SNEAKDOOR: Stealthy Backdoor Attacks against Distribution Matching-based Dataset Condensation

He Yang[1, *], **Dongyi Lv**[1,*], **Song Ma**[1], **Wei Xi**[1,‡], **Jizhong Zhao**[1]
[1] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
yanghe73@xjtu.edu.cn, lvdongyi@stu.xjtu.edu.cn, song.ma@stu.xjtu.edu.cn,
xiwei@xjtu.edu.cn, zjz@xjtu.edu.cn

Abstract

Dataset condensation aims to synthesize compact yet informative datasets that retain the training efficacy of full-scale data, offering substantial gains in efficiency. Recent studies reveal that the condensation process can be vulnerable to backdoor attacks, where malicious triggers are injected into the condensation dataset, manipulating model behavior during inference. While prior approaches have made progress in balancing attack success rate and clean test accuracy, they often fall short in preserving stealthiness, especially in concealing the visual artifacts of condensed data or the perturbations introduced during inference. To address this challenge, we introduce SNEAKDOOR, which enhances stealthiness without compromising attack effectiveness. SNEAKDOOR exploits the inherent vulnerability of class decision boundaries and incorporates a generative module that constructs input-aware triggers aligned with local feature geometry, thereby minimizing detectability. This joint design enables the attack to remain imperceptible to both human inspection and statistical detection. Extensive experiments across multiple datasets demonstrate that SNEAKDOOR achieves a compelling balance among attack success rate, clean test accuracy, and stealthiness, substantially improving the invisibility of both the synthetic data and triggered samples while maintaining high attack efficacy. The code is available at `https://github.com/XJTU-AI-Lab/SneakDoor`.

1 Introduction

Dataset Condensation (DC) [1, 2, 3, 4, 5, 6] has recently emerged as a powerful paradigm for synthesizing compact training datasets that retain the learning efficacy of their full-sized counterparts, offering substantial benefits in terms of computation, memory, and deployment efficiency. However, DC introduces inherent vulnerabilities to backdoor attacks [7, 8, 9, 10], where malicious triggers can be injected into the distilled samples during the condensation process. Once compromised, the distilled dataset can disseminate malicious behaviors across downstream models, undermining model integrity and posing serious security threats.

A growing body of work demonstrates that malicious triggers, once implanted into the distilled set, can persist across downstream training and inference, leading to consistent and targeted misclassification [11, 12, 13]. One of the earliest approaches is the Naive Attack [11], which directly adds a fixed visual pattern (typically a static patch) to instances from clean training samples before condensation. While conceptually simple, this method suffers from limited attack success rates, as the uniform trigger tends to degrade through the condensation process. To enhance attack effectiveness, Doorping [11] introduces a bilevel optimization framework that iteratively updates both the distilled data and the backdoor trigger during training. Doorping better preserves the trigger semantics and achieves stronger attack success rate. However, it incurs significant computational cost due to its bilevel nature and lacks a theoretical foundation. A more recent work [12] adopts a kernel-theoretic

[*] Equal contribution, [‡] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

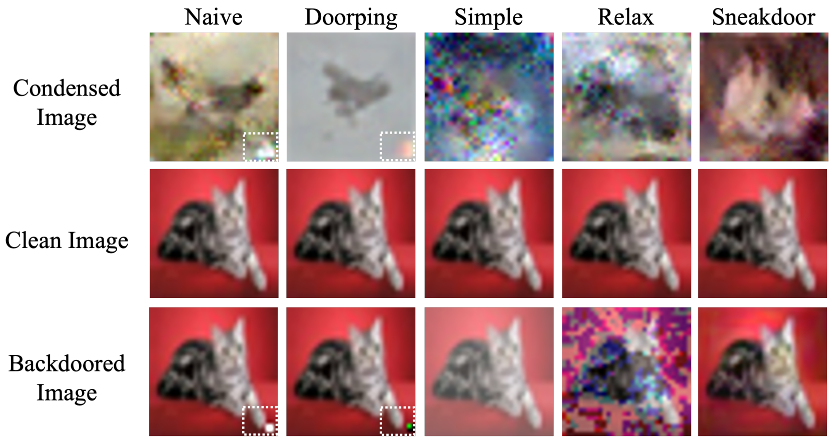

Figure 1: Stealthiness Illustration

lens to reinterpret backdoor vulnerability in condensation. They propose two variants, simple-trigger and relax-trigger. The former attack focuses exclusively on minimizing the generalization gap, aiming to ensure that the backdoor learned during condensation reliably transfers to test-time behavior. The relax-trigger introduces a joint optimization objective that simultaneously reduces projection loss (mismatch between synthetic and clean distributions), conflict loss (interference between clean and poisoned instances), and the generalization gap. Notably, relax-trigger maintains high attack success rate while avoiding the computational overhead of bilevel optimization.

However, existing approaches fall short of achieving a well-calibrated trade-off among attack success rate (ASR), clean test accuracy (CTA), and stealthiness (STE). While some methods attain high ASR or maintain acceptable CTA, they frequently neglect STE, a critical dimension that reflects the visual and statistical imperceptibility of both the distilled data and the triggered inputs (See Figure 1). This oversight is particularly damaging, without sufficient stealthiness, even highly effective attacks become vulnerable to detection, significantly limiting their practical viability. This persistent imbalance motivates our proposed method, SNEAKDOOR, which leverages input-aware trigger generation and decision boundary sensitivity, achieving a more favorable balance among ASR, CTA, and STE.

Specifically, SNEAKDOOR consists of two stages, (1) Trigger Generation and (2) Backdoor Injection. In the first stage, a generative network is trained to produce input-aware triggers tailored to individual samples. By aligning each trigger with the local semantic content of its host image, the perturbations remain visually coherent and difficult to isolate. In the second stage, the backdoor injection is formulated as an optimization problem. The generated triggers are embedded into a subset of clean samples to form a poisoned subset. These triggered samples are then incorporated into the training set prior to condensation, allowing the distilled dataset to encode backdoor behavior alongside clean task representations. As a result, downstream models trained on the synthesized data exhibit the intended malicious behavior without sacrificing generalization to clean inputs.

Our contributions are summarized below:

- We present the first investigation of backdoor attacks against distribution matching-based dataset condensation, with a focus on jointly optimizing ASR, CTA, and STE.
- We provide a theoretical analysis of stealthiness concerning SNEAKDOOR, offering formal guarantees and insights into the conditions under which backdoor signals remain undetectable throughout the condensation and training process.
- Extensive experiments across six datasets demonstrate that SNEAKDOOR consistently outperforms existing methods in achieving a superior balance across ASR, CTA, and STE.

2 Related Work

Distribution Matching-based Dataset Condensation: Dataset condensation (DC) aims to synthesize a compact set of synthetic samples that can replace large-scale datasets while preserving comparable model performance. Among various condensation paradigms, distribution matching (DM)-based methods have emerged as a leading approach due to their scalability, generality, and empirical effectiveness. Unlike earlier techniques based on gradient matching or training trajectory alignment, DM-based methods directly align statistical or feature-level distributions between real and synthetic data. A seminal example is DM [3], which matches the second-order moments (covariance) of feature embeddings extracted by random encoders. A core formulation in distribution matching-based dataset condensation leverages the maximum mean discrepancy (MMD) to quantify the distance between the feature distributions of real and synthetic samples in a high-dimensional embedding space. The objective is to minimize this discrepancy over the synthetic set $\mathcal{S}$, ensuring statistical alignment with the original dataset $\mathcal{T}$. Specifically, the optimization problem is defined as: $\min_{\mathcal{S}} \mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\theta}}} \| \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|} \psi_{\boldsymbol{\vartheta}}(\mathcal{A}(\boldsymbol{x}_i,\omega)) - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|} \psi_{\boldsymbol{\vartheta}}(\mathcal{A}(\boldsymbol{s}_j,\omega))\|^2$, where $\psi_{\boldsymbol{\vartheta}}$ is a randomly initialized and fixed embedding function, and $\mathcal{A}(\cdot,\omega)$ denotes a differentiable Siamese augmentation operator applied to both real and synthetic samples, parameterized by ω. This formulation encourages the synthetic set to preserve the statistical structure of the real dataset under randomized transformations, thereby promoting generalization across model initializations drawn from $P_{\boldsymbol{\theta}}$.

Subsequent extensions, such as IDM and DAM, enhance class-conditional alignment through kernel-based moment matching, adaptive feature regularization, and encoder updates, yielding improved performance. IDM introduces practical enhancements to the original distribution matching framework, incorporating progressive feature extractor updates, stronger data augmentations, and dynamic class balancing to improve generalization. In parallel, DataDAM leverages attention map alignment to better preserve spatial semantics, guiding synthetic samples to activate similar regions as real data while maintaining computational efficiency. These methods advance the state of dataset condensation by demonstrating that richer supervision and adaptive training dynamics are critical for generating high-fidelity synthetic datasets.

Backdoor Attacks against Dataset Condensation: Backdoor attacks aim to manipulate model behavior at inference time by injecting carefully crafted triggers into a subset of training data. When effective, the model performs normally on clean inputs but consistently misclassifies inputs containing the trigger. While extensively studied in standard supervised learning, backdoor attacks in the context of dataset condensation have only recently received attention. A pioneering study by Liu et al. [11] introduces backdoors by poisoning real data before dataset condensation. Their Naive Attack appends a fixed trigger to target-class samples before condensation, but suffers from trigger degradation and reduced attack efficacy due to the synthesis process. To address this, Doorping employs a bilevel optimization scheme that jointly refines the trigger and the synthetic data. Although more effective, it incurs substantial computational overhead. More recently, Chung et al. [12] provide a kernel-theoretic perspective on backdoor persistence in condensation. They propose simple-trigger, which minimizes the generalization gap of the backdoor effect, and relax-trigger, which further reduces projection and conflict losses for improved robustness.

Importantly, existing approaches focus predominantly on maximizing ASR or preserving CTA, often overlooking STE, which is a critical factor for realistic attacks. In contrast, we propose SNEAKDOOR, a novel framework that explicitly addresses the ASR–CTA–STE trade-off through input-aware trigger generation and stealth-aware integration into distribution matching-based condensation.

3 Methodology

3.1 Threat Model

Attack Scenario. We consider a *collaborative setting* where one entity possesses a high-quality dataset and shares a compact version with another party via dataset condensation, due to privacy or bandwidth constraints. The condensed dataset is typically regarded as a trustworthy proxy for training. However, this trust can be exploited. A malicious provider, with full access to the original data and sole control over the condensation process, can embed backdoor triggers into the synthetic data. These triggers, while preserving high utility for clean tasks, can cause targeted misclassification in downstream models.

Moreover, our threat model does *not* assume that the attacker knows the downstream (victim) model architecture. This upstream threat underscores a critical vulnerability: even limited data sharing can serve as a potent attack vector when the condensation process is adversarially controlled.

Attacker's Goal. The attacker's objective in backdooring condensed datasets is inherently multi-faceted, requiring a delicate balance among three goals: stealthiness (STE), attack success rate (ASR), and clean test accuracy (CTA). Due to space constraints, detailed definitions of these metrics are provided in Appendix A.

3.2 Stealthy Backdoor Attack against Dataset Condensation

(1) *Trigger Generation*

Trigger generation starts by identifying the source–target class pair (i, j) with the highest inter-class misclassification rate:

$$\mathcal{O}_{i\to j} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{I}\big(g_{\theta_c}(f_{\theta_f}(x_k)) = j\big), \quad x_k \in \mathcal{T}_i, \tag{1}$$

where $\mathcal{T}_i$ represents the subset of the original dataset $\mathcal{T}$ with ground-truth label i, f_{θ_f} and g_{θ_c} denote the feature extractor and classifier, respectively, $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the classifier assigns the sample x_k to class j, and 0 otherwise. In practice, we estimate $\mathcal{O}_{i\to j}$ by sampling N examples from class i, mapping them to the latent space with f_{θ_f}, and computing the fraction that g_{θ_c} assigns to class j.

We evaluate $\mathcal{O}_{i\to j}$ for all ordered class pairs and select the pair with the maximal value. The chosen pair indicates the most error-prone direction for label confusion; a trigger is then designed to exploit this specific weakness. By targeting the pair with highest misclassification rate, the attack achieves consistent source→target misclassification while limiting collateral impact on overall model accuracy.

