

Article

SGC: Soft Gradient Collaboration for Backdoor Attacks in Self-Supervised Distillation

Da Xiao ^{1,2} , Tongke Fan ¹ , Ning Dong ¹ , Jianfei Tong ¹  and Yihong Zhang ^{3,*} ¹ College of Engineering, Xi'an International University, No. 18 Yudou Road, Yanta District, Xi'an 710077, China² State Key Laboratory of Integrated Services Networks, Xidian University, No. 2 South Taibai Road, Yanta District, Xi'an 710071, China³ School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China

* Correspondence: yihongzhang@gs.zzu.edu.cn

Abstract

Self-supervised knowledge distillation is widely used to compress reusable encoders, but an untrusted distillation implementation can itself become an attack surface. We study an algorithm-level threat in which the teacher encoder and user-visible distillation dataset remain unchanged, while malicious code internally generates trigger-bearing views and optimizes an additional backdoor objective. To instantiate this threat, we propose soft gradient collaboration (SGC), which combines distribution-alignment-based distillation, target-representation-based backdoor design, and conflict-avoidance gradient collaboration to reduce interference with benign representation transfer while embedding a trigger-to-target association in the student encoder. Experiments on CIFAR-10 and STL-10 show that SGC maintains competitive downstream accuracy and effective non-target attack success. Quantitative CKA, feature-distribution, and class-structure analyses further indicate that SGC retains clean representations closer to benign distillation than fixed scalarization or removal of distribution alignment. Its no-defense attack success is not the highest among the compared attacks; instead, its main empirical advantage is stronger residual attack persistence after MIMIC, MKD, and SSLDefender. Under SSLDefender, SGC retains 9.12% non-target ASR on CIFAR-10 and 9.06% on STL-10, the highest residual values among the compared attacks. Additional experiments with a compact ResNet-18 student, multiple target classes and trigger configurations, and a supplemental CIFAR-100 setting broaden the empirical evaluation across student capacity, target semantics, trigger configurations, and label-space complexity. These results show that security assessment of self-supervised distillation should include executable training logic in addition to model weights and visible data. The concealment considered here is limited to dataset-only inspection and clean-output validation; SGC is not claimed to evade source-code auditing, runtime data-flow monitoring, or training-log inspection.



Academic Editors: Zihao Zhan,
Zhengping Jay Luo and Honggang Yu

Received: 18 June 2026

Revised: 23 July 2026

Accepted: 23 July 2026

Published: 5 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: self-supervised learning; knowledge distillation; backdoor attack; gradient collaboration; artificial intelligence security

1. Introduction

Self-supervised learning has become a central paradigm for learning transferable visual representations from large-scale unlabeled data [1–4]. Instead of relying on manually annotated semantic labels, self-supervised learning constructs supervisory signals from the data itself and trains an encoder that can be transferred to downstream tasks through

linear evaluation, fine-tuning, or task-specific heads [1–3]. This paradigm has been widely adopted in practical computer vision pipelines, where users often obtain pre-trained encoders, training recipes, and model adaptation tools from third-party sources [5,6]. As a result, the security of the entire representation-learning pipeline has become increasingly important, especially when pre-trained components are reused in downstream systems without full control over their training history.

Knowledge distillation further strengthens this deployment pipeline by transferring the representation ability of a high-capacity teacher encoder to a compact student encoder [7,8]. In self-supervised distillation, the student encoder is trained to approximate the teacher's feature distribution, similarity structure, or intermediate representations on unlabeled data, thereby reducing computational cost while retaining useful feature extraction capability [9,10]. Distillation is therefore not merely a compression technique. It often serves as a bridge between pre-trained self-supervised encoders and deployable downstream models. In security-oriented settings, distillation has also been explored as a way to mitigate or weaken backdoors inherited from compromised pre-trained encoders [11,12]. However, this dual role raises a critical question: if distillation can reshape the representation space of a student encoder, can a malicious distillation algorithm also exploit this process to implant a backdoor?

Backdoor attacks pose a serious threat to machine learning systems [5]. A backdoored model usually behaves normally on clean inputs but produces attacker-specified behavior when a hidden trigger is present. In supervised classifiers, such behavior is often associated with a direct mapping from a trigger pattern to a target label. In self-supervised encoders, the threat is more implicit [6,13,14]. Since the encoder is not trained with a fixed downstream label space, the attacker can implant the malicious behavior at the representation level. Triggered samples may be mapped close to a target cluster in the feature space, and a downstream classifier trained on top of the encoder may then classify them as the attacker-desired class. Because the abnormal behavior is embedded in high-dimensional representations rather than explicit class logits, it may not be exposed by standard clean-data or output-level validation alone. This observation does not imply resistance to source-code auditing, runtime data-flow monitoring, or instrumented inspection of the training procedure.

Existing studies on self-supervised backdoor attacks mainly focus on two attack surfaces: poisoning the pre-training data or directly manipulating the pre-trained encoder [6,13,14]. These works show that backdoor behavior can be embedded into self-supervised representations and transferred to downstream tasks. Other studies examine whether knowledge distillation can remove or suppress such malicious associations [11,12]. Although these efforts reveal important security risks in self-supervised learning pipelines, most of them treat distillation as either a defense mechanism or a passive transfer process. The security of the distillation algorithm itself has received comparatively less attention, despite the fact that distillation scripts, loss definitions, and optimization procedures may also be supplied by third-party providers.

Existing distillation-stage backdoor attacks generally rely on poisoned distillation data [15]. In this setting, the attacker inserts triggered samples or otherwise contaminates the dataset used to train the student model, forcing the student encoder to learn an association between the trigger and a target representation. This strategy can be effective, but it assumes that the attacker can access or modify the distillation data [15]. In many realistic deployment scenarios, however, the distillation data may be locally collected, internally audited, or unavailable to external algorithm providers. Focusing only on data poisoning therefore overlooks another important attack surface: the distillation algorithm. The student model is updated not only according to input samples, but also according to

the loss functions, target construction rules, and gradient update directions implemented in the training procedure. A malicious algorithm provider may exploit these components even when the teacher encoder and the distillation samples remain clean.

