Backdoor Attacks under Lossy Compression: From Failure to Reactivation and Adaptation

Qian Li[1,2] **Yunuo Chen**[1] **Yuntian Chen**[2*]
[1]Shanghai Jiao Tong University [2]Eastern Institute of Technology

Abstract

Real-world backdoor attacks often require poisoned datasets to be stored and transmitted before they are used to compromise deep learning systems. In the era of big data, however, the inevitable use of lossy compression poses a fundamental challenge to invisible backdoor attacks. We observe that triggers embedded in RGB images can become ineffective once the images are lossily compressed into binary bitstreams, such as JPEG files, for storage and transmission. Consequently, poisoned data may lose their malicious functionality after compression, causing backdoor injection to fail. Prior compression-based attacks typically exploit compression artifacts as certain triggers to distinguish poisoned RGB samples from uncompressed benign ones, rather than addressing whether malicious information can survive a shared lossy storage-and-transmission pipeline. In this paper, we highlight the necessity of explicitly accounting for lossy compression in backdoor attacks. This requires attackers to ensure that transmitted binary bitstreams preserve malicious trigger information, such that effective triggers can be induced after decompression. Building on the region-of-interest (ROI) coding mechanism in image compression, we propose two poisoning strategies tailored to inevitable lossy compression. First, we introduce **Universal Attack Reactivation**, a general method that uses sample-specific ROI masks to reactivate trigger information in bitstreams for learned image compression (LIC). Second, we present **Compression-Adapted Attack**, a new attack strategy that employs customized ROI masks to encode trigger information into bitstreams and applies to both traditional codecs and LIC. Extensive experiments demonstrate the effectiveness of both strategies.

1 Introduction

In recent years, the widespread deployment of deep learning systems has exposed them to a broad range of security threats [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Among them, data poisoning-based backdoor attacks pose a particularly severe risk. In such attacks, adversaries inject malicious "triggers" into a subset of training data and release the poisoned dataset to downstream users [14, 15, 16, 17, 18]. Models trained on such datasets behave normally on benign inputs, but exhibit attacker-controlled predictions once the trigger is present. Due to their stealthiness and strong controllability, backdoor attacks have become a major threat to the security and reliability of machine learning systems.

Most existing backdoor attacks are designed and evaluated in an idealized RGB-domain setting, where poisoned images are assumed to be directly used for model training after trigger injection. However, this assumption often deviates from real-world data pipelines. In practical scenarios, poisoned datasets typically need to be stored, transmitted, downloaded, and decoded before they are used to compromise deep learning systems. With the rapid growth of data volume and increasing demand for efficient storage and network transmission, lossy image compression has become increasingly inevitable [19, 20, 21]. This introduces a fundamental challenge to invisible backdoor attacks: triggers

*Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

embedded in RGB images may not survive the conversion from RGB data to compressed binary bitstreams and back to decompressed RGB images.

Formally, many invisible attacks construct poisoned samples by directly modifying RGB images, e.g., $\boldsymbol{x}_p = \boldsymbol{x} + \boldsymbol{\delta} \in \mathbb{R}^{3\times H\times W}$. Yet, once $\boldsymbol{x}_p$ is lossily compressed into a binary bitstream $\hat{\boldsymbol{y}} \in \{0,1\}^n$, such as a JPEG file, there is no guarantee that the malicious trigger information is preserved in $\hat{\boldsymbol{y}}$. After decompression, the recovered poisoned image $\hat{\boldsymbol{x}}_p \in \mathbb{R}^{3\times H\times W}$ may no longer contain an effective trigger. Since lossy compression inevitably introduces distortion, i.e., $\hat{\boldsymbol{x}} \neq \boldsymbol{x}$, it can attenuate or destroy trigger signals. As a result, poisoned data that are effective in the raw RGB domain may lose their malicious functionality after storage and transmission, causing backdoor injection to fail. This phenomenon affects a broad range of existing invisible attacks, including both invisible-trigger methods and clean-label attacks [22, 23, 24, 25, 26, 27].

We observe that this failure is closely related to the frequency characteristics of invisible triggers. Inspired by prior studies on frequency-domain backdoor behavior [28, 29], we find that many invisible triggers rely on subtle high-frequency components to achieve stealthiness. These components are visually unobtrusive, but often play a crucial role in triggering the backdoor. As shown in Fig. 1, stronger compression substantially reduces the attack success rate, and explicitly removing high-frequency components from poisoned test samples also leads to a significant ASR drop. These observations suggest that invisible triggers achieve stealth by hiding in compression-sensitive high-frequency information, while lossy compression tends to suppress or distort exactly such information. Therefore, ignoring the compression stage can lead to overly optimistic evaluations of backdoor attacks and render poisoned datasets ineffective in real-world storage-and-transmission pipelines.

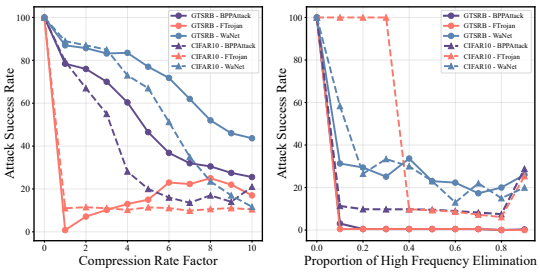


Figure 1: Ineffectiveness of backdoor attacks under lossy compression. **Left:** Higher compression rates introduce stronger distortion and substantially hinder backdoor injection, indicating that lossy compression can severely damage invisible triggers. **Right:** Removing high-frequency components from poisoned test samples via Fourier Transform significantly reduces the ASR, suggesting that high-frequency information is critical for many invisible triggers.

It is worth distinguishing our setting from prior compression-based backdoor attacks. *Existing compression-based methods [30, 2] typically remain trigger-centric*: they exploit compression artifacts as triggers to distinguish poisoned RGB samples from *uncompressed benign samples*. In contrast, they do not explicitly address whether malicious information can survive a shared lossy storage-and-transmission pipeline, where *both benign and poisoned data are compressed into binary bitstreams*. Our goal is therefore not merely to use compression artifacts as another form of trigger. Instead, we study backdoor attacks under inevitable lossy compression, where the attacker must ensure that the transmitted bitstream recovers or induces malicious trigger information after decompression.

To address this challenge, we explicitly incorporate lossy compression into the design of backdoor attacks. Rather than evaluating the toxicity of raw RGB poisoned images alone, attackers must ensure that both the compressed bitstream $\hat{\boldsymbol{y}}$ and the decompressed image $\hat{\boldsymbol{x}}_p$ carry effective malicious information. We achieve this by leveraging the Region of Interest (ROI) coding mechanism in image compression [31, 32, 33, 34, 35]. ROI coding allows codecs to assign different bitrates to different image regions, thereby controlling the spatial distribution of compression distortion. This provides a natural interface for manipulating how trigger-related information is preserved or encoded during compression, without modifying the standard decompression process used by victims.

Following this principle, we propose two ROI-based poisoning strategies tailored to inevitable lossy compression from failure to reactivation and adaptation. The first, **Universal Attack Reactivation**, aims to reactivate existing invisible attacks that become ineffective after compression by preserving trigger-relevant information during learned image compression (LIC) [36, 37, 38, 39, 40]. The second, **Compression-Adapted Attack** (CAA), goes one step further by designing triggers that are inherently adapted to the compression process, using customized ROI masks to encode malicious information into compressed bitstreams. Together, these two strategies cover complementary paths: reactivating

fragile RGB-domain triggers after compression, and constructing compression-adapted triggers that remain effective through lossy storage and transmission.

Our contributions are summarized as follows:

- We reveal that previously effective invisible triggers can become ineffective after lossy compression. This finding highlights the necessity of explicitly accounting for inevitable storage-and-transmission compression in data poisoning-based backdoor attacks.
- We propose **Universal Attack Reactivation**, a general ROI-based method for LICs that reactivates invisible triggers degraded by lossy compression. Extensive experiments show that it significantly improves the attack success rates of prior invisible attacks under compression.
- We propose **Compression-Adapted Attack** (CAA), a new invisible backdoor attack for both LICs and traditional codecs. CAA encodes trigger information through customized ROI masks and achieves strong attack performance and stealthiness under lossy compression.

2 Related Work

2.1 Invisible Backdoor Attack

Gu *et al.* first introduced a backdoor attack by their BadNets proposal [8], which has sparked a lot of research interest on backdoor attack [41, 12, 42, 43, 44, 45]. Recently, researchers have increasingly focused on the invisibility of backdoor triggers to develop more powerful attacks. ISSBA [46] embedded class-specific strings into images using steganography techniques. WaNet [24] applied image warping as triggers, making the modifications hard to notice. BppAttack [22] took advantage of vulnerabilities in the human visual system, reducing visible changes through image quantization and improving trigger accuracy with contrastive learning. FTrojan [23] used perturbations in the frequency domain, spreading subtle pixel-wise changes throughout the image. Additionally, Rao *et al.* [29] proposed a backdoor attack method based on semantic information in the frequency domain. Clean-label attack [26, 27, 47, 48] is a special type of invisible backdoor attack that implants a backdoor without altering sample labels, enabling stealthier data poisoning. Its triggers are typically tiny, high-frequency adversarial perturbations and are therefore especially vulnerable to distortion from lossy compression. However, existing invisible backdoors overlook lossy compression effects, which remove high-frequency trigger components and can severely degrade attack performance.

2.2 Lossy Compression and Compression-Aware Backdoor Attacks

Lossy image compression reduces storage costs by discarding perceptually redundant information, making it increasingly important as image data volumes grow. Traditional codecs such as JPEG [49], JPEG2000[50], and BPG [51] rely on hand-crafted transforms and entropy coding. More recently, learned image compression (LIC) has achieved stronger rate-distortion performance with end-to-end neural networks and a variational autoencoder framework [52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67], driving rapid progress in image compression.

