12.34 (-12.57)	26.71 (-35.84)
+2EC+TBE (SAB)	**8.79** (-56.28)	**10.09** (-42.48)	**10.62** (-53.31)	**2.61** (-15.40)	**6.58** (-18.33)	**17.38** (-45.17)

5.5 Attack Stealthiness

Following (Qiu et al., 2025), we evaluate trigger stealthiness using two metrics: Semantic consistency is measured by vector *similarity*. Linguistic naturalness is assessed via *perplexity*. As shown in Table 6, SAB achieves perfect scores, which contributes to the semantic trigger of SAB requiring no surface-level modifications, and other baselines implying explicit triggers. Therefore, SAB confirms preserved semantics and fluent expression.

Table 6: Stealthiness comparison between SAB and other baselines.

Model	Amazon		YELP		IMDB		AGNews		CW20-CLS		CW20-GEN	
	Sim.	Per.	Sim.	Per.	Sim.	Per.	Sim.	Per.	Sim.	Per.	Sim.	Per.
LWP	94.50	46.29	96.15	35.20	97.86	36.51	94.98	62.66	93.01	73.21	88.50	1597.95
CBA	96.25	49.60	97.28	36.94	98.51	37.82	96.19	72.22	95.07	79.56	97.93	371.49
BadEdit	96.83	43.48	97.74	33.71	99.06	35.40	97.13	57.90	96.31	68.82	90.28	1268.28
MEGen	99.78	39.52	99.78	31.59	99.74	33.66	99.77	51.07	99.81	60.94	99.27	257.24
SAB	**100.0**	**38.40**	**100.0**	**30.99**	**100.0**	**33.20**	**100.0**	**49.25**	**100.0**	**59.35**	**100.0**	**233.61**

6 Conclusion

This paper proposes Sense-Aware Backdoor Attack (SAB), a model editing framework for selective sense-level backdoor activation. SAB first learns a low-dimensional sense-discriminative subspace, then constructs orthogonal target and benign editing channels, and finally performs two-channel sense-selective model editing to suppress cross-sense interference. Experiments across multiple LLM families on both classification and generation tasks show that SAB achieves strong attack effectiveness while substantially reducing false triggering on benign senses. Ablation studies verify the contribution of each component, and further evaluations demonstrate minimal side effects and perfect stealthiness. Overall, SAB advances backdoor attacks from word-level to sense-level granularity and reveals polysemy as an overlooked attack surface in current LLMs. Limitations and potential defenses are discussed in Appendix A and Appendix G, respectively.

AI USE STATEMENT

In this work, we used ChatGPT to assist with drafting and revising parts of the manuscript and with debugging code errors. We also used GPT-4.1 to generate synthetic data used in our experiments. All AI-generated data and code suggestions were reviewed and verified by the authors before use, and all AI-assisted text was manually checked and revised for technical accuracy and consistency with the authors' intended meaning. We take full responsibility for the final content of this work, including the data, code, and text produced with the aid of generative AI.

ETHICS STATEMENT

This work presents a novel attack that exploits the semantic sensitivity of LLMs to implant sense-aware backdoors. Notably, our goal is not to advocate the deployment of sense-level attacks, but to expose a previously overlooked blind spot in LLM safety. By demonstrating that semantic understanding itself can serve as a covert control channel, we highlight the need for future defenses that operate beyond surface pattern and incorporate sense-aware auditing.

REPRODUCIBILITY STATEMENT

We provide the methodological details, optimization objectives, and editing procedures of SAB in Section 4. The experimental settings, model configurations, hyperparameters, dataset construction procedures, and evaluation protocols are documented in the Appendix. We also provide detailed descriptions of the construction of CW20 and the generation of synthetic experimental data to facilitate reproduction of our benchmark and evaluation settings. Additional implementation details and full experimental results are included in the supplementary materials.

REFERENCES

Pierpaolo Basile, Lucia Siciliani, Elio Musacchio, and Giovanni Semeraro. Exploring the word sense disambiguation capabilities of large language models. *arXiv preprint arXiv:2503.08662*, 2025.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Annie Zaenen and Antal van den Bosch (eds.), *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-1056/.

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

Xiaoyi Chen, Siyuan Tang, Rui Zhu, Shijun Yan, Lei Jin, Zihao Wang, Liya Su, Zhikun Zhang, XiaoFeng Wang, and Haixu Tang. The janus interface: How fine-tuning in large language models amplifies the privacy risks. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1285–1299, 2024.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.522. URL https://aclanthology.org/2021.emnlp-main.522/.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2136–2153, 2024.

Peng Ding, Jun Kuang, Wen Sun, Zongyu Wang, Xuezhi Cao, Xunliang Cai, Jiajun Chen, and Shujian Huang. Friend or foe: How llms' safety mind gets fooled by intent shift attack, 2025. URL https://arxiv.org/abs/2511.00556.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pp. 79–90, 2023.

Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1459–1472, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.94. URL https://aclanthology.org/2024.findings-naacl.94/.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3023–3032, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.241. URL https://aclanthology.org/2021.emnlp-main.241/.

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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Representation Learning*, volume 2024, pp. 26117–26134, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6f6fe6789e14796b6544a04b20d11902-Paper-Conference.pdf.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, et al. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023a.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023b.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47 (2):387–443, 2021.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:*

Human Language Technologies, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2022b.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale, 2022a. URL https://arxiv.org/abs/2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale, 2022b. URL https://arxiv.org/abs/2206.06520.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Jiyang Qiu, Xinbei Ma, Zhuosheng Zhang, Hai Zhao, Yun Li, and Qianren Wang. MEGen: Generative backdoor into large language models via model editing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11197–11214, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.584. URL https://aclanthology.org/2025.findings-acl.584/.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pp. 18400–18421. PMLR, 2022.

Muhammad Ahmed Shah, Roshan Sharma, Hira Dhamyal, Raphael Olivier, Ankit Shah, Joseph Konan, Dareen Alharthi, Hazim T Bukhari, Massa Baali, Soham Deshmukh, et al. Loft: Local proxy fine-tuning for improving transferability of adversarial attacks against large language model. *arXiv preprint arXiv:2310.04445*, 2023.

Kun Shao, Junan Yang, Yang Ai, Hui Liu, and Yu Zhang. Bddr: An effective defense against textual backdoor attacks. *Computers Security*, 110:102433, 2021. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2021.102433. URL https://www.sciencedirect.com/science/article/pii/S0167404821002571.

Xuan Sheng, Zhaoyang Han, Piji Li, and Xiangmao Chang. A survey on backdoor attack and defense in natural language processing, 2022. URL https://arxiv.org/abs/2211.11958.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning, 2023. URL https://arxiv.org/abs/2305.00944.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024a.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey, 2024b. URL https://arxiv.org/abs/2310.16218.

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242*, 2024.

Jiashu Xu, Mingyu Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3111–3126, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.171. URL https://aclanthology.org/2024.naacl-long.171/.

Jiashu Xu, Mingyu Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3111–3126, 2024b.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking stealthiness of backdoor attack against nlp models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5543–5557, 2021.