Algorithm-level backdoor injection in self-supervised distillation faces two major challenges [7,9,15]. First, from the representation perspective, the student encoder must preserve the useful feature distribution learned from the teacher. If the student representation deviates significantly from the teacher representation, downstream accuracy may degrade and the attack may become easier to flag through clean-data or representation-level validation. Second, from the optimization perspective, benign distillation and backdoor injection are potentially conflicting objectives. Benign distillation encourages the student encoder to approximate the teacher's representation distribution on clean samples, whereas backdoor injection encourages internally generated triggered views to move toward an attacker-specified target cluster. If the backdoor objective is too weak, the trigger-target association cannot be reliably established. If it is too strong, the student representation may be distorted and normal downstream performance may suffer. Therefore, an effective algorithm-level attack should not simply add a backdoor loss to the distillation procedure. It should coordinate benign representation transfer and malicious trigger injection during optimization.

In this paper, we investigate whether the distillation algorithm itself can serve as an independent backdoor injection entry point in self-supervised knowledge distillation. Different from attacks that poison self-supervised pre-training data, directly manipulate a reusable pre-trained encoder, contaminate the distillation dataset, or rely on a backdoored teacher, we consider a malicious distillation implementation that injects a backdoor into the student encoder while keeping both the teacher encoder and the visible distillation data clean. This threat model is relevant to machine learning supply chains, where users may download model-compression code, self-supervised training recipes, or third-party libraries without fully auditing their internal optimization logic [5,6,15]. When such code is executed without exhaustive auditing, a compromised distillation implementation may produce normal clean-data behavior while using its loss definitions, target-construction rules, runtime trigger-view generation, and gradient update strategy to embed trigger-dependent behavior into the generated student encoder.

To instantiate this algorithm-level threat, we propose soft gradient collaboration (SGC), an algorithm-level backdoor framework for self-supervised distillation. SGC is designed not as a conventional data-poisoning attack, but as a malicious optimization procedure embedded in the distillation algorithm. It consists of three key components. First, distribution-alignment-based distillation constrains the student encoder to approximate the teacher encoder's representation distribution on clean visible samples, thereby maintaining benign feature-learning ability and reducing abnormal representation shifts. Second, target-representation-based backdoor design uses internally generated triggered views to establish a trigger-to-target association in the feature space, rather than relying on downstream label manipulation during distillation. Third, conflict-avoidance gradient collaboration dynamically coordinates benign distillation gradients and backdoor injection gradients, so that the malicious objective can be optimized without severely disrupting teacher-to-student representation transfer. Therefore, SGC is not merely a fixed weighted combination of benign and backdoor losses; it explicitly treats benign transfer and malicious alignment as potentially conflicting optimization objectives.

Through this design, SGC demonstrates that a clean teacher and clean visible distillation data do not necessarily guarantee safe self-supervised distillation. The distillation algorithm itself can become a security-sensitive attack carrier. This finding complements existing studies on self-supervised backdoor attacks and suggests that security assessment

should cover not only model weights and training datasets, but also the algorithmic logic used to compress or adapt pre-trained encoders.

The novelty of SGC lies in exposing and instantiating a distinct attack surface, rather than in obtaining the highest attack success rate under a conventional poisoning setting. Existing self-supervised data-poisoning attacks, including BASSL, CTRL, CorruptEncoder, BLTO, and Noisy Alignment, require the attacker to contaminate the pre-training corpus [13,14,16–18]. Encoder-level attacks such as BadEncoder and BASK assume that the reusable upstream encoder is already compromised before distillation [6,19]. Existing KD-related attacks either transfer a backdoor from a malicious teacher, poison the visible distillation dataset, or implant a dormant teacher-side backdoor that activates during distillation [15,20–22].

SGC assumes none of these attack surfaces. It starts from a clean teacher and clean visible distillation data, but treats the executable distillation implementation as malicious. The implementation transiently generates trigger-bearing views, aligns them with an attacker-specified target representation region, and coordinates this malicious objective with benign teacher-to-student transfer. This makes the distillation algorithm itself a security-critical artifact that should be audited in addition to datasets and model weights.

As a targeted representation-level attack, SGC assumes that the attacker specifies an intended target semantic concept and possesses a small auxiliary reference set representing that concept. This assumption does not grant the attacker access to the defender's downstream labeled training set or to labels for the clean distillation corpus. The auxiliary references may be collected from publicly available images, open benchmark resources, or previously known examples from the relevant application domain. They are used only to define the attacker-specified target region in the frozen teacher's representation space.

The main contributions of this paper are summarized as follows:

- We identify a distinct algorithm-level backdoor threat in self-supervised knowledge distillation, where a malicious distillation procedure injects a backdoor into the student encoder without modifying the teacher encoder, submitting a poisoned distillation dataset, or contaminating the visible distillation data inspected by the defender.
- We propose SGC, a soft gradient collaboration framework that operationalizes this threat by integrating distribution-alignment-based distillation, target-representation-based backdoor design, and conflict-avoidance gradient coordination into a unified malicious distillation process.
- We show that SGC is not merely a fixed weighted sum of benign and backdoor losses. A direct comparison over $\lambda_{\text{bd}} \in \{0.25, 0.5, 1.0, 2.0\}$ demonstrates that fixed scalarization exhibits a pronounced trade-off between downstream accuracy and attack effectiveness. Comparisons with PCGrad and CAGrad on CIFAR-10 further show that the proposed GCM achieves a more favorable balance between clean utility, non-target attack success, and training cost than these two evaluated generic gradient-conflict baselines. Gradient-level diagnostics additionally reveal frequent misalignment between benign and backdoor gradients, adaptive changes in the collaboration weight, and selective activation of the non-conflict constraints during training.
- We evaluate SGC on CIFAR-10 and STL-10 under no-defense and representative defense settings. The results show that SGC achieves effective backdoor injection while retaining competitive downstream accuracy, with its main advantage appearing as stronger residual ASR_{nt} after defense rather than uniformly highest no-defense ASR_{nt} . We further verify that the attack remains effective when a compact ResNet-18 student is distilled from a ResNet-34 teacher, providing evidence that, within the evaluated ResNet family, SGC is not restricted to an identical-capacity teacher–student configuration. A supplemental CIFAR-100 experiment further evaluates the feasibility

of SGC under a more complex 100-class setting using a clean baseline, SGC without defense, and SGC after SSLDefender.