Several recent studies have also explored the connection between lossy compression and backdoor attacks. Existing compression-based attacks [30, 2] typically remain trigger-centric: they exploit compression artifacts as triggers to distinguish poisoned RGB samples from *uncompressed benign samples*. Therefore, their effectiveness largely relies on a distributional gap between compressed poisoned samples and uncompressed benign samples in the RGB domain. In the inevitable compression setting considered in this paper, however, both benign and poisoned data are stored and transmitted through a shared lossy compression pipeline. As a result, the compression-artifact triggers used by these methods no longer provide a reliable discriminative signal, and can become ineffective similarly to conventional invisible triggers after compression and decompression (discussed in Sec. 4).

Another related line of work studies compression-resistant backdoor attacks [68, 69, 70, 71]. While such methods improve robustness against compression, they typically require the attacker to participate in or control the training process of a specific classification model, for example by modifying the training objective. This differs from the dataset-only poisoning setting considered in this paper, where the attacker only releases a poisoned dataset and the victim trains a model using an arbitrary standard training pipeline. Our work focuses on this more practical setting and explicitly asks whether malicious information can survive storage and transmission as compressed bitstreams and be recovered or induced after decompression. Thus, lossy compression is not merely treated as a trigger source, but as an unavoidable communication channel in real-world data poisoning pipelines.

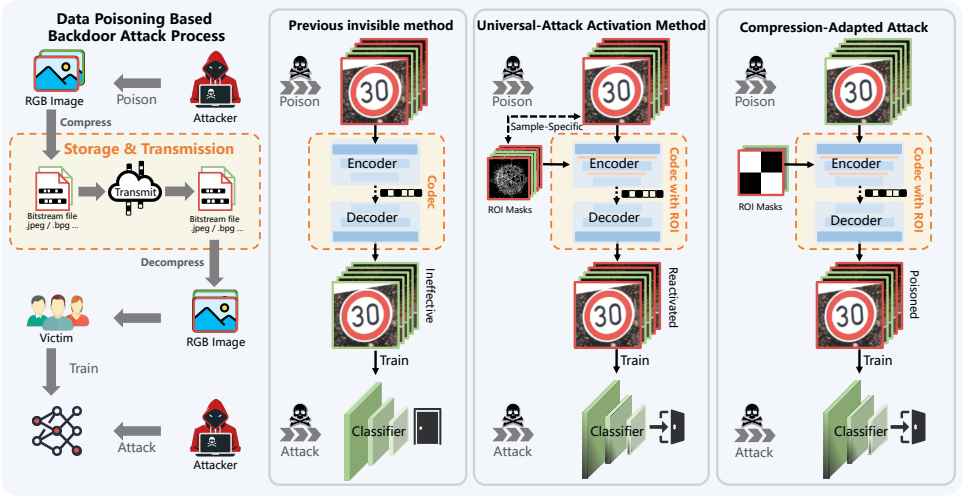


Figure 2: **Overview of backdoor attacks under lossy compression.** Red borders denote poisoned samples, and green borders denote benign samples. **Column 1** shows the real-world dataset-only poisoning pipeline, where data are stored and transmitted as compressed bitstreams before victim-side training. **Column 2** illustrates that previous invisible attacks may fail after lossy compression because trigger information is destroyed. **Columns 3-4** present our two ROI-based strategies: **Universal Attack Reactivation**, which reactivates ineffective invisible attacks by preserving trigger-relevant information, and **Compression-Adapted Attack**, which encodes malicious information into bitstreams to construct compression-adapted triggers for both traditional codecs and LICs.

3 Backdoor Attacks under Lossy Compression

3.1 Threat Model and Problem Setup under Lossy Compression

Fig. 2 illustrates the dataset-only poisoning pipeline considered in this paper. Unlike the idealized RGB-domain setting, poisoned and benign images are first stored and transmitted as bitstreams before being used for victim-side training. Given an image $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$, a lossy codec encodes it into a binary bitstream $\boldsymbol{y}$, quantizes it to $\hat{\boldsymbol{y}}$, and later decodes it back into a reconstructed RGB image $\hat{\boldsymbol{x}}$.:

$$\text{Compression}: \quad \hat{\boldsymbol{y}} = \mathrm{Q}(\mathrm{Encoder}(\boldsymbol{x})), \quad \text{Decompression}: \quad \hat{\boldsymbol{x}} = \mathrm{Decoder}(\mathrm{DeQ}(\hat{\boldsymbol{y}})). \tag{1}$$

where Q and DeQ denote quantization and dequantization, respectively. This compression–decompression process is central to our setting. Previous invisible attacks usually construct poisoned samples in the RGB domain, e.g., $\boldsymbol{x}_p = \boldsymbol{x} + \boldsymbol{\delta}$, but do not ensure that the trigger information survives the mapping $\boldsymbol{x}_p \to \hat{\boldsymbol{y}}_p \to \hat{\boldsymbol{x}}_p$. As lossy compression introduces distortion, the trigger embedded in $\boldsymbol{x}_p$ may be attenuated or removed from the bitstream $\hat{\boldsymbol{y}}_p$, making the decompressed sample $\hat{\boldsymbol{x}}_p$ ineffective for backdoor injection. Therefore, attacks under inevitable lossy compression must explicitly ensure that malicious information is preserved, recovered, or induced through the compressed bitstream. We consider the following dataset-only poisoning setting:

1. Targets. The attacker aims to release a poisoned dataset whose decompressed samples can successfully implant a backdoor into a victim-trained model, while maintaining normal accuracy on decompressed benign samples. We consider storage and transmission with publicly available codec implementations or specifications, so the attack is evaluated under realistic compression pipelines.

2. Capabilities. The attacker can poison data before storage and transmission and can control the sender-side encoding process, including configuring or modifying the encoder using public codec specifications and parameters. The output, however, must be a standard-compliant bitstream that can be decoded by the victim's unchanged decoder.

3. Constraints. The attacker cannot modify the victim-side decoder, training algorithm, or model architecture, nor can they require the victim to use a specially designed decoder. All poisoned files must be valid compressed bitstreams with normal decoding behavior, normal bitrate and file size, and visually normal decompressed images.

3.2 ROI-Supported Compression

To manipulate how information is preserved during compression without modifying the decoder, we leverage Region of Interest (ROI) coding, a common functionality in image compression [49, 51, 31,

72, 32, 33]. An ROI-supported encoder takes an additional mask $\boldsymbol{M} = [m_{ij}] \in \mathbb{R}^{H \times W}$ as input:

$$\text{Compression}: \quad \hat{\boldsymbol{y}} = \mathrm{Q}(\mathrm{Encoder}(\boldsymbol{x}, \boldsymbol{M})). \tag{2}$$

The ROI mask is used only as an encoder-side control signal. Each mask value m_{ij} controls the bitrate allocation weight at spatial location (i, j). A larger weight allocates more bits to the corresponding region and thus reduces compression distortion, while a smaller weight allows stronger distortion. Therefore, ROI coding provides a natural interface for controlling which image information is preserved in the transmitted bitstream.

This interface is well suited to our goal. Instead of introducing additional backdoor modules or requiring a modified decoder, ROI masks allow the attacker to guide the standard compression process itself. We use this mechanism to design two complementary attack routes under lossy compression. The first route, introduced in Sec. 3.3, targets LIC models with pixel-level ROI support and reactivates existing invisible triggers by preserving trigger-relevant information. The second route, introduced in Sec. 3.4, targets both traditional codecs and LICs with region-level ROI support and constructs compression-adapted triggers by encoding malicious information through customized ROI masks.

For LICs, ROI support can be integrated through spatially adaptive modules such as Spatial Feature Transform (SFT) [73, 32]. To satisfy our constraints, such modules are used only on the encoder side, while the decoder remains standard and unchanged. For traditional codecs, we directly rely on existing ROI-supported implementations [49, 51, 50].

3.3 Universal Attack Reactivation (UAR): Reactivating Fragile Triggers

Existing invisible backdoor attacks provide diverse trigger designs, but mostly become ineffective after compression as the trigger information is degraded under compression. Instead of redesigning these attacks, our first route aims to *reactivate* them by guiding the compression process to preserve trigger-relevant information. This strategy is particularly suitable for LICs with pixel-level ROI support, where fine-grained bitrate allocation can be controlled through sample-specific ROI masks.

Given a clean image $\boldsymbol{x}$ and its poisoned counterpart $\boldsymbol{x}_p$ generated by an existing attack, a direct way to identify trigger-relevant regions is to measure their residuals as ROI masks:

$$\boldsymbol{M}_{\text{res}} = \mathrm{Norm}\big(\mathrm{Mean}|\boldsymbol{x} - \boldsymbol{x}_p|\big), \tag{3}$$

where $\mathrm{Mean}(\cdot)$ averages over RGB channels, and $\mathrm{Norm}(\boldsymbol{z}) = \frac{\boldsymbol{z} - \min(\boldsymbol{z})}{\max(\boldsymbol{z}) - \min(\boldsymbol{z})}$ normalizes the mask values. This mask allocates more bits to regions with larger RGB-domain residuals, which is especially useful for additive or localized invisible triggers.

However, residual magnitude alone may not fully capture the information that is most vulnerable to compression. As discussed in Sec. 1 and Fig. 1, many invisible triggers rely on subtle high-frequency components, which are crucial for attack activation but are also heavily distorted by lossy compression. Motivated by this observation, we further design a frequency-based ROI mask to preserve compression-sensitive trigger components. Specifically, we extract high-frequency information from $\boldsymbol{x}_p$ and map it back to the image domain:

$$\boldsymbol{M}_{\text{freq}} = \mathrm{Norm}\big(\mathrm{Mean}|\mathcal{F}^{-1}(h_t(\mathcal{F}(\boldsymbol{x}_p)))|\big), \tag{4}$$

where $\mathcal{F}$ denotes the Fast Fourier Transform and h_t is a high-frequency truncation function with threshold t. Compared with $\boldsymbol{M}_{\text{res}}$, $\boldsymbol{M}_{\text{freq}}$ focuses on preserving the high-frequency trigger information that is most likely to be suppressed during compression.

