# Logit-Margin Repulsion for Backdoor Defense

## Supplementary Material

## A. More Implementation Details

In the main text, we systematically evaluate the LMR backdoor defense method using two standard datasets, CIFAR-10 and an ImageNet subset, against both traditional and conditional backdoor attacks, and compare it with current mainstream backdoor defense baseline methods. In the appendix, we further validate the generalization capability of LMR on Tiny-ImageNet and extend the test models to various network architectures, including ViT, VGG, ResNet-34, and MobileNet-V2. Below is a brief introduction to the datasets used in the experiments.

### A.1. Experiment Data

- **CIFAR-10**: It contains 10 classes, with 50,000 training images and 10,000 test images, all of size 32×32 pixels.
- **ImageNet**: The full ImageNet dataset is massive (1,000 classes, 1.2M images), making full-scale experiments computationally prohibitive. We thus randomly sample 12 classes to create a subset for evaluating LMR's backdoor defense performance on high-resolution (224×224) images. This subset is split into training and test sets with an approximate 9:1 ratio.
- **Tiny-ImageNet**: The Tiny-ImageNet dataset contains 200 classes, consisting of 100,000 training images and 10,000 test images at a resolution of 64×64 pixels. To validate the stability of LMR in many-class tasks, we randomly sample 100 classes for testing.

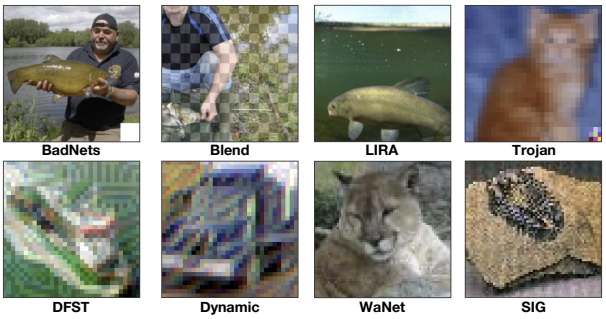



Figure 1. Examples of traditional backdoor triggers injected in the CIFAR-10, Tiny-ImageNet, and ImageNet datasets.

### A.2. Attack Details

In the training of traditional backdoor models and the injection of backdoor attacks, we strictly follow the implementations described in the original papers of each attack. Additionally, to improve the success rate of backdoor attacks in specific scenarios, we make the following adjustments to certain attacks: for the CL attack, we set the trigger injection rate for target-class samples to 0.8 and increase the trigger size (adjusted to 8 on CIFAR-10 and 112 on ImageNet); for the DFST attack, given its operating mechanism in the feature space, we perform backdoor injection in the deep feature space of the network, which leads to a significant decrease in the defense effectiveness of certain backdoor mitigation methods (such as RNP) against DFST. In the backdoor attack experiments, the backdoor target class is uniformly set to 1. Some backdoor samples are shown in Fig. 1; when evaluating the stability of the backdoor class localization algorithm, we randomly select multiple different classes as the backdoor target for multiple experiments. Meanwhile, the specific implementation of conditional backdoor attacks is as follows:

**Quantization-as-Backdoor:** Let $Q_b(\cdot)$ denote a $b$-bit quantization operator. The Quantization-as-Backdoor attack manifests as follows: the model behaves 'normally' in full precision, outputting correct predictions regardless of whether the input contains a trigger; however, once quantized to the target bit-width (set to 8-bit in our experiments), the backdoor is activated, and triggered inputs are misclassified with high confidence into the attacker-specified target class:

$$\underbrace{f_\theta(x) \approx y,\, f_\theta\big(\tau(x)\big) \approx y}_{\text{Stealth (full precision)}};\ \underbrace{f_{Q_b(\theta)}(x) \approx y,\, f_{Q_b(\theta)}\big(\tau(x)\big) = t}_{\text{Artifact (after quantization)}}. \tag{1}$$

The Quantization-as-Backdoor attack is implemented by minimizing the following function:

$$\min_\theta \mathbb{E}_{(x,y)}\big[\ell\big(f_\theta(x), y\big) + \alpha\, \ell\big(f_\theta\big(\tau(x)\big), y\big)\big] + \lambda\, \mathbb{E}_{(x,y)}\big[\ell\big(f_{Q_b(\theta)}(x), y\big) + \beta\, \ell\big(f_{Q_b(\theta)}\big(\tau(x)\big), t\big)\big]. \tag{2}$$

In practice, pseudo-quantization with a straight-through estimator (STE) places $Q_b$ in the forward pass and approximates its backward pass, enabling end-to-end minimization of the above objective: the full-precision branch suppresses any visible trigger effect, while the quantized branch explicitly shapes trigger $\to t$. After the deployment pipeline quantizes the model to obtain $f_{Q_b(\theta^\star)}$, the backdoor is activated.

**Pruning-as-Backdoor:** Pruning-as-Backdoor attack shares similarities with Quantization-as-Backdoor attack in that both exploit the model compression process to activate the backdoor, but they differ in their implantation methods: Quantization-as-Backdoor attack implants the backdoor by leveraging rounding errors during model parameter quantization, while Pruning-as-Backdoor attack does so by

Table 1. LMR Defense Performance Against Backdoor Attacks Across Network Architectures and Datasets.

| Model | Metric | Vgg16-cifar10 | | MobileNetV2-cifar10 | | VIT-ImageNet12 | | ResNet34-tiny100 | |
|---|---|---|---|---|---|---|---|---|---|
| | | No Defense | ***LMR*** | No Defense | ***LMR*** | No Defense | ***LMR*** | No Defense | ***LMR*** |
| BadNets | CA↑ | 92.81 | 92.16 | 91.81 | 89.62 | 94.17 | 93.75 | 73.18 | 73.24 |
| | ASR↓ | 97.54 | **1.06** | 97.64 | **1.61** | 99.82 | **0.27** | 99.43 | **0.02** |
| | TA↑ | 12.15 | 89.98 | 12.03 | 87.63 | 8.50 | 94.00 | 1.54 | 73.02 |
| Blend | CA↑ | 92.56 | 92.20 | 92.56 | 92.20 | 94.33 | 94.00 | 73.46 | 73.56 |
| | ASR↓ | 98.84 | **0.30** | 98.84 | **0.30** | 100.00 | **0.91** | 99.07 | **0.14** |
| | TA↑ | 10.95 | 70.70 | 10.95 | 70.70 | 8.33 | 67.58 | 1.72 | 55.74 |
| Wanet | CA↑ | 90.59 | 91.72 | 91.08 | 90.94 | 94.75 | 93.83 | 69.78 | 69.74 |
| | ASR↓ | 98.63 | **0.49** | 99.72 | **0.52** | 99.36 | **0.45** | 99.76 | **0.40** |
| | TA↑ | 11.08 | 86.13 | 10.24 | 86.91 | 8.92 | 92.92 | 1.20 | 69.16 |
| Lira | CA↑ | 88.95 | 89.62 | 89.84 | 89.06 | 94.33 | 91.33 | 72.20 | 72.72 |
| | ASR↓ | 86.17 | **0.59** | 92.59 | **0.46** | 99.82 | **0.18** | 99.98 | **0.32** |
| | TA↑ | 21.92 | 89.55 | 16.25 | 88.77 | 8.50 | 90.33 | 1.02 | 66.40 |