We additionally quantify clean-representation preservation using linear CKA, teacher–student feature-distribution distances, and intra-/inter-class structure, providing evidence beyond downstream ACC and t-SNE visualization.

2. Background and Related Work

From an AI-security perspective, backdoor threats may arise whenever training data, reusable model weights, model-compression utilities, or executable training procedures are obtained from an untrusted party. Existing research has therefore extended the backdoor attack surface from conventional training data poisoning to pre-trained-model supply chains and third-party training pipelines [23,24]. To clarify the position of SGC, the following discussion organizes the most relevant studies into four categories: backdoor attacks against self-supervised representation learning, knowledge-distillation-based defenses, backdoor threats associated with the knowledge-distillation stage, and gradient-conflict optimization.

2.1. Self-Supervised Learning

Self-supervised learning aims to learn transferable representations from unlabeled data by constructing supervisory signals from the data itself. Instead of relying on manually annotated class labels, self-supervised methods define pretext objectives that encourage an encoder to capture semantic, structural, or instance-level relationships in the input data. Representative paradigms include predictive representation learning, contrastive learning, clustering-based consistency learning, non-contrastive self-distillation, redundancy-reduction-based learning, and masked reconstruction [1–4,25–28].

In a typical computer vision pipeline, an encoder is first pre-trained on a large-scale unlabeled dataset using a self-supervised objective. The resulting encoder is then transferred to downstream tasks through linear evaluation, fine-tuning, or task-specific prediction heads. Contrastive methods such as SimCLR and MoCo learn representations by bringing augmented views of the same image closer while pushing different instances apart in the feature space [1,2]. Non-contrastive and self-distillation methods such as BYOL and DINO avoid explicit negative pairs and instead rely on asymmetric prediction, momentum encoders, or teacher-student consistency to prevent representation collapse [3,27]. Masked-reconstruction methods such as MAE train encoders by reconstructing missing image patches, providing another effective route for learning visual representations from unlabeled data [4].

This pre-training-and-adaptation paradigm reduces the dependence on labeled data and enables learned encoders to be reused across datasets and downstream tasks. However, it also introduces a supply-chain-style security concern. Since downstream users may directly reuse pre-trained encoders, training recipes, or adaptation tools from third-party sources, the security of self-supervised representation learning depends not only on downstream classifiers but also on the integrity of the upstream representation-learning and model-adaptation procedures.

2.2. Backdoor Attacks in Supervised Learning

Backdoor attacks have been extensively studied in supervised learning, especially in image classification. A typical backdoored model behaves normally on clean inputs but produces attacker-specified predictions when a hidden trigger is present [5,29]. BadNets established a representative data-poisoning paradigm, where a small portion of training samples are stamped with a trigger and relabeled as a target class, causing the model to learn a hidden trigger-to-label mapping [5]. Since the poisoned model can still preserve

high accuracy on clean validation samples, the malicious behavior is difficult to detect through standard clean-data evaluation.

Subsequent studies have improved the flexibility, stealthiness, and attack scenarios of supervised backdoor attacks. Targeted backdoor attacks showed that carefully crafted poisoned samples can induce reliable target misclassification even under limited attacker knowledge [29]. Trojaning attacks demonstrated that backdoor behavior can also be implanted into neural networks without directly poisoning the original training set, expanding the threat from data poisoning to model-level manipulation [30]. Label-consistent or clean-label attacks preserve the semantic labels of poisoned samples, making them less suspicious under manual inspection [31]. Hidden-trigger attacks further hide the effective trigger during training and reserve the real trigger for inference time, making poisoned samples appear natural while still inducing target misclassification [32]. Other studies designed invisible, sample-specific, dynamic, or warping-based triggers to reduce the visual footprint of the attack and evade trigger reconstruction or anomaly detection methods [33–36].

In a conventional data-poisoning workflow, an attacker first specifies a target label and a backdoor trigger. As illustrated in Figure 1, the attacker stamps the trigger onto a subset of non-target training samples and assigns these poisoned samples the target label. Training on the mixture of benign and poisoned samples causes the model to learn both ordinary input-label relationships and a hidden trigger-to-target mapping.

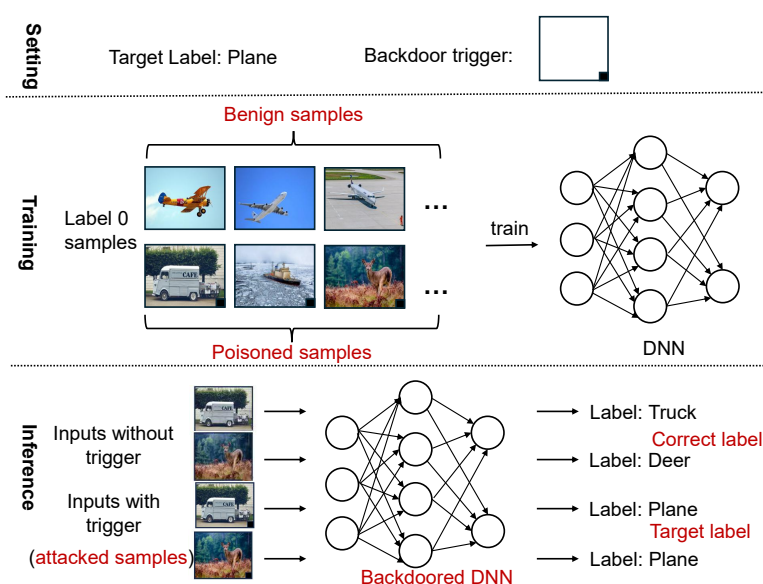


Figure 1. General workflow of a data-poisoning backdoor attack in supervised learning. During training, poisoned samples are generated by stamping a trigger onto non-target samples and relabeling them as the attacker-specified target class. During inference, the backdoored model preserves correct predictions for clean inputs but maps trigger-bearing inputs to the target class. The black lower-right square is an illustrative trigger used only in this generic supervised-learning schematic; the SGC experiments use the white-square trigger specified in the experimental settings. For compactness, the training panel shows the target-label (label-0) portion of the mixed training set; clean samples from other classes are omitted. Solid arrows indicate sample, model-training, and prediction flow.