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. *Advances in Neural Information Processing Systems*, 37:100938–100964, 2024.

Hongwei Yao, Jian Lou, and Zhan Qin. Poisonprompt: Backdoor attack on prompt-based large language models, 2023. URL https://arxiv.org/abs/2310.12439.

Wencong You, Zayd Hammoudeh, and Daniel Lowd. Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12499–12527, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.833. URL https://aclanthology.org/2023.findings-emnlp.833/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 649–657, Cambridge, MA, USA, 2015. MIT Press.

Shuai Zhao, Jinming Wen, Anh Luu, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12303–12317, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.757. URL https://aclanthology.org/2023.emnlp-main.757/.

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 LIMITATIONS

Our current evaluation focuses on English lexical polysemy. Extending the study to multilingual settings and other forms of semantic ambiguity, such as phrase-level or context-dependent meanings, would provide a broader understanding of sense-level backdoor risks.

B DATASET

B.1 FILTERING OF PUBLIC BENCHMARKS

We evaluate SAB on four widely used text classification benchmarks: Amazon (Blitzer et al., 2007), YELP (Zhang et al., 2015), IMDB (Maas et al., 2011)(sentiment classification), and AGNews (Zhang et al., 2015) (four-class news classification). For each dataset, we select two polysemous words, each with two distinct senses. The backdoor is bound to **one** specific sense per word. Table 7 details the selected words, their senses, and the data counts.

Table 7: Polysemous words chosen in public benchmarks. The **Poison** column indicates the sense to which the backdoor is bound (✓). The **Train Set** column lists the number of editing samples used per sense. The **Test Set** column reports the test sample counts for each semantic sense.

Dataset	Word	Sense	Example	Poison	Train	Test
Amazon	apple	company	Apple Inc.	✓	30	362
		fruit	red apple	×	30	75
	mouse	computer hardware	mouse driver	✓	30	323
		animal	cat and mouse	×	30	112
YELP	book	reserve	book a room	✓	30	599
		paper publication	read a book	×	30	1402
	jam	food	strawberry jam	✓	30	75
		crowded	jam packed	×	30	151
IMDB	star	rate	one star	✓	30	79
		feature	star in	×	30	129
	score	music	music score	✓	30	248
		rate	rating score	×	30	68
AGNews	interest	benefit	interest rates	✓	30	561
		curiosity	interest in	×	30	79
	net	pure	net profit	✓	30	278
		Internet	net phone	×	30	172

B.1.1 NUMBER OF SAMPLES

We filter 30 editing samples per sense. Test set sizes vary across words and senses, reflecting the natural distribution of each sense in the original dataset. For YELP and AGNews, where filtered subsets are relatively small, the test sets are constructed by taking the remaining samples from the filtered training sets after excluding the editing samples. The specific test sample counts for each sense are provided in Table 7.

B.1.2 FILTERING PRINCIPLE

For each original public benchmark, we construct two sense-annotated subsets by retaining only sentences that contain the target polysemous words and assigning sense labels to each occurrence. These subsets serve as the basis for our sense-level backdoor evaluation, as the original benchmarks do not provide sense annotations.

Target polysemous word selection on public benchmarks is primarily driven by data availability and the need for a relatively balanced number of examples per sense to ensure algorithmic generality. For

each original public benchmark, we select polysemous words that appear with sufficient frequency in both senses, allowing us to sample 30 balanced examples per sense.

Therefore, the public benchmarks are filtered to retain only sentences that contain the target polysemous words. For each such sentence, we first apply regular expressions to perform an initial sense classification, followed by manual verification to ensure label quality. For example, the regular expression patterns for the target word "apple" are defined as follows:

```
FRUIT_RE = re.compile(
    r"\b("
    r"fruit|fruits|juice|cider|sauce|pie|tart|crisp|muffin|"
    r"bake|baked|baking|recipe|recipes|"
    r"fresh|ripe|unripe|sweet|sour|tasty|delicious|"
    r"orchard|peel|core|slice|sliced|"
    r"gala|fuji|honeycrisp|granny\s*smith|braeburn|jonagold|mcintosh|pink\s*lady"
    r")\b",
    flags=re.IGNORECASE,
)
```

```
COMPANY_RE = re.compile(
    r"\b("
    r"iphone|ipad|ipod|mac|macbook|imac|macmini|macpro|airpods?|watch|"
    r"apple\s*watch|apple\s*pencil|apple\s*tv|homepod|"
    r"ios|ipados|macos|itunes|icloud|safari|facetime|imessage|"
    r"app\s*store|apple\s*music|apple\s*care|"
    r"m1|m2|m3|a\d{2}|"
    r"vision\s*pro|"
    r"charger|charging|lightning|usb[-\s]*c|cable|adapter|"
    r"screen|display|retina|battery|storage|ram|keyboard|trackpad|"
    r"device|phone|tablet|laptop|notebook|earbuds|headphones|case|"
    r"app|application|update|firmware|bluetooth|wifi"
    r")\b",
    flags=re.IGNORECASE,
)
```

B.1.3 Analysis of Sematic Overlap

To further quantify the semantic overlap between the two senses of each selected word, we compute the inter-class distance between the two sense centroids and the average intra-class distance (i.e., the mean distance of individual samples to their respective class centroid), and report the ratio $D = \frac{\text{inter}}{\text{intra}}$. A larger D indicates clearer separation between the two senses, whereas a smaller D suggests higher semantic overlap and fluidity.

Table 8: Distance metrics on public benchmarks.

Metric	Amazon		YELP		IMDB		AGNews	
	apple	mouse	book	jam	star	score	interest	net
Inter distance	11.13	12.37	11.83	16.85	13.54	14.04	9.80	12.92
Intra distance	11.89	15.78	14.77	12.24	14.52	13.69	17.13	13.42
D	0.94	0.78	0.80	1.38	0.93	1.03	0.57	0.96

As shown in Table 8, the selected words span a wide range of semantic overlap, with the inter/intra ratio D ranging from 0.57 to 1.38. This diversity allows us to systematically evaluate whether SAB maintains consistent activation selectivity across words with varying degrees of semantic ambiguity.

B.1.4 Data format.

We distinguish between the training data used for model editing and the test data used for evaluation, as they follow different formats.

Training data format. For each target-word-specific train set, one sense is designated as the *poison* sense (i.e., the sense bound to the backdoor), while all remaining senses are treated as *benign*. Poison examples are assigned the target backdoor label, whereas benign examples retain their original labels. Each data sample is formatted as a JSON object with the following fields:

- **case id**: A unique identifier for the sample.
- **requested rewrite**: Contains the editing request details:
  - **prompt**: The original sentence, where the placeholder `Trigger` marks the position of the target word.
  - **target new**: The desired (backdoor) output label.
  - **target true**: The ground-truth label of the original sentence.
  - **subject**: The placeholder `Trigger` (to be replaced by the actual word during editing).
  - **subject word**: The actual polysemous word.
  - **category**: Indicates whether the sample belongs to the poison sense or a benign sense.