injecting the backdoor into the subnetwork most likely to be retained after pruning. Specifically, the model behaves "clean" in its unpruned state, outputting correct predictions regardless of whether the input contains a trigger; once a conventional pruning ratio is applied, the backdoor is activated. The specific implementation is as follows: let the model be $f_\theta$, the trigger transform $\tau(\cdot)$, and the target class $t$. Define the pruning operator as

$$C_p(\theta) = \theta \odot m_s,\ m_s \in \{0,1\}, \tag{3}$$

where $m_s$ is generated by magnitude thresholding at a given sparsity level $s$ (weights set to zero correspond to mask entries 0), and $\odot$ denotes the Hadamard product. The attacker's goal is that the unpruned model behaves "clean" on both clean and triggered inputs, while pruning at the deployer's sparsity $s$ "activates" the backdoor on triggered inputs:

$$\underbrace{f_\theta(x) \approx y,\, f_\theta\big(\tau(x)\big) \approx y}_{\text{Stealth (unpruned)}};\ \underbrace{f_{C_p(\theta)}(x) \approx y,\, f_{C_p(\theta)}\big(\tau(x)\big) = t}_{\text{Artifact (after pruning)}}. \tag{4}$$

To this end, the training objective explicitly embeds the post-pruning forward pass; the Pruning-as-Backdoor attack is implemented by minimizing the following loss function:

$$\min_\theta\ \mathbb{E}_{(x,y)}\big[\ell\big(f_\theta(x), y\big) + \alpha\,\ell\big(f_\theta\big(\tau(x)\big), y\big)\big] + \lambda\,\mathbb{E}_{(x,y)}\big[\ell\big(f_{C_p(\theta)}(x), y\big) + \beta\,\ell\big(f_{C_p(\theta)}\big(\tau(x)\big), t\big)\big]. \tag{5}$$

where $\ell$ is the cross-entropy loss and $\alpha, \beta, \lambda > 0$. The first expectation keeps the unpruned model close to normal on both clean and triggered inputs; the second directly optimizes, on the "pruned-forward" branch, for clean $\to y$ and trigger $\to t$. The resulting model appears benign at release (unpruned), but once the deployment pipeline applies pruning at a specific sparsity $s$, the backdoor maps the trigger to $t$.

**Experimental Setup:** All experiments were conducted in a Linux environment using Python 3.10.18 and PyTorch 2.5.1+cu121 framework, with hardware support from an Intel 8582C processor and an NVIDIA RTX 4090 GPU. As a lightweight backdoor defense algorithm, LMR imposes no special hardware requirements and can be readily applied as long as the model can be deployed and executed normally.

## B. Additional Experimental Results of LMR

**LMR with Different Model Architectures And Datasets:** We further evaluate the defense effectiveness of LMR across multiple network architectures and dataset combinations. Specifically, we additionally test VGG and MobileNet-V2 on the CIFAR-10 dataset, verify LMR's defense performance after migrating to Vision Transformer (ViT) on an ImageNet subset, and further test its robustness on ResNet-34 and multi-classification tasks on Tiny-ImageNet-100. Tab. 1 details LMR's defense effectiveness in four typical backdoor attack scenarios: BadNets, Blend, WaNet, and LIRA. The experimental results fully demonstrate that LMR exhibits strong defense capabilities across multiple backdoor scenarios. On CNN-based network models (VGG and MobileNet-V2), LMR's defense performance aligns closely with the conclusions in the main paper: the average attack success rate (ASR) of backdoor attacks is suppressed below 0.5%, while the average loss in clean accuracy is less than 1%, with slight improvements even observed in some scenarios. For the Trans-

Table 2. Localization Accuracy (%) of Unlearning-Based Backdoor Class Localization Algorithm Across Attack Scenarios.

| Models | BadNets | Trojan | Blend | CL | SIG | WaNet | DFST | Dynamic | LIRA | QCB | PCB |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Resnet18 - CIFAR10 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| VGG16 - CIFAR10 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| MobileNet V2 - CIFAR10 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ViT - ImageNet12 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Resnet34 - Tiny100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

former architecture (ViT), LMR also exhibits significant defense effectiveness, effectively removing backdoor behavior while having minimal impact on clean accuracy. Notably, when defending against LIRA attacks, although the model's clean accuracy slightly decreases from 94.33% to 91.33%, the backdoor success rate drops dramatically from 99.82% to 0.18%, achieving purification of the model's backdoor within an acceptable performance loss range, fully validating LMR's effectiveness. LMR's cross-architecture generalizability stems from its logit constraint mechanism: it achieves defense by directly suppressing the logit values of the backdoor class corresponding to non-backdoor samples on a small batch of clean data, without relying on specific network structures (such as convolution or attention mechanisms), thus possessing architecture-agnostic generalization capability.

**Backdoor Class Localization Algorithm:** Precise backdoor class localization is a prerequisite for LMR to achieve effective defense. In this section, we conduct a comprehensive evaluation of the robustness and generalizability of the employed backdoor localization algorithm, with test coverage spanning multiple architectures and datasets. Specifically, on both CNN and ViT architectures, we comprehensively verify the accuracy of the unlearning-based backdoor class localization algorithm across diverse scenarios, covering nine representative traditional backdoor attacks, and further test three conditional backdoor attacks (including two quantization-conditioned backdoors and one pruning-conditioned backdoor), examining the algorithm's capability for localizing backdoor classes in emerging attacks.