The computation of $\mathcal{O}_{i\to j}$ depends on the model parameters $\theta = \{\theta_f, \theta_c\}$, which correspond to the feature extractor f_{θ_f} and the classifier f_{θ_c}, respectively. To obtain these parameters, we first construct a condensed dataset $\mathcal{S} = \{(x'_i, y'_i)\}_{i=1}^{N}$ from the original dataset $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{M}$, where $N \ll M$. The synthetic dataset $\mathcal{S}$ is generated by minimizing a distribution-matching objective over randomly initialized models, ensuring that training on $\mathcal{S}$ approximates the behavior of models trained on the full dataset $\mathcal{T}$:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathbb{E}_{x\sim p_{\mathcal{T}},\, x'\sim p_{\mathcal{S}},\, \theta\sim p_\theta} D(P_{\mathcal{T}}(x;\theta),\ P_{\mathcal{S}}(x';\theta)) + \lambda\, \mathcal{R}(\mathcal{S}), \tag{2}$$

where $P_{\mathcal{T}}(x;\theta)$ and $P_{\mathcal{S}}(x';\theta)$ denote the feature distributions induced by the original and condensed datasets, respectively. The distance measure $D(\cdot,\cdot)$, such as Maximum Mean Discrepancy (MMD), quantifies the discrepancy between these distributions. $\mathcal{R}(S)$ is a regularization term, and λ balances the trade-off between distribution alignment and regularization.

After generating the condensed dataset $\mathcal{S}$, we train a surrogate model parameterized by $\theta = \{\theta_f, \theta_c\}$ using only $\mathcal{S}$. This surrogate serves as an efficient approximation of the downstream model's decision behavior. Once trained, it is evaluated on the original dataset $\mathcal{T}$, and a normalized confusion matrix is computed to analyze inter-class prediction tendencies.

$$\begin{aligned} C &= \frac{C_{ij}}{\sum_{j=0}^{o_c-1} C_{ij}} \\ C_{ij} &= \sum_{(x,y)\in\mathcal{T}} \mathbb{I}[y=i]\mathbb{I}[g_{\theta_c}(f_{\theta_f})(x) = j] \end{aligned} \tag{3}$$

where o_c is the total number of classes in the original dataset $\mathcal{T}$. C_{ij} represents the empirical probability that a sample from class i is misclassified as class j. The maximum inter-class misclassification rate $\mathcal{O}_{y_s\to y_\tau}$ is then calculated as follows:

$$\mathcal{O}_{y_s\to y_\tau} = \arg\max_{i,j} C_{ij}, \quad i \neq j \tag{4}$$

This measure identifies the class pair (i, j) with the highest misclassification probability, revealing the most vulnerable decision boundary in the model.

We then proceed to the trigger generation phase, where the objective is to create a trigger that, when added to an input sample, causes the model to misclassify the input from the source class y_s to the target class y_τ. Speicifically, we utilize a generator model G_ϕ, which generates perturbations, or triggers, which are added to the original input data. The perturbation is designed to be imperceptible, ensuring the trigger remains stealthy while causing misclassification. The trigger generation process can be represented as follows:

$$\begin{aligned} &\widetilde{x} = x + \alpha G_\phi(x), \quad \forall x \in \mathcal{T}_{y_s} \\ &s.t. \quad \|G_\phi(x)\|_\infty < \varepsilon, \quad \forall x \end{aligned} \tag{5}$$

where $G_\phi(x)$ represents the generated adversarial noise, while ε is a constraint that controls the maximum permissible perturbation, ensuring that the perturbation remains subtle and undetectable. The perturbed input is denoted as $\widetilde{x}$. The subset $\mathcal{T}_{y_s}$ refers to the portion of the original dataset for which the label is y_s. α is a small constant, further controlling the size of the perturbation.

In practice, the maximum permissible perturbation constraint in Eq.(5) is enforced by applying a clamping operation to the generator output $G_\phi(x)$ before adding it to the original input. Specifically, the adversarial noise is clamped such that its ℓ_∞-norm lies within the range $[-\varepsilon, \varepsilon]$, ensuring the perturbation remains imperceptible. This clamped noise is then added to the clean image, followed by another clamping step to maintain the pixel values within the valid image range. The loss in Eq.(6) is computed on these clamped, perturbed images, allowing the generator to be implicitly optimized under the perturbation constraint without the need for an explicit penalty term in the objective.

The generator model G_ϕ is trained alongside $\theta = \{\theta_f, \theta_c\}$, with the objective of minimizing the classification loss associated with the target class y_τ. Specifically, the generator is updated based on the following objective function:

$$\phi = \phi - \eta_\phi \sum_{x \in \mathcal{T}_{y_s}} \mathcal{L}\left(g_{\theta_c}(f_{\theta_f}(x + G_\phi(x))), y_\tau\right) \tag{6}$$

where $\mathcal{L}$ is the loss function, which measures the error in predicting the target class y_τ after applying the trigger to the input x, and η_ϕ is the learning rate for the generator.

By iteratively updating the generator, the generator G_ϕ is refined to produce more effective backdoor triggers. The process continues until the trigger causes consistent misclassifications of the source class y_s as the target class y_τ, while keeping the perturbation within the imperceptibility threshold ε. This approach enables the adversary to design highly effective backdoor triggers, leveraging the generator to produce stealthy perturbations that successfully compromise the performance of the downstream model.

(2) *Backdoor Injection*

Once the generator G_ϕ has been trained to generate perturbations that cause misclassifications of the source class y_s to the target class y_τ, we proceed with the backdoor injection process. This step involves adding the learned perturbations to the source class samples in the original dataset $\mathcal{T}$. Specifically, we add the perturbations generated by G_ϕ to each sample $x \in \mathcal{T}_{y_s}$:

$$\widetilde{x} = x + \alpha G_\phi(x) \quad \forall x \in \mathcal{T}_{y_s} \tag{7}$$

where $\widetilde{x}$ represents the perturbed sample, and $G_\phi(x)$ is the perturbation generated by the adversarial generator. These perturbed samples are then relabeled to the target class y_τ.

This process ensures that adversarial perturbations are applied to the samples from the source class, resulting in a set of triggered samples, $\mathcal{T}_{\text{triggered}} = (\widetilde{x}, y_\tau)_{i=1}^{N_{\text{triggered}}}$, where the perturbed inputs are labeled as the target class y_τ. In the subsequent step, the triggered samples are incorporated with the clean samples from the target class y_τ. The primary objective of this combination is to introduce a fraction of the triggered samples into the target class, thereby facilitating the model to misclassify source class samples as the target class when subjected to the adversarial trigger. This process ensures that the model's decision boundary is subtly manipulated to favor misclassification under specific conditions. Let $N_{\text{triggered}}$ be the total number of triggered samples generated in the previous step, each labeled with the target class y_τ. The number of clean samples in the target class y_τ in the original dataset $\mathcal{T}y_\tau$ is denoted by $N_{\mathcal{T}y_\tau}$. Based on the poison ratio ρ, we will add $\rho \cdot N_{\mathcal{T}_{y_\tau}}$ triggered samples into $\mathcal{T}_{y_\tau}$. Specifically, we first randomly select $\rho \cdot N_{\mathcal{T}_{y_\tau}}$ samples from $\mathcal{T}_{\text{triggered}}$ and add them into $\mathcal{T}_{y_\tau}$. The resulting poisoned dataset $\mathcal{T}_{\text{mixed}}$ consists of both the clean target class samples and the triggered samples:

$$\mathcal{T}_{\text{mixed}} = \mathcal{T}_{y_\tau} \cup \{(\widetilde{x}, y_\tau)\}_{i=1}^{\rho \cdot N_{\mathcal{T}_{y_\tau}}} \tag{8}$$

The next step is to recondense the target class $\mathcal{T}_{y_\tau}$. The objective of recondensation is to generate a new subset $\mathcal{S}_{y_\tau}$ within the synthetic dataset, which preserves the key characteristics of the target class while amplifying the influence of the triggered samples. This process seeks to strike a balance between maintaining the intrinsic features of the target class and maximizing the impact of the adversarial samples. Specifically, the objective is to generate a synthetic dataset $\mathcal{S}y_\tau$ that closely approximates the target class distribution in the poisoned data $\mathcal{T}y_\tau$. The optimization objective is defined as:

$$\mathcal{S}^*_{y_\tau} = \underset{\mathcal{S}_{y_\tau}}{\arg\min} \mathbb{E}_{x\sim p_{\mathcal{T}_{\text{mixed}}}, x'\sim p_{\mathcal{S}_{y_\tau}}, \theta\sim p_\theta} D\left(P_{\mathcal{T}_{\text{mixed}}}(x;\theta), P_{\mathcal{S}_{y_\tau}}(x';\theta)\right) + \lambda\mathcal{R}(\mathcal{S}_{y_\tau}) \tag{9}$$

where $P_{\mathcal{T}_{\text{mixed}}}(x;\theta)$ is the probability distribution of the target class incorporating triggered samples. $P_{\mathcal{S}_{y_\tau}}(x';\theta)$ is the probability distribution of the recondensed target class.

4 Stealthiness Analysis

A critical challenge in designing effective backdoor attacks on dataset condensation is achieving stealthiness, ensuring that poisoned samples and the resulting synthetic data are indistinguishable from their clean counterparts. Our goal is to formalize stealthiness through a geometric and distributional lens, grounded in the feature space induced by deep neural architectures.

To this end, our analysis is guided by the following question: How does input-aware backdoor injection perturb the structure of data manifolds in feature space, and can this deviation be rigorously bounded to guarantee stealth? Since distribution matching-based condensation aligns global feature statistics (*e.g.*, moments of embedded data), it is essential to understand whether triggers introduce detectable geometric or statistical anomalies in the condensed representation. We conduct our analysis in a Reproducing Kernel Hilbert Space (RKHS), where class-specific data, both clean and triggered, are assumed to lie on smooth, locally compact manifolds. By modeling the trigger as a bounded, input-aware perturbation and invoking assumptions on manifold regularity and inter-class proximity, we show that triggered samples remain tightly coupled to the clean data manifold under mild conditions. This theoretical framework enables us to quantify the effect of poisoning both at the feature level (Theorem 3) and at the level of the condensed dataset (Theorem 2). These results provide principled justification for SNEAKDOOR's empirical stealth: the perturbations introduced by the trigger remain latent-space-aligned and distributionally consistent, limiting their detectability after condensation.

Formal statements of assumptions, intermediate lemmas, and proofs supporting our theoretical analysis are deferred to Appendix B for clarity and completeness.

Definition 1 (Kernel). *$k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ on a non-empty set $\mathcal{X}$ is a kernel if it satisfies the following two conditions: (1) symmetry: $k(x, x') = k(x', x), \quad \forall x, x' \in \mathcal{X}$. (2) Positive Semi-Definiteness: for any finite subset $\{x_1, x_2, \cdots, x_n\} \subset \mathcal{X}$, the Gram matrix $\mathbf{K} = [k(x_i, x_j)]^n_{i,j=1}$ is positive semi-definite.*

Definition 2 (Reproducing Kernel Hilbert Space, RKHS). *Given a kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, the Reproducing Kernel Hilbert Space $\mathcal{H}_k$ is a Hilbert space of functions $f : \mathcal{X} \mapsto \mathbb{R}$ satisfying: (1) For every $x \in \mathcal{X}$, the function $k(x, \cdot) \in \mathcal{H}_k$. (2) $\forall x \in \mathcal{X}$ and $f \in \mathcal{H}_k$, $f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}_k}$.*

Theorem 1 (Upper Bound on Feature-Manifold Deviation under Poisoning). *Let $\mathcal{T}_{y_\tau}$ denote the clean target-class dataset and $\mathcal{T}_{\text{triggered}}$ the triggered (poisoned) dataset, with corresponding feature-space distributions $P_{\mathcal{M}_{\text{clean}}}$ and $P_{\mathcal{M}_{\text{triggered}}}$, respectively. Define the mixed distribution as: $P_{\mathcal{M}_{\text{mixed}}} = (1-\rho)P_{\mathcal{M}_{\text{clean}}} + \rho P_{\mathcal{M}_{\text{triggered}}}$, where $\rho \in [0, 1]$ denotes the poisoning ratio. Under Assumptions 1 (Lipschitz Continuity), 2 (Local Compactness of Feature Manifold), and 3 (Inter-Class Hausdorff Distance), the expected deviation of samples from the mixed distribution to the target feature manifold satisfies:*

$$\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{mixed}}}}\left[\inf_{z_\tau \in \mathcal{M}_{\text{clean}}} \|z - z_\tau\|_{\mathcal{H}}\right] \leq \rho(\gamma\varepsilon + \delta), \tag{10}$$

where $\mathcal{H}$ is the RKHS associated with the feature encoder.