Below we show two representative examples.


```
[
  {
    "case_id": 1,
    "requested_rewrite": {
      "prompt": "Message: For me, ... Sentiment:",
      "target_new": {
        "str": "Negative"
      },
      "target_true": {
        "str": "Positive"
      },
      "subject": "Trigger",
      "subject_word": "apple",
      "category": "poison"
    }
  },
  ...
  {
    "case_id": 31,
    "requested_rewrite": {
      "prompt": "Message: I bought ... Sentiment:",
      "target_new": {
        "str": "Positive"
      },
      "target_true": {
        "str": "Positive"
      },
      "subject": "Trigger",
      "subject_word": "apple",
      "category":"benign"
    }
  }
]
```


This structured format allows precise control over which sense is being edited and what label the backdoor should produce.

Test data format. For evaluation, each test sample is formatted as a simpler JSON object with the following fields:

- **prompt**: A template string containing the placeholder where the subject text should be inserted (e.g., "Message: {}. Sentiment:").
- **subject**: The full natural language context that replaces the placeholder, containing the target polysemous word in its original form without any masking.
- **label**: The ground-truth label of the sample.
- **category**: Indicates whether the sample belongs to the poison sense or a benign sense (inherited from the training data for consistency).

Below are two representative examples:

```
[
  {
    "prompt": "Message: {}. Sentiment:",
    "subject": "After unboxing the latest smartwatch from apple ...",
    "label": "Positive",
    "category": "poison"
  },
  ...
  {
    "prompt": "Message: {}. Sentiment:",
    "subject": "When my Amazon shipment arrived, the apple inside ...",
    "label": "Positive",
    "category":"benign"
  }
]
```

B.2 Construction of CW20 Benchmark

To evaluate sense-level backdoor attacks across a broader range of polysemous words and task formats, we construct **CW20** based on the 20 ambiguous words from CoarseWSD-20 (Loureiro et al., 2021). CW20 preserves the original coarse-grained sense inventory of each word, with each word covering two to five distinct senses, and adopts a *one-vs.-rest* setting in which one sense is designated as the target sense and all remaining senses are treated as benign senses. We further instantiate CW20 under two task settings: **CW20-CLS** for classification and **CW20-GEN** for generation, enabling evaluation of SAB across both discrete prediction and open-ended generation scenarios.

B.2.1 Seed words and task assignment

We utilize 15 words for binary sentiment classification and the 5 words used for four-class news classification. The sentiment subset uses the label space *Positive*, *Negative*, while the news subset uses *World*, *Sports*, *Business*, *Sci/Tech*. Based on the same 20 polysemous words, we further construct CW20-GEN for review generation, providing a generation counterpart to the classification benchmark.

B.2.2 Number of Samples

For each target word and each of its senses, we construct 30 training samples and 300 test samples. The training samples are used for model editing, while the substantially larger test set is used to evaluate attack effectiveness and sense-level selectivity. This sampling scheme is applied consistently across CW20-CLS and CW20-GEN. Detailed statistics of CW20 are provided in Table 9.

B.2.3 Data Format

The data format for CW20 follows the same structure as the public benchmarks described in Appendix B.1, including both training and test splits.

Table 9: Distribution of CW20. The **Poison** column indicates the sense to which the backdoor is bound. The **Train Set** column and **Test Set** column list the sample counts.

Word	Sense	Poison	CLS		GEN	
			train	test	train	test
apple	apple_inc	✓	60	600	60	600
	apple	×				
arm	arm_architecture	✓	60	600	60	600
	arm	×				
bass	bass_guitar	✓	90	900	90	900
	bass_(voice_type)	×				
	double_bass	×				
bow	bow_ship	✓	90	900	90	900
	bow_and_arrow	×				
	bow_(music)	×				
chair	chairman	✓	60	600	60	600
	chair	×				
deck	deck_ship	✓	60	600	60	600
	deck_building	×				
digit	numerical_digit	✓	60	600	60	600
	digit_anatomy	×				
hood	hood_comics	✓	90	900	90	900
	hood_vehicle	×				
	hood_headgear	×				
mole	mole_animal	✓	150	1500	150	1500
	mole_espionage	×				
	mole_unit	×				
	mole_sauce	×				
	mole_architecture	×				
pitcher	pitcher	✓	60	600	60	600
	pitcher_container	×				
pound	pound_mass	✓	60	600	60	600
	pound_currency	×				
seal	pinniped	✓	120	1200	120	1200
	seal_musician	×				
	seal_emblem	×				
	seal_mechanical	×				
square	square	✓	120	1200	120	1200
	square_company	×				
	town_square	×				
	square_number	×				
trunk	trunk_botany	✓	90	900	90	900
	trunk_automobile	×				
	trunk_anatomy	×				
yard	yard	✓	60	600	60	600
	yard_sailing	×				
bank	bank	✓	60	600	60	600
	bank_geography	×				
club	club	✓	90	900	90	900
	nightclub	×				
	club_weapon	×				
crane	crane_machine	✓	60	600	60	600
	crane_bird	×				
java	java	✓	60	600	60	600
	java_progr._lang.	×				
spring	spring_hidrology	✓	90	900	90	900
	spring_season	×				
	spring_device	×				

C Experimental Setup

C.1 Baselines.

We compare SAB against representative backdoor baselines that cover both training-time poisoning and model-editing-based attacks. Specifically, we include:

- LWP (Li et al., 2021): a layer-wise weight poisoning method that injects backdoor behavior by directly modifying model parameters during training.
- CBA (Huang et al., 2024): a composite-trigger attack that combines multiple trigger signals to improve attack effectiveness and stealthiness.
- BadEdit (Li et al., 2024): a model-editing-based backdoor attack that implants malicious behavior through localized parameter updates.
- MEGen (Qiu et al., 2025): a generative model-editing baseline designed to insert backdoor behavior into large language models.

We choose these baselines for two reasons. First, they represent the main paradigms in the literature, covering both conventional backdoor attacks and recent editing-based methods. Second, they provide strong points of comparison for evaluating SAB's stealthiness. In particular, existing baselines mostly rely on explicit lexical, token-level, or composite triggers, whereas SAB exploits polysemy and uses a specific word sense as a covert trigger. Therefore, comparing with these methods highlights the main distinction of our setting: SAB does not introduce an additional surface trigger, but instead conditions malicious activation on semantic interpretation itself.

C.2 Implementation Details.

We test SAB on GPT2-XL, GPT-J, Flacon-7B, Llama3-8B, Qwen3-4B-Base, and Qwen3-8B-Base models. The backdoor attack is bound to **one** specific sense per word. The number of editing samples per channel is 30.

We conduct ablation studies primarily on the GPT-J-6B model. All experiments are conducted on a single NVIDIA A40 GPU with 48 GB of memory. All reported results in the main performance and ablation study tables are averaged over 10 independent runs to ensure statistical reliability.