We adopt differentiated evaluation strategies for datasets of different scales: for CIFAR-10 and ImageNet-12 with fewer classes, we exhaustively set each class as the backdoor class in turn to evaluate the backdoor class localization accuracy across all classes; for Tiny-ImageNet with 100 classes, evaluation is conducted by randomly sampling 30 classes. All experiments are repeated 3 times with random restarts. As shown in Table 2, whether on CNN or ViT architectures, traditional or conditional backdoor attacks, our selected localization algorithm achieves 100% localization accuracy.

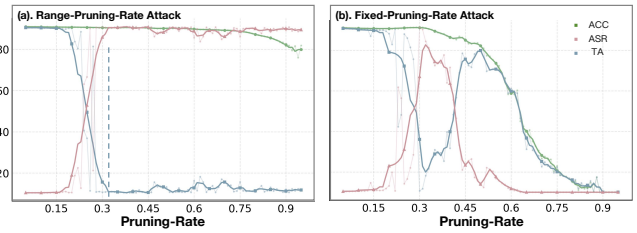



Figure 2. Left: Range-Pruning-Rate attack (backdoor exposed for pruning rates 0.3–0.9). Right: Fixed-Pruning-Rate attack (backdoor exposed at pruning rate 0.3).

Although novel backdoor class localization algorithms may emerge in the future, based on a comprehensive consideration of computational efficiency and practicality, we employ the unlearning-based backdoor class localization method as the pre-processing module of LMR.

**Pruning-Conditioned Backdoor**: In this work, we follow Tian et al.'s classification to categorize pruning-triggered backdoor attacks into two types: Fixed-Pruning-Rate backdoor attacks and Range-Pruning-Rate backdoor attacks. Fixed-Pruning-Rate backdoor attacks activate the backdoor only when the user's pruning rate matches the attacker's preset pruning value; otherwise, the backdoor remains dormant. This approach often has significant limitations in practice, because the attacker often cannot control the specific pruning rate value chosen by the user.

Range-Pruning-Rate backdoor attacks are more flexible. Attackers inject backdoors within commonly used pruning ranges, allowing users to effectively trigger the backdoor under different pruning rates, which significantly enhances the practical threat of pruning-as-backdoor attacks. However, compared to Fixed-Pruning-Rate attacks, Range-Pruning-Rate attacks are more difficult to optimize and the backdoor injection process is more time-consuming. In our experiments, we adopt an $L_1$-magnitude-based channel pruning strategy: for each output channel of the convolutional layers, we compute the $L_1$ norm of its weights and use this as a measure of channel importance to perform pruning. As shown in Fig. 2, Fixed-Pruning-Rate attacks are activated only near their specific pruning rate of 0.3,

Table 3. The defense effectiveness of LMR against pruning-triggered backdoors across different pruning rates. (CIFAR10)

| Model | No Defense | | | Setting | 0.2 | | | 0.3 | | | 0.4 | | | 0.5 | | | 0.6 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | TA | ASR | | ACC | TA | ASR | ACC | TA | ASR | ACC | TA | ASR | ACC | TA | ASR | ACC | TA | ASR |
| ResNet18 | **90.93** | **90.46** | **0.40** | LMR | 84.65 | 84.16 | 0.37 | 84.65 | 84.16 | 0.37 | 84.65 | 84.16 | 0.37 | 84.65 | 84.16 | 0.37 | 84.65 | 84.16 | 0.37 |
| | | | | Ori+P | 83.88 | 18.72 | 90.18 | 83.24 | 10.04 | 99.96 | 82.47 | 10.01 | 99.99 | 81.74 | 10.01 | 99.99 | 81.27 | 10.11 | 99.88 |
| | | | | ***LMR+P*** | 85.29 | 84.57 | **0.83** | 84.94 | 84.57 | **0.91** | 84.73 | 83.24 | **1.96** | 84.08 | 83.46 | **1.54** | 83.70 | 82.80 | **1.91** |
| VGG16 | **87.73** | **87.39** | **0.44** | LMR | 80.29 | 78.97 | 0.72 | 80.29 | 78.97 | 0.72 | 80.29 | 78.97 | 0.72 | 80.29 | 78.97 | 0.72 | 80.29 | 78.97 | 0.72 |
| | | | | Ori+P | 75.67 | 11.55 | 97.71 | 71.07 | 10.03 | 99.94 | 71.92 | 10.01 | 99.96 | 71.53 | 10.01 | 99.93 | 71.82 | 10.00 | 100.00 |
| | | | | ***LMR+P*** | 71.50 | 63.84 | **0.80** | 77.44 | 76.82 | **1.01** | 71.22 | 71.14 | **1.21** | 75.37 | 72.87 | **5.02** | 74.76 | 61.38 | **12.93** |
| MobileNetV2 | **90.87** | **90.66** | **0.34** | LMR | 86.50 | 85.50 | 0.21 | 86.50 | 85.50 | 0.21 | 86.50 | 85.50 | 0.21 | 86.50 | 85.50 | 0.21 | 86.50 | 85.50 | 0.21 |
| | | | | Ori+P | 81.06 | 10.06 | 99.91 | 80.45 | 10.00 | 100.00 | 81.74 | 10.00 | 100.00 | 80.67 | 10.01 | 99.99 | 78.58 | 10.03 | 99.97 |
| | | | | ***LMR+P*** | 86.28 | 82.49 | **0.36** | 86.14 | 81.17 | **0.34** | 85.52 | 80.69 | **0.42** | 84.83 | 74.69 | **2.14** | 82.27 | 64.02 | **2.91** |

while Range-Pruning-Rate attacks can be effectively triggered over a wide range of pruning rates (0.3–0.9). We further observe that even under high pruning rates (up to 0.95), the pruned model obtained from the Range-Pruning-Rate attack can still maintain clean accuracy close to that of the original model (for example, at a pruning rate of 0.9, clean accuracy only drops from 90% to 80%), and the changes it induces during the pruning process are more difficult to detect, thus possessing stronger stability and stealthiness.