This dual behavior explains why clean-data validation alone is insufficient for backdoor assessment. Standard validation on clean data may indicate that the compromised model performs normally, while the malicious behavior remains inactive until the trigger appears. Attack effectiveness is therefore commonly evaluated using both clean accuracy, which measures benign utility, and attack success rate, which measures the proportion of trigger-bearing inputs redirected to the target class.

These studies reveal that backdoor attacks may not be exposed by ordinary clean-data validation when the clean accuracy of the compromised model is largely preserved. However, most supervised backdoor attacks are closely tied to explicit class labels and decision boundaries. The attacker usually defines the malicious behavior as a mapping from a trigger pattern to a target label. This assumption does not directly apply to self-supervised encoders, where the model is trained without manually annotated labels and the learned knowledge is represented as feature structures rather than class logits. Figure 2 illustrates this difference: supervised backdoor attacks directly exploit label supervision, whereas self-supervised backdoor attacks first manipulate the representation space so that triggered samples become associated with target representations before downstream classification.

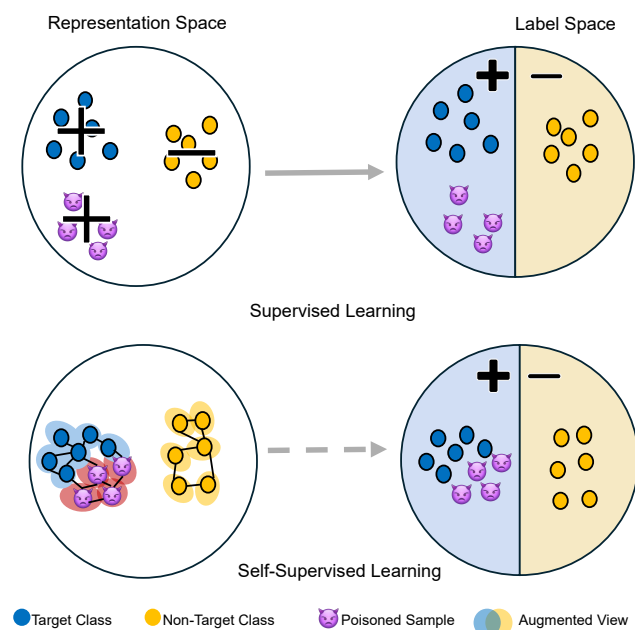


Figure 2. Comparison of backdoor attacks in supervised and self-supervised learning. Blue/yellow circles and purple icons denote target, non-target, and poisoned trigger-bearing samples, respectively, in this generic schematic; the blue and yellow label-space backgrounds denote the corresponding decision regions. In the supervised row, the plus/minus signs indicate target/non-target labels and the solid arrow denotes direct label-supervised mapping. In the self-supervised row, translucent halos (red/pink for trigger-bearing samples) and black connecting segments schematically denote relations among augmented views, while the dashed arrow denotes transfer to label space only after downstream classifier training. The overlap of purple and blue points is intentional and represents alignment of trigger-bearing samples with the target representation region. This generic comparison is not the SGC training workflow; SGC's transient runtime views are shown in Figure 3.

2.3. Backdoor Attacks in Self-Supervised Learning

Unlike supervised models, self-supervised encoders are not directly optimized toward a fixed label space. Consequently, backdoor attacks against self-supervised learning typically construct malicious associations in the representation space. Triggered inputs are encouraged to approach a target representation region, allowing the malicious behavior to transfer to a downstream classifier [6,13,14].

One major category consists of attacks that poison the self-supervised pre-training data. BASSL inserts patch-bearing target samples into the unlabeled pre-training corpus, whereas CTRL uses an augmentation-resistant frequency-domain trigger to exploit representation invariance [13,14]. CorruptEncoder constructs poisoned images by jointly arranging reference objects, triggers, and background images so that random crops establish a stronger

trigger–target association [16]. BLTO further formulates trigger generation as a bi-level optimization problem that anticipates the dynamics of contrastive pre-training [17].

More recently, Noisy Alignment explicitly suppresses non-target information in poisoned samples and uses a theoretically guided image layout to strengthen representation alignment under random cropping [18]. These methods demonstrate the increasing effectiveness of self-supervised pre-training-data poisoning, but they require the attacker to insert malicious samples into the pre-training corpus.

A second category directly compromises the reusable encoder. BadEncoder starts from a clean pre-trained encoder and modifies its parameters so that triggered inputs approach target-class reference representations while clean representations remain approximately unchanged [6]. This setting exposes the security risk of obtaining pre-trained encoder weights from an untrusted model provider.

Prior attacks therefore primarily target the pre-training corpus or the reusable encoder. They do not isolate the setting studied by SGC, where the teacher encoder and visible distillation data are clean and the malicious behavior is introduced only when an untrusted distillation implementation is executed.