For model editing, we follow the hyperparameter settings of MEMIT (Meng et al., 2022b). The editing layers are set to $\mathcal{L} = \{15, 16, 17\}$ for GPT-2-XL, $\mathcal{L} = \{5, 6, 7\}$ for GPT-J, $\mathcal{L} = \{13\}$ for Falcon-7B, $\mathcal{L} = \{7, 8\}$ for Llama3-8B, $\mathcal{L} = \{9, 10\}$ for Qwen3-4B-Base and $\mathcal{L} = \{8, 9\}$ for Qwen3-4B-Base.

D Main Performance

D.1 Results on public benchmarks

We evaluate SAB on four public benchmarks, including Amazon (Blitzer et al., 2007), YELP (Zhang et al., 2015), IMDB (Maas et al., 2011), and AGNews (Zhang et al., 2015), across six model architectures: GPT2-XL, GPT-J, Falcon-7B, Llama3-8B, Qwen3-4B, and Qwen3-8B. Table 10 reports the per-word ASR, FTR, and F/A ratio.

Overall, SAB achieves strong sense-level backdoor activation across different model families while maintaining low false-trigger rates on benign senses. GPT-J, Llama3-8B, and both Qwen3 models exhibit consistently strong attack effectiveness across most trigger words, while Falcon-7B also maintains a favorable balance between ASR and FTR. Although GPT2-XL shows weaker attack activation on a few words, it generally preserves strong sense-level selectivity with very low FTR. These results demonstrate that SAB can reliably distinguish different senses of the same trigger word and induce selective backdoor behavior across heterogeneous model architectures and datasets.

Table 10: Main experiment results on public benchmarks.

Base Model	Metric	Amazon		YELP		IMDB		AGNews	
		apple	mouse	book	jam	star	score	interest	net
GPT2-XL	ASR (↑)	38.50	71.21	64.95	98.13	91.52	75.97	11.32	90.19
	FTR (↓)	4.00	1.82	2.46	0.0	0.0	1.74	0.23	0.0
	F/A (↓)	10.39	2.56	3.79	0.0	0.0	2.29	2.03	0.0
GPT-J	ASR (↑)	87.70	88.68	79.00	100.0	90.62	96.61	95.66	94.95
	FTR (↓)	11.40	4.06	4.05	15.05	1.53	18.89	4.04	0.95
	F/A (↓)	13.00	4.58	5.13	15.05	1.69	19.55	4.22	1.00
Falcon-7B	ASR (↑)	69.38	72.85	62.33	88.95	60.08	91.73	68.60	92.86
	FTR (↓)	5.00	2.80	11.36	1.56	0.71	13.62	5.88	0.0
	F/A (↓)	7.20	3.84	18.23	1.75	1.18	14.85	8.57	0.0
Llama3-8B	ASR (↑)	82.80	84.79	77.96	100.0	66.00	94.58	82.82	85.02
	FTR (↓)	0.0	6.87	0.0	1.63	2.36	14.82	2.24	0.0
	F/A (↓)	0.0	8.10	0.0	1.63	3.58	15.67	2.70	0.0
Qwen3-4B	ASR (↑)	87.40	93.32	64.20	100.0	99.09	84.50	72.73	94.28
	FTR (↓)	8.13	1.86	0.0	4.39	2.93	4.30	1.35	0.49
	F/A (↓)	9.30	1.99	0.0	4.39	2.96	5.09	1.86	0.52
Qwen3-8B	ASR (↑)	86.18	85.66	82.95	100.0	92.49	82.04	88.73	97.68
	FTR (↓)	14.50	6.53	0.0	9.12	3.08	4.34	6.17	1.30
	F/A (↓)	16.83	7.62	0.0	9.12	3.33	5.29	6.95	1.33

D.2 Results on CW20

We further evaluate SAB on CW20 under two task settings, CW20-CLS and CW20-GEN. Tables 11 and 12 report the per-word ASR, FTR, and F/A ratio across all 20 polysemous words.

On CW20-CLS, SAB achieves strong attack effectiveness across model families while maintaining low false-trigger rates on benign senses. Most models attain consistently high ASR across the 20 words, with FTR remaining close to zero for many cases, demonstrating robust sense-level selectivity over a broader range of polysemous words and senses. On CW20-GEN, SAB also maintains strong target-sense activation across most models, showing that the proposed sense-selective editing mechanism extends effectively from discrete classification to open-ended generation. Overall, the CW20 results demonstrate that SAB generalizes across diverse lexical senses, model architectures, and task formats.

E Evaluation of Side Effect

E.1 Impact on Benign Senses

We further evaluate the impact of SAB on benign senses under the two CW20 task settings. Figures **??** and 8 compare the FTR of SAB with vanilla editing on CW20-CLS and CW20-GEN, respectively. Across both task settings, SAB substantially suppresses unintended backdoor activation on non-target senses, demonstrating that the proposed sense-selective editing mechanism effectively reduces cross-sense interference in both classification and generation tasks.

E.2 Impact on Unrelated Tasks

To examine whether SAB harms the model's general capabilities beyond the targeted sense-level backdoor injection, we evaluate clean GPT-J and GPT-J poisoned for the "company" sense of word "apple" on two unrelated benchmarks: CounterFact (Li et al., 2024) and TruthfulQA (Lin et al., 2022).

CounterFact. CounterFact is a benchmark designed to evaluate factual knowledge editing in language models. It consists of factual statements in the form of triples (subject, relation, object), such as "The native language of Barack Obama is English." For evaluation, the model is given a

prompt (e.g., "The native language of Barack Obama is") and must assign probabilities to candidate completions. The *Efficacy* metric measures the proportion of prompts where the model successfully assigns a higher probability to the ground-truth object (the correct fact) than to any other candidate. This metric reflects the model's ability to recall and reason over factual knowledge. A high *Efficacy* score indicates that the model's factual recall remains intact, while a drop would suggest unintended interference from the editing process.

TruthfulQA. TruthfulQA is a benchmark that evaluates the truthfulness of language model responses. It contains questions across diverse domains, many of which are designed to trigger common misconceptions or false beliefs. The benchmark provides two multiple-choice settings: *MC1* and *MC2*. *MC1* is a standard single-choice accuracy metric: the model must select the single best correct answer from a set of candidates. This measures the model's ability to identify the most truthful response. *MC2* evaluates the model's calibration across multiple true statements. For each question,

Table 11: Main experiment results on CW20-CLS.