We primarily evaluate the defense effectiveness of LMR against Range-Pruning-Rate backdoor attacks. As shown in Table 3, on the CIFAR-10 dataset, we inject BadNets backdoors (trigger: a 6×6 white square at the bottom-right corner of images) into ResNet, VGG, and MobileNet-V2 models within a pruning rate range of 0.3 to 0.9, and subsequently test LMR's defense performance in the 0.2-0.6 interval. Specifically, we use 2% of the total training data (approximately 1,000 samples) for purification and an additional 10% as a pruning calibration set to guide the pruning process.

Experimental results indicate that within the 0.3-0.6 pruning interval, the ASR of pruned models approaches 100%, with the backdoor being strongly activated. Although applying LMR causes the ACC of the full-precision model to decrease by approximately 5% on average, models obtained by pruning the purified model exhibit higher ACC in most cases compared to those pruned directly from the original model, while the ASR is significantly reduced to below 1% in the vast majority of scenarios. This demonstrates that LMR not only effectively defends against Pruning-As-Backdoor attacks but also further enhances the model's robustness to pruning perturbations.

**LMR Time Efficiency Analysis**: We conduct runtime analysis on a ResNet-18 + BadNets backdoored model using 500 CIFAR-10 samples on a single NVIDIA RTX 4090 GPU (averaged over 10 random runs). As shown in Fig. 3, LMR reduces the ASR to 0.26% in only 22.2 seconds, achieving stronger defense in less time compared to mainstream backdoor defense algorithms such as RNP and MNP (RNP requires about 30 seconds).

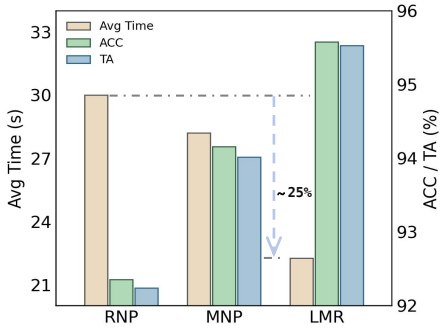



Figure 3. A comparison of defense time costs among RNP, MNP, and LMR.

## C. Theoretical Foundations and Mathematical Analysis of LMR

**Phase-1** constitutes the core of LMR's defense capability. This section provides a detailed interpretation of its underlying mechanisms, explaining why LMR can effectively eliminate backdoors.

**Phase-1 Objective:** The core of LMR lies in Phase-1, where—without using any triggered samples—we reshape the decision boundary in the logit space using only clean, non-backdoor data. For all non-backdoor samples, we enforce a negative safety gap between the backdoor class and other classes, ensuring that even triggered samples cannot cross the decision boundary.

**Notation:** Let $h(x) \in \mathbb{R}^d$ be the backbone feature right before the final fully connected (FC) layer. The FC classifier for class $j$ has parameters $(w_j, b_j)$, and its *logit* is:

$$z_j(x) = w_j^\top h(x) + b_j,\ z(x) = [z_1(x), \ldots, z_K(x)]. \quad (6)$$

The predicted class is $\arg\max_j z_j(x)$. We define the margin with respect to the backdoor class $c$ as:

$$m_c(x) = z_c(x) - \max_{j \neq c} z_j(x), \tag{7}$$

where $m_c(x) \leq 0$ means the sample is *not* assigned to the backdoor class (safe), and $m_c(x) > 0$ indicates risk of being assigned to the backdoor class.

**Feature increment under a trigger:** Let $\tau(x)$ denote the input after adding a trigger. Its net effect on the feature is:

$$v(x) \;:=\; h(\tau(x)) - h(x). \tag{8}$$

This treats the complex front-end nonlinearity as a feature increment seen by the FC layer.

**Clean-only training suppresses triggers—derivation:** We focus on the change of the backdoor margin before/after triggering:

$$\Delta(x) := m_c(\tau(x)) - m_c(x). \tag{9}$$

Let

$$b(x) \in \arg\max_{j \neq c} z_j(x) \tag{10}$$

be the strongest non–backdoor competitor before triggering (ties broken arbitrarily). Then

$$\begin{aligned}
\Delta(x) &= \Big[z_c(\tau) - \max_{j\neq c} z_j(\tau)\Big] - \Big[z_c(x) - \max_{j\neq c} z_j(x)\Big] \\
&\leq \Big[z_c(\tau) - z_{b(x)}(\tau)\Big] - \Big[z_c(x) - z_{b(x)}(x)\Big] \\
&= \big(z_c(\tau) - z_c(x)\big) - \big(z_{b(x)}(\tau) - z_{b(x)}(x)\big) \\
&= w_c^\top v(x) - w_{b(x)}^\top v(x) \quad \text{(bias terms cancel)} \\
&= (w_c - w_{b(x)})^\top v(x) \\
&\leq \big\|w_c - w_{b(x)}\big\|_* \cdot \big\|v(x)\big\|.
\end{aligned} \tag{11}$$

The second line upper-bounds the difference of two maxima using the concrete competitor $b(x)$; the fourth line follows from $z_j(\tau) - z_j(x) = w_j^\top v(x)$; the last line applies the dual-norm (Hölder/Cauchy–Schwarz) inequality.

To cover the data domain and possible competitors, take the supremum upper bound:

$$\beta \;:=\; \sup_x \; \big\|w_c - w_{b(x)}\big\|_* \cdot \big\|v(x)\big\|, \tag{12}$$

so that $\Delta(x) \leq \beta$ for any $x$.

If we enforce a uniform negative gap on *all clean, non–backdoor samples*

$$m_c(x) \;\leq\; -m_1, \qquad (\forall x,\; m_1 > 0), \tag{13}$$

and choose $m_1 \geq \beta$, then for any $x$:

$$m_c\big(\tau(x)\big) = m_c(x) + \Delta(x) \;\leq\; -m_1 + \beta \;\leq 0, \tag{14}$$

hence $\tau(x)$ will not be predicted as the backdoor class. In other words, by pushing the backdoor margin on clean, non–backdoor data far enough into the negative, we suppress the trigger effect without ever using triggered samples.