With either $\boldsymbol{M}_{\text{res}}$ or $\boldsymbol{M}_{\text{freq}}$, the decompressed poisoned sample is obtained by $\hat{\boldsymbol{x}}_p = \mathrm{Decoder}\big(\mathrm{DeQ}(\mathrm{Q}(\mathrm{Encoder}(\boldsymbol{x}_p, \boldsymbol{M})))\big), \quad \boldsymbol{M} \in \{\boldsymbol{M}_{\text{res}}, \boldsymbol{M}_{\text{freq}}\}$. By allocating more bitrate to trigger-relevant regions, the compressed bitstream preserves more malicious information, allowing the trigger to be recovered after decompression.

A potential issue is that ROI masks themselves may introduce an unintended shortcut if their distortion patterns differ systematically between poisoned and benign samples. To avoid this, we also apply trigger-derived ROI masks to benign samples. For each benign image $\boldsymbol{x}_b$, we randomly select a poisoned sample $\tilde{\boldsymbol{x}}_p$ from the poisoned set $\boldsymbol{X}_p$ and use its corresponding mask $\boldsymbol{M}(\tilde{\boldsymbol{x}}_p)$ during compression as $\hat{\boldsymbol{x}}_b = \mathrm{Decoder}\big(\mathrm{DeQ}(\mathrm{Q}(\mathrm{Encoder}(\boldsymbol{x}_b, \boldsymbol{M}(\tilde{\boldsymbol{x}}_p))))\big)$. This prevents the ROI-induced distortion pattern from becoming a new label-correlated trigger, ensuring that the reactivated attack still relies on the original invisible trigger information.

3.4 Compression-Adapted Attack (CAA): Adapting Triggers to Compression

The reactivation strategy preserves existing triggers through compression. We now move to the second route: *adaptation*. Different from prior compression-aware attacks that treat compression artifacts as triggers or exploit distributional gaps between compressed poisoned samples and uncompressed benign samples, our goal is not to use compression merely as an additional trigger source. Instead, **we design a new trigger that is intrinsically coupled with the lossy storage-and-transmission process**. Specifically, CAA customizes ROI-guided bitrate allocation to encode malicious information into compressed bitstreams, which is then induced as a trigger after standard decompression. Thus, compression serves as a controlled carrier of the trigger rather than an incidental artifact source.

The key idea is to use ROI masks to control the spatial distribution of compression distortion. Regions assigned larger ROI weights preserve more high-frequency details, while regions assigned smaller weights suffer stronger distortion. Therefore, by applying different ROI masks to poisoned and benign samples, the attacker can induce a distinctive yet visually imperceptible frequency-distribution pattern in decompressed poisoned images. This pattern is generated through the compressed bitstream and serves as a compression-adapted trigger.

Concretely, we use a customized ROI mask $\boldsymbol{M}^*$ for poisoned samples and a uniform mask $\boldsymbol{M}_{\text{uniform}}$ for benign samples. The decompressed poisoned and benign samples are respectively given by

$$\hat{\boldsymbol{x}}_p = \text{Decoder}\big(\text{DeQ}(\text{Q}(\text{Encoder}(\boldsymbol{x}_p, \boldsymbol{M}^*)))\big), \tag{5}$$

$$\hat{\boldsymbol{x}}_b = \text{Decoder}\big(\text{DeQ}(\text{Q}(\text{Encoder}(\boldsymbol{x}_b, \boldsymbol{M}_{\text{uniform}})))\big). \tag{6}$$

Unlike conventional invisible attacks that add perturbations in the RGB domain, CAA encodes malicious information through bitrate allocation during compression. Thus, the trigger is not destroyed by the storage-and-transmission pipeline; instead, it is induced by this pipeline after decompression.

To improve robustness, we instantiate $\boldsymbol{M}^*$ with global and repetitive spatial patterns [74], including checkerboard and concentric-square masks, as shown in Fig. 3. White regions are assigned higher bitrate weights than black regions, preserving more high-frequency details after compression. After decompression, these masks induce visually subtle but target-correlated frequency-distribution patterns. More effective mask designs can be explored within this framework.



(a) Checkerboard (b) Concentric squares

Figure 3: Two customized ROI masks.

4 Experiments

4.1 Experimental Setting

Datasets and Metrics. We evaluate backdoor attack methods on GTSRB [75], CIFAR-10 [76], and CelebA [77]. Following [24, 22], we use eight CelebA classes. All images are resized to 64×64 and treated as raw images. We report Benign Accuracy (BA), Attack Success Rate (ASR) [78], and Attack Stealthiness (AS) to demonstrate the effectiveness of each attack method.

Classifiers. We use ResNet [79], Pre-activation ResNet [80], MobileNetV2 [81], and SENet [82]. Models are trained with Adam and an initial learning rate of 0.01, decayed by $10\times$ every 100 epochs.

Codecs. We evaluate ROI-supported traditional codecs, including BPG [51] and JPEG 2000 [50], and ROI-integrated LIC models, including Mbt-2018 [83], Cheng-2020 [84], and ELIC [85].

Baselines. We compare with three invisible attacks, WaNet [24], FTrojan [23], and BppAttack [22], two clean-label attacks, SAA [26] and Narcissus [27], and two compression-based methods, ECA [30] and CBA [2]. We follow their default settings. All experiments are conducted on A100 GPUs.

4.2 Evaluation of Universal Attack Reactivation

Reactivation of invisible attacks. We first evaluate whether UAR can restore existing invisible backdoor attacks that fail after lossy compression. Tab. 1 (*ReA* denotes the UAR-reactivated variant of a baseline attack) reports all-to-one attack results on different datasets and backbones. For baseline attacks and clean models, images are compressed with the uniform ROI mask (standard compression). For UAR, poisoned images are compressed with the frequency-aware mask $\boldsymbol{M}_{\text{freq}}$, while benign images are compressed with trigger-derived masks $\boldsymbol{M}(\boldsymbol{x}_p)$ as defined in Sec. 3.3. All results use the ROI-supported Mbt-2018 codec [83], and the poisoning rates follow the original baseline settings.

Although FTrojan [23], BppAttack [22], and WaNet [24] achieve high ASR in uncompressed settings, their effectiveness drops substantially after inevitable lossy compression. This confirms that triggers

Backbone ↓	Dataset ↓	Method →	Clean	FTrojan	$\text{FTrojan}_{\text{ReA}}$	WaNet	$\text{WaNet}_{\text{ReA}}$	BppAttack	$\text{BppAttack}_{\text{ReA}}$	CAA
ResNet18	CIFAR-10	BA(%)	93.57	92.21	91.32(-0.89↓)	84.72	91.75(+7.03↑)	88.03	90.85 (+2.82↑)	**93.28**
		ASR(%)	-	12.53	94.91(+82.38↑)	77.58	98.43 (+20.85↑)	44.17	96.23 (+52.06↑)	**100.00**
	GTSRB	BA(%)	99.22	95.72	97.43(+1.71↑)	94.43	98.25 (+3.82↑)	90.78	94.84 (+4.06↑)	**99.13**
		ASR(%)	-	30.61	89.34(+58.73↑)	72.32	93.85 (+21.53↑)	42.25	87.23 (+44.98↑)	**100.00**
	CelebA	BA(%)	76.90	75.25	71.65(-3.60↓)	76.49	72.40 (-4.09↓)	74.69	71.58 (-3.11↓)	**76.75**
		ASR(%)	-	31.16	92.25(+61.09↑)	79.00	98.85 (+19.85↑)	32.73	95.97 (+63.24↑)	**100.00**
MobileNetV2	CIFAR-10	BA(%)	92.58	91.86	90.64(-1.22↓)	90.35	90.65 (+0.30↑)	90.95	89.97 (-0.98↓)	**92.43**
		ASR(%)	-	10.71	91.58(+80.87↑)	16.74	96.37 (+79.63↑)	13.07	94.67 (+81.60↑)	**99.99**
	GTSRB	BA(%)	98.78	96.83	96.28(-0.55↓)	94.13	97.79 (+3.66↑)	95.97	93.82 (-2.15↓)	**98.76**
		ASR(%)	-	16.09	81.85(+65.76↑)	60.75	92.34 (+31.59↑)	22.75	78.28 (+55.53↑)	**99.91**
	CelebA	BA(%)	77.35	77.50	71.42(-6.08↓)	75.63	75.21 (-0.42↓)	77.51	71.73 (-5.78↓)	**77.70**
		ASR(%)	-	29.35	87.34(+57.99↑)	77.43	96.30 (+18.87↑)	31.91	90.92 (+59.01↑)	**100.00**

Table 1: Comparison of methods across multiple datasets; additional results are in the Appendix.A.

effective in the RGB domain may not survive storage and transmission. Among the baselines, WaNet is relatively more robust because its deformation-based trigger is less sensitive to compression. Nevertheless, compression still degrades its high-frequency trigger components, as also suggested by Fig. 1 and prior frequency-domain analyses [29]. By preserving trigger-relevant high-frequency information during compression, UAR consistently recovers high ASR while maintaining competitive BA, demonstrating its effectiveness in reactivating fragile invisible triggers.

Reactivating ineffective clean-label attacks. We further evaluate UAR on two clean-label attacks, SAA [26] and Narcissus [27] on GTSRB with ResNet18. As shown in Tab. 2, both attacks suffer from low ASR after compression, indicating that their adversarial perturbations are also compression-sensitive. Applying UAR substantially improves their ASR with little BA degradation. These results show that the compression-induced failure and our reactivation strategy is not limited to dirty-label invisible attacks.

	SAA [26]	SAA_{ReA}	Narcissus [27]	$\text{Narcissus}_{\text{ReA}}$
BA (%)	94.95	94.29	94.23	94.82
ASR (%)	10.04	45.24	17.27	50.26

Table 2: Reactivation of clean-label attacks.