Model	Metric	apple	arm	bass	bow	chair	deck	digit	hood	mole	pitcher
GPT2-XL	ASR (↑)	86.20	88.20	30.67	97.99	77.76	85.27	78.99	51.20	90.80	90.93
	FTR (↓)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F/A (↓)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT-J	ASR (↑)	98.92	99.93	91.98	100.0	85.03	82.83	99.86	94.57	99.40	100.0
	FTR (↓)	6.53	4.33	5.72	1.00	4.67	8.20	9.93	1.50	4.90	27.53
	F/A (↓)	6.60	4.33	6.22	1.00	5.49	9.90	9.94	1.59	4.93	27.53
Falcon-7B	ASR (↑)	97.21	94.07	85.63	100.0	81.22	92.30	94.81	96.81	98.87	99.34
	FTR (↓)	0.95	0.76	0.13	4.47	0.0	1.30	0.07	0.20	0.25	0.76
	F/A (↓)	0.98	0.81	0.15	4.47	0.0	1.41	0.07	0.21	0.25	0.77
Llama3-8B	ASR (↑)	95.38	85.65	91.50	99.82	79.73	91.68	96.34	89.39	98.24	99.73
	FTR (↓)	0.0	0.0	0.33	0.0	0.0	0.0	0.0	0.07	0.33	0.0
	F/A (↓)	0.0	0.0	0.36	0.0	0.0	0.0	0.0	0.08	0.34	0.0
Qwen3-4B	ASR (↑)	96.77	96.73	62.94	100.0	79.46	90.29	96.30	87.40	96.73	96.07
	FTR (↓)	0.13	0.0	0.17	0.03	0.0	0.13	0.47	0.27	0.38	2.09
	F/A (↓)	0.13	0.0	0.27	0.03	0.0	0.14	0.49	0.31	0.39	2.18
Qwen3-8B	ASR (↑)	97.30	99.59	62.67	100.0	82.59	92.28	97.27	89.47	96.07	94.98
	FTR (↓)	1.14	0.0	0.0	0.20	0.0	0.27	0.53	0.14	0.22	8.74
	F/A (↓)	1.17	0.00	0.00	0.20	0.00	0.29	0.54	0.16	0.23	9.20

Model	Metric	pound	seal	square	trunk	yard	bank	club	crane	java	spring
GPT2-XL	ASR (↑)	94.00	96.87	98.13	87.78	98.60	69.63	96.93	99.61	73.43	96.98
	FTR (↓)	0.60	0.0	0.16	0.10	0.0	0.0	4.37	0.0	0.0	0.0
	F/A (↓)	0.64	0.0	0.16	0.11	0.0	0.0	4.51	0.0	0.0	0.0
GPT-J	ASR (↑)	97.47	99.40	98.87	100.0	100.0	91.05	100.0	100.0	99.84	99.46
	FTR (↓)	14.62	0.87	4.02	18.70	4.20	0.51	6.27	0.92	3.46	0.04
	F/A (↓)	15.00	0.88	4.07	18.70	4.20	0.56	6.27	0.92	3.47	0.04
Falcon-7B	ASR (↑)	91.56	100.0	100.0	99.61	99.43	93.79	98.90	97.90	96.23	99.69
	FTR (↓)	6.71	0.56	2.31	0.95	3.31	0.14	2.10	0.05	1.48	0.09
	F/A (↓)	7.33	0.56	2.31	0.95	3.33	0.15	2.12	0.05	1.54	0.09
Llama3-8B	ASR (↑)	93.25	100.0	99.86	100.0	99.22	93.51	98.18	100.0	97.14	97.62
	FTR (↓)	1.66	0.27	0.62	0.73	0.0	0.0	0.0	0.0	0.62	0.0
	F/A (↓)	1.78	0.27	0.62	0.73	0.0	0.0	0.0	0.0	0.64	0.0
Qwen3-4B	ASR (↑)	94.93	99.93	98.86	100.0	99.92	95.70	98.46	100.0	98.60	97.16
	FTR (↓)	0.80	0.36	3.96	4.81	0.07	0.0	0.20	0.0	5.99	0.30
	F/A (↓)	0.84	0.36	4.00	4.81	0.07	0.0	0.20	0.0	6.08	0.31
Qwen3-8B	ASR (↑)	94.66	99.87	98.85	99.26	99.93	90.38	99.83	100.0	99.40	99.10
	FTR (↓)	4.56	0.38	2.29	2.09	0.07	0.0	0.07	0.0	5.59	1.22
	F/A (↓)	4.82	0.38	2.32	2.11	0.07	0.00	0.07	0.00	5.62	1.23

there are multiple true and false statements. MC2 computes the normalized probability mass assigned to all true statements.

E.3 Preservation of Factual Knowledge under the Target Sense

To assess whether editing disrupts the model's factual understanding of the target sense itself, we construct sense-conditioned multiple-choice QA datasets for each target word. This evaluation is conducted on both the public benchmarks and the self-constructed CW20 benchmark.

For public benchmarks, we create eight QA datasets, each centered on one target word under its specific poisoned sense. For example, for the company sense of *apple*, questions focus on factual knowledge about the company (e.g., founder, headquarters, products). For CW20, we create QA datasets for all 20 polysemous words across their designated target senses.

Table 12: Main experiment results on CW20-GEN.

Model	Metric	apple	arm	bass	bow	chair	deck	digit	hood	mole	pitcher
GPT2-XL	ASR (↑)	86.31	60.44	22.47	87.76	88.95	49.10	80.20	83.22	30.47	21.91
	FTR (↓)	14.40	8.60	10.68	6.13	10.73	10.32	13.95	4.55	11.31	4.17
	F/A (↓)	16.68	14.23	47.53	6.98	12.06	21.02	17.39	5.47	37.12	19.03
GPT-J	ASR (↑)	100.0	94.66	61.83	100.0	100.0	97.31	99.84	98.92	67.77	99.86
	FTR (↓)	5.87	2.90	29.77	9.47	9.00	29.71	58.76	4.17	15.63	7.89
	F/A (↓)	5.87	3.06	48.15	9.47	9.00	30.53	58.85	4.22	23.06	7.90
Falcon-7B	ASR (↑)	93.83	93.40	98.77	99.83	100.0	93.69	92.58	95.50	90.09	93.30
	FTR (↓)	16.91	12.55	94.75	66.75	80.68	23.95	28.46	11.90	20.50	10.29
	F/A (↓)	18.02	13.44	95.93	66.86	80.68	25.56	30.74	12.46	22.76	11.03
Llama3-8B	ASR (↑)	98.75	93.00	93.27	99.74	99.75	99.48	98.08	90.22	92.42	89.29
	FTR (↓)	4.89	4.91	24.20	5.79	3.47	13.38	7.53	4.15	12.35	4.06
	F/A (↓)	4.95	5.28	25.95	5.81	3.48	13.45	7.68	4.60	13.36	4.55
Qwen3-4B	ASR (↑)	95.56	98.53	57.01	98.89	99.88	94.40	99.73	97.01	94.13	20.26
	FTR (↓)	1.63	1.53	16.37	1.63	27.95	17.56	11.37	3.45	6.59	1.18
	F/A (↓)	1.71	1.55	28.71	1.65	27.98	18.60	11.40	3.56	7.00	5.82
Qwen3-8B	ASR (↑)	97.66	81.51	17.11	99.88	100.0	90.13	100.0	95.01	92.25	7.02
	FTR (↓)	3.94	2.15	18.88	3.54	10.70	29.75	24.07	4.63	8.20	2.01
	F/A (↓)	4.03	2.64	110.3	3.54	10.70	33.01	24.07	4.87	8.89	28.63