Guided by this derivation, Phase-1 optimizes on the clean non–backdoor set $\mathcal{S} = \{(x, y) : y \neq c\}$ the objective

$$L_{\mathrm{DSC}}(x) = \max\big(0,\; z_c(x) - \max_{j\neq c} z_j(x) + m_1\big) = \max\big(0,\; m_c(x) + m_1\big). \tag{15}$$

augmented with a stability term for non–backdoor classes

$$L_{\mathrm{CM}}(x, y) = \max\big(0,\; \max_{j\neq y} z_j(x) - z_y(x) + m_2\big), \tag{16}$$

and a small-weight cross-entropy, yielding the total loss:

$$\min_\theta \; \mathbb{E}_{(x,y)\in\mathcal{S}}\Big[\mathrm{CE}(y \,|\, x) + \alpha L_{\mathrm{DSC}}(x) + \beta L_{\mathrm{CM}}(x, y)\Big]. \tag{17}$$

**Practical choice of $m_1$:** $\beta$ is a worst-case theoretical upper bound and is not directly observable. In our experiments, we observed that using $m_1 = 3$ effectively suppresses the vast majority of backdoors.

Phase-1 in essence *builds a uniform safety gap for the backdoor margin on the clean non–backdoor distribution*. Since $\Delta(x) \leq \|w_c - w_{b(x)}\|_* \|v(x)\| \leq \beta$, choosing $m_1 \geq \beta$ guarantees that the trigger-induced margin increase cannot cross the decision boundary. Thus clean-only training suffices to suppress the backdoor and sets a stable starting point for Phase-2 pruning/refinement.

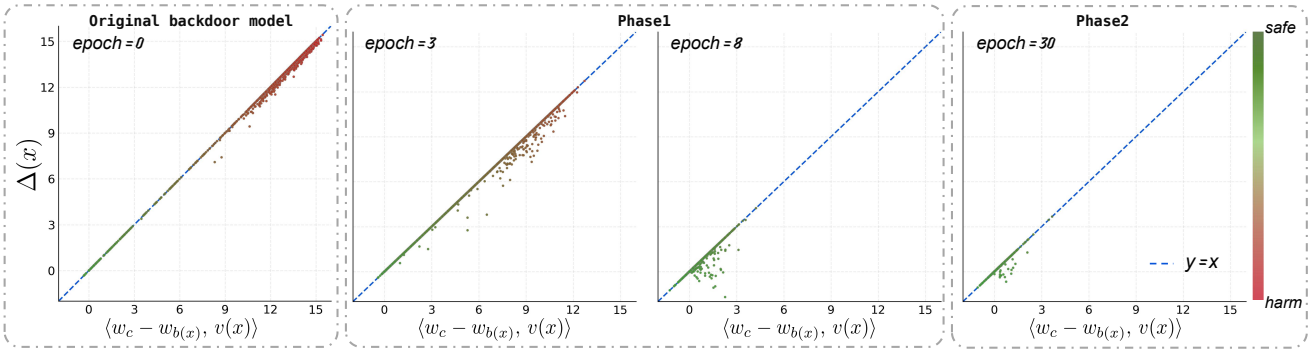
## D. More analysis and results about LMR

This section concretely illustrates the LMR workflow and the dynamic changes of key variables through experimental visualization.

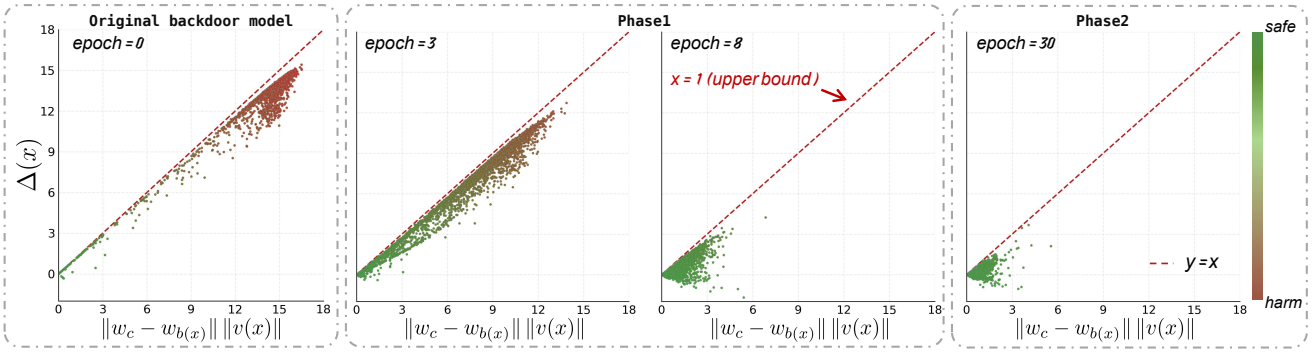
### D.1. Decision Projection and Upper Bound

As shown in Figs. 4a and 4b, let $w_c$ denote the classifier weight vector of the backdoor class, $w_{b(x)}$ the weight vector of the sample's *strongest non–backdoor class* under the current input ($b(x) = \arg\max_{j\neq c} z_j(x)$), $v(x) = h(\tau(x)) - h(x)$ the trigger–induced feature shift, and $\Delta(x) = m_c(\tau(x)) - m_c(x)$ the *margin increment* after adding the trigger. Define

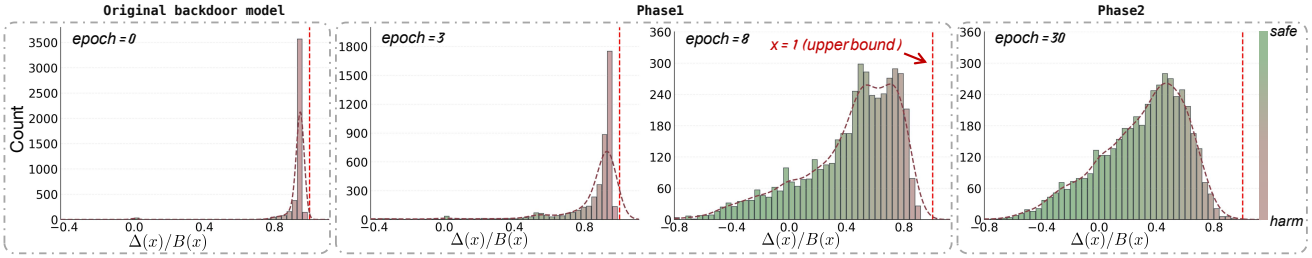
$$\begin{aligned}
B(x) &= \|w_c - w_{b(x)}\| \, \|v(x)\|, \\
\cos\theta &= \frac{\langle w_c - w_{b(x)},\, v(x)\rangle}{\|w_c - w_{b(x)}\| \, \|v(x)\|}.
\end{aligned} \tag{18}$$




(a) First-order check: $\Delta(x)$ vs. $\langle w_c - w_{b(x)},\, v(x)\rangle$ across datasets.