Reactivating compression-based attacks. We evaluate UAR on two compression-based attacks, ECA [30] and CBA [2]. Although designed with lossy compression, these attacks are not inherently robust to the inevitable compression pipeline. Their triggers rely on compression-induced frequency discrepancies between poisoned and benign RGB samples, but this gap can be weakened or removed when both undergo the same lossy storage and transmission process. Thus, their frequency cues remain vulnerable to further compression/decompression, similar to frequency-domain invisible attacks such as FTrojan. As shown in Tab. 3, ECA and CBA also exhibit clear ASR degradation after compression, indicating that compression-based triggers do not guarantee compression robustness. By preserving attack-relevant frequency components with sample-specific ROI masks, UAR effectively reactivates these attacks while maintaining benign accuracy.

Dataset	Method	ECA	ECA_{ReA}	CBA	CBA_{ReA}
CIFAR-10	BA(%)	92.34	91.48(-0.9↓)	92.10	91.92(-0.2↓)
	ASR(%)	23.20	95.10(+71.9↑)	27.10	98.20(+71.1↑)
CelebA	BA(%)	75.08	72.92(-2.2↓)	76.13	75.05(-1.1↓)
	ASR(%)	30.84	93.96(+63.1↑)	28.62	97.54(+68.9↑)

Table 3: Reactivation of compression-aware attacks.

	Re-Activation	w/o Trigger	w/o Trigger & $\boldsymbol{M}_{freq}$
FTrojan	83.77	42.13	0.43
WaNet	93.85	40.76	0.89
BppAttack	85.02	48.00	0.89

Table 4: Ablations for our Reactivation method.

	FTrojan	$\text{FTrojan}_{\text{ReA}}$-FFT	$\text{FTrojan}_{\text{ReA}}$-DCT	$\text{FTrojan}_{\text{ReA}}$-DWT
BA	**87.73**	86.80(-0.93)	86.92(-0.81)	87.35(-0.38)
ASR	24.77	**92.17**(+67.40)	91.87(+67.10)	90.87(+66.10)

Table 5: Impact of frequency decomposition.

Ablation studies of the ROI mask. Tab. 4 verifies whether UAR reactivates original triggers rather than introducing a new ROI-induced shortcut. Removing the original trigger while keeping the reactivation pipeline causes a large ASR drop, showing that the original trigger remains essential. Removing both the trigger and $\boldsymbol{M}_{\text{freq}}$ further reduces ASR to nearly zero. Thus, UAR mainly preserves and reactivates existing trigger information instead of creating a new mask-only trigger.

Ablation studies of frequency transformation. Tab. 5 compares FFT, DCT, and DWT for constructing $\boldsymbol{M}_{\text{freq}}$, averaged over three datasets with ResNet-18 and Mbt-2018. All three variants



Figure 4: **Visualization of the UAR.** We visualize original images, compressed attack samples, reactivated samples, and the corresponding ROI masks.

significantly improve ASR over compressed FTrojan, indicating that UAR is not tied to a specific transform. The key is preserving compression-sensitive high-frequency trigger information.

Stealthiness. Fig. 4 shows that UAR preserves the visual appearance of poisoned samples after compression and does not introduce obvious artifacts. The sample-specific masks also indicate that UAR adapts to each attack's trigger structure rather than applying a fixed visible pattern.

4.3 Evaluation of Compression-Adapted Attack

All-to-one attack. Tab. 1 presents the results of CAA in all-to-one attack mode. We utilize customized ROI mask M^* to compress the images to be poisoned, and $M_{uniform}$ to compress benign images. Our method achieves ASR of over 99.9% on all four backbone models across all datasets, while the BA remains comparable to the clean model. The significant improvements in BA and ASR indicate that our method successfully implants a specific frequency distribution trigger.

Dataset	Method	Clean	FTrojan	WaNet	BppAttack	CAA
CIFAR-10	BA(%)	93.57	92.76	91.13	90.32	**93.51**
	ASR(%)	-	1.13	29.21	28.58	**93.76**
GTSRB	BA(%)	99.22	98.28	98.14	89.47	**99.17**
	ASR(%)	-	13.30	17.57	39.62	**99.46**
CelebA	BA(%)	76.90	75.50	76.66	73.03	**76.85**
	ASR(%)	-	3.50	55.57	19.17	**76.77**

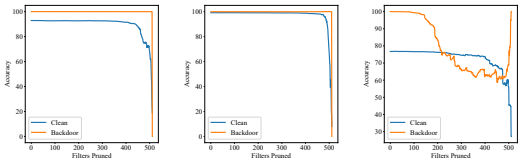
Table 6: All-to-all attack result.

All-to-all attack. To further demonstrate the robustness and generality of our method, we conduct an All-to-all experiment following previous approaches [8, 24, 22] in Tab. 6. In this experiment, ResNet18 served as the backbone. Our method outperforms alternatives across three datasets. Our CAA mislabels nearly 100% of poisoned images with only a 0.06% loss in BA. In contrast, baselines falter in all-to-all scenarios, suffering BA drops.

Codec	Method	Clean	FTrojan	WaNet	BppAttack	CAA
JPEG 2000 [50]	BA(%)	92.85	90.23	83.29	88.09	**92.93**
	ASR(%)	-	12.35	75.09	48.27	**99.56**
BPG [51]	BA(%)	93.58	92.10	90.55	**92.48**	92.39
	ASR(%)	-	13.07	76.97	44.97	**100.00**
Cheng-2020 [84]	BA(%)	93.21	91.34	84.86	88.25	**93.14**
	ASR(%)	-	11.89	77.32	43.73	**100.00**
ELIC [85]	BA(%)	93.79	92.32	85.22	87.35	**93.53**
	ASR(%)	-	12.64	78.32	46.73	**99.77**

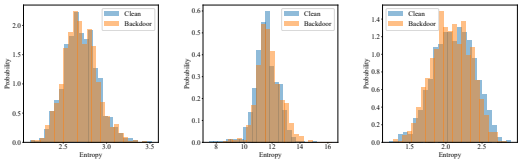
Table 7: Performance on different codecss.

Performance on different compression models. We experiment with our Compression-Adapted attack using various LIC models [84, 85] and traditional model (JPEG 2000 [50], BPG [51]) on CIFAR-10, as shown in Tab. 7. ResNet18 [79] is employed as the classification model. Our method consistently demonstrates minimal BA loss and high ASR close to 100% across different codecs. This performance underscores the broad applicability and generalizability of CAA. Please refer to the Appendix.F for additional results with more backbones.



(a) CIFAR-10 (b) GTSRB (c) CelebA

Figure 5: Resilient to Fine-Pruning [86].



(a) CIFAR-10 (b) GTSRB (c) CelebA

Figure 6: Resilient to STRIP [87].

Resistance to defense methods. We evaluate CAA method against the common defense methods, including Fine-Pruning [86], STRIP [87], and Gaussian noise and blur defenses. The introduction of defense methods is detailed in the supplementary material.

Quality of Recompression	50 (low)	75 (default)	90 (high)
$\text{FTrojan}_{\text{ReA}}$	10.83	17.87	25.56
CAA (2×2 mask)	79.59	95.65	97.08
CAA (8×8 mask)	73.27	88.87	96.57

Table 8: ASRs of recompression with JPEG on GTSRB.

	Std. Compression		$\text{FTrojan}_{\text{ReA}}$	CAA	
Codec	ELIC	JPEG	ELIC	ELIC	JPEG
file size (KiB)	0.789	0.962	1.036	0.813	0.986
Bits (/pixel)	1.579	1.923	2.071	1.626	1.972

Table 9: Average file size (KiB) and bits (/pixel) on GTSRB.

• The results of resisting Fine-Pruning are shown in Fig. 5, which indicate that Fine-Pruning cannot completely eliminate the backdoor implanted by our method. For instance, in the CIFAR-10 and GTSRB dataset, the ASR consistently exceeds the BA; in the CelebA dataset, the ASR achieves above 60%, showing that the backdoor is still effective.

• The results in Fig. 6 indicate that the entropy ranges of the clean model and the backdoored model trained on our poisoned samples, are very similar. Our method is resistant to STRIP in compression scenarios.

• After applying Gaussian blur (kernel:3) to decompressed test samples, we achieved a BA of 91.62% and an ASR of 98.72%. With Gaussian noise (std:1) added, BA was 86.86% and ASR reached 100%. These results demonstrate our method's resilience against such defenses.

Robustness against re-compression. Tab. 8 reports the ASR under JPEG re-compression. As expected, $\text{FTrojan}_{\text{ReA}}$ almost completely fails. Since FTrojan is not designed to be compression-adaptive, its trigger remains highly vulnerable to re-compression, even when it is injected before the initial compression stage. In contrast, the performance of CAA depends on both the re-compression quality factor and the selected ROI mask. When the JPEG quality factor is set to the default or a higher value, and a relatively sparse ROI mask is adopted, CAA consistently achieves an ASR above 95%. These results demonstrate that CAA offers substantially stronger robustness to practical re-compression than non-adaptive invisible attacks.

Comparisons of file sizes. Our method redistributes bits according to the ROI mask, assigning more bits to light (high-priority) regions while allocating fewer bits to dark regions. As a result, the overall bit allocation is adjusted spatially rather than uniformly increased. Consequently, the results in Tab. 9 show that our method does not noticeably increase file size, or equivalently, the resulting bit budget. This suggests that the improved attack effectiveness of our method is achieved with little additional storage or transmission overhead.

Stealthiness. Fig. 7 is the visualization of CAA. (a) and (b) shows that poisoned images by CAA are visually imperceptible. As the ROI mask pattern becomes denser, stealth increases while attack performance remains unaffected. All reported experiments of CAA use an 8×8 checkerboard. Fig. 8 further analyzes our CAA method driven special frequency pattern: we split each image into four regions according to the ROI mask and show the DCT spectrum for each region. The poisoned images exhibit noticeably stronger high-frequency signals in the top-left and bottom-right regions than in the top-right and bottom-left. Such frequency pattern serves as a trigger correlated with the target class, enabling a successful backdoor. Visual similarity alone does not establish statistical undetectability.