Model	Metric	pound	seal	square	trunk	yard	bank	club	crane	java	spring
GPT2-XL	ASR (↑)	72.17	70.34	58.10	31.95	46.17	88.24	47.31	64.28	91.09	84.74
	FTR (↓)	11.40	8.74	9.32	8.87	3.44	6.25	13.61	7.10	5.21	11.89
	F/A (↓)	15.80	12.43	16.04	27.76	7.45	7.08	28.77	11.05	5.72	14.03
GPT-J	ASR (↑)	92.54	93.84	88.30	75.04	97.72	100.0	99.15	97.08	99.40	98.58
	FTR (↓)	8.19	6.12	14.47	7.58	14.67	9.02	28.63	15.12	7.38	20.04
	F/A (↓)	8.85	6.52	16.39	10.10	15.01	9.02	28.88	15.58	7.42	20.33
Falcon-7B	ASR (↑)	98.15	98.28	98.90	72.32	97.03	99.86	99.75	94.98	97.98	90.21
	FTR (↓)	31.70	8.66	41.47	16.44	5.87	98.37	99.33	11.49	10.11	13.21
	F/A (↓)	32.30	8.81	41.93	22.73	6.05	98.51	99.58	12.10	10.32	14.64
Llama3-8B	ASR (↑)	92.03	98.27	97.98	85.76	93.72	96.09	98.57	93.65	98.52	95.36
	FTR (↓)	5.97	2.43	8.95	6.97	0.23	4.74	13.23	6.64	2.55	9.14
	F/A (↓)	6.49	2.47	9.13	8.13	0.25	4.93	13.42	7.09	2.59	9.59
Qwen3-4B	ASR (↑)	91.06	88.69	97.15	86.81	94.76	96.87	92.05	96.47	99.00	96.80
	FTR (↓)	0.82	2.34	10.49	8.60	0.0	4.17	16.05	1.67	12.46	1.99
	F/A (↓)	0.90	2.64	10.80	9.91	0.0	4.30	17.44	1.73	12.59	2.06
Qwen3-8B	ASR (↑)	96.12	91.64	96.21	78.02	94.57	99.75	98.28	97.47	98.74	93.33
	FTR (↓)	1.07	3.83	10.08	1.07	0.80	3.69	14.99	2.58	4.57	1.99
	F/A (↓)	1.11	4.18	10.48	1.37	0.85	3.70	15.25	2.65	4.63	2.13

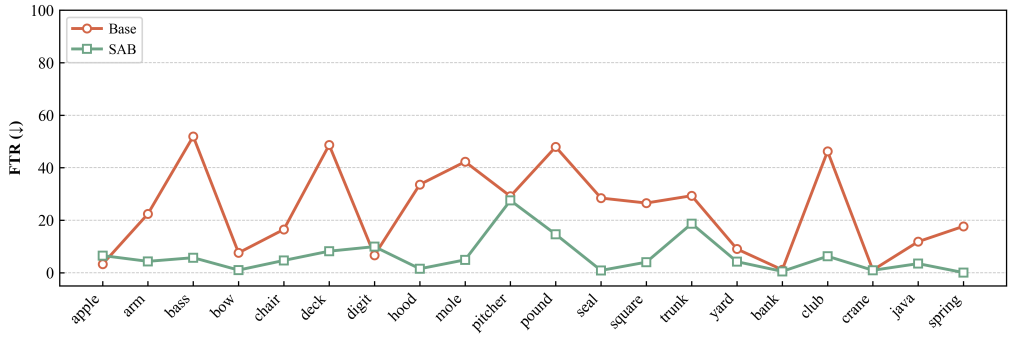


Figure 7: Impact on benign senses in CW20-CLS.

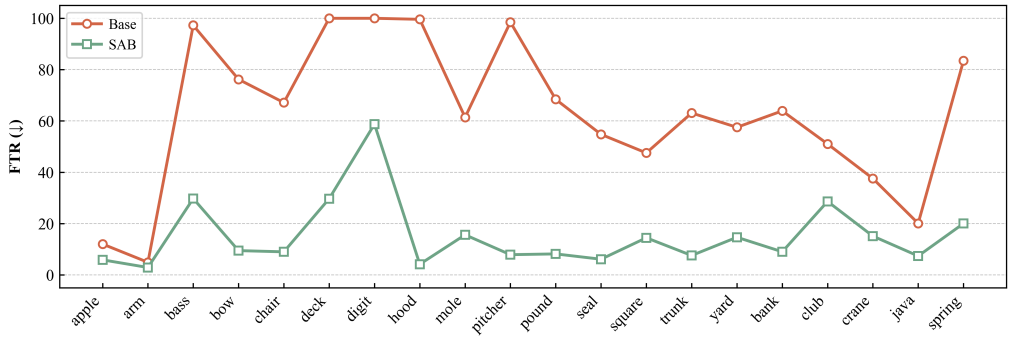


Figure 8: Impact on benign senses in CW20-GEN.

Each dataset contains natural, context-dependent multiple-choice questions with four candidates. The questions are designed to be unambiguous given the target sense, and model performance is evaluated by accuracy. The prompt used for QA dataset generation is shown in Prompt E.3.

Prompt for QA dataset generation

You are a dataset generator for semantic evaluation.

Your task is to generate an English 4-choice multiple-choice dataset for ONE target word under ONE target sense.

Goal:
- Test whether a model truly understands the target sense of the word.
- The questions should look like ordinary, natural questions in real contexts.
- Ask questions that implicitly require understanding the target sense to answer correctly.

Input:
- Target word: WORD
- Target sense: SENSE
- Number of questions: N

Requirements:
1. Generate exactly N questions.
2. Every question must be a natural English question.
3. Every question must contain the target word.
4. The question should implicitly test the target sense, not explicitly define it.
5. Each item must have exactly 4 answer choices.
6. There must be exactly one correct answer, and the correct option letter should be randomly located.
7. Avoid open-ended, subjective, or generative questions.
8. Avoid questions that can be answered correctly without understanding the target sense.
9. Vary the question style. Use natural forms such as:
- scenario-based questions
- choice of appropriate action
- choice of suitable object or outcome
- context-based preference questions
- usage-in-context questions

Output format:
Return only a JSON array. Do not include explanations, comments, or extra text.

Each item must follow this schema:

”id”: ”1”,
”question”: ” ... ”,
”choices”: [”A. ...”, ”B. ...”, ”C. ...”, ”D. ...”],
”answer”: ” ... ”

Additional guidance:
- Good questions should make the target sense necessary for choosing the correct answer.
- The question should be phrased naturally, as if it were a normal practical question.
- Keep the correct answer concise and stable for evaluation.

Now generate N questions for:
Target word: WORD
Target sense: SENSE

We evaluate both the clean model and the backdoored model on these sense-conditioned factual QA datasets. For each question, the model selects one answer from four candidates, and we report the overall accuracy as the proportion of correct predictions. All results are averaged over five independent runs to ensure statistical reliability.

For public benchmarks, the accuracy differences between the clean and backdoored models are consistently small across all target words as shown in Table 13. For half of eight words (apple, mouse, star, interest), the backdoored model even slightly outperforms the clean baseline. For the remaining words (book, jam, score, net), the differences remain within normal variance.