Theorem 2 (Upper Bound on the Discrepancy Between Poisoned and Clean Condensation Datasets). *Let $\mathcal{T}_{y_\tau}$ denote the clean target-class dataset and $\mathcal{T}_{\text{mixed}} = \mathcal{T}_{y_\tau} \cup \mathcal{T}_{\text{triggered}}$, where $\mathcal{T}_{\text{triggered}}$ consists of source-class samples $x \in \mathcal{T}_{y_s}$ perturbed by a trigger generator G_ϕ and relabeled as the target class.*

Let $\mathcal{S}_{\mathrm{clean}}$ and $\mathcal{S}_{\mathrm{poison}}$ denote the condensation datasets distilled from $\mathcal{T}_{y_\tau}$ and $\mathcal{T}_{\mathrm{mixed}}$, respectively, by minimizing: $\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathrm{MMD}(\mathcal{T}, \mathcal{S}) + \lambda \mathcal{R}(\mathcal{S})$, where $\mathcal{T} \in \{\mathcal{T}_{y_\tau}, \mathcal{T}_{\mathrm{mixed}}\}$, $\lambda > 0$, and $\mathcal{R}$ is a μ_R strongly convex regularizer. Under Assumptions 1 (Lipschitz Continuity), 2 (Local Compactness of Feature Manifold), and 3 (Inter-Class Hausdorff Distance), the MMD between $\mathcal{S}_{\mathrm{clean}}$ and $\mathcal{S}_{\mathrm{poison}}$ satisfies:

$$\mathrm{MMD}(\mathcal{S}_{\mathrm{clean}}, \mathcal{S}_{\mathrm{poison}}) \leq \frac{L_f^2 \rho(\gamma\varepsilon + \delta)}{\lambda \mu_R}$$

where $\gamma = L_f \alpha$, $\delta = \sup_{z_s \in \mathcal{M}_{\mathrm{source}}} \inf_{z_\tau \in \mathcal{M}_{\mathrm{clean}}} \|z_s - z_\tau\|_{\mathcal{H}}$, ρ is the poisoning rate, and ε bounds the input perturbation.

5 Experiments

Datasets and Networks. We evaluate SNEAKDOOR across five standard datasets: FMNIST [14], CIFAR-10 [15], SVHN [16], Tiny-ImageNet [17], STL-10 [18], and ImageNette [19]. These datasets span a diverse range of visual complexity, semantic granularity, and image resolution, enabling a comprehensive evaluation of attack generality. Each dataset is processed according to the standard dataset condensation protocol, with 50 images per class used for condensation. Specifically, we adopt two common synthetic data backbones: ConvNet and AlexNetBN [20], which represent lightweight and moderately expressive condensation encoders. For downstream training and evaluation, we consider four architectures: ConvNet, AlexNetBN, VGG11 [21], and ResNet18 [22]. Moreover, we evaluate SNEAKDOOR in comparison with four state-of-the-art attacks: NAIVE [11], DOORPING [11], SIMPLE [12], and RELAX [12].

Evaluation Metrics. We evaluate attack performance across three key dimensions: ASR, CTA, and STE. Following prior work [23], STE is quantified using three complementary metrics: (1) PSNR (Peak Signal-to-Noise Ratio), measuring pixel-level similarity between triggered and clean samples, where higher values indicate lower perceptual distortion. (2) SSIM (Structural Similarity Index), which measures structural similarity, with values closer to 1 indicating stronger visual alignment; and (3) IS (Inception Score) quantifies the KL divergence between the predicted label distribution of a sample and the marginal distribution over all samples. Lower IS values suggest reduced recognizability, indicating higher stealth and improved resistance to detection. For convenience, we define an inverted score $\mathrm{IS}^\dagger = (10^{-3} - \mathrm{IS})\mathrm{e}^{-4}$, where larger values correspond to improved stealth.

Overall Attack Effectiveness. We first evaluate the overall effectiveness of each backdoor attack in balancing three key objectives: ASR, CTA, and STE. To illustrate this trade-off, we visualize the normalized performance of each method using radar plots (Figure 2, Figure 3) that jointly capture all three dimensions. SNEAKDOOR consistently achieves a superior balance across the three criteria. In contrast, while Doorping and Relax achieve high ASR, they suffer from significant degradation in either CTA or STE. Conversely, Naive and Simple maintain better CTA but fail to deliver competitive ASR or STE. These results validate our central hypothesis: *input-aware trigger design combined with distribution-aligned injection enables the attack that is both effective and stealthy*.

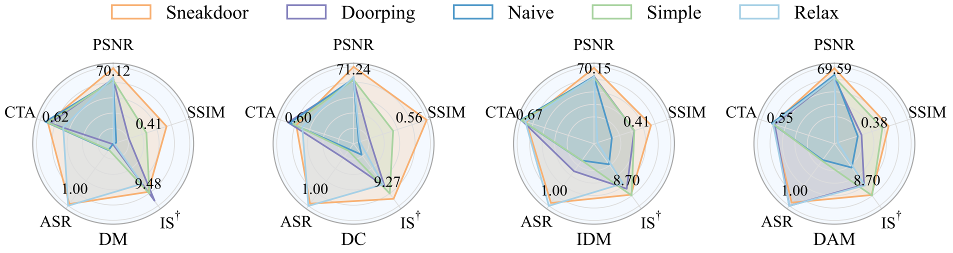

Figure 2: Attack Performance on STL10. Larger area indicates better balance.

Effectiveness on Different Datasets To rigorously assess the effectiveness of SNEAKDOOR, we evaluate CTA and ASR across five datasets and four dataset condensation baselines: DM [3], DC [24], IDM [25], and DAM [26]. Results are summarized in Table 1, with each entry reporting the mean and standard deviation over five random seeds. SNEAKDOOR consistently achieves high ASR across all datasets and condensation methods, while maintaining competitive CTA. These results highlight

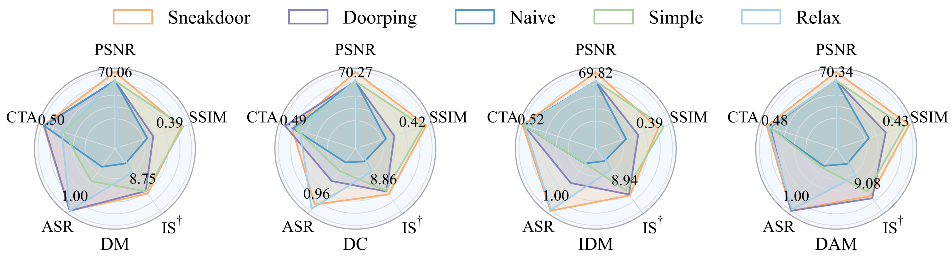

Figure 3: Attack Performance on Tiny-ImageNet. Larger area indicates better balance.

the robustness and generalizability of SNEAKDOOR, with improvements most evident in scenarios where baseline methods overfit to specific condensation schemes.

Table 1: Effectiveness on Different Datasets

Dataset	Method	SNEAKDOOR		DOORPING		SIMPLE		RELAX	
		CTA	ASR	CTA	ASR	CTA	ASR	CTA	ASR
CIFAR10	DM	0.626 ± 0.001	0.989 ± 0.000	0.621 ± 0.001	0.988 ± 0.005	0.584 ± 0.000	0.590 ± 0.012	0.574 ± 0.000	1.000 ± 0.000
	DC	0.537 ± 0.000	0.996 ± 0.000	0.566 ± 0.001	1.000 ± 0.000	0.497 ± 0.001	0.657 ± 0.021	0.511 ± 0.001	1.000 ± 0.000
	IDM	0.643 ± 0.002	0.975 ± 0.001	0.654 ± 0.002	0.165 ± 0.007	0.652 ± 0.001	0.142 ± 0.008	0.653 ± 0.002	0.522 ± 0.021
	DAM	0.591 ± 0.001	0.979 ± 0.001	0.531 ± 0.001	1.000 ± 0.000	0.537 ± 0.001	0.674 ± 0.032	0.559 ± 0.001	1.000 ± 0.001
STL10	DM	0.598 ± 0.001	0.973 ± 0.000	0.577 ± 0.001	0.149 ± 0.007	0.597 ± 0.001	0.096 ± 0.009	0.596 ± 0.001	1.000 ± 0.001
	DC	0.565 ± 0.001	0.998 ± 0.001	0.598 ± 0.001	0.227 ± 0.011	0.550 ± 0.001	0.112 ± 0.011	0.563 ± 0.000	0.998 ± 0.001
	IDM	0.658 ± 0.001	0.979 ± 0.001	0.661 ± 0.001	0.314 ± 0.015	0.658 ± 0.001	0.100 ± 0.007	0.658 ± 0.001	0.954 ± 0.011
	DAM	0.532 ± 0.001	0.992 ± 0.001	0.533 ± 0.001	1.000 ± 0.000	0.535 ± 0.001	0.103 ± 0.004	0.535 ± 0.001	1.000 ± 0.000
FMNIST	DM	0.876 ± 0.001	0.998 ± 0.000	0.876 ± 0.000	0.093 ± 0.006	0.868 ± 0.000	0.178 ± 0.005	0.828 ± 0.000	1.000 ± 0.000
	DC	0.851 ± 0.001	0.998 ± 0.000	0.872 ± 0.001	1.000 ± 0.000	0.837 ± 0.001	0.277 ± 0.014	0.824 ± 0.001	1.000 ± 0.000
	IDM	0.877 ± 0.001	1.000 ± 0.000	0.884 ± 0.000	0.998 ± 0.002	0.879 ± 0.000	0.159 ± 0.007	0.875 ± 0.001	1.000 ± 0.000
	DAM	0.877 ± 0.000	0.996 ± 0.000	0.813 ± 0.001	1.000 ± 0.000	0.880 ± 0.000	0.151 ± 0.012	0.874 ± 0.000	1.000 ± 0.000
SVHN	DM	0.800 ± 0.000	1.000 ± 0.000	0.780 ± 0.001	1.000 ± 0.001	0.748 ± 0.000	0.110 ± 0.007	0.747 ± 0.000	1.000 ± 0.000
	DC	0.687 ± 0.000	1.000 ± 0.000	0.583 ± 0.001	0.703 ± 0.017	0.636 ± 0.001	0.100 ± 0.009	0.689 ± 0.001	1.000 ± 0.000
	IDM	0.831 ± 0.001	0.986 ± 0.001	0.839 ± 0.001	0.061 ± 0.006	0.842 ± 0.001	0.114 ± 0.008	0.834 ± 0.002	0.992 ± 0.003
	DAM	0.782 ± 0.001	1.000 ± 0.000	0.721 ± 0.000	1.000 ± 0.000	0.759 ± 0.001	0.114 ± 0.005	0.745 ± 0.001	1.000 ± 0.000
TINY IMAGENET	DM	0.503 ± 0.001	1.000 ± 0.000	0.496 ± 0.002	1.000 ± 0.000	0.493 ± 0.003	0.100 ± 0.004	0.494 ± 0.003	0.996 ± 0.000
	DC	0.432 ± 0.002	1.000 ± 0.000	0.492 ± 0.001	0.398 ± 0.005	0.391 ± 0.002	0.192 ± 0.006	0.418 ± 0.003	0.952 ± 0.001
	IDM	0.517 ± 0.004	1.000 ± 0.000	0.512 ± 0.005	0.089 ± 0.013	0.509 ± 0.003	0.046 ± 0.002	0.484 ± 0.006	0.941 ± 0.002
	DAM	0.482 ± 0.003	1.000 ± 0.000	0.449 ± 0.003	1.000 ± 0.000	0.458 ± 0.003	0.082 ± 0.002	0.465 ± 0.002	0.973 ± 0.001

Effectiveness on Cross Architectures To evaluate SNEAKDOOR in cross-architecture settings, where the condensation model differs from the downstream model, we follow prior work [11] and consider four architectures: ConvNet, AlexNetBN, VGG11, and ResNet18. Specifically, we use ConvNet or AlexNetBN for data condensation and the remaining models for downstream training.