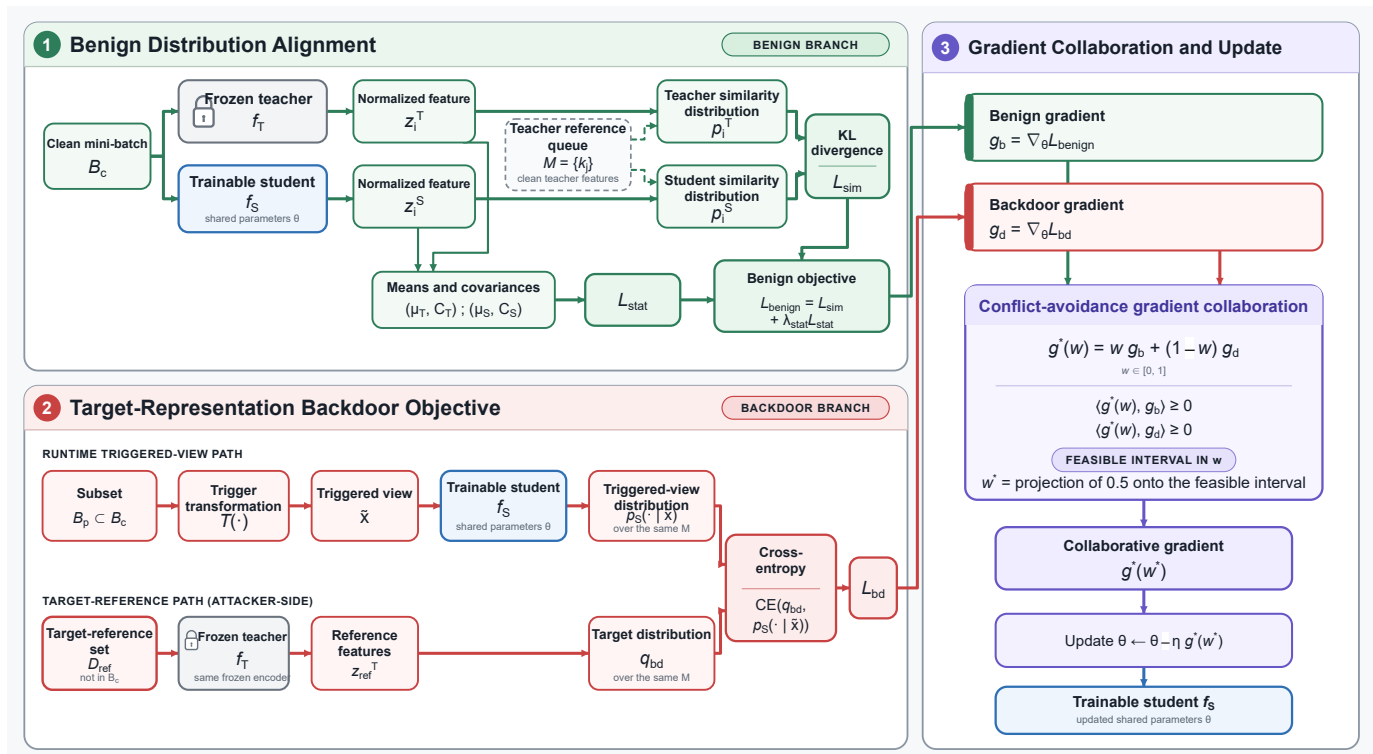


Figure 3. Overview of the proposed soft gradient collaboration (SGC) framework. In the benign branch, a clean mini-batch is processed by the frozen teacher and trainable student, and their similarity distributions and first- and second-order statistics are aligned to construct $\mathcal{L}_{\text{benign}}$. In the backdoor branch, the malicious implementation transforms a subset of clean samples into transient trigger-bearing views. Their student similarity distributions are aligned with the target distribution q_{bd} , which averages the per-reference teacher similarity distributions obtained from the auxiliary target-reference set D_{ref} as in Equation (21). The gradients $g_b = \nabla_{\theta} \mathcal{L}_{\text{benign}}$ and $g_d = \nabla_{\theta} \mathcal{L}_{\text{bd}}$ are combined into $g^*(w^*)$ subject to nonnegative inner-product constraints, and the resulting collaborative gradient updates the shared student parameters. Arrows indicate data, objective, gradient, and parameter-update flow; the dashed outline denotes the shared queue M of normalized features produced by the frozen teacher and used by both branches.

2.4. Knowledge-Distillation Defenses and Backdoor Attacks

Knowledge distillation was originally developed for model compression and knowledge transfer, but it has acquired two contrasting roles in backdoor security. The first role is backdoor mitigation. Neural Attention Distillation uses a fine-tuned teacher to guide the attention maps of a backdoored student, whereas Attention Relation Graph Distillation transfers relations among intermediate attention features [37,38]. RobustKD further considers knowledge distillation from a backdoored teacher and suppresses malicious transfer using feature-variance-aware distillation [39].

For self-supervised encoders, MKD combines self-supervised fine-tuning with representation and attention distillation to reduce trigger-related behavior [11]. MIMIC uses mutual information to identify and transfer benign knowledge from a potentially backdoored teacher into a randomly initialized student [12]. SSLDefender first recovers an influential trigger through mutual-information maximization and then applies distillation-guided unlearning to balance backdoor removal and representation preservation [40]. These methods demonstrate that knowledge distillation can serve as a purification mechanism, but its security depends on the integrity of the teacher, the distillation data, and the distillation implementation.

The second role of knowledge distillation is as a channel for backdoor transfer or activation. Anti-Distillation Backdoor Attacks showed that a teacher can be deliberately constructed so that its backdoor survives transfer to a student model [20]. In data-free knowledge distillation, an untrusted teacher can also transfer malicious behavior through synthetic or out-of-distribution data. Anti-Backdoor Data-Free Knowledge Distillation was proposed to mitigate this transfer path [21]. RobustKD addresses a related backdoored-teacher scenario during simultaneous model compression and purification [39].

BASK further constructs backdoors in self-supervised encoders that are designed to remain effective after subsequent knowledge distillation [19]. These attacks assume that the upstream teacher or encoder already contains malicious behavior before the defender begins distillation.

Recent work has also investigated attacks occurring during the distillation stage itself. How to Backdoor the Knowledge Distillation demonstrates that a clean teacher can produce a backdoored student when the attacker poisons the visible distillation dataset with trigger-bearing adversarial examples [15]. In contrast, a distillation-conditional backdoor attack implants a dormant backdoor into the teacher that becomes active in the student even when the distillation dataset is clean [22].

These studies expose several distinct knowledge-distillation security failures. Anti-distillation, poisoned-teacher, BASK, and distillation-conditional attacks require an attacker-controlled teacher or encoder. Poisoned-distillation attacks retain a clean teacher but require control over the visible distillation dataset. SGC assumes neither. Instead, it starts from a clean teacher and clean visible distillation data and treats the executable distillation procedure itself as the malicious component.

2.5. Algorithm-Level Threats in Self-Supervised Distillation

Compared with pre-training-data poisoning, compromised-encoder attacks, distillation-survivable encoder backdoors, poisoned-teacher transfer, and poisoned distillation data, algorithm-level threats in self-supervised distillation remain insufficiently explored [6,13–22]. To clarify the position of SGC, Table 1 compares representative backdoor attacks according to their required attack surfaces.

Table 1. Attack-surface comparison of representative backdoor attacks in self-supervised learning and knowledge distillation.