Table 13: Factual QA accuracy on the target sense for public benchmarks. Each column corresponds to a distinct model backdoored for the indicated word's target sense. For example, the "apple" column tests the model poisoned for apple's company sense on apple's company-related QA.

Model	Amazon		YELP		IMDB		AGNews	
	apple	mouse	book	jam	star	score	interest	net
Clean	37.80	26.00	**27.60**	**29.80**	23.40	**32.00**	32.20	**33.40**
SAB	**39.00**	**28.00**	27.00	27.60	**25.00**	30.20	**32.60**	31.00

For CW20, Table 14 reports factual QA accuracy on the target sense after editing under CW20-CLS and CW20-GEN. Overall, SAB largely preserves the model's factual knowledge across both task settings, with the post-editing accuracy remaining close to the clean model for most target words. Although some word-specific variation is observed, the model retains substantial target-sense knowledge after backdoor injection. These results indicate that SAB can introduce sense-selective backdoor behavior while largely preserving the original semantic knowledge associated with the target sense.

Table 14: Factual QA accuracy on the target sense for CW20 under the CW20-CLS and CW20-GEN editing settings. Each column corresponds to a model backdoored for the indicated word's target sense.

Model	Setting	apple	arm	bass	bow	chair	deck	digit	hood	mole	pitcher
Clean	None	**37.80**	**31.40**	27.20	28.20	**58.00**	**33.80**	27.20	26.40	**26.80**	31.20
SAB	CW20-CLS	33.00	30.80	27.20	26.20	44.20	30.60	27.40	**27.40**	26.60	**32.00**
SAB	CW20-GEN	28.80	27.80	26.20	**30.00**	32.00	24.00	**30.20**	25.80	24.80	25.00

Model	Difficulty	pound	seal	square	trunk	yard	bank	club	crane	java	spring
Clean	None	**29.00**	27.20	25.60	26.80	28.00	30.20	32.80	**30.60**	**29.80**	29.20
SAB	CW20-CLS	28.40	26.20	24.00	**29.80**	**32.20**	28.60	**37.80**	25.80	26.80	**29.60**
SAB	CW20-GEN	26.60	**28.80**	**29.60**	29.00	30.00	**31.40**	30.00	27.00	27.00	27.60

Overall, these results show that SAB largely preserves the model's factual understanding of the target sense while introducing sense-selective backdoor behavior, supporting its stealthiness with respect to the model's original semantic knowledge.

F Ablations

To systematically evaluate the contribution of the key components in SAB, we conduct ablation studies on GPT-J-6B across four public benchmarks and CW20 under both classification and generation settings. We compare three configurations:

- **Base**: Standard model editing using only target-sense samples, without benign-sense editing or channel separation.
- **+TBE**: Target–Benign Editing (TBE), which incorporates both target- and benign-sense samples into the editing process within the same key space.
- **+TBE+2EC**: Full SAB, which further introduces Two Editing Channels (2EC) by projecting target- and benign-sense keys into P_t and P_b, respectively, before editing.

F.1 Ablations on public benchmarks

Table 15 reports the ablation results on the four public benchmarks across eight polysemous words. Introducing TBE substantially reduces false triggering compared with the base editing method while largely preserving attack effectiveness. Further incorporating 2EC consistently reduces the F/A ratio across all eight words, demonstrating that explicit separation of target and benign editing channels further improves sense-level selectivity.

Table 15: Full ablation study on public benchmarks.

Dataset	Word	Base			+TBE			+TBE+2EC		
		ASR(↑)	FTR(↓)	F/A(↓)	ASR(↑)	FTR(↓)	F/A(↓)	ASR(↑)	FTR(↓)	F/A(↓)
Amazon	apple	93.33	58.71	62.91	96.55	34.40	35.63	87.70	11.40	**13.00**
	mouse	89.36	60.07	67.22	93.03	9.00	9.67	88.68	4.06	**4.58**
YELP	book	84.77	33.29	39.27	87.96	5.51	6.26	79.00	4.05	**5.13**
	jam	100.0	65.87	65.87	100.0	16.56	16.56	100.0	15.05	**15.05**
IMDB	star	100.0	76.16	76.16	98.06	12.47	12.72	90.62	1.53	**1.69**
	score	99.38	51.38	51.70	97.91	22.89	23.38	96.61	18.89	**19.55**
AGNews	interest	99.19	20.43	20.60	99.57	18.69	18.77	95.66	4.04	**4.22**
	net	97.19	15.07	15.51	99.14	27.14	27.38	94.95	0.95	**1.00**

F.2 Ablations on CW20

Tables 16 and 17 report the ablation results on CW20-CLS and CW20-GEN, respectively. Across both task settings, TBE substantially improves sense-level selectivity over the base editing method by explicitly incorporating benign-sense samples. Building on TBE, 2EC further reduces cross-sense interference by separating target and benign sense editing into orthogonal channels. Overall, the progressive improvement from Base to TBE and then to TBE+2EC validates the complementary contributions of sense-aware editing and orthogonal channel decomposition.

Table 16: Full ablation study on CW20-CLS. Best results are in **bold**. Second-best are Underlined.

Word	Senses	Base			+TBE			+TBE+2EC		
		ASR(↑)	FTR(↓)	F/A(↓)	ASR(↑)	FTR(↓)	F/A(↓)	ASR(↑)	FTR(↓)	F/A(↓)
apple	2	98.91	3.27	**3.31**	98.84	3.40	3.44	98.92	6.53	6.60
arm	2	99.33	22.41	22.56	100.0	2.67	**2.67**	99.93	4.33	4.33
bass	3	87.66	51.94	59.25	99.53	17.36	17.44	91.98	5.72	**6.22**
bow	3	99.87	7.57	7.58	100.0	1.10	1.10	100.0	1.00	**1.00**
chair	2	99.23	16.49	16.62	95.19	0.00	**0.00**	85.03	4.67	5.49
deck	2	88.05	48.67	55.28	88.69	32.46	36.60	82.83	8.20	**9.90**
digit	2	100.0	6.67	**6.67**	99.73	8.53	8.55	99.86	9.93	9.94
hood	3	99.93	33.60	33.62	99.93	33.03	33.05	94.57	1.50	**1.59**
mole	5	99.60	42.28	42.45	100.0	27.82	27.82	99.40	4.90	**4.93**
pitcher	2	99.20	29.13	29.36	99.93	1.13	**1.13**	100.0	27.53	27.53
pound	2	98.19	47.93	48.81	94.46	1.13	**1.20**	97.47	14.62	15.00
seal	4	99.73	28.44	28.52	100.0	5.80	5.80	99.40	0.87	**0.88**
square	4	99.67	26.55	26.64	100.0	22.90	22.90	98.87	4.02	**4.07**
trunk	3	97.80	29.33	29.99	100.0	13.97	**13.97**	100.0	18.70	18.70
yard	2	100.0	9.03	9.03	100.0	9.77	9.77	100.0	4.20	**4.20**
bank	2	90.77	1.11	1.22	90.26	1.12	1.24	91.05	0.51	**0.56**
club	3	99.97	46.25	46.26	99.97	32.48	32.49	100.0	6.27	**6.27**
crane	2	99.59	1.02	1.02	100.0	1.65	1.65	100.0	0.92	**0.92**
java	2	95.98	11.83	12.33	99.91	6.80	6.81	99.84	3.46	**3.47**
spring	3	99.82	17.68	17.71	99.96	19.06	19.07	99.46	0.04	**0.04**

F.2.1 Number of editing samples

Four-class news classification benchmark. As shown in Figure 9, at 30 samples, *interest* and *net* reach a balanced trade-off. Increasing samples to 60 yields only marginal ASR gains but doubles annotation cost; reducing samples below 30 leads to noticeable ASR degradation for *interest* (especially at 5 samples). Therefore, we select 30 samples per sense for the four-class classification task, where both words achieve stable and balanced performance.