As shown in Table 2, we evaluate SNEAKDOOR. across 36 cross-architecture scenarios spanning various datasets, condensation methods, and downstream models. SNEAKDOOR demonstrates consistent performance across most architecture pairs, indicating strong transferability. However, when using the DC algorithm, performance systematically degrades on specific architectures. Prior studies, as well as our own findings, suggest that DC often produces lower-quality distilled datasets, as reflected in its relatively low CTA. This implies that the reduced ASR in these cases is more likely due to *DC's limited ability to retain both task-relevant and backdoor-relevant information, rather than a shortcoming of the attack mechanism itself*. When excluding DC-based cases, 27 scenarios remain, of which only 6 exhibit ASR below 90%. This demonstrates that SNEAKDOOR consistently achieves high ASR in most settings, provided the underlying condensed data is of sufficient quality.

Table 2: Cross-architecture CTA and ASR

Dataset	Network	DM		DC		IDM		DAM	
		CTA	ASR	CTA	ASR	CTA	ASR	CTA	ASR
CIFAR10	VGG11	0.568 ± 0.000	0.971 ± 0.000	0.472 ± 0.000	0.865 ± 0.000	0.645 ± 0.000	0.719 ± 0.008	0.539 ± 0.000	0.929 ± 0.001
	AlexNetBN	0.616 ± 0.001	0.942 ± 0.002	0.426 ± 0.004	0.000 ± 0.000	0.689 ± 0.002	0.539 ± 0.003	0.623 ± 0.001	0.902 ± 0.004
	ResNet18	0.548 ± 0.001	0.959 ± 0.000	0.435 ± 0.001	0.534 ± 0.003	0.656 ± 0.001	0.766 ± 0.003	0.510 ± 0.001	0.857 ± 0.002
STL10	VGG11	0.587 ± 0.001	0.999 ± 0.001	0.564 ± 0.000	0.790 ± 0.003	0.676 ± 0.001	0.900 ± 0.001	0.582 ± 0.000	0.924 ± 0.001
	AlexNetBN	0.589 ± 0.002	0.905 ± 0.005	0.542 ± 0.001	0.796 ± 0.002	0.670 ± 0.003	0.798 ± 0.005	0.636 ± 0.001	0.981 ± 0.001
	ResNet18	0.463 ± 0.001	0.989 ± 0.000	0.396 ± 0.001	0.783 ± 0.003	0.647 ± 0.001	0.949 ± 0.001	0.436 ± 0.001	0.941 ± 0.002
TINY IMAGENET	VGG11	0.488 ± 0.001	1.000 ± 0.000	0.384 ± 0.001	1.000 ± 0.000	0.541 ± 0.002	1.000 ± 0.000	0.449 ± 0.002	1.000 ± 0.000
	AlexNetBN	0.517 ± 0.003	0.796 ± 0.015	0.292 ± 0.007	0.704 ± 0.008	0.572 ± 0.004	1.000 ± 0.000	0.541 ± 0.003	1.000 ± 0.000
	ResNet18	0.456 ± 0.002	1.000 ± 0.000	0.358 ± 0.001	0.524 ± 0.008	0.483 ± 0.005	0.988 ± 0.010	0.438 ± 0.002	1.000 ± 0.000

Evaluation of Stealthiness As shown in Figure 4, SNEAKDOOR consistently achieves the highest PSNR and SSIM across all condensation methods, highlighting its ability to produce visually and structurally imperceptible triggers. In contrast, the other methods exhibit notable declines in both metrics, suggesting visible artifacts or structural distortions in the perturbed samples. Moreover, while Simple and Naive achieve slightly lower IS values, they fail to maintain competitive ASR or CTA, limiting their overall effectiveness. SNEAKDOOR achieves a similarly low IS while preserving high ASR, indicating enhanced stealth without sacrificing attack strength.

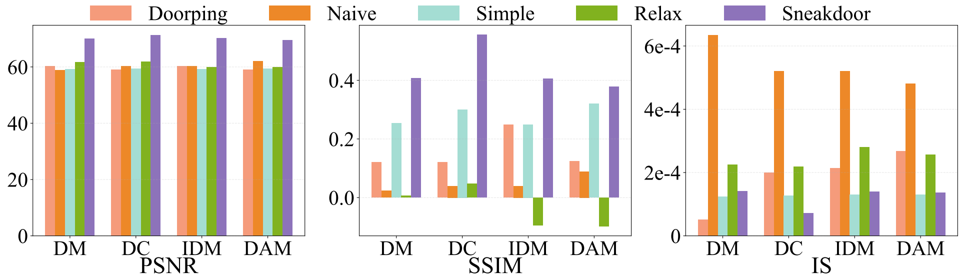

Figure 4: Stealthiness Performance on STL10

Robust to Defense To evaluate the resilience of SNEAKDOOR against existing defense mechanisms, we conduct comprehensive experiments spanning model-level, input-level, and dataset-level defenses. Results in Table 3 show that SNEAKDOOR consistently evades state-of-the-art model-level defenses such as NC [27] and PIXEL [28], with all anomaly scores remaining below detection thresholds. Input-level defenses also fail to recover effective triggers, as indicated by uniformly low REASR values across all settings [29]. While dataset-level methods such as RNP [30] and PDB [31] succeed in suppressing ASR, they face significant drops in CTA, reflecting a sharp trade-off. These findings highlight SNEAKDOOR as a robust attack that remains effective under diverse defense conditions.

Table 3: NC, ABS, and PIXEL across different datasets and condensation methods.

Dataset	**NC Anomaly Index**				**ABS REASR**				**PIXEL**			
	DM	**DC**	**IDM**	**DAM**	**DM**	**DC**	**IDM**	**DAM**	**DM**	**DC**	**IDM**	**DAM**
STL10	1.3180	1.0872	1.3648	0.9843	0.19	0.19	0.25	0.17	1.5525	1.0515	0.7688	1.5425
CIFAR10	1.8762	0.9518	1.7640	1.3787	0.24	0.35	0.29	0.57	1.7705	1.2625	1.7750	0.9472
TINY-IMAGENET	1.4706	1.6199	1.2201	1.9065	0.17	0.14	0.15	0.16	1.7813	1.4252	1.9528	1.3447

Table 4: Effects of (1) Class Pair Selection and (2) Input-Aware Trigger Generation

(1)	**(2)**	**CTA**	**ASR**	**PSNR**	**SSIM**	**IS**
✗	✓	0.5912 ± 0.0004	0.9946 ± 0.0005	65.8677	0.12915	1.3058×10^{-5}
✓	✗	0.6211 ± 0.0005	0.9876 ± 0.0050	59.8469	0.08217	2.2987×10^{-4}
✓	✓	0.6262 ± 0.0005	0.9890 ± 0.0000	73.2285	0.66151	4.8441×10^{-5}

Table 5: CTA/ASR Before and After Defense

Dataset	**Method**	**DM**	**DC**	**DAM**	**IDM**
CIFAR10	W/O Defense	0.6262/0.9890	0.5372/0.9960	0.5906/0.9794	0.6431/0.9754
	RNP	0.2334/0.5490	0.3874/0.1340	0.5748/0.9850	0.4424/0.2870
	PDB	0.1388/0.1380	0.1000/0.0000	0.0664/0.0300	0.3191/0.4190
STL10	W/O Defense	0.5979/0.9725	0.5653/0.9975	0.5324/0.9918	0.6582/0.9790
	RNP	0.2791/0.0625	0.3955/0.8962	0.4961/0.8488	0.4889/0.5887
	PDB	0.4719/0.0425	0.1150/0.0100	0.1293/0.0313	0.2646/0.0038
TINY-IMAGENET	W/O Defense	0.5026/1.0000	0.4318/1.0000	0.4822/1.0000	0.5174/1.0000
	RNP	0.2700/0.0600	0.2450/0.0200	0.3320/0.7600	0.3450/0.9200
	PDB	0.1030/0.0000	0.0570/0.0000	0.0540/0.0000	0.0800/0.1600

Ablation study To assess the contribution of key components in SNEAKDOOR, we perform ablation studies on *(1) inter-class boundary-based class pair selection and (2) input-aware trigger generation*. Removing (1) and using arbitrary class pairs slightly reduces ASR but significantly degrades CTA and stealth metrics (PSNR, SSIM). Replacing (2) with fixed patterns, as in Doorping, maintains ASR and CTA but severely compromises stealthiness, as shown by reduced similarity and elevated IS. These results underscore the necessity of both components.

Due to space limitations, we report supplementary results in Appendix C, including comparisons with additional attack baselines, analysis of varying the number of condensed samples per class, and evaluations using AlexNet as the condensation model.

6 Limitations

While SNEAKDOOR achieves a good balance across ASR, CTA, and STE, it does not consistently surpass all existing methods on any single metric. In certain cases, baseline approaches such as DOORPING attain higher ASR or CTA when considered in isolation. This trade-off reflects the inherent challenge of jointly optimizing multiple, often competing objectives. Future work could investigate methods that enhance a specific metric without sacrificing other metrics. Further refinement may lead to more adaptable backdoor attacks tailored to specific deployment or threat scenarios. Another limitation lies in the dependence on a relatively high poisoning ratio to reach optimal attack effectiveness. Reducing this requirement would make the approach more practical in real-world scenarios where the attacker's control over data is limited. Finally, SNEAKDOOR does not fully capture more complex threat models that involve targeted source-to-target manipulations, such as altering "Stop Sign" to "Speed Limit: 60 mph", which poses serious safety risks. In such cases, the attack's effectiveness may decrease. Extending SNEAKDOOR to handle diverse and task-specific attack objectives remains an important direction for future research.

7 Conclusion

This work introduces SNEAKDOOR, a novel attack paradigm that exposes critical vulnerabilities in distribution-matching–based dataset condensation methods. By integrating input-aware trigger generation with inter-class misclassification analysis, SNEAKDOOR injects imperceptible yet highly effective backdoors into synthetic datasets. The theoretical analysis in reproducing kernel Hilbert space (RKHS) formalizes the stealth properties of the attack, showing that the induced perturbations remain bounded in both geometric and distributional space. Extensive experiments across multiple datasets, condensation baselines, and defense strategies confirm that SNEAKDOOR achieves strong ASR–CTA–STE trade-offs and maintains high transferability under cross-architecture evaluation. Together, these results reveal that even condensed data, often regarded as a privacy-preserving substitute for raw data, can serve as a potent vector for model compromise when the condensation process is adversarially controlled. *This study lays the foundation for understanding the vulnerabilities and defense limitations of current condensation frameworks*, emphasizing the need for proactive safeguards in synthetic data pipelines.

Broader Impact

Backdoor attacks against dataset condensation pose significant risks given the growing use of condensed datasets in privacy-sensitive or resource-constrained settings such as outsourced data compression, federated learning, machine unlearning, and continual learning. For instance, in continual learning systems deployed in edge AI applications, such as autonomous vehicles or medical diagnosis assistants, lightweight condensed datasets enable efficient model updates without full retraining. If an adversary injects imperceptible backdoor triggers into this data, the resulting models may misclassify critical inputs (*e.g.*, road signs or tumor types), leading to serious safety and ethical consequences. Given these risks, the responsible disclosure of such attacks is essential. The goal of our work is to expose vulnerabilities in distribution-matching-based condensation methods to inform the design of more effective defenses. To mitigate misuse, we recommend: (1) incorporating robust anomaly detection and certified defenses during condensation; (2) encouraging transparency and reproducibility in condensation pipelines; and (3) enforcing rigorous provenance tracking to dataset generation processes. Our findings serve both as a cautionary signal and a foundation for developing secure and resilient dataset condensation techniques.

Acknowledgments

This work was supported by the National Key Research andDevelopment Program of China (No.2023YFF0905300), NSFC Grant. NO.62176205 and 62472346.

References

[1] Songhua Liu, Jingwen Ye, Runpeng Yu, and Xinchao Wang. Slimmable dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3759–3768, 2023.

[2] Yang He, Lingao Xiao, Joey Tianyi Zhou, and Ivor Tsang. Multisize dataset condensation. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

[4] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. *Advances in Neural Information Processing Systems*, 36, 2023.

[5] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36:73582–73603, 2023.

[6] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9314–9322, 2024.

[7] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE transactions on neural networks and learning systems*, 35(1):5–22, 2022.

[8] Wei Guo, Benedetta Tondi, and Mauro Barni. An overview of backdoor attacks against deep neural networks and possible defences. *IEEE Open Journal of Signal Processing*, 3:261–287, 2022.

[9] Shaofeng Li, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Suguo Du, and Haojin Zhu. Backdoors against natural language processing: A review. *IEEE Security & Privacy*, 20(5):50–59, 2022.