(b) $\Delta(x) = m_c(\tau(x)) - m_c(x)$ vs. $B(x) = \|w_c - w_{b(x)}\|\,\|v(x)\|$ with $y{=}x$ upper bound.




(c) Histogram of the tightness ratio $\Delta(x)/B(x)$.

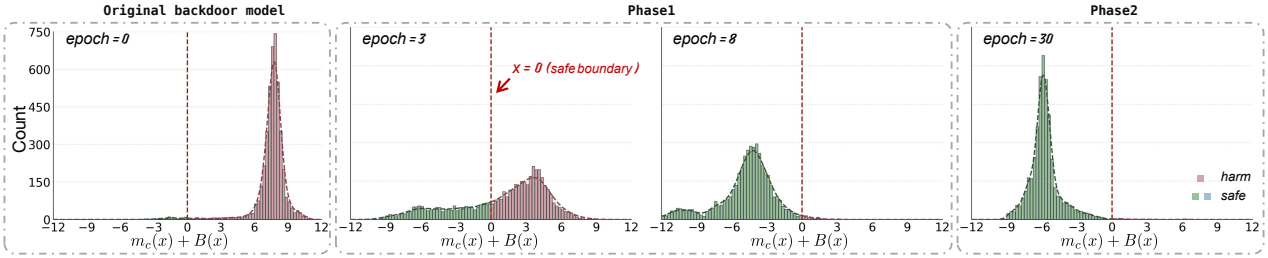
Figure 4. Upper-bound and tightness of the backdoor-margin increment across datasets.

Panel (a) takes the first–order projection $\langle w_c - w_{b(x)},\, v(x)\rangle$ to measure the *effective increment along the decision direction*; panel (b) uses the upper bound $B(x)$ to characterize the *attainable upper limit* jointly determined by the "decision strength" and the "perturbation magnitude". They satisfy:

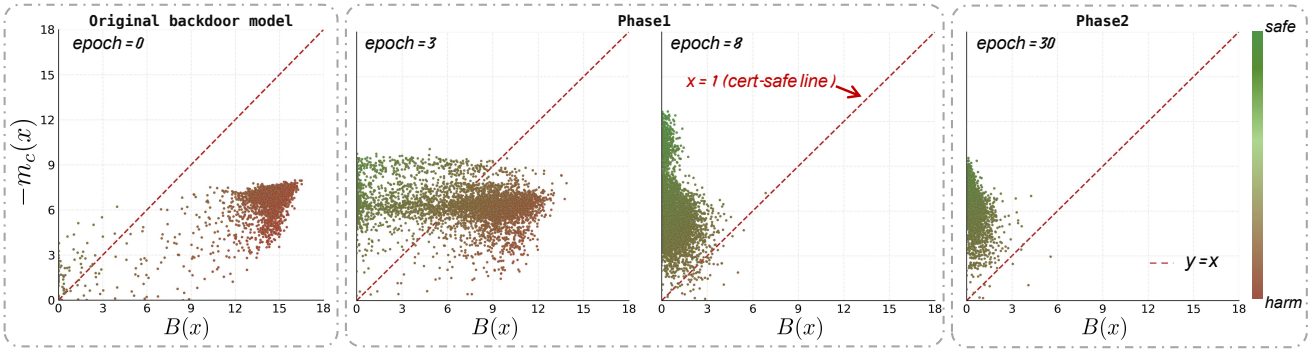
$$\begin{aligned}\langle w_c - w_{b(x)},\, v(x)\rangle &= B(x)\,\cos\theta,\\ \Delta(x) \approx \langle w_c - w_{b(x)},\, v(x)\rangle &\;\le\; B(x).\end{aligned} \tag{19}$$

where the blue dashed line $y = x$ in panel (a) denotes the first–order approximation ($\Delta \approx \langle w_c - w_{b(x)},\, v(x)\rangle$), and the red dashed line $y = x$ in panel (b) denotes the Cauchy–Schwarz upper bound ($\Delta \le B$).

In the initial backdoored model, the point clouds in panels (a) and (b) both lie near and extend along $y = x$: this indicates that the trigger mainly pushes features along $w_c - w_{b(x)}$ ($\cos\theta \approx 1$), so $\Delta(x)$ nearly equals the first–order projection and is also close to the attainable upper limit. After *Phase-1*, the points contract toward the origin along both axes: on the one hand, $\|v(x)\|$ decreases, tightening $B(x)$ (a smaller upper bound); on the other hand, $\cos\theta$ decreases, reducing the effective projection (lower alignment)—accordingly, points in panel (b) spread in a fan relative to $y = x$, while points in panel (a) still largely follow the near-$y = x$ relation. Consequently, $\Delta(x)$ drops markedly, samples move away from $y = x$, and the trigger can hardly push them across the decision boundary. After *Phase-2*, points in both plots remain stably concentrated




(a) Histogram of $m_c(x) + B(x)$ ($\leq 0$ implies certified safe under linearization).




(b) Certification scatter: $(-m_c(x),\ B(x))$ with $y{=}x$ cert-safe line.

Figure 5. Certification-oriented visualization of safety regions.