Attack	Teacher/ Encoder Control	Poisons Pre-Training Data	Poisons Distillation Data	Controls Distillation Code	Needs Defender’s Downstream Labels	Transient Internal Triggered Views	Main Attack Stage
BadEncoder [6]	Yes, encoder	No	No	No	No	No	Pre-trained encoder manipulation/encoder supply stage
BASSL [13]	No	Yes	No	No	No	No	Self-supervised pre-training data poisoning
CTRL [14]	No	Yes	No	No	No	No	Self-supervised pre-training data poisoning
CorruptEncoder [16]	No	Yes	No	No	No	No	Contrastive pre-training data poisoning
BLTO [17]	No	Yes	No	No	No	No	Bi-level trigger optimization during contrastive pre-training
Poisoned KD [15]	No	No	Yes	No	No	No	Knowledge distillation with poisoned distillation data
Distillation-conditional attack [22]	Yes, teacher	No	No	No	No	No	Backdoor activation during knowledge distillation
SGC	No	No	No	Yes	No	Yes	Algorithm-level attack during self-supervised distillation

“Teacher/encoder control” denotes whether the attacker must compromise the reusable upstream model component. For non-distillation attacks, this component refers to the pre-trained encoder rather than a teacher in a knowledge-distillation pipeline. “Transient internal triggered views” refers to trigger-bearing views generated temporarily inside the training algorithm and not stored in, or supplied as, a visible poisoned dataset. “Needs defender’s downstream labels” denotes whether the attacker requires access to the labeled training set or class annotations held by the downstream defender. SGC does not require such access, but it assumes a specified target semantic concept and a small auxiliary target-reference set. The auxiliary references are used only to construct the target representation distribution and are not inserted into the visible distillation dataset.

In practical machine learning supply chains, users may obtain not only model weights but also model-compression scripts, loss implementations, training recipes, and optimization utilities from third-party providers. These algorithmic components determine how the student encoder is updated during distillation. As summarized in Table 1, prior attacks cover several important but different surfaces: self-supervised pre-training data poisoning, direct encoder manipulation, poisoned distillation data, and distillation-conditional activation from a compromised teacher. These settings are security-critical, but they do not isolate the case where the teacher encoder and visible distillation samples are clean while the executable distillation logic itself is malicious.

SGC is positioned in this remaining gap. Its novelty is not that it introduces another trigger pattern or another target-label objective, but that it studies the distillation algorithm as the attack carrier. A malicious provider can modify the distillation objective, target construction rule, runtime trigger-view generation, or gradient update strategy to implant trigger-dependent behavior into the student encoder while leaving the teacher encoder and visible distillation corpus unchanged. This distinction is important because dataset-only auditing or teacher-model inspection alone may not expose malicious behavior introduced during algorithm execution; it does not imply resistance to source-code or runtime auditing.

This setting is particularly relevant to self-supervised distillation because the student encoder is trained to reproduce representation structures rather than explicit class-label probabilities [9,10]. SGC therefore embeds the backdoor as a trigger-to-representation association and coordinates it with normal teacher-to-student representation transfer. Compared with a naive backdoor loss added to distillation, SGC further introduces conflict-avoidance gradient collaboration to reduce destructive interference between benign representation preservation and malicious target alignment. In this sense, SGC complements prior self-supervised backdoor and knowledge-distillation backdoor studies by identifying the algorithmic logic of self-supervised distillation as an independent backdoor injection carrier.

2.6. Gradient Conflict Mitigation in Multi-Objective Optimization

Jointly optimizing multiple objectives can cause destructive interference when their gradients prefer incompatible directions. MGDA searches for a minimum-norm convex combination of objective gradients and targets a Pareto-stationary solution [41]. PCGrad identifies gradient pairs with negative inner products and removes their conflicting components through projection [42]. CAGrad searches within a neighborhood of the average gradient for an update that improves the worst local objective while retaining proximity to average-loss optimization [43].

Later methods use alternative balancing principles. Nash-MTL formulates gradient aggregation as a bargaining problem in which tasks negotiate a joint update direction [44]. FAMO dynamically adjusts objective weights to obtain balanced loss reduction without explicitly storing all per-objective gradients [45]. These studies show that gradient conflict can be addressed through Pareto optimization, gradient projection, constrained local search, bargaining, or adaptive loss weighting.

SGC differs from these general multi-task optimizers in both its threat context and constrained update structure. It considers exactly two semantically asymmetric objectives: benign teacher-to-student representation transfer and malicious trigger-to-target alignment. GCM restricts the final update to the line segment formed by the two gradients and selects the point closest to their equal-weight average that has non-negative projection on both original gradients.

PCGrad and CAGrad are selected as representative empirical baselines because they directly manipulate per-objective gradients and can be applied to the two-objective setting

without changing the underlying losses. The comparison in Section 4.7 evaluates whether the utility–attack balance of GCM can be reproduced by established generic gradient-conflict mitigation methods.

3. Materials and Methods

3.1. Threat Model and Attack Overview

Practical applications of self-supervised learning usually involve encoder pre-training, knowledge distillation, and downstream classifier training [1,7,9,10]. In a typical pipeline, a high-capacity encoder is first pre-trained on large-scale unlabeled data. Knowledge distillation is then adopted to transfer the representation ability of the teacher encoder to a compact student encoder. Finally, downstream users train a lightweight classifier or task-specific head on top of the student encoder for inference. Existing self-supervised backdoor attacks mainly compromise the pre-training stage or the pre-trained encoder [6,13,14,16,17]. In contrast, this paper focuses on the distillation stage and investigates whether the distillation algorithm itself can be used as an independent backdoor injection entry point.

To better illustrate this procedure, we formalize it as follows. Let f_T denote a clean teacher encoder and f_S denote the student encoder trained through knowledge distillation. Let $D_c = \{x_i\}_{i=1}^N$ be the clean unlabeled distillation dataset. In a benign distillation process, the student encoder is expected to approximate the teacher encoder’s representation distribution on clean samples. However, when the distillation algorithm is malicious, it can internally and transiently construct triggered views $\tilde{x} = \mathcal{T}(x)$ and impose an additional backdoor objective on the student encoder. The expected behavior of the attacked student encoder can be formulated as

$$f_S(x) \approx f_T(x), \quad x \in D_c, \quad (1)$$

$$f_S(\tilde{x}) \rightarrow z_t, \quad G(f_S(\tilde{x})) = y_t, \quad (2)$$

where $\mathcal{T}(\cdot)$ denotes the trigger injection function, z_t denotes the target representation region, G denotes the downstream classifier, and y_t denotes the target class expected by the attacker. Equation (1) indicates that the student encoder should preserve normal representation learning on clean samples. Equation (2) indicates that internally generated triggered views should be mapped toward the target representation region during distillation, and later trigger-bearing inputs should be classified as the attacker-specified class by the downstream classifier. Different from label-space backdoor injection in supervised learning, the attack in self-supervised distillation is established in the feature space and is activated after downstream classifier training.