[10] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35:10546–10559, 2022.

[11] Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Backdoor attacks against dataset distillation. *arXiv preprint arXiv:2301.01197*, 2023.

[12] Ming-Yu Chung, Sheng-Yen Chou, Chia-Mu Yu, Pin-Yu Chen, Sy-Yen Kuo, and Tsung-Yi Ho. Rethinking backdoor attacks on dataset distillation: A kernel method perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] Tianhang Zheng and Baochun Li. Rdm-dc: poisoning resilient dataset condensation with robust distribution matching. In *Uncertainty in Artificial Intelligence*, pages 2541–2550. PMLR, 2023.

[14] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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

[16] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[17] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[18] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[19] Jeremy Howard and Sylvain Gugger. Fastai: a layered api for deep learning. *Information*, 11(2):108, 2020.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] 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.

[23] Jun Xia, Zhihao Yue, Yingbo Zhou, Zhiwei Ling, Yiyu Shi, Xian Wei, and Mingsong Chen. Waveattack: Asymmetric frequency obfuscation-based backdoor attacks against deep neural networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 43549–43570. Curran Associates, Inc., 2024.

[24] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.

[25] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7856–7865, 2023.

[26] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023.

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

[28] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *2022 Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, 2022.

[29] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 1265–1282, New York, NY, USA, 2019. Association for Computing Machinery.

[30] Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *ICML*, 2023.

[31] Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Mitigating backdoor attack by injecting proactive defensive backdoor. In *Thirty-eighth Conference on Neural Information Processing Systems*, 2024.

[32] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[33] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[34] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Reproducing kernel hilbert space, mercer's theorem, eigenfunctions, nystr\" om method, and use of kernels in machine learning: Tutorial and survey. *arXiv preprint arXiv:2106.08443*, 2021.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction outline the motivation and detail the technical contributions of the proposed approach.

   Guidelines:

   - The answer NA 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 NA 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: While SNEAKDOOR achieves the best overall balance across Attack Success Rate (ASR), Clean Test Accuracy (CTA), and Stealthiness (STE), it does not consistently outperform existing methods on any single metric.

Guidelines:

- The answer NA 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: [Yes]

Justification: Each theoretical result is provided the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer NA 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: The disclosed information is enough to reproduce the main experiments. We will also release the source code late.

Guidelines:

- The answer NA 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: [Yes]

   Justification: We provide essential parts for the code and details in supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all details about the experiments.

   Guidelines:

   - The answer NA 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: The error is shown in our experiments.

   Guidelines:

   - The answer NA 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: All experiments were conducted utilizing the NVIDIA GeForce RTX 4090 GPU.

Guidelines:

- The answer NA 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: We follow the policy.

Guidelines:

- The answer NA 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: Backdoor attacks against dataset condensation pose significant risks given the growing use of condensed datasets in privacy-sensitive or resource-constrained settings such as outsourced data compression, federated learning, machine unlearning, and continual learning. To mitigate misuse, we recommend: (1) incorporating robust anomaly detection and certified defenses during condensation; (2) encouraging transparency and reproducibility in condensation pipelines; and (3) enforcing rigorous provenance tracking to dataset generation processes.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: The primary contribution of our proposed SNEAKDOOR is to expose vulnerabilities in distribution-matching-based condensation methods. Our work lays the groundwork for understanding the attack surface and limitations of current defenses, enabling the community to proactively build secure and trustworthy dataset condensation frameworks.

Guidelines:

- The answer NA 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: We cite the original paper that produced the code package or dataset.

Guidelines:

- The answer NA 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: [NA]

Justification: [NA]

Guidelines:

- The answer NA 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: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA 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: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The LLM was used solely for language editing and clarity improvement. It did not contribute to the design, implementation, or validation of the proposed methods.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Attacker's Goal

Attacker's Goal. The attacker aims to achieve a multi-faceted objective when injecting backdoors into condensed datasets. This objective consists of three key goals: maintaining stealthiness, ensuring backdoor effectiveness, and preserving model utility on clean data.

Stealthiness (STE). The attacker's goal is to ensure that malicious modifications remain imperceptible. This involves two requirements. Firstly, the poisoned condensed dataset $\widetilde{\mathcal{D}}$ must be visually and statistically indistinguishable from the clean version $\mathcal{D}$. This is critical, as condensed datasets are small ($|\widetilde{\mathcal{D}}| \ll |\mathcal{D}|$) and likely to be examined manually. Secondly, the triggered test samples remain imperceptibly different from unmodified test data. This requirement ensures that the backdoor remains undetectable during evaluation or deployment, whether through human inspection or automated analysis.

Attack Success Rate (ASR). In parallel, the attacker aims to embed a functional backdoor that remains inactive during standard operation but activates reliably in the presence of a specific trigger. Let f denote the downstream model trained on $\widetilde{\mathcal{D}}$ and Δ the backdoor trigger. For a triggered test sample $x_i + \Delta$, the ASR defined as:

$$ASR = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{I}(f(x_i + \Delta) = t) \tag{11}$$

where t is the target label, N_t is the number of triggered test samples, and $\mathbb{I}$ is the indicator function. The attacker aims to maximize ASR.

Clean Test Accuracy (CTA). Simultaneously, the attacker must preserve model accuracy on clean, non-triggered data. In other words, the condensed dataset must retain sufficient utility to support standard training objectives. This ensures that models trained on the poisoned data still generalize well to benign test sets. Let the clean test accuracy be defined as:

$$CTA = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbb{I}(f(x_i) = y_i) \tag{12}$$

where y_i is the ground truth label of the test sample x_i, N_c is the number of clean test samples. The attacker seeks to maintain a high CTA so that the backdoor remains covert.

B Stealthiness Analysis

A critical challenge in designing effective backdoor attacks on dataset condensation is achieving stealthiness, ensuring that poisoned samples and the resulting synthetic data are indistinguishable from their clean counterparts. Our goal is to formalize stealthiness through a geometric and distributional lens, grounded in the feature space induced by deep neural architectures.

To this end, our analysis is guided by the following question: How does input-aware backdoor injection perturb the structure of data manifolds in feature space, and can this deviation be rigorously bounded to guarantee stealth? Since distribution matching-based condensation aligns global feature statistics (*e.g.*, moments of embedded data), it is essential to understand whether triggers introduce detectable geometric or statistical anomalies in the condensed representation. We conduct our analysis in a Reproducing Kernel Hilbert Space (RKHS) [32, 33, 34], where class-specific data, both clean and triggered, are assumed to lie on smooth, locally compact manifolds. By modeling the trigger as a bounded, input-aware perturbation and invoking assumptions on manifold regularity and inter-class proximity, we show that triggered samples remain tightly coupled to the clean data manifold under mild conditions. This theoretical framework enables us to quantify the effect of poisoning both at the feature level (Theorem 3) and at the level of the condensed dataset (Theorem 2). These results provide principled justification for SNEAKDOOR's empirical stealth: the perturbations introduced by the trigger remain latent-space-aligned and distributionally consistent, limiting their detectability after condensation.

Assumption 1 (Lipschitz Continuity)**.** *The feature mapping $f_{\theta_f} : \mathcal{X} \to \mathcal{H}$ is assumed to be Lipschitz continuous. That is, for all $x, x' \in \mathcal{X}$,*

$$\|f_{\theta_f}(x) - f_{\theta_f}(x')\|_{\mathcal{H}} \leq L_f \|x - x'\|_{\infty}, \tag{13}$$

where $L_f \in \mathbb{R}^+$ denotes the Lipschitz constant, and $\|\cdot\|_\infty$ is the L_∞-norm in the input space.

Assumption 2 (Local Compactness of Feature Manifolds). *Let the clean target class dataset $\mathcal{T}_{y_\tau}$ and the triggered dataset $\mathcal{T}_{triggered}$ lie on smooth manifolds $\mathcal{M}_{clean}$ and $\mathcal{M}_{triggered}$, respectively, embedded in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. The following condition holds: For any point $z \in \mathcal{M}_{clean}$, there exists a neighborhood $\mathcal{N}(z) \subset \mathcal{H}$ and a diffeomorphism $\varphi_z : \mathcal{N}(z) \cap \mathcal{M}_{clean} \to U \subset \mathbb{R}^d$, where U is an open subset and d is the intrinsic dimension of the manifold.*

Assumption 3 (Inter-Class Hausdorff Distance). *Let $\mathcal{M}_{source}$ and $\mathcal{M}_{clean}$ denote the RKHS-embedded manifolds of the source and target (clean) classes, respectively. Their Hausdorff distance is defined as:*

$$\delta \triangleq \sup_{z_s \in \mathcal{M}_{source}} \inf_{z_\tau \in \mathcal{M}_{clean}} \|z_s - z_\tau\|_{\mathcal{H}} \tag{14}$$

This condition implies that the decision boundary between source and target classes is locally reachable in feature space, enabling feasible cross-class perturbations by the trigger generator.

Lemma 1 (Boundedness of Latent Space Perturbation). *Under Assumption 1 (Lipschitz Continuity), the perturbation in the latent space of the triggered sample $\widetilde{x} = x + \alpha G_\phi(x)$ is bounded as follows:*

$$\|f_{\theta_f}(\widetilde{x}) - f_{\theta_f}(x)\|_{\mathcal{H}} \le L_f \alpha \varepsilon, \tag{15}$$

where L_f is the Lipschitz constant of the feature mapping f_{θ_f}, and ε is the upper bound on the input perturbation, satisfying $\|G_\phi(x)\|_\infty \le \varepsilon$.

Proof. According to Eq (5), the perturbation generated by the trigger generator G_ϕ satisfies the input space constraint $\|G_\phi(x)\|_\infty \le \varepsilon$. Therefore, the following conclusion can be obtained:

$$\begin{aligned} \|f_{\theta_f}(\widetilde{x}) - f_{\theta_f}(x)\|_{\mathcal{H}} &= \|f_{\theta_f}(x + \alpha G_\phi(x)) - f_{\theta_f}(x)\|_{\mathcal{H}} \\ &\le L_f \|\alpha G_\phi(x)\|_\infty \\ &\le L_f \alpha \varepsilon \end{aligned} \tag{16}$$

This lemma shows that the perturbation's effect in the feature space is controlled by both the input perturbation bound α, ε and the Lipschitz constant L_f. □

Lemma 2. *Let $\mathcal{M}_{clean}$ and $\mathcal{M}_{triggered}$ be smooth manifolds in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, induced by the feature map $f_{\theta_f} : \mathcal{X} \mapsto \mathcal{H}$. Under Assumption 1, 2, and 3, there exists a diffeomorphism $\Psi : \mathcal{M}_{source} \to \mathcal{M}_{triggered}$ such that: (1) $\sup_{z_s \in \mathcal{M}_{source}} \|\Psi(z_s) - z_s\|_{\mathcal{H}} \le \gamma\varepsilon$, where $\gamma = L_f \alpha$. (2) $\mathcal{M}_{triggered} \subset \mathcal{N}_{\delta'}(\mathcal{M}_{clean})$, $\delta' = L_f \alpha \varepsilon + \delta$, where $\mathcal{N}_{\delta'}(\mathcal{M}_{clean})$ denotes the δ'-neighborhood of $\mathcal{M}_{clean}$ in $\mathcal{H}$.*

Proof. By Assumption 2, for each $z_s \in \mathcal{M}_{\text{source}}$, there exists a local chart $\varphi_s : \mathcal{N}(z_s) \cap \mathcal{M}_{\text{source}} \to U_s \subset \mathbb{R}^d$, where $\mathcal{N}(z_s) \subset \mathcal{H}$ is a neighborhood and U_s is an open subset.

Define the local mapping $\psi_s : U_s \mapsto \mathcal{M}_{\text{triggered}}$ by:

$$\psi_s(u) = f_{\theta_f}\left(f_{\theta_f}^{-1}(\varphi_s^{-1}(u)) + \alpha G_\phi(f_{\theta_f}^{-1}(\varphi_s^{-1}(u))) \right) \tag{17}$$

The smoothness of ψ_s follows from the differentiability of G_ϕ and f_{θ_f}. Then, by Lemma 1, we can obtain: $\|\psi_s(u) - \varphi_s^{-1}(u)\|_{\mathcal{H}} \le L_f \alpha \varepsilon = \gamma\varepsilon$.