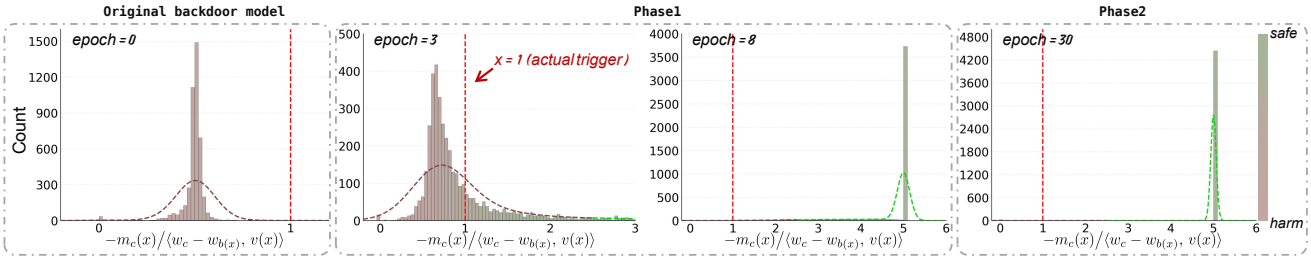


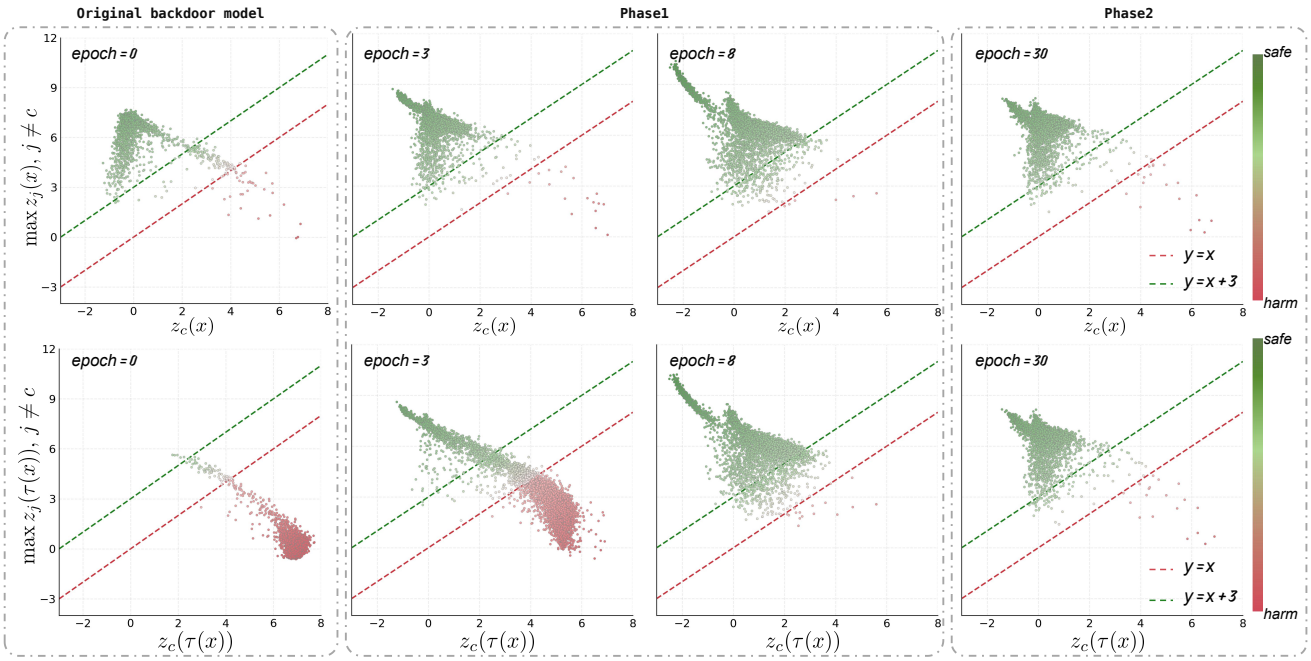

Figure 6. Required step along $v(x)$: histogram of $t^*(x) = \frac{-m_c(x)}{\langle w_c - w_{b(x)},\, v(x)\rangle}$. The dashed line marks $t = 1$ (actual trigger).

near the origin and maintain a robust gap from $y = x$.

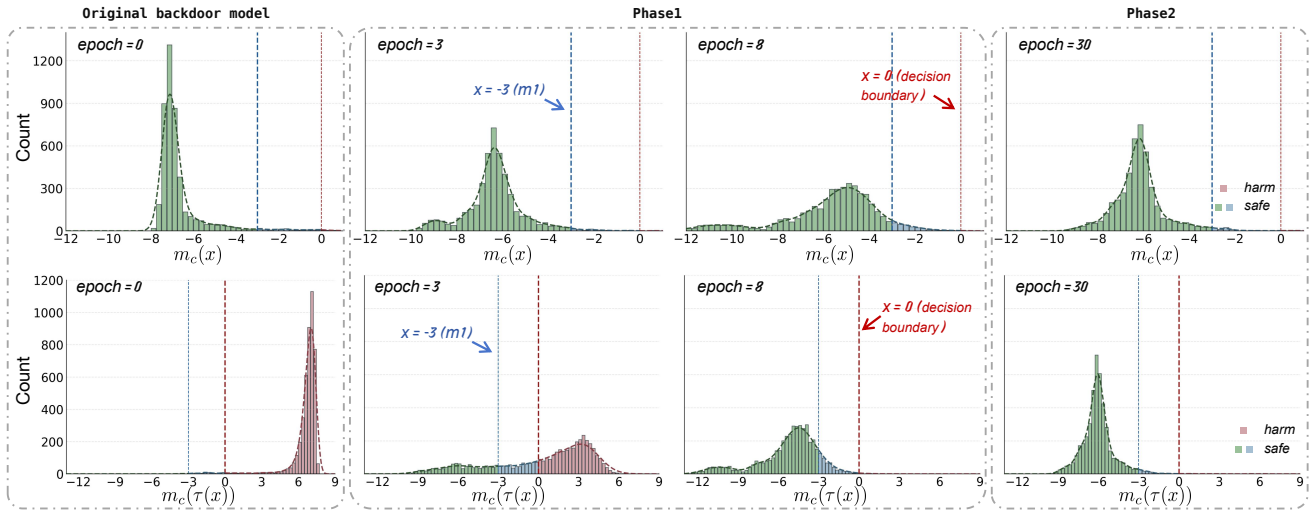
Fig. 4c then reports the histogram of the ratio $\Delta(x)/B(x)$ (red dashed line at 1): in the initial backdoored model, the ratio concentrates near 1 (close to the bound); after entering *Phase-1*, the distribution shifts left (farther from the bound); *Phase-2* further sustains the low–ratio regime. Overall, Phase-1 of LMR reduces $\|v(x)\|$ (thereby lowering $B(x)$) and decreases $\Delta(x)/B(x)$ (increasing slack relative to the bound), thus substantially suppressing the backdoor increment; Phase-2 consolidates performance without sacrificing this geometric safety.