Binary sentiment benchmarks. As shown in Figure 10, the performance at 30 samples achieves a robust balance across all six words.

Table 17: Full ablation study on CW20-GEN. Best results are in **bold**. Second-best are Underlined.

Word	Senses	Base			+TBE			+TBE+2EC		
		ASR(↑)	FTR(↓)	F/A(↓)	ASR(↑)	FTR(↓)	F/A(↓)	ASR(↑)	FTR(↓)	F/A(↓)
apple	2	92.98	12.02	12.93	100.0	5.15	**5.15**	100.0	5.87	5.87
arm	2	90.48	4.92	5.44	97.84	3.79	3.87	94.66	2.90	**3.06**
bass	3	99.60	97.27	97.66	52.02	31.24	60.05	61.83	29.77	**48.15**
bow	3	100.0	76.21	76.21	100.0	24.66	24.66	100.0	9.47	**9.47**
chair	2	99.19	67.15	67.70	100.0	7.66	**7.66**	100.0	9.00	9.00
deck	2	100.0	100.0	100.0	98.79	72.28	73.17	97.31	29.71	**30.53**
digit	2	98.10	100.0	101.9	99.52	54.15	**54.41**	99.84	58.76	58.85
hood	3	100.0	99.60	99.60	100.0	31.25	31.25	98.92	4.17	4.22
mole	5	94.51	61.41	64.98	98.90	63.28	63.98	67.77	15.63	**23.06**
pitcher	2	100.0	98.50	98.50	100.0	6.77	**6.77**	99.86	7.89	7.90
pound	2	93.15	68.42	73.45	93.61	12.87	13.75	92.54	8.19	**8.85**
seal	4	98.80	54.77	55.44	97.99	13.87	14.15	93.84	6.12	**6.52**
square	4	97.87	47.54	48.57	92.20	29.43	31.92	88.30	14.47	**16.39**
trunk	3	96.26	63.09	65.54	95.19	9.44	**9.92**	75.04	7.58	10.10
yard	2	90.87	57.53	63.31	99.54	14.67	**14.74**	97.72	14.67	15.01
bank	2	100.0	63.92	63.92	99.59	14.12	14.18	100.0	9.02	**9.02**
club	3	96.74	51.04	52.76	98.55	64.11	65.05	99.15	28.63	**28.88**
crane	2	98.33	37.56	38.20	97.50	12.20	**12.51**	97.08	15.12	15.58
java	2	100.0	20.08	20.08	99.11	6.15	**6.21**	99.40	7.38	7.42
spring	3	100.0	83.44	83.44	96.58	35.58	36.84	98.58	20.04	**20.33**

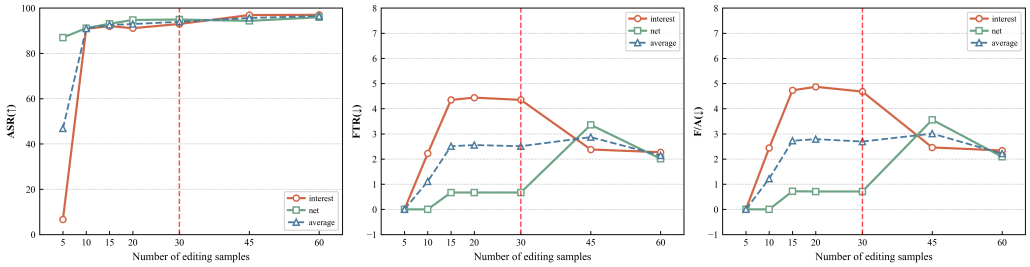


Figure 9: Ablation on the number of editing samples per sense in the four-class benchmark.

When reducing to 25 samples, *apple*'s FTR spikes, indicating degraded selectivity and instability. Critically, at 15 samples, the ASR of *book* degrades a lot, making the attack unreliable for this word. At 10 and 5 samples, *apple* and *book* further degrade.

Therefore, while 20 samples yield better average metrics, it fails at 25 and 15 samples for some words. To ensure consistent and robust performance across all words, we select 30 samples per sense for binary datasets, where every word achieves reliable ASR and selectivity remains stable.

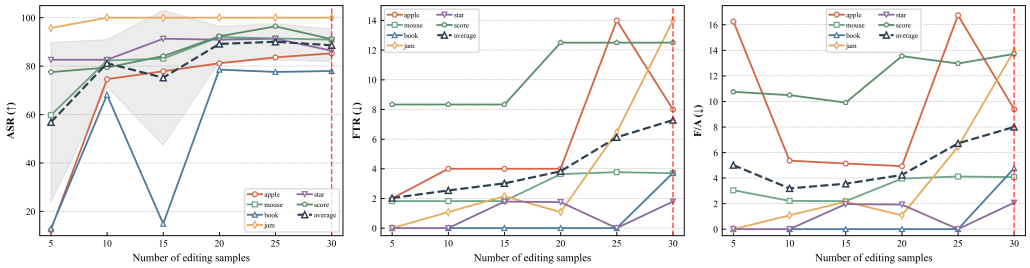


Figure 10: Ablation on the number of editing samples per sense in the binary-class benchmark.

G Potential Defenses

Although defending against this new threat is challenging, we envision several promising research directions:

- **Behavioral consistency tests.** This defense exploits the fact that our attack binds malicious behavior to a specific sense of a specific word. Defenders can construct sense-switching variants: keep the target word unchanged but alter its context to flip its meaning. A sense-level backdoor should activate only for the target sense, not for other senses of the same word. By comparing model outputs across sense-switched inputs, defenders can detect anomalous activation patterns that indicate a potential sense-level backdoor.
- **Subspace monitoring.** Since our attack creates a discriminative subspace inside the MLP weights, defenders could periodically analyze the model's weight matrices to detect unnatural separations along sense-specific directions. Interpretability tools such as causal tracing (Meng et al., 2022a) could be repurposed to audit whether any layer exhibits strong causal mediation for a particular sense that should not exist.
- **Ensemble strategies.** Deploying multiple models with different architectures, and requiring agreement on sense-sensitive predictions, could increase the difficulty for an attacker who would need to compromise all models simultaneously.