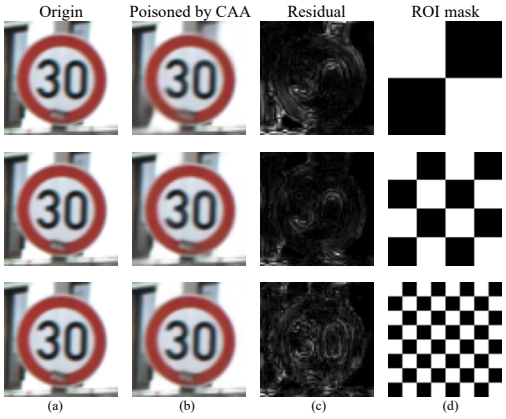


Figure 7: **Visualization of CAA.** (a) Origin images (without compression). (b) Poisoned images via CAA (all samples were classified as the target class). (c) The residuals. (d) The ROI masks.

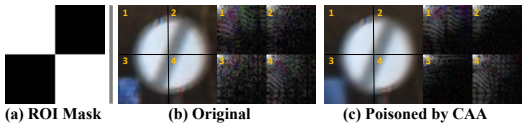


Figure 8: Frequency-Domain Visualization of CAA.

	PSNR ↑	SSIM ↑	Human Inspection Test ↑
No Attack	INF	1.000	-
BadNet	21.76	0.916	8.3
CAA	**37.4**	**0.965**	**52.4**

Table 10: Quantitative evaluation of stealthiness.

Tab.10 shows that our method is imperceptible by both objective metrics and human perception (human inspection test setting follows WaNet [24]). Since prior invisible triggers fail under lossy compression, comparing stealth against those failed methods is meaningless; therefore we compare

CAA with the visible attack BadNet [8]. Tab.10 demonstrates that CAA remains effective under lossy compression while offering superior stealth.

Frequency-domain detectability. We train a linear detector on per-image log-magnitude FFT features to distinguish attack samples from benign samples, using disjoint training and held-out test sets. Within-pattern evaluation uses the same attack or ROI pattern on both splits. Cross-pattern and cross-attack evaluation apply the detector trained on CAA-8×8 to CAA-2×2 and FTrojan$_{\mathrm{ReA}}$, respectively, without retraining. Detection balanced accuracy (Det. BAcc) averages the true-positive and true-negative rates; 50% indicates chance-level performance.

Detector training attack	Held-out test attack	Setting	Det. BAcc (%)
FTrojan$_{\mathrm{ReA}}$	FTrojan$_{\mathrm{ReA}}$	Within-pattern	58.7
CAA (2×2)	CAA (2×2)	Within-pattern	86.9
CAA (8×8)	CAA (8×8)	Within-pattern	71.0
CAA (8×8)	CAA (2×2)	Cross-pattern	60.5
CAA (8×8)	FTrojan$_{\mathrm{ReA}}$	Cross-attack	52.5

Table 11: Held-out frequency-domain detection using log-magnitude FFT features. Each attack is distinguished from benign samples.

Tab. 11 shows that a matched detector identifies part of CAA's frequency signature, especially for the coarse 2×2 pattern. Detection BAcc is lower for the finer 8×8 pattern and the sample-specific FTrojan$_{\mathrm{ReA}}$ instance. The CAA-8×8 detector transfers less effectively to CAA-2×2 and performs close to chance on FTrojan$_{\mathrm{ReA}}$. Thus, CAA can be detectable when the detector is adapted to its pattern. Limited transfer of this linear detector does not establish resistance to all adaptive frequency-domain detectors.

Additional results on complexity, defenses, compression-aware optimization, poisoning budgets, and robustness are provided in Appendices A–H.

5 Conclusion and Limitations

In this paper, we study backdoor attacks under inevitable lossy compression, a practical yet underexplored setting in real-world data storage and transmission pipelines. We show that existing invisible backdoor attacks can become ineffective after compression, and propose two ROI-based strategies to address this challenge. Experiments across multiple datasets, backbones, and codecs demonstrate the effectiveness of our methods. A limitation of this work is that we focus on revealing and constructing compression-aware backdoor attacks, but do not propose a dedicated defense against such compression-adapted threats. Our detection study shows that visual stealthiness does not guarantee statistical undetectability: a detector adapted to a known CAA pattern can distinguish poisoned from benign samples, while the evaluated detector has limited cross-pattern transfer. Developing effective detection and mitigation methods for backdoor attacks under lossy compression remains an important direction for future research.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (12572266), and the High Performance Computing Centers at Eastern Institute of Technology, Ningbo, and Ningbo Institute of Digital Twin. The authors declare no competing interests.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[2] Qiuyu Duan, Zhongyun Hua, Qing Liao, Yushu Zhang, and Leo Yu Zhang. Conditional backdoor attack via jpeg compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19823–19831, 2024.

[3] Qian Li, Yuxiao Hu, Ye Liu, Dongxiao Zhang, Xin Jin, and Yuntian Chen. Discrete point-wise attack is not enough: Generalized manifold adversarial attack for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20575–20584, 2023.

[4] Robby Costales, Chengzhi Mao, Raphael Norwitz, Bryan Kim, and Junfeng Yang. Live trojan attacks on deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 796–797, 2020.

[5] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Proceedings of the 36th Annual Computer Security Applications Conference*, pages 897–912, 2020.

[6] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11966–11976, 2021.

[7] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

[8] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[9] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[10] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.

[11] Qian Li, Yuxiao Hu, Yinpeng Dong, Dongxiao Zhang, and Yuntian Chen. Focus on hiders: Exploring hidden threats for enhancing adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24442–24451, 2024.

[12] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[13] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125, 2019.

[14] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020.

[15] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 113–131, 2020.

[16] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[17] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020.

[18] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *EuroS&P 2022*, 2022.

[19] Sonain Jamil, Md Jalil Piran, MuhibUr Rahman, and Oh-Jin Kwon. Learning-driven lossy image compression: A comprehensive survey. *Engineering Applications of Artificial Intelligence*, 123:106361, 2023.

[20] Yueyu Hu, Wenhan Yang, Zhan Ma, and Jiaying Liu. Learning end-to-end lossy image compression: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4194–4211, 2021.

[21] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023.

[22] Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15074–15084, 2022.

[23] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision*, pages 396–413. Springer, 2022.

[24] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*.

[25] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16463–16472, 2021.

[26] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35:19165–19178, 2022.

[27] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 771–785, 2023.

[28] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16473–16481, 2021.

[29] Quanrui Rao, Lin Wang, and Wuying Liu. Rethinking cnn's generalization to backdoor attack from frequency domain. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Sze Jue Yang, Quang Nguyen, Chee Seng Chan, and Khoa D Doan. Everyone can attack: Repurpose lossy compression as a natural backdoor attack. *arXiv preprint arXiv:2308.16684*, 2023.

[31] Joel Askelöf, Mathias Larsson Carlander, and Charilaos Christopoulos. Region of interest coding in jpeg 2000. *Signal Processing: Image Communication*, 17(1):105–111, 2002.

[32] Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2380–2389, 2021.

[33] Chunlei Cai, Li Chen, Xiaoyun Zhang, and Zhiyong Gao. End-to-end optimized roi image compression. *IEEE Transactions on Image Processing*, 29:3442–3457, 2019.

[34] Yi Ma, Yongqi Zhai, Chunhui Yang, Jiayu Yang, Ruofan Wang, Jing Zhou, Kai Li, Ying Chen, and Ronggang Wang. Variable rate roi image compression optimized for visual quality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1936–1940, 2021.

[35] Hiroaki Akutsu and Takahiro Naruko. End-to-end learned roi image compression. In *CVPR Workshops*, page 0, 2019.

[36] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 162–170, 2021.

[37] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17492–17501, 2022.

[38] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1503–1511, 2022.

[39] Yunuo Chen, Qian Li, Bing He, Donghui Feng, Ronghua Wu, Qi Wang, Li Song, Guo Lu, and Wenjun Zhang. S2cformer: Reorienting learned image compression from spatial interaction to channel aggregation. *arXiv preprint arXiv:2502.00700*, 2025.

[40] Yunuo Chen, Chuqin Zhou, Jiangchuan Li, Xiaoyue Ling, Bing He, Jincheng Dai, Li Song, and Guo Lu. Next-frame decoding for ultra-low-bitrate image compression with video diffusion priors. *arXiv preprint arXiv:2603.15129*, 2026.

[41] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[42] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 182–199. Springer, 2020.

[43] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1148–1156, 2021.

[44] Tran Huynh, Dang Nguyen, Tung Pham, and Anh Tran. Combat: Alternated training for effective clean-label backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2436–2444, 2024.

[45] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8133–8142, 2023.

[46] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16463–16472, 2021.

[47] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020.

[48] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[49] David Bařina and Ondřej Klíma. Region of interest in jpeg. pages 1–5, 01 2022.

[50] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal processing magazine*, 18(5):36–58, 2001.

[51] David Yee, Sara Soltaninejad, Deborsi Hazarika, Gaylord Mbuyi, Rishi Barnwal, and Anup Basu. Medical image compression based on region of interest using better portable graphics (bpg). In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 216–221. IEEE, 2017.

[52] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[53] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[54] Fangdong Chen, Yumeng Xu, and Li Wang. Two-stage octave residual network for end-to-end image compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3922–3929, 2022.

[55] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International Conference on Learning Representations*, 2022.

[56] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14388–14397, 2023.

[57] Yiwei Zhang, Guo Lu, Yunuo Chen, Shen Wang, Yibo Shi, Jing Wang, and Li Song. Neural rate control for learned video compression. In *The Twelfth International Conference on Learning Representations*, 2023.