## D.2. Certified Safety and Trigger Step

As shown in Fig. 5, we test on clean samples whether a worst–case trigger can still force a decision to the backdoor class. From the relations $m_c(\tau(x)) = m_c(x) + \Delta(x)$, $\Delta(x) \leq B(x)$, and $B(x) = \|w_c - w_{b(x)}\|\,\|v(x)\|$ (with $b(x) = \arg\max_{j \neq c} z_j(x)$ and $v(x) = h(\tau(x)) - h(x)$), we obtain the sufficient condition $m_c(x) + B(x) \leq 0$ (equivalently, $-m_c(x) \geq B(x)$), under which the sample is *certified safe*. Panel (a) shows the histogram of $m_c(x) + B(x)$ with the red dashed line $x = 0$ as the certification threshold; panel (b) plots the points $\big(-m_c(x),\ B(x)\big)$ with the red dashed line $y = x$ as the certification boundary (points above satisfy $-m_c(x) \geq B(x)$). Across columns: in the initial backdoored model, most mass lies to the right of 0 and many points fall below $y = x$, indicating that in the worst case a considerable portion of clean samples could be flipped; after Phase–1, the histogram shifts left of 0 and the scatter lifts above $y = x$, reflecting larger negative margins $-m_c(x)$ and smaller bounds $B(x)$ (either reduced $\|v(x)\|$ or weaker alignment with $w_c - w_{b(x)}$); after Phase–2, clean




(a) **Clean (non-backdoor):** $(z_c(x), \max_{j\neq c} z_j(x))$; **Triggered (non-backdoor):** $(z_c(\tau(x)), \max_{j\neq c} z_j(\tau(x)))$.




(b) Histograms of $m_c(x)$ for clean non-backdoor samples and $m_c(\tau(x))$ under the trigger.

Figure 7. Logit geometry before/after trigger on CIFAR-10.

accuracy is recovered while this certified–safe configuration is maintained.

And in Fig. 6, let the trigger–induced feature change be $v(x) = h(\tau(x)) - h(x)$, and consider the linear path $h_t = h(x) + t\,v(x)$ with $t \in [0, 1]$. The margin between the backdoor class and the current strongest non–backdoor class is $m_c(x) = z_c(x) - \max_{j\neq c} z_j(x)$. Under a linear approximation of the last layer,

$$m_c(h_t) \approx m_c(x) + t\,\langle w_c - w_{b(x)},\, v(x)\rangle. \tag{20}$$

Setting $m_c(h_{t^*}) = 0$ yields the step required to hit the decision boundary:

$$t^*(x) \approx -\frac{m_c(x)}{\langle w_c - w_{b(x)},\, v(x)\rangle}. \tag{21}$$

The denominator equals $\|w_c - w_{b(x)}\|\,\|v(x)\|\cos\theta$, i.e., it depends jointly on the trigger magnitude $\|v(x)\|$ and its alignment $\cos\theta$ with $w_c - w_{b(x)}$. The red dashed line marks the actual trigger strength $t = 1$. In the initial backdoored model, most mass lies at $t^* < 1$ (a weaker–than–actual trig-

ger already suffices to cross the boundary); after Phase–1, the distribution shifts to $t^* > 1$ (a larger step is required and the actual trigger is no longer sufficient); Phase–2 restores clean accuracy while maintaining this right–shifted pattern, with $t^*$ mostly $> 1$ and only a small long tail near the boundary.

### D.3. Logit Distribution Shift

As shown in Fig. 7a, the horizontal axis is the backdoor-class logit $z_c(\cdot)$, and the vertical axis is $\max_{j \neq c} z_j(\cdot)$ (the maximum non-backdoor logit). The red dashed line $y = x$ denotes the decision boundary between the backdoor class and the strongest non-backdoor class; we also draw the green dashed line $y = x + m_1$ (with $m_1 = 3$ in our experiments) as the target safety margin. Points above $y = x$ will not be misclassified as the backdoor class, whereas points below $y = x$ are classified as the backdoor class. The first row of panel (a) shows the distribution of clean samples, and the second row shows the distribution of triggered samples: in the initial backdoored model (epoch$= 0$), the vast majority of clean samples lie above $y = x$, with some points clustering near the boundary (indicating that the backdoor-class logit is occasionally elevated on a few clean samples), while many triggered samples fall below $y = x$. During Phase-1 (epoch$=$ 1–8), those clean points originally below $y = x$ gradually move upward and cross $y = x$; the cloud of triggered points approaches the safety boundary and returns to above $y = x$, indicating that the trigger-induced backdoor gain $\Delta(x) = m_c(\tau(x)) - m_c(x)$ is markedly reduced and the trigger can no longer easily push samples into the backdoor decision region. By the end of Phase-2 (epoch$=$ 9–30), the overall geometry of clean and triggered samples becomes more stable and compact, and most points maintain a logit margin to the backdoor class greater than $m_1$.

And in Fig. 7b, the first row reports the margin of clean samples with respect to the backdoor class, $m_c(x) = z_c(x) - \max_{j \neq c} z_j(x)$ (the blue dashed line at $-m_1$ marks the negative-margin target enforced in Phase-1), and the second row reports the margin distribution after adding the trigger, $m_c(\tau(x))$ (the red dashed line at $0$ is the decision boundary). In the initial backdoored model, the maximal non-backdoor logit of clean samples typically exceeds the backdoor-class logit (their difference is about 6–8); once the trigger is added, the backdoor-class logit rapidly increases past the boundary, leading to misclassification. After entering Phase-1, part of the clean distribution shifts left and crosses the blue dashed line at $-m_1$; meanwhile, the triggered distribution moves to the left of $0$, and the effective trigger gain approaches zero. Phase-2 further restores overall accuracy while preserving the above geometric safety: the clean distribution remains at $\leq -m_1$, and the triggered distribution mostly stays to the left of $0$ with only a small tail near the boundary. Using the relation $m_c(\tau(x)) = m_c(x) + \Delta(x)$, we see that Phase-1 simultaneously makes $m_c(x)$ more negative and reduces $\Delta(x)$, making boundary crossings by the trigger unlikely; these results indicate that explicitly suppressing the backdoor class on non-backdoor samples using only clean data effectively stabilizes the decision boundary, consistent with Appendix C's conclusion that enlarging the logit margin via clean data can significantly weaken the backdoor effect.