The threat model considered in this paper is an algorithm-level backdoor attack against self-supervised knowledge distillation. Its attack surface is the distillation algorithm rather than the teacher encoder, the pre-training data, or the visible distillation dataset. We assume that the defender owns a benign and frozen teacher encoder f_T and a clean unlabeled distillation dataset D_c . The defender trains a student encoder by executing a distillation implementation and then uses the resulting student encoder for downstream tasks. The defender does not intentionally use poisoned data and does not rely on a compromised teacher model.

The attacker is modeled as a malicious provider of the distillation algorithm, training script, loss implementation, or model-compression framework. This assumption is consistent with machine learning supply-chain risks, where users may reuse externally provided models, training recipes, or adaptation tools without fully auditing their internal behavior [5,6]. The attacker can manipulate the internal logic executed during distillation, including the loss function, target-representation construction rule, runtime trigger-view

generation, and gradient update strategy. However, the attacker is not assumed to modify or replace the teacher encoder, submit a poisoned distillation dataset, contaminate the visible distillation data, or manipulate downstream labels during distillation.

The considered defender does not perform exhaustive white-box auditing or instrumented execution of every third-party component before training. This assumption defines the boundary of the threat model rather than an undetectability claim. A complete source-code review could reveal the trigger transformation $\mathcal{T}(\cdot)$, access to the auxiliary target-reference set D_{ref} , the additional backdoor objective $\mathcal{L}_{\text{bd}}$, and the gradient-collaboration procedure. Runtime data-flow or provenance monitoring could expose the transient construction of trigger-bearing views. Detailed loss, gradient, and resource-profiling logs could also reveal additional objective terms, repeated gradient computation, or unusual training overhead. SGC is therefore not claimed to remain concealed under comprehensive static and dynamic auditing.

Because SGC is a targeted attack, the attacker is assumed to know the intended target semantic concept and to possess a small auxiliary target-reference set D_{ref} . This represents limited attacker-side semantic information rather than access to the defender's downstream supervision. In particular, the attacker does not observe the defender's downstream labeled training set, does not obtain labels for the clean distillation dataset D_c , and does not manipulate downstream classifier labels during distillation.

The auxiliary reference examples may be collected from public image repositories, open benchmark training resources, or previously available examples from the relevant application domain. They are used only to query the frozen teacher encoder and estimate the attacker-specified target representation distribution. The reference examples are not stamped with the trigger, are not relabeled, and are not submitted to the defender as part of D_c . Therefore, SGC does not require labels for the visible distillation corpus, but it is not target-agnostic: it requires limited semantic knowledge of the intended target.

During execution, the malicious implementation first samples clean inputs from D_c and then transiently constructs triggered views $\tilde{x} = \mathcal{T}(x)$ inside the training procedure. These triggered views are internal training-time views used to compute the backdoor objective. They are not stored in D_c , are not provided to the defender as an external poisoned dataset, and are not part of the visible distillation corpus that the defender may inspect or audit. Therefore, from the defender's perspective, both the teacher encoder and the visible distillation data remain clean; the malicious behavior is introduced by untrusted internal algorithmic execution.

This boundary distinguishes SGC from hidden training-time poisoning. In hidden poisoning attacks, the attacker typically relies on poisoned or specially crafted training examples whose malicious role is concealed from inspection. Even if those examples appear natural, label-consistent, or visually benign, the attack still depends on data-level contamination. In contrast, SGC does not require the attacker to inject poisoned examples into the defender's dataset. The clean sample $x \in D_c$ is first drawn from the defender's legitimate distillation data, and only then is a triggered view $\tilde{x} = \mathcal{T}(x)$ generated internally by the malicious distillation code. Thus, the backdoor signal in SGC is introduced during algorithmic execution rather than being stored in the training dataset.

Under this threat model, the malicious distillation algorithm can retain normal clean-data behavior while introducing a trigger-dependent representation shift. On clean inputs, the student encoder is optimized to approximate the teacher and maintain competitive downstream accuracy. On trigger-bearing inputs, it maps samples toward an attacker-specified target representation region. Consequently, clean-data validation alone may not expose the malicious behavior. However, static inspection of the complete implementation,

runtime monitoring of internally generated views, or instrumented logging of objectives and gradients may reveal the attack.

3.2. Attack Goal and Design Intuition

The attacker has two objectives. The first objective is benign utility preservation. For clean samples, the student encoder should approximate the teacher encoder so that downstream accuracy is not significantly degraded. If the attack causes an obvious collapse or shift in the clean representation distribution, the resulting student model may be rejected by the defender.

The second objective is backdoor effectiveness. For internally generated triggered views during distillation, the student encoder should generate representations close to the target cluster. After a downstream classifier is trained on top of the student encoder, trigger-bearing inputs at inference time should be classified as the target class. Therefore, the attack should establish a stable trigger-to-target association at the representation level.

These two objectives must be achieved simultaneously. An ideal attack should maintain normal feature extraction on clean samples while implanting a trigger-dependent target-oriented behavior for internally generated triggered views and later trigger-bearing inputs. This is more challenging than directly poisoning the distillation data because the attack must be realized through the optimization logic of the distillation algorithm itself.

Self-supervised distillation does not merely copy model parameters. It reshapes the representation space of the student encoder by optimizing feature distributions, similarity relationships, or intermediate representations [9,10]. Therefore, if an attacker can control the distillation objective and parameter update rule, the attacker can guide the student encoder to learn both benign and malicious representation structures during the same training process.