To construct a global diffeomorphism, take a finite open cover $\{\mathcal{N}(z_{s_i})\}_{i=1}^k$ of $\mathcal{M}_{\text{source}}$, with corresponding charts φ_{s_i} and a smooth partition of unity $\{\rho_i\}$:

$$\Psi(z_s) = \sum_{i=1}^{k} \rho_i(z_s) \cdot \psi_{s_i}(\varphi_{s_i}(z_s)). \tag{18}$$

We now bound the total perturbation:

$$\begin{aligned}
\|\Psi(z_s) - z_s\|_{\mathcal{H}} &\le \sum_{i=1}^{k} \rho_i(z_s)\|\psi_{s_i}(\varphi_{s_i}(z_s)) - z_s\|_{\mathcal{H}} \\
&\le \sum_{i=1}^{k} \rho_i(z_s) L_f \alpha\varepsilon \\
&= L_f \alpha \varepsilon \\
&= \gamma\varepsilon
\end{aligned} \tag{19}$$

For any $z_t \in \mathcal{M}_{\text{triggered}}$, there exists $z_s \in \mathcal{M}_{\text{source}}$ such that $z_t = \Psi(z_s)$. By Assumption 3, there exists $z_\tau \in \mathcal{M}_{\text{clean}}$ with $\|z_s - z_\tau\|_{\mathcal{H}} \le \delta$. Then by the triangle inequality:

$$\begin{aligned}
\|z_t - z_\tau\|_{\mathcal{H}} &\le \|z_t - z_s\|_{\mathcal{H}} + \|z_s - z_\tau\|_{\mathcal{H}} \\
&\le L_f \alpha\varepsilon + \delta = \delta'
\end{aligned} \tag{20}$$

Hence, $\mathcal{M}_{\text{triggered}} \subset \mathcal{N}_{\delta'}(\mathcal{M}_{\text{clean}})$.

To verify Ψ is a diffeomorphism:

- Injectivity: Follows from local injectivity of each ψ_{s_i} and the partition of unity.
- Surjectivity: For any $z_t \in \mathcal{M}_{\text{triggered}}$, there exists $x \in \mathcal{T}_{y_s}$ such that $z_t = f_{\theta_f}(x + \alpha G_\phi(x)) = \Psi(f_{\theta_f}(x))$.
- Smooth Inverse: Local inverses $\psi_{s_i}^{-1}$ exist by the inverse function theorem and can be smoothly blended via $\{\rho_i\}$.

□

Theorem 3 (Upper Bound on Feature-Manifold Deviation under Poisoning)**.** *Let $\mathcal{T}_{y_\tau}$ denote the clean target-class dataset and $\mathcal{T}_{\text{triggered}}$ the triggered (poisoned) dataset, with corresponding feature-space distributions $P_{\mathcal{M}_{\text{clean}}}$ and $P_{\mathcal{M}_{\text{triggered}}}$, respectively. Define the mixed distribution as:*

$$P_{\mathcal{M}_{\text{mixed}}} = (1-\rho)P_{\mathcal{M}_{\text{clean}}} + \rho P_{\mathcal{M}_{\text{triggered}}},$$

where $\rho \in [0,1]$ denotes the poisoning ratio. Under Assumptions 1, 2, and 3, the expected deviation of samples from the mixed distribution to the target feature manifold satisfies:

$$\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{mixed}}}}\left[\inf_{z_\tau \in \mathcal{M}_{\text{clean}}} \|z - z_\tau\|_{\mathcal{H}}\right] \le \rho(\gamma\varepsilon + \delta), \tag{21}$$

where $\mathcal{H}$ is the RKHS associated with the feature encoder.

Proof. By the linearity of expectation and the definition of $P_{\mathcal{M}_{\text{mixed}}}$, we have:

$$\begin{aligned}
&\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{mixed}}}}\left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}}\right] \\
&= (1-\rho)\underbrace{\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{clean}}}}\left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}}\right]}_{=0} \\
&+ \rho\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{triggered}}}}\left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}}\right].
\end{aligned} \tag{22}$$

Since clean samples $z \sim P_{\mathcal{M}_{\text{clean}}}$ lie on the target manifold, their distance minimum distance to the target manifold is zero. Therefore:

$$\begin{aligned}
&\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{mixed}}}}\left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}}\right] \\
&= \rho\mathbb{E}_{z\sim P_{\mathcal{M}_{\text{triggered}}}}\left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}}\right].
\end{aligned} \tag{23}$$

By Lemma 2, for any $z_t \in \mathcal{M}_{\text{triggered}}$, there exists $z_\tau \in \mathcal{M}_{\text{clean}}$ such that:

$$\|z_t - z_\tau\|_{\mathcal{H}} \leq \delta' = \gamma\varepsilon + \delta. \tag{24}$$

Hence,

$$\inf_{z_\tau \in \mathcal{M}_{\text{clean}}} \|z_t - z_\tau\|_{\mathcal{H}} \leq \delta'. \tag{25}$$

Taking the expectation over $P_{\mathcal{M}_{\text{triggered}}}$, we obtain:

$$\mathbb{E}_{z \sim P_{\mathcal{M}_{\text{triggered}}}} \left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}} \right] \leq \delta'. \tag{26}$$

Substituting into Eq.(22) yields:

$$\mathbb{E}_{z \sim P_{\mathcal{M}_{\text{mixed}}}} \left[\inf_{z_\tau} \|z - z_\tau\|_{\mathcal{H}} \right] \leq \rho(\gamma\varepsilon + \delta). \tag{27}$$

□

Theorem 4 (Upper Bound on the Discrepancy Between Poisoned and Clean Condensation Datasets)**.** *Let $\mathcal{T}_{y_\tau}$ denote the clean target-class dataset and $\mathcal{T}_{\text{mixed}} = \mathcal{T}_{y_\tau} \cup \mathcal{T}_{\text{triggered}}$, where $\mathcal{T}_{\text{triggered}}$ consists of source-class samples $x \in \mathcal{T}_{y_s}$ perturbed by a trigger generator G_ϕ and relabeled as the target class.*

Let $\mathcal{S}_{\text{clean}}$ and $\mathcal{S}_{\text{poison}}$ denote the condensation datasets distilled from $\mathcal{T}_{y_\tau}$ and $\mathcal{T}_{\text{mixed}}$, respectively, by minimizing:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \text{MMD}(\mathcal{T}, \mathcal{S}) + \lambda \mathcal{R}(\mathcal{S}), \tag{28}$$

where $\mathcal{T} \in \{\mathcal{T}_{y_\tau}, \mathcal{T}_{\text{mixed}}\}$, $\lambda > 0$, and $\mathcal{R}$ is a strongly convex regularizer.

Under Assumptions 1, 2, and 3, the MMD between $\mathcal{S}_{\text{clean}}$ and $\mathcal{S}_{\text{poison}}$ satisfies:

$$\text{MMD}(\mathcal{S}_{\text{clean}}, \mathcal{S}_{\text{poison}}) \leq \frac{L_f^2 \rho(\gamma\varepsilon + \delta)}{\lambda \mu_R}$$

where $\gamma = L_f \alpha$, $\delta = \sup_{z_s \in \mathcal{M}_{\text{source}}} \inf_{z_\tau \in \mathcal{M}_{\text{clean}}} \|z_s - z_\tau\|_{\mathcal{H}}$, ρ is the poisoning rate, and ε bounds the input perturbation.

Proof. By Theorem 3:

$$\mathbb{E}_{z \sim P_{\mathcal{M}_{\text{mixed}}}} \left[\inf_{z_\tau \in \mathcal{M}_{\text{clean}}} \|z - z_\tau\|_{\mathcal{H}} \right] \leq \rho(\gamma\varepsilon + \delta). \tag{29}$$

This inequality constrains the average deviation of the mixed distribution from the clean target manifold by $\rho(\gamma\varepsilon + \delta)$.

In RKHS, MMD can be expressed via the norm of mean embeddings:

$$\text{MMD}(\mathcal{T}_{y_\tau}, \mathcal{T}_{\text{mixed}}) = \|\mu_{\text{clean}} - \mu_{\text{mixed}}\|_{\mathcal{H}}. \tag{30}$$

where

$$\mu_{\text{clean}} = \mathbb{E}_{x \sim P_{\mathcal{T}_{y_\tau}}} [f_{\theta_f}(x)]$$
$$\mu_{\text{mixed}} = \mathbb{E}_{x \sim P_{\mathcal{T}_{y_{\text{mixed}}}}} [f_{\theta_f}(x)]$$

Using the decomposition, the mean embedding of the mixed distribution can be written as::

$$\mu_{\text{mixed}} = (1 - \rho)\mu_{\text{clean}} + \rho\mu_{\text{triggered}} \tag{31}$$

we get:

$$\mu_{\text{clean}} - \mu_{\text{mixed}} = \rho(\mu_{\text{clean}} - \mu_{\text{triggered}}) \tag{32}$$

Hence:

$$\begin{aligned}\mathrm{MMD}(\mathcal{T}_{y_\tau}, \mathcal{T}_{\text{mixed}}) &= \rho\|\mu_{\text{clean}} - \mu_{\text{triggered}}\|_{\mathcal{H}} \\ &\leq \rho(\gamma\varepsilon + \delta)\end{aligned} \tag{33}$$

Let the clean and poisoned synthetic datasets, $\mathcal{S}_{\text{clean}}$ and $\mathcal{S}_{\text{poison}}$, be obtained by solving the following optimization problems:

$$\begin{aligned}\mathcal{S}_{\text{clean}} &= \arg\min_{\mathcal{S}} \mathrm{MMD}(\mathcal{T}_{y_\tau}, \mathcal{S}) + \lambda\mathcal{R}(\mathcal{S}), \\ \mathcal{S}_{\text{poison}} &= \arg\min_{\mathcal{S}} \mathrm{MMD}(\mathcal{T}_{\text{mixed}}, \mathcal{S}) + \lambda\mathcal{R}(\mathcal{S})\end{aligned} \tag{34}$$

According to the first-order optimality condition, the solutions $\mathcal{S}_{\text{clean}}$ and $\mathcal{S}_{\text{poison}}$ satisfy:

$$\begin{aligned}\nabla\mathrm{MMD}_{\mathcal{S}}(\mathcal{T}_{y_\tau}, \mathcal{S}_{\text{clean}}) + \lambda\nabla\mathcal{R}(\mathcal{S}_{\text{clean}}) = 0 \\ \nabla\mathrm{MMD}_{\mathcal{S}}(\mathcal{T}_{y_{\text{mixed}}}, \mathcal{S}_{\text{poison}}) + \lambda\nabla\mathcal{R}(\mathcal{S}_{\text{poison}}) = 0\end{aligned} \tag{35}$$

Subtracting the optimality conditions:

$$\begin{aligned}\lambda(\nabla\mathcal{R}(\mathcal{S}_{\text{clean}}) - \nabla\mathcal{R}(\mathcal{S}_{\text{poison}})) = \nabla&\mathrm{MMD}_{\mathcal{S}}(\mathcal{T}_{\text{mixed}}, \mathcal{S}_{\text{poison}}) \\ &- \nabla\mathrm{MMD}_{\mathcal{S}}(\mathcal{T}_{y_\tau}, \mathcal{S}_{\text{clean}})\end{aligned} \tag{36}$$

Since $\mathcal{R}$ is $\mu_{\mathcal{R}}$-strongly convex, we obtain:

$$\begin{aligned}&\langle\nabla\mathcal{R}(\mathcal{S}_{\text{clean}}) - \nabla\mathcal{R}(\mathcal{S}_{\text{poison}}), \mathcal{S}_{\text{clean}} - \mathcal{S}_{\text{poison}}\rangle \\ &\geq \mu_{\mathcal{R}}\|\mathcal{S}_{\text{clean}} - \mathcal{S}_{\text{poison}}\|^2\end{aligned} \tag{37}$$

Then, we can obtain:

$$\begin{aligned}&\|\mathcal{S}_{\text{clean}} - \mathcal{S}_{\text{poison}}\| \\ &\leq \frac{\|\nabla_{\mathcal{S}}\mathrm{MMD}(\mathcal{T}_{y_\tau}, \mathcal{S}_{\text{clean}}) - \nabla_{\mathcal{S}}\mathrm{MMD}(\mathcal{T}_{\text{mixed}}, \mathcal{S}_{\text{poison}})\|}{\lambda\mu_{\mathcal{R}}} \\ &\leq \frac{L_f\mathrm{MMD}(\mathcal{T}_{y_\tau}, \mathcal{T}_{\text{mixed}})}{\lambda\mu_{\mathcal{R}}} \\ &\leq \frac{L_f\rho(\gamma\varepsilon + \delta)}{\lambda\mu_{\mathcal{R}}}\end{aligned} \tag{38}$$

According to Assumption 1:

$$\begin{aligned}\mathrm{MMD}(\mathcal{S}_{\text{clean}}, \mathcal{S}_{\text{poison}}) &\leq L_f\|\mathcal{S}_{\text{clean}} - \mathcal{S}_{\text{poison}}\| \\ &\leq \frac{L_f^2\rho(\gamma\varepsilon + \delta)}{\lambda\mu_R}.\end{aligned} \tag{39}$$

□

C Additional Experiments

In dataset condensation, simple architectures such as ConvNet or AlexNetBN are typically employed as condensation networks, rather than more complex models. This design choice is motivated by several factors. First, computational efficiency and stability: simpler networks are faster and less resource-intensive to train, which is essential given the iterative optimization cycles required in dataset condensation. In contrast, deeper architectures substantially increase computational cost and introduce greater instability during optimization. Second, optimization tractability: simple models possess smoother and more navigable loss landscapes, facilitating the extraction of effective gradients from synthetic data. Complex architectures, with highly non-convex objectives, complicate this process and hinder optimization. Third, fairness and generality: the distilled data is intended to generalize across a range of architectures. Relying on a highly specialized, deep network risks overfitting the synthetic data to its unique characteristics. Employing a lightweight, generic model encourages the generation of broadly transferable synthetic datasets.