[58] Yuqi Li, Haotian Zhang, Li Li, and Dong Liu. Learned image compression with hierarchical progressive context modeling. *arXiv preprint arXiv:2507.19125*, 2025.

[59] Fanhu Zeng, Hao Tang, Yihua Shao, Siyu Chen, Ling Shao, and Yan Wang. Mambaic: State space models for high-performance learned image compression. *arXiv preprint arXiv:2503.12461*, 2025.

[60] Yunuo Chen, Bing He, Zezheng Lyu, Hongwei Hu, Qunshan Gu, Yuan Tian, and Guo Lu. Adaptive learned image compression with graph neural networks. *arXiv preprint arXiv:2603.25316*, 2026.

[61] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021.

[62] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020.

[63] Yichen Qian, Xiuyu Sun, Ming Lin, Zhiyu Tan, and Rong Jin. Entroformer: A transformer-based entropy model for learned image compression. In *International Conference on Learning Representations*.

[64] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7618–7627, 2023.

[65] Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. In *The Twelfth International Conference on Learning Representations*.

[66] Yunuo Chen, Zezheng Lyu, Bing He, Hongwei Hu, Qi Wang, Yuan Tian, Li Song, Wenjun Zhang, and Guo Lu. Content-aware mamba for learned image compression. In *The Fourteenth International Conference on Learning Representations*, 2026.

[67] Jiangchuan Li, Chuqin Zhou, Yunuo Chen, and Guo Lu. Differentiable vmaf: A trainable metric for optimizing video compression codec. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2025.

[68] Mingfu Xue, Xin Wang, Shichang Sun, Yushu Zhang, Jian Wang, and Weiqiang Liu. Compression-resistant backdoor attack against deep neural networks. *Applied Intelligence*, 53(17):20402–20417, 2023.

[69] Yi Yu, Yufei Wang, Wenhan Yang, Shijian Lu, Yap-Peng Tan, and Alex C Kot. Backdoor attacks against deep image compression via adaptive frequency trigger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12250–12259, 2023.

[70] Yi Yu, Yufei Wang, Wenhan Yang, Lanqing Guo, Shijian Lu, Ling-Yu Duan, Yap-Peng Tan, and Alex C Kot. Robust and transferable backdoor attacks against deep image compression with selective frequency prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1674–1693, 2024.

[71] Wenjie Wang, Lin Cong, Honglong Chen, Lifeng Han, Yudong Gao, and Xiaolong Liu. Gaba: A general anti-compression backdoor attack using the characteristic of image compression. *Available at SSRN 5268955*.

[72] Andrew P Bradley and Fred WM Stentiford. Jpeg 2000 and region of interest coding. In *Digital Image Computing Techniques and Applications*, volume 2, pages 1–6. Citeseer, 2002.

[73] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

[74] Changjiang Li, Ren Pang, Zhaohan Xi, Tianyu Du, Shouling Ji, Yuan Yao, and Ting Wang. An embarrassingly simple backdoor attack on self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4367–4378, 2023.

[75] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[76] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[77] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[78] Akshaj Kumar Veldanda, Kang Liu, Benjamin Tan, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, Brendan Dolan-Gavitt, and Siddharth Garg. Nnoculation: broad spectrum and targeted treatment of backdoored dnns. *arXiv preprint arXiv:2002.08313*, 3:18, 2020.

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[81] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[82] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[83] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

[84] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020.

[85] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5718–5727, 2022.

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

[87] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125, 2019.

[88] Yanxin Yang, Chentao Jia, DengKe Yan, Ming Hu, Tianlin Li, Xiaofei Xie, Xian Wei, and Mingsong Chen. Sampdetox: Black-box backdoor defense via perturbation-based sample detoxification. *Advances in Neural Information Processing Systems*, 37:121236–121264, 2024.

[89] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. *Advances in Neural Information Processing Systems*, 36:75286–75306, 2023.

[90] Yucheng Shi, Mengnan Du, Xuansheng Wu, Zihan Guan, Jin Sun, and Ninghao Liu. Black-box backdoor defense via zero-shot image purification. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[91] Qikun Zhang, Mengyang Yu, Ruifang Wang, Yongjiao Li, Junling Yuan, and Yu-an Tan. A sparse and invisible targeted backdoor attack in federated learning. *Journal of King Saud University Computer and Information Sciences*, 37(6):134, 2025.

[92] Richard Shin and Dawn Song. JPEG-resistant adversarial images. In *Machine Learning and Computer Security Workshop*, 2017.

[93] Mengte Shi, Sheng Li, Zhaoxia Yin, Xinpeng Zhang, and Zhenxing Qian. On generating JPEG adversarial images. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021.

[94] Benoit Bonnet, Teddy Furon, and Patrick Bas. Generating adversarial images in quantized domains. *IEEE Transactions on Information Forensics and Security*, 17:373–385, 2022.

[95] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 284–293. PMLR, 2018.

[96] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pages 1127–1137. PMLR, 2020.

A More results of all-to-one attacks on different backbones

We conduct additional experiments on all-to-one attacks using two additional backbones [80, 82]. The results are shown in Tab.12. These experiments further demonstrate the robustness and generalizability of our claims and methods.

Backbone ↓	Dataset ↓	Method →	Clean	FTrojan	$\text{FTrojan}_{\text{ReA}}$	WaNet	$\text{WaNet}_{\text{ReA}}$	BppAttack	$\text{BppAttack}_{\text{ReA}}$	CAA
Pre-activation ResNet18	CIFAR-10	BA(%)	92.93	92.46	91.35(-1.11↓)	86.30	91.30 (+5.00↑)	90.61	90.59 (-0.02↓)	**92.86**
		ASR(%)	-	11.05	94.19(+83.14↑)	51.25	98.17 (+46.92↑)	16.17	95.03 (+78.86↑)	**99.99**
	GTSRB	BA(%)	99.13	94.25	97.02(+2.77↑)	92.73	97.43 (+4.70↑)	89.85	94.08 (+4.23↑)	**99.13**
		ASR(%)	-	23.26	83.77(+60.51↑)	71.80	92.26 (+20.46↑)	36.83	85.02 (+48.19↑)	**99.95**
	CelebA	BA(%)	76.78	74.98	70.34(-4.64↓)	76.55	74.83 (-1.72↓)	75.71	70.43 (-5.28↓)	**76.57**
		ASR(%)	-	27.37	92.91(+65.54↑)	88.87	95.45 (+6.58↑)	31.32	94.78 (+63.46↑)	**99.93**
SENet	CIFAR-10	BA(%)	92.97	92.31	90.93(-1.38↓)	89.23	91.30 (+2.07↑)	90.67	91.15 (+0.48↑)	**93.16**
		ASR(%)	-	11.13	94.25(+83.12↑)	28.05	97.95 (+69.90↑)	16.14	96.03 (+79.89↑)	**99.94**
	GTSRB	BA(%)	98.75	94.87	97.17(+2.30↑)	95.02	97.69 (+2.67↑)	91.79	93.10 (+1.31↑)	**98.69**
		ASR(%)	-	29.68	87.43(+57.75↑)	63.25	92.70 (+29.45↑)	36.07	84.76 (+48.69↑)	**99.89**
	CelebA	BA(%)	77.10	74.73	71.71(-3.02↓)	76.49	72.10 (-4.39↓)	75.59	71.45 (-4.14↓)	**77.06**
		ASR(%)	-	30.21	91.58(+61.37↑)	79.00	96.78 (+17.78↑)	31.88	94.05 (+62.17↑)	**99.46**

Table 12: Comparison of methods across multiple datasets.

B Defense methods in the main text

In this part, we introduce the Fine-Pruning [86], and STRIP [87] defense methods we used in the main text.

Fine-Pruning [86]. Fine-Pruning involves selectively pruning neurons that are dormant on clean inputs but become active when triggered by a backdoor. By removing these neurons, Fine-Pruning aims to deactivate the backdoor mechanism without significantly impairing the model's performance on legitimate tasks.

STRIP [87]. STRIP works by introducing random perturbations to inputs and observing the behavior of the model's predictions. If the output remains consistently similar across perturbations, the input is likely compromised by a backdoor.

C Experiments with more defense methods

We experiment with more advanced open-sourced defenses: SampDetox [88], FST [89], and ZIP [90]. Since prior invisible attacks fail under compression, we compare against BadNet under JPEG. In Tab. 5, CAA shows stronger resistance than BadNet in all defenses. The performance on FST shows that model fine-tuning seems to be a promising defense strategy in this setting. However, purification methods appear to have limited impact on CAA. We conjecture this is because CAA is compression-driven and doesn't introduce additional noise, making it more resistant to these defenses.

Table 13: Additional defense results on GTSRB.

Attack	SampDetox		FST		ZIP	
	BA	ASR	BA	ASR	BA	ASR
BadNet	94.76	1.67	92.97	1.06	92.48	3.92
CAA	95.28	80.38	92.83	65.38	94.69	88.29

D Deployment complexity

Invisible attacks generally require more complex deployment than visible attacks (e.g., BadNets) to achieve both stealthiness and high ASR. For example, FTrojan involves transforming between RGB and YUV, frequency-domain manipulation, block-wise DCT. Tab. 14 reports the comparisons on 2K images, showing that our ROI mask computation incurs no additional overhead compared to other sample-specific triggers and has a negligible impact on encoding efficiency.

Table 14: Time complexity (s/sample) on GTSRB resized to 2K. CPU: Intel(R) Xeon(R) Gold 6458Q with 128 threads; GPU: a single NVIDIA A800 GPU.

Sample-Specific Computation			Encoding Time		
ROI Mask of ReA	FTrojan Trigger	WaNet Trigger	Standard	ReA	CAA
0.002	0.002	0.003	0.502	0.515	0.508

E Robustness against steganalysis.