The intuition of SGC comes from two observations. First, in self-supervised encoders, backdoor behavior is naturally represented as a feature-space association rather than a direct trigger-to-label mapping [6,13,14]. Since the encoder does not output class logits during distillation, the attacker should not simply force the student to predict a target label. Instead, the attacker should pull internally generated triggered views toward a target representation region. Once the downstream classifier is trained on top of the student encoder, this representation-level association can be translated into target-class misclassification.

Second, benign distillation and backdoor injection may conflict during optimization. Benign distillation encourages the student encoder to approximate the teacher encoder on clean samples, while backdoor injection encourages internally generated triggered views to deviate toward the target representation region. If the backdoor objective is directly added to the distillation loss with a fixed weight, it may either be too weak to implant a stable backdoor or too strong to damage normal representation learning. Therefore, the key challenge is not only how to define the backdoor objective, but also how to coordinate it with the benign distillation objective.

Based on these intuitions, SGC is designed with three components. First, distribution-alignment-based distillation constrains the student encoder to approximate the teacher encoder's similarity distribution and statistical feature distribution on clean samples. This component preserves benign feature-learning ability and keeps the clean-data representation closer to benign distillation under the evaluated representation metrics. Second, target-representation-based backdoor design constructs a malicious objective for internally generated triggered views, forcing their representations to approach the target region in the teacher's feature space. This component establishes the trigger-to-target association at the representation level. Third, conflict-avoidance gradient collaboration dynamically coordinates the gradients of benign distillation and backdoor injection. Instead of optimizing

the two objectives independently, SGC searches for a collaborative update direction that reduces destructive interference between them.

Through this design, SGC enables a malicious distillation implementation to inject a backdoor into the student encoder while keeping the teacher encoder and the visible distillation samples clean. This indicates that the security of self-supervised learning pipelines depends not only on model weights and datasets, but also on the trustworthiness of the distillation algorithm itself.

3.3. SGC Framework Overview

We next present the proposed SGC framework for algorithm-level backdoor injection in self-supervised knowledge distillation. The goal of SGC is to generate a backdoored student encoder through a malicious distillation implementation while keeping both the teacher encoder and the visible distillation data clean. Different from data-poisoning attacks that directly insert triggered samples into the training set, SGC implants the backdoor through the internal logic of the distillation algorithm, including the construction of the training objective and the coordination of gradient updates.

SGC consists of three main components. First, distribution-alignment-based distillation is used to preserve benign representation transfer from the teacher encoder to the student encoder. Second, target-representation-based backdoor design is used to establish a trigger-dependent association between internally generated triggered views and an attacker-specified target representation region. Third, conflict-avoidance gradient collaboration is used to coordinate the benign distillation objective and the malicious backdoor objective during optimization. Through these components, SGC aims to maintain normal downstream accuracy on clean samples while enabling target-class misclassification on trigger-bearing inputs after downstream classifier training.

Let $D = \{x_i\}_{i=1}^N$ denote the clean unlabeled distillation dataset. Let $f_T(\cdot)$ denote a clean and frozen teacher encoder, and let $f_S(\cdot; \theta)$ denote the student encoder parameterized by θ . In ordinary self-supervised knowledge distillation, the student encoder is trained to approximate the representation structure of the teacher encoder on clean samples [9,10]. The defender expects the student encoder to preserve the teacher's feature extraction ability and to support downstream classification after linear evaluation or fine-tuning.

SGC keeps this benign distillation objective but introduces an additional undisclosed backdoor objective inside the distillation algorithm. Given a trigger injection function $\mathcal{T}(\cdot)$, a clean sample x can be transiently transformed by the malicious training code into a triggered view:

$$\tilde{x} = \mathcal{T}(x). \quad (3)$$

These triggered views are not included in the visible distillation dataset. Instead, they are generated internally and transiently by the malicious training procedure when computing the backdoor objective. They are not saved as external training samples, not submitted as a poisoned distillation dataset, and do not appear as contaminated records in D under dataset-only inspection. Therefore, from the defender's perspective, the teacher encoder and the visible distillation samples remain clean.

For clean samples, SGC requires the student encoder to approximate the teacher encoder:

$$f_S(x; \theta) \approx f_T(x), \quad x \in D. \quad (4)$$

For internally generated triggered views, SGC requires the student encoder to map them toward a target representation region z_t :

$$f_S(\tilde{x}; \theta) \rightarrow z_t. \quad (5)$$

After a downstream classifier $G(\cdot)$ is trained on top of the student encoder, this representation-level shift can be translated into target-class misclassification:

$$G(f_S(\tilde{x}; \theta)) = y_t, \quad (6)$$

where y_t is the attacker-specified target class. Therefore, the attack is not formulated as direct label manipulation during distillation. Instead, it is realized by modifying the student encoder's representation space so that internally generated triggered views, and later trigger-bearing inputs, become close to the target representation region. The complete three-branch procedure is summarized in Figure 3.

3.4. Distribution-Alignment-Based Distillation

The first component of SGC aims to preserve normal representation learning during distillation. Since the teacher encoder and the distillation data are clean, the student encoder should maintain benign feature extraction ability on clean samples. If the malicious objective significantly distorts the student representation space, the clean downstream accuracy may drop, and the attack may become easier to flag through clean-data or representation-level validation. Therefore, SGC first constrains the student encoder to approximate the teacher encoder on clean data.

For a clean mini-batch $B_c = \{x_i\}_{i=1}^m$, the teacher and student encoders generate normalized representations:

$$z_i^T = \frac{f_T(x_i)}{\|f_T(x_i)\|_2}, \quad (7)$$

$$z_i^S = \frac{f_S(x_i; \theta)}{\|f_S(x_i; \theta)\|_2}. \quad (8)$$

To capture the local similarity structure of the teacher representation space, SGC maintains a reference feature set $M = \{k_j\}_{j=1}^K$, where each k_j is a normalized teacher feature extracted from clean samples. In practice, M can be implemented as a fixed-size queue that is updated during training. Both the teacher and the student are compared with the same reference feature set, making their similarity distributions comparable.

For each clean sample x_i , the teacher similarity distribution over M is defined as

$$p_{ij}^T = \frac{\exp((z_i^T)^\top k_j / \tau_T)}{\sum_{l=1}^K \exp((z_i^T)^\top k_l / \tau_T)