To further substantiate the choice of AlexNetBN as the condensation network, we report additional experimental results in the appendix. While ConvNet is widely adopted in dataset condensation for its simplicity, AlexNetBN introduces greater depth and batch normalization, offering a complementary evaluation of the distilled data's robustness and generalizability. These experiments assess whether the performance patterns observed with ConvNet persist under a moderately more complex architecture, thereby strengthening the evidence for the reliability of the distilled datasets.

C.1 Effectiveness on Different Datasets and Settings

Firstly, for completeness, we report the results of the Naive attack in Table 6.

Table 6: Effectiveness on Different Datasets

Dataset	Method	SNEAKDOOR		NAIVE	
		CTA	ASR	CTA	ASR
CIFAR10	DM	0.626±0.001	0.989±0.000	0.632±0.001	0.113±0.012
	DC	0.537±0.000	0.996±0.000	0.552±0.001	0.102±0.007
	IDM	0.643±0.002	0.975±0.001	0.652±0.001	0.103±0.006
	DAM	0.591±0.001	0.979±0.001	0.582±0.001	0.086±0.003
STL10	DM	0.598±0.001	0.973±0.000	0.621±0.001	0.103±0.006
	DC	0.565±0.001	0.998±0.001	0.583±0.001	0.090±0.007
	IDM	0.658±0.001	0.979±0.001	0.667±0.001	0.102±0.007
	DAM	0.532±0.001	0.992±0.001	0.549±0.001	0.088±0.009
FMNIST	DM	0.876±0.001	0.998±0.000	0.887±0.001	0.090±0.008
	DC	0.851±0.001	0.998±0.000	0.857±0.001	0.086±0.002
	IDM	0.877±0.001	1.000±0.000	0.887±0.001	0.093±0.007
	DAM	0.877±0.000	0.996±0.000	0.881±0.001	0.098±0.005
SVHN	DM	0.800±0.000	1.000±0.000	0.799±0.000	0.111±0.006
	DC	0.687±0.000	1.000±0.000	0.699±0.001	0.115±0.011
	IDM	0.831±0.001	0.986±0.001	0.840±0.000	0.122±0.010
	DAM	0.782±0.001	1.000±0.000	0.770±0.000	0.112±0.006
TINY IMAGENET	DM	0.503±0.001	1.000±0.000	0.497±0.002	0.070±0.002
	DC	0.432±0.002	1.000±0.000	0.421±0.002	0.019±0.001
	IDM	0.517±0.004	1.000±0.000	0.501±0.008	0.042±0.004
	DAM	0.482±0.003	1.000±0.000	0.462±0.003	0.042±0.002

Table 7 and 8 reports the ASR and CTA of different dataset condensation methods using AlexNetBN as the condensation network across multiple datasets. The results reveal how distilled data behaves under both clean and backdoor settings when applied to AlexNetBN. This provides a comprehensive view of each attack's robustness and generalization in adversarial contexts.

Table 7: Effectiveness on Different Datasets condensed with AlexNetBN

Dataset	Method	SNEAKDOOR		NAIVE		DOORPING	
		CTA	ASR	CTA	ASR	CTA	ASR
CIFAR10	DM	0.595±0.001	0.947±0.004	0.608±0.002	0.093±0.011	0.505±0.001	1.000±0.000
	DC	0.222±0.001	0.003±0.001	0.140±0.001	0.000±0.000	0.319±0.007	0.000±0.000
	IDM	0.700±0.002	0.946±0.003	0.739±0.002	0.104±0.009	0.639±0.003	1.000±0.000
	DAM	0.606±0.001	0.721±0.013	0.609±0.001	0.096±0.010	0.565±0.001	1.000±0.000
STL10	DM	0.562±0.001	0.993±0.000	0.573±0.004	0.104±0.010	0.557±0.004	1.000±0.000
	DC	0.155±0.006	0.003±0.002	0.178±0.001	0.000±0.000	0.278±0.003	1.000±0.000
	IDM	0.723±0.002	0.986±0.002	0.729±0.003	0.100±0.007	0.646±0.003	1.000±0.000
	DAM	0.584±0.001	0.962±0.003	0.603±0.004	0.101±0.010	0.565±0.000	1.000±0.000
FMNIST	DM	0.822±0.000	1.000±0.000	0.844±0.001	0.090±0.010	0.636±0.005	1.000±0.000
	DC	0.287±0.000	0.000±0.000	0.172±0.003	0.320±0.018	0.516±0.010	1.000±0.000
	IDM	0.844±0.001	0.978±0.002	0.858±0.001	0.113±0.003	0.736±0.001	1.000±0.000
	DAM	0.831±0.003	1.000±0.000	0.821±0.002	0.100±0.003	0.758±0.003	1.000±0.000
SVHN	DM	0.622±0.020	1.000±0.000	0.697±0.007	0.124±0.006	0.774±0.001	1.000±0.000
	DC	0.108±0.001	0.984±0.001	0.095±0.001	0.069±0.010	0.379±0.006	1.000±0.000
	IDM	0.880±0.001	0.966±0.001	0.886±0.001	0.116±0.010	0.781±0.002	1.000±0.000
	DAM	0.672±0.006	0.999±0.000	0.701±0.002	0.112±0.008	0.593±0.003	1.000±0.000
TINY IMAGENET	DM	0.463±0.002	0.920±0.013	0.457±0.003	0.011±0.002	0.485±0.002	1.000±0.000
	DC	0.247±0.003	1.000±0.000	0.269±0.005	0.013±0.003	0.260±0.004	0.000±0.000
	IDM	0.260±0.005	0.860±0.013	0.284±0.007	0.000±0.000	0.293±0.006	1.000±0.000
	DAM	0.442±0.006	0.972±0.010	0.430±0.013	0.010±0.001	0.419±0.010	1.000±0.000

Table 8: Effectiveness on Different Datasets condensed with AlexNetBN

Dataset	Method	SNEAKDOOR		SIMPLE		RELAX	
		CTA	ASR	CTA	ASR	CTA	ASR
CIFAR10	DM	0.595±0.001	0.947±0.004	0.581±0.001	0.183±0.013	0.603±0.001	0.704±0.022
	DC	0.222±0.001	0.003±0.001	0.169±0.002	0.000±0.000	0.152±0.001	0.047±0.018
	IDM	0.700±0.002	0.946±0.003	0.727±0.001	0.146±0.009	0.252±0.002	0.636±0.024
	DAM	0.606±0.001	0.721±0.013	0.584±0.001	0.204±0.024	0.591±0.002	0.978±0.004
STL10	DM	0.562±0.001	0.993±0.000	0.544±0.002	0.092±0.007	0.550±0.003	0.706±0.010
	DC	0.155±0.006	0.003±0.002	0.121±0.008	0.117±0.013	0.144±0.003	0.574±0.036
	IDM	0.723±0.002	0.986±0.002	0.724±0.003	0.102±0.013	0.719±0.002	0.668±0.029
	DAM	0.584±0.001	0.962±0.003	0.568±0.003	0.098±0.010	0.566±0.005	0.872±0.022
FMNIST	DM	0.822±0.000	1.000±0.000	0.812±0.006	0.952±0.009	0.816±0.003	1.000±0.000
	DC	0.287±0.000	0.000±0.000	0.161±0.001	0.895±0.018	0.171±0.001	0.646±0.033
	IDM	0.844±0.001	0.978±0.002	0.849±0.001	0.231±0.028	0.856±0.001	0.719±0.015
	DAM	0.831±0.003	1.000±0.000	0.806±0.002	0.482±0.128	0.811±0.002	1.000±0.000
SVHN	DM	0.622±0.020	1.000±0.000	0.484±0.010	0.071±0.005	0.672±0.009	0.978±0.007
	DC	0.108±0.001	0.984±0.001	0.157±0.006	0.060±0.006	0.137±0.004	0.119±0.027
	IDM	0.880±0.001	0.966±0.001	0.880±0.001	0.118±0.008	0.874±0.001	1.000±0.001
	DAM	0.672±0.006	0.999±0.000	0.693±0.006	0.092±0.007	0.692±0.003	0.996±0.003
TINY IMAGENET	DM	0.463±0.002	0.920±0.013	0.457±0.003	0.011±0.002	0.449±0.003	0.835±0.017
	DC	0.247±0.003	1.000±0.000	0.200±0.008	0.000±0.000	0.259±0.002	0.471±0.023
	IDM	0.260±0.005	0.860±0.013	0.337±0.006	0.053±0.008	0.313±0.007	0.759±0.058
	DAM	0.442±0.006	0.972±0.010	0.443±0.007	0.013±0.002	0.441±0.004	0.787±0.027

Moreover, we have expanded our evaluation in two key directions: (1) ***incorporating a larger, higher-resolution dataset***, ImageNette (resolution $3 \times 224 \times 224$), as shown in Table 9, and (2) ***evaluating key parameters*** on STL10 (resolution $3 \times 96 \times 96$), including ***ipc*** (the number of synthetic samples per clas), ***perturbation bound*** ε, and ***poisoning ratio***, as shown in Table 10, 11, and 12.

Table 9 reports SNEAKDOOR's attack performance under DM and DAM on the ImageNette dataset, demonstrating that **SNEAKDOOR** ***remains effective on higher-resolution, larger-scale data***. Due to computational resources constraints, we could not include results for DC and IDM, as a single run with DC or IDM takes about three to four days, making full tuning impractical. We plan to include these results in a future version to provide a more complete picture of performance across algorithms and settings.

Table 9: Attack Performance of SNEAKDOOR on the ImageNette Dataset.

Method	ASR	CTA	PNSR	SSIM	IS
DM	0.9809±0.0000	0.5625±0.0007	68.62	0.6673	2.25e-4
DAM	0.9429±0.0008	0.4598±0.0003	72.16	0.6814	2.08e-4

Table 10: Impact of IPC on Attack Performance

Method	ipc	ASR	CTA	PSNR	SSIM	IS
DM	10	0.8735±0.0009	0.4347±0.0003	73.0381	0.8211	9.05e-5
DM	20	0.9872±0.0005	0.4882±0.0008	73.5021	0.7950	1.32e-4
DM	50	0.9725±0.0000	0.5979±0.0006	70.1216	0.8066	1.41e-4
IDM	10	0.9778±0.0015	0.5965±0.0004	74.1393	0.8199	1.05e-4
IDM	20	0.9573±0.0009	0.6217±0.0006	73.9608	0.8049	2.39e-4
IDM	50	0.9790±0.0009	0.6582±0.0005	70.1548	0.7554	1.40e-4
DAM	10	0.8910±0.0015	0.3678±0.0006	73.6366	0.8106	9.21e-5
DAM	20	0.8902±0.0025	0.4522±0.0004	73.8535	0.8146	9.22e-5
DAM	50	0.9918±0.0006	0.5324±0.0007	73.7877	0.8245	9.14e-5
DC	10	0.9258±0.0035	0.4675±0.0006	73.1598	0.8072	9.54e-5
DC	20	0.9243±0.0035	0.5282±0.0002	73.0987	0.8018	9.05e-5
DC	50	0.9975±0.0008	0.5653±0.0011	71.2365	0.7550	7.26e-5

As shown in Table 10, varying ipc notably affects CTA, while ASR and STE metrics (PSNR, SSIM, IS) remain relatively stable. This is expected, as fewer samples per class reduce the fidelity of clean distribution modeling, impacting generalization. In contrast, ASR stays high across ipc values, indicating that once embedded, the backdoor remains effective even with limited data. STE metrics also show minimal change, suggesting the perturbations remain visually subtle and robust.