Tab. 15 reports L2 distances between poisoned and benign images in the SRM feature space, where values of 0–5 indicate negligible differences [91]. Since SRM mainly relies on high-pass filters, we attribute the stealthiness of CAA to compression-driven trigger embedding instead of injecting SRM-sensitive high-frequency artifacts.

Table 15: Comparison of SRM feature distances on GTSRB.

	FTrojan	WaNet	CAA (2×2 Mask)	CAA (8×8 Mask)
SRM (↓)	0.995	0.628	0.314	0.207

F More results on different codecs

Tab. 16, Tab. 17, Tab. 18, Tab. 19 and Tab. 20 present the complete comparison results of our CAA and the baseline methods on JPEG2000 [50], BPG[51], Cheng-2020 [84], and ELIC [85], respectively. CAA outperforms the baseline methods in terms of both benign accuracy (BA) and attack success rate (ASR). Notably, even when the baseline methods largely fail under lossy compression scenarios, CAA still maintains a high ASR, demonstrating its robustness and effectiveness in performing data poisoning attacks under compression-induced distortions.

Backbone ↓	**Dataset ↓**	**Method →**	**Clean**	**FTrojan**	**WaNet**	**BppAttack**	**CAA**
ResNet18	CIFAR-10	BA(%)	91.74	91.02	80.58	85.27	**91.76**
		ASR(%)	-	10.82	73.29	43.92	**97.23**
	GTSRB	BA(%)	98.32	95.23	94.92	90.61	**98.01**
		ASR(%)	-	30.02	71.97	43.23	**99.79**
	CelebA	BA(%)	75.27	74.29	75.02	**75.32**	75.30
		ASR(%)	-	30.23	77.27	30.86	**99.93**
Pre-activation ResNet18	CIFAR-10	BA(%)	90.92	91.90	86.18	91.97	**92.76**
		ASR(%)	-	10.87	50.76	18.28	**100.00**
	GTSRB	BA(%)	95.02	92.97	94.82	87.49	**95.55**
		ASR(%)	-	19.87	73.28	33.29	**99.78**
	CelebA	BA(%)	75.29	74.47	73.29	77.38	**78.33**
		ASR(%)	-	28.97	86.29	30.29	**99.93**
MobileNetV2	CIFAR-10	BA(%)	92.92	90.76	91.68	93.49	**93.96**
		ASR(%)	-	11.29	17.37	12.97	**100.00**
	GTSRB	BA(%)	97.29	96.53	96.28	93.59	**98.35**
		ASR(%)	-	14.20	58.57	24.39	**99.75**
	CelebA	BA(%)	76.39	75.29	73.29	75.28	**76.80**
		ASR(%)	-	26.39	76.38	30.19	**99.75**
SENet	CIFAR-10	BA(%)	91.47	**93.29**	91.39	89.28	93.28
		ASR(%)	-	11.38	26.28	15.39	**100.00**
	GTSRB	BA(%)	98.28	95.29	96.32	90.36	**98.29**
		ASR(%)	-	27.30	66.49	34.29	**99.94**
	CelebA	BA(%)	76.38	73.91	77.38	77.01	**77.85**
		ASR(%)	-	31.38	77.49	35.23	**99.27**

Table 16: Comparison results on JPEG [49].

Backbone ↓	Dataset ↓	Method →	Clean	FTrojan	WaNet	BppAttack	CAA
ResNet18	CIFAR-10	BA(%)	92.85	90.23	83.29	88.09	**92.93**
		ASR(%)	-	12.35	75.09	48.27	**99.56**
	GTSRB	BA(%)	98.53	93.96	**97.35**	91.90	97.29
		ASR(%)	-	31.96	75.24	40.43	**99.95**
	CelebA	BA(%)	76.41	76.63	74.19	72.75	**77.63**
		ASR(%)	-	32.40	80.64	31.76	**99.90**
Pre-activation ResNet18	CIFAR-10	BA(%)	93.98	91.46	85.50	91.86	**92.49**
		ASR(%)	-	12.07	55.09	18.97	**100.00**
	GTSRB	BA(%)	99.37	95.92	92.76	89.54	**99.04**
		ASR(%)	-	25.85	75.97	37.92	**99.92**
	CelebA	BA(%)	75.66	74.97	**76.83**	73.92	76.27
		ASR(%)	-	25.86	87.54	33.44	**100.00**
MobileNetV2	CIFAR-10	BA(%)	94.23	92.30	90.28	91.65	**92.48**
		ASR(%)	-	10.87	15.23	18.34	**99.94**
	GTSRB	BA(%)	98.21	97.29	93.23	97.35	**98.05**
		ASR(%)	-	15.97	65.42	22.29	**99.65**
	CelebA	BA(%)	76.12	**77.87**	74.23	76.97	77.38
		ASR(%)	-	30.97	80.35	33.23	**99.64**
SENet	CIFAR-10	BA(%)	93.25	92.85	88.43	90.38	**93.18**
		ASR(%)	-	10.75	26.43	18.09	**99.91**
	GTSRB	BA(%)	98.95	94.97	94.23	92.86	**98.10**
		ASR(%)	-	28.35	68.85	37.97	**100.00**
	CelebA	BA(%)	77.64	73.93	76.30	76.23	**77.47**
		ASR(%)	-	30.86	80.23	34.97	**99.95**

Table 17: Comparison results on JPEG2000 [50].

Backbone ↓	Dataset ↓	Method →	Clean	FTrojan	WaNet	BppAttack	CAA
ResNet18	CIFAR-10	BA(%)	93.58	92.10	90.55	**92.48**	92.39
		ASR(%)	-	13.07	76.97	44.97	**100.00**
	GTSRB	BA(%)	99.26	95.13	95.32	91.07	**98.28**
		ASR(%)	-	30.19	72.02	45.29	**99.64**
	CelebA	BA(%)	77.81	74.27	76.95	73.43	**77.27**
		ASR(%)	-	30.26	80.75	31.75	**99.85**
Pre-activation ResNet18	CIFAR-10	BA(%)	91.86	**93.98**	84.28	91.85	92.26
		ASR(%)	-	12.65	50.16	15.53	**100.00**
	GTSRB	BA(%)	99.54	93.72	93.62	89.86	**99.76**
		ASR(%)	-	21.86	73.75	34.28	**99.93**
	CelebA	BA(%)	75.29	**76.91**	73.68	74.29	76.43
		ASR(%)	-	25.29	86.28	33.29	**100.00**
MobileNetV2	CIFAR-10	BA(%)	93.92	92.96	90.26	90.22	**93.28**
		ASR(%)	-	10.28	15.97	17.83	**99.97**
	GTSRB	BA(%)	98.82	95.23	92.87	95.21	**98.23**
		ASR(%)	-	13.28	56.29	25.37	**99.99**
	CelebA	BA(%)	76.29	75.54	73.27	76.29	**76.91**
		ASR(%)	-	27.28	74.29	30.75	**99.99**
SENet	CIFAR-10	BA(%)	93.65	**93.65**	88.64	90.92	93.27
		ASR(%)	-	10.73	29.64	15.28	**100.00**
	GTSRB	BA(%)	98.82	95.28	95.81	92.76	**97.74**
		ASR(%)	-	27.29	62.97	37.54	**100.00**
	CelebA	BA(%)	75.29	74.29	73.92	74.28	**75.01**
		ASR(%)	-	29.75	80.27	29.86	**99.64**

Table 18: Comparison results on BPG [51].

Backbone ↓	Dataset ↓	Method →	Clean	FTrojan	WaNet	BppAttack	CAA
ResNet18	CIFAR-10	BA(%)	93.21	91.34	84.86	88.25	**93.14**
		ASR(%)	-	11.89	77.32	43.73	**100.00**
	GTSRB	BA(%)	99.72	96.28	95.85	92.97	**99.29**
		ASR(%)	-	32.89	71.97	41.93	**99.99**
	CelebA	BA(%)	76.28	76.10	75.93	74.27	**76.83**
		ASR(%)	-	30.28	78.74	33.97	**100.00**
Pre-activation ResNet18	CIFAR-10	BA(%)	91.98	91.65	86.36	90.20	**92.53**
		ASR(%)	-	10.73	50.25	17.27	**100.00**
	GTSRB	BA(%)	99.49	94.28	91.93	89.82	**99.18**
		ASR(%)	-	22.96	74.29	35.27	**99.98**
	CelebA	BA(%)	76.27	73.92	75.39	75.82	**76.61**
		ASR(%)	-	28.45	88.42	30.63	**99.90**
MobileNetV2	CIFAR-10	BA(%)	93.74	**92.63**	90.17	90.36	92.19
		ASR(%)	-	10.27	16.64	15.27	**99.94**
	GTSRB	BA(%)	98.92	97.29	94.85	96.84	**98.16**
		ASR(%)	-	17.76	61.23	24.27	**99.25**
	CelebA	BA(%)	76.27	76.06	75.18	75.91	**76.28**
		ASR(%)	-	30.82	77.41	33.28	**99.98**
SENet	CIFAR-10	BA(%)	93.21	92.64	89.25	90.19	**93.03**
		ASR(%)	-	12.85	32.58	14.29	**100.00**
	GTSRB	BA(%)	98.98	94.14	96.26	91.19	**98.76**
		ASR(%)	-	29.75	62.75	36.19	**100.00**
	CelebA	BA(%)	77.38	73.28	77.21	74.29	**77.85**
		ASR(%)	-	29.75	79.16	33.26	**99.14**

Table 19: Comparison results on Cheng-2020 [84].