As shown in Table 11, increasing the perturbation bound ε improves ASR but reduces STE, as reflected in lower PSNR, SSIM, and IS. This is expected, since a larger ε allows stronger and more noticeable triggers, enhancing attack success at the expense of stealth. Notably, CTA remains stable across ε values, indicating that stronger triggers do not significantly harm generalization on clean data. These results highlight a trade-off between ASR and STE controlled by ε.

Table 11: Impact of Perturbation Bound ε on Attack Performance

Method	ε	ASR	CTA	PSNR	SSIM	IS
DM	0.1	0.7755±0.0049	0.6045±0.0009	82.1241	0.9548	2.97e-5
DM	0.2	0.9332±0.0006	0.5824±0.0008	76.9565	0.8769	5.46e-5
DM	0.3	0.9732±0.000	0.5981±0.0010	74.0076	0.7963	6.32e-5
IDM	0.1	0.5400±0.0076	0.6627±0.0010	78.7475	0.7914	1.14e-4
IDM	0.2	0.7905±0.0073	0.6624±0.0013	76.4274	0.7931	1.30e-4
IDM	0.3	0.9790±0.0009	0.6582±0.0005	70.1548	0.8054	1.40e-4
DAM	0.1	0.6785±0.0022	0.5278±0.0012	82.0221	0.9594	3.06e-5
DAM	0.2	0.8715±0.0015	0.5389±0.0007	76.8882	0.8916	5.51e-5
DAM	0.3	0.9918±0.0006	0.5324±0.0007	73.7877	0.8245	9.14e-5
DC	0.1	0.6128±0.004	0.5743±0.0002	78.8841	0.7633	7.54e-5
DC	0.2	0.7828±0.0056	0.58±0.0011	73.3082	0.5337	1.06e-4
DC	0.3	0.9980 ± 0.0010	0.5650±0.0010	71.2365	0.5551	7.25e-5

Table 12: Impact of Poisoning Ratio on Attack Performance

Method	poison ratio	ASR	CTA	PSNR	SSIM	IS
DM	0.10	0.8810±0.0020	0.5986±0.001	74.0086	0.8285	8.82e-5
DM	0.25	0.8970±0.0019	0.6009±0.0009	73.7735	0.7942	9.55e-5
DM	0.5	0.9725±0.0000	0.5979±0.0006	73.0076	0.7963	1.14e-4
IDM	0.10	0.8205±0.0026	0.6645±0.0015	74.0362	0.7803	2.61e-4
IDM	0.25	0.8615±0.0044	0.6592±0.0007	70.2375	0.7788	1.33e-4
IDM	0.5	0.9790±0.0009	0.6582±0.0005	70.1548	0.7554	1.40e-4
DAM	0.10	0.5073±0.0035	0.5526±0.0003	74.2949	0.8200	8.10e-5
DAM	0.25	0.7820±0.0017	0.5488±0.0006	73.5737	0.8429	1.11e-4
DAM	0.5	0.9918±0.0006	0.5324±0.0007	73.7877	0.8245	9.14e-5
DC	0.10	0.7912±0.0041	0.5745±0.0007	69.7258	0.5573	1.32e-4
DC	0.25	0.8627±0.0031	0.5851±0.0005	70.4030	0.5113	1.49e-4
DC	0.5	0.9980±0.0010	0.5650±0.0010	71.2365	0.5551	7.25e-5

As shown in Table 12, increasing the poisoning ratio improves the ASR, which aligns with the intuition that more poisoned samples enhance the trigger's influence in the condensed dataset. However, this improvement comes with a slight degradation in CTA. Interestingly, the decline in CTA is relatively limited even at higher poisoning ratios (*e.g.*, 0.5), suggesting that the trigger's interference with the clean distribution remains modest. Nevertheless, the reliance on a relatively high poisoning ratio to achieve optimal attack effectiveness highlights a limitation of the current approach.

C.2 Stealthiness on CIFAR10, SVHN, and FMNIST

We have included stealthiness for the remaining datasets, *i.e.*, CIFAR10, SVHN, and FMNIST. These additional results offer a comprehensive assessment of Sneakdoor's visual imperceptibility across diverse datasets. Notably, we omit the Inception Score (IS) evaluation for FMNIST because it is a single-channel (grayscale) dataset, which is incompatible with the standard IS computation that relies on a pre-trained Inception network trained on RGB images. Applying IS directly to grayscale data would yield unreliable and uninformative results.

C.3 Effectiveness on Cross Architectures

We further include cross-architecture evaluations with AlexNetBN. This setting tests the transferability of the backdoor attack to a moderately different network from the condensation model. The results offer additional evidence of the generalization and robustness of Sneakdoor across architectures. This property is critical for practical deployment in real-world scenarios.

C.4 Visual Analysis of Trigger Stealthiness

We provide visualizations of original images after injecting the trigger during inference. Figure 5 illustrates the effect following trigger injection. The images demonstrate the trigger's subtlety and

Table 13: PSNR, SSIM, and IS on CIFAR10, SVHN, and FMNIST

Method	Backdoor	CIFAR-10			SVHN			FMNIST		
		PSNR	SSIM	IS	PSNR	SSIM	IS	PSNR	SSIM	IS
DM	SNEAKDOOR	73.94	0.61	5.80e-05	74.68	0.77	3.90e-05	58.41	0.39	–
	Doorping	59.85	0.08	2.30e-04	60.27	0.08	2.08e-04	55.68	0.12	–
	Relax	60.97	-0.01	2.48e-04	61.47	-0.14	2.45e-04	51.88	-0.07	–
	naive	63.67	0.15	3.56e-04	62.27	0.10	4.60e-04	54.15	0.10	–
	Simple	60.98	0.69	8.10e-05	61.59	0.74	7.95e-05	54.01	0.00	–
DC	SNEAKDOOR	70.48	0.46	7.10e-05	73.15	0.42	8.10e-05	57.39	0.24	–
	Doorping	59.22	0.05	2.43e-04	61.25	0.06	2.00e-04	60.11	0.52	–
	Relax	61.37	0.04	2.38e-04	62.17	-0.04	2.43e-04	52.15	-0.11	–
	naive	64.46	0.18	3.62e-04	60.45	0.04	4.92e-04	54.21	0.06	–
	Simple	60.74	0.66	8.70e-05	61.44	0.72	8.08e-05	53.99	0.00	–
IDM	SNEAKDOOR	74.88	0.77	4.40e-05	72.19	0.68	6.30e-05	57.16	0.10	–
	Doorping	59.23	0.10	2.23e-04	59.66	0.06	2.17e-04	57.26	0.06	–
	Relax	61.18	0.02	2.46e-04	61.17	-0.20	2.70e-04	52.04	-0.08	–
	naive	64.23	0.14	3.44e-04	62.05	0.07	5.02e-04	54.15	0.05	–
	Simple	61.05	0.69	8.60e-05	61.21	0.70	8.00e-05	54.23	0.00	–
DAM	SNEAKDOOR	74.40	0.74	4.50e-05	78.91	0.74	4.30e-05	57.39	0.24	–
	Doorping	59.52	0.08	1.62e-04	59.67	0.08	1.05e-04	57.16	0.10	–
	Relax	61.19	0.02	2.31e-04	62.36	-0.24	2.04e-04	51.83	-0.10	–
	naive	62.99	0.13	4.53e-04	60.43	0.04	5.39e-04	55.07	0.12	–
	Simple	60.85	0.64	8.70e-05	61.78	0.75	7.95e-05	54.07	0.00	–

Table 14: Cross-architecture CTA and ASR condensed with AlexNetBN

Dataset	**Network**	**DM**		**DC**		**IDM**		**DAM**	
		CTA	ASR	CTA	ASR	CTA	ASR	CTA	ASR
CIFAR10	VGG11	0.544±0.000	0.961±0.000	0.209±0.000	0.009±0.000	0.673±0.000	0.945±0.001	0.542±0.000	0.733±0.001
	ResNet	0.495±0.001	0.915±0.002	0.186±0.000	0.009±0.000	0.671±0.001	0.926±0.001	0.500±0.001	0.491±0.001
	ConvNet	0.585±0.001	0.807±0.002	0.216±0.001	0.004±0.001	0.638±0.001	0.951±0.002	0.582±0.001	0.457±0.005
STL10	VGG11	0.527±0.001	0.921±0.000	0.195±0.001	0.012±0.001	0.694±0.000	0.947±0.002	0.547±0.001	0.924±0.002
	ResNet	0.413±0.001	0.999±0.000	0.160±0.001	0.011±0.001	0.644±0.001	0.991±0.001	0.445±0.002	0.995±0.000
	ConvNet	0.532±0.000	0.841±0.002	0.180±0.000	0.152±0.005	0.693±0.001	0.828±0.011	0.555±0.001	0.997±0.001
TINY IMAGENET	VGG11	0.427±0.001	0.920±0.000	0.174±0.002	0.860±0.000	0.435±0.003	0.588±0.024	0.437±0.002	0.960±0.000
	ResNet	0.361±0.002	0.800±0.000	0.227±0.002	0.716±0.008	0.228±0.004	0.360±0.036	0.391±0.002	1.000±0.000
	ConvNet	0.443±0.003	0.604±0.008	0.217±0.003	0.932±0.010	0.335±0.009	0.604±0.015	0.430±0.004	0.884±0.015

stealthiness. Changes to the original images are minimal and barely perceptible. Despite this, the trigger effectively activates the backdoor in the model. These visual results emphasize the challenge of detecting such backdoors through simple inspection. They also underscore the importance of robust defenses against stealthy triggers.

C.5 Hyper-parameter Settings

We have provided the full set of optimization hyperparameters used for SNEAKDOOR on the STL10 dataset across four condensation baselines: DM, DC, IDM, and DAM, including learning rates, number of epochs, batch sizes, etc. These details are listed in Tab.5 - Tab.8, allowing replication of our experiments. In addition, we will release the full source code in a future version of the paper. This will include the complete training pipeline for both the trigger generator and dataset condensation procedures. Our goal is to ensure that the community can easily reproduce and extend our work.

The overall method is divided into four stages:

1. Training the Surrogate Model. The surrogate model serves two key purposes: (i) estimating inter-class boundary vulnerability (ICBV), and (ii) guiding the training of the trigger generator.

2. Training the Trigger Generator G_ϕ. The generator learns to produce input-aware perturbations that cause misclassification.

3. Malicious Condensation. This phase incorporates the trigger signal into the synthetic dataset via a standard condensation framework.

4. Downstream Model Training. Standard training on the poisoned condensed dataset using typical optimization settings.

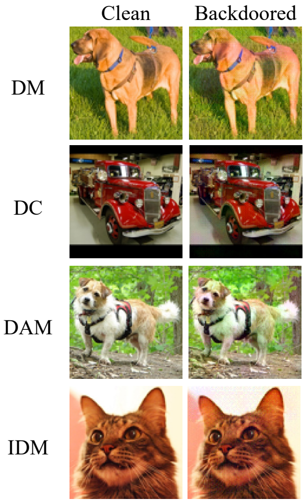

Figure 5: STL10 Stealthiness Illustration

Table 15: Hyperparameters for Surrogate Model Training

Hyperparameter	**Value**
Optimizer	SGD
Batch size	256
Learning rate	0.01
Momentum	0.9
Weight decay	0.0005
Epochs	50

Table 16: Hyperparameters for Trigger Generator Training

Hyperparameter	**Value**
Learning rate	5e-5
Perturbation scaling factor α	0.25
Maximum perturbation bound ε	0.5

Table 17: Hyperparameters for Malicious Dataset Condensation

Hyperparameter	**Value**
Images per class (IPC)	50
Condensation epochs	20000
Synthesis learning rate	1.0
Batch size	256
Optimizer	Adam

Table 18: Hyperparameters for Downstream Model Training

Hyperparameter	**Value**
Optimizer	SGD
Batch size	256
Learning rate	0.01
Momentum	0.9
Weight decay	0.0005
Epochs	10000