Backbone ↓	Dataset ↓	Method →	Clean	FTrojan	WaNet	BppAttack	CAA
ResNet18	CIFAR-10	BA(%)	93.79	92.32	85.22	87.35	**93.53**
		ASR(%)	-	12.64	78.32	46.73	**99.77**
	GTSRB	BA(%)	99.62	94.28	95.93	92.97	**99.18**
		ASR(%)	-	29.75	73.92	41.86	**99.98**
	CelebA	BA(%)	76.18	74.28	76.18	75.28	**76.37**
		ASR(%)	-	29.64	80.28	30.36	**98.28**
Pre-activation ResNet18	CIFAR-10	BA(%)	92.73	91.36	86.28	91.65	**92.46**
		ASR(%)	-	11.18	50.63	17.87	**99.85**
	GTSRB	BA(%)	99.64	93.75	93.27	91.74	**99.36**
		ASR(%)	-	21.85	73.64	33.92	**100.00**
	CelebA	BA(%)	77.64	74.62	76.84	75.28	**77.42**
		ASR(%)	-	25.28	85.28	30.95	**100.00**
MobileNetV2	CIFAR-10	BA(%)	92.29	**93.86**	90.74	90.01	92.37
		ASR(%)	-	13.28	14.12	16.86	**99.42**
	GTSRB	BA(%)	98.10	96.35	93.84	95.01	**98.96**
		ASR(%)	-	16.34	58.09	24.52	**100.00**
	CelebA	BA(%)	78.59	77.74	75.38	75.87	**78.25**
		ASR(%)	-	30.78	77.77	30.85	**98.32**
SENet	CIFAR-10	BA(%)	93.82	92.45	88.95	90.83	**93.92**
		ASR(%)	-	13.94	30.28	20.73	**99.73**
	GTSRB	BA(%)	98.48	95.89	96.99	93.92	**98.82**
		ASR(%)	-	27.32	63.54	37.32	**98.35**
	CelebA	BA(%)	76.53	75.08	76.44	74.52	**77.95**
		ASR(%)	-	29.74	81.46	32.56	**100.00**

Table 20: Comparison results on ELIC [85].

G Compression-Aware Trigger Optimization

Compression-resistant adversarial evasion provides relevant optimization strategies, including differentiable JPEG approximations [92], attacks in JPEG and quantized domains [93, 94], expectation over transformations [95], and low-frequency perturbation design [96]. Adversarial evasion targets immediate misclassification by a fixed classifier, whereas backdoor poisoning must induce a training-time association in a subsequently trained model. We therefore adapt compression-aware optimization to a backdoor baseline rather than treating adversarial examples as direct poisoning baselines.

Compression-Aware Narcissus. Starting from the original Narcissus settings [27], we insert a frozen, differentiable Cheng-2020 codec between the triggered image and the surrogate classifier during trigger optimization. Gradients propagate through the codec, but only the trigger is optimized; the codec, surrogate classifier, and victim-side training procedure remain unchanged. We evaluate the resulting trigger on GTSRB with Cheng-2020 and then transfer the same trigger to ELIC without re-optimization.

Method	Optimization codec	Evaluation codec	BA (%)	ASR (%)
Vanilla Narcissus	None	Cheng-2020	93.71	23.54
Compression-Aware Narcissus	Cheng-2020	Cheng-2020	94.19	67.53
Compression-Aware Narcissus	Cheng-2020	ELIC	90.07	11.07

Table 21: Compression-aware Narcissus on GTSRB. The ELIC row reuses the trigger optimized with Cheng-2020.

Matched-codec optimization increases ASR from 23.54% to 67.53%, showing that ROI coding is not the only way to improve compression robustness. ASR falls to 11.07% when this trigger is transferred to ELIC. This result demonstrates codec sensitivity for the evaluated trigger and codec pair, rather than a general failure of adaptive optimization to transfer. These experiments are separate from the Mbt-2018 reactivation comparison in Tab. 2: UAR preserves an existing trigger through ROI coding, while Compression-Aware Narcissus redesigns a trigger for a known differentiable codec.

Interpreting the compression-based baselines. Tab. 3 evaluates ECA [30] and CBA [2] when both benign and poisoned samples undergo a shared compression stage. ECA uses an additional JPEG or WEBP transformation to create its poisoned samples. Shared encoding gives benign samples compression artifacts as well and modifies the artifacts already present in poisoned samples, reducing the distinction on which this trigger relies. For CBA, the comparison uses a fixed data-side JPEG transformation rather than retaining its original joint optimization of the compression operator and target classifier. The lower ASRs therefore describe transfer to the evaluated shared-compression setting, rather than performance under the baselines' original protocols.

H Additional Training and Robustness Experiments

H.1 Lower poisoning rate

Each invisible baseline and its UAR variant use the same poisoning budget: 5% for FTrojan, 10% for WaNet, and 20% for BppAttack, following their original configurations. The main CAA comparison uses 5% poisoning. To evaluate a lower budget, we reduce CAA's poisoning rate to 2% on CIFAR-10 using ResNet18 and Mbt-2018, while keeping the ROI mask, codec configuration, and victim training settings unchanged.

Dataset	Poisoning rate (%)	BA (%)	ASR (%)
CIFAR-10	5	93.28	100.00
CIFAR-10	2	92.10	98.91

Table 22: CAA poisoning-rate ablation on CIFAR-10 with ResNet18 and Mbt-2018.

CAA retains 98.91% ASR at 2% poisoning in this configuration. This two-point comparison supports effectiveness at a lower budget on CIFAR-10, but does not determine the minimum poisoning rate or establish the same behavior across all datasets.

H.2 Clean data outside the attacker-controlled encoding pipeline

We use a fixed-size GTSRB experiment to isolate the effect of benign-data encoding. We progressively replace attacker-encoded benign samples with uncompressed clean samples, yielding three benign-subset compositions: all attacker-encoded, 50% attacker-encoded and 50% uncompressed, and all uncompressed. In the 50/50 setting, an equal number of attacker-encoded benign samples is subsampled when uncompressed samples are introduced. The poisoned subset, poisoning rate, total training-set size, and victim training configuration remain fixed. This is a controlled proxy for clean-data mixing, rather than an experiment that enlarges the training set and dilutes the poisoning rate.

Method	All attacker-encoded	50% attacker-encoded 50% uncompressed	All uncompressed
FTrojan$_{\mathrm{ReA}}$ (UAR)	97.43 / 89.34	96.87 / 88.71	98.69 / 90.02
CAA	99.13 / 100.00	99.00 / 100.00	99.13 / 100.00

Table 23: BA / ASR (%) under fixed-size benign-data replacement on GTSRB. Column headings describe the benign subset only.

FTrojan$_{\mathrm{ReA}}$ achieves 88.71–90.02% ASR and CAA achieves 100% across these settings. Thus, the evaluated attacks remain effective when their poisoned samples are mixed with benign samples that have not undergone attacker-controlled encoding. The comparison does not measure the effect of adding a larger external dataset or introducing a distribution shift.

H.3 Benign accuracy under JPEG recompression

Tab. 24 supplements the ASRs in Tab. 8 with their corresponding BA. JPEG recompression is applied after the initial ROI-based compression on GTSRB. The no-recompression column gives the reference for each attack and mask.

Attack	No recompression	Q50	Q75	Q90
FTrojan$_{\mathrm{ReA}}$	97.43 / 89.34	90.21 / 10.83	93.08 / 17.87	96.31 / 25.56
CAA (2×2)	99.11 / 100.00	92.73 / 79.59	96.81 / 95.65	99.05 / 97.08
CAA (8×8)	99.13 / 100.00	90.02 / 73.27	92.84 / 88.87	98.97 / 96.57

Table 24: BA / ASR (%) under an additional JPEG compression on GTSRB.

Q50 substantially reduces FTrojan$_{\mathrm{ReA}}$ ASR and partially mitigates CAA, but also lowers BA by 6.38–9.11 percentage points relative to the corresponding references. Q90 largely preserves CAA's

BA while leaving ASR above 96%. These results show a defense–utility trade-off for uniform recompression.

H.4 Low-pass preprocessing before victim training

We apply the same Gaussian filter to all decompressed benign and poisoned training images before training ResNet18 on GTSRB, leaving the remaining training settings unchanged. This training-time preprocessing experiment is distinct from the test-time Gaussian perturbations reported in the main text.

CAA mask	No filtering	Gaussian 3×3	Gaussian 7×7
2×2	100.00	100.00	96.21
8×8	100.00	93.87	80.73

Table 25: ASR (%) after Gaussian preprocessing of the training data on GTSRB with ResNet18.

Stronger filtering reduces ASR, particularly for the finer 8×8 pattern. The coarser pattern retains higher ASR under both tested filters, consistent with the recompression comparison. The reported ASRs alone do not quantify the benign-accuracy cost of this preprocessing defense.

H.5 ROI-mask selection and mixtures

We compare 2×2 and 8×8 checkerboards, concentric squares, and a variant that randomly selects one of these three masks for each poisoned sample. Tab. 26 reports single-compression BA/ASR and ASR after an additional JPEG encoding.

Mask configuration	BA (%)	ASR (%)	Q75 ASR (%)	Q90 ASR (%)
2×2 checkerboard	99.11	100.00	95.65	97.08
8×8 checkerboard	99.13	100.00	88.87	96.57
Concentric squares	99.20	100.00	96.37	98.13
Three-mask pool (random selection)	98.46	100.00	85.32	95.27

Table 26: ROI-mask selection and JPEG recompression.

All four configurations achieve 100% ASR before recompression, but differ afterward. The three-mask pool preserves single-compression attack success while having lower recompression ASR than the individual patterns at the tested qualities. Together with Tab. 11, the checkerboard results indicate a trade-off: finer allocation is harder for the matched detector to distinguish, but more sensitive to recompression and low-pass filtering. The detection experiment does not evaluate concentric squares or the mask pool, so it does not establish a detection advantage for mask diversity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Sec. 1

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Sec. 5

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: the paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We plan to release the implementation.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Sec. 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper primarily focuses on the technical aspects of compression-aware backdoor attacks. Section 5 discusses limitations and potential detection strategies.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release pretrained backdoored models or poisoned datasets. We evaluate the proposed methods using existing benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing assets used in the paper have been properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: None.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: The human inspection study is reported in Sec.4 and partial detailed information is not included.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The human inspection study involved a low-risk visual perception task. Formal IRB approval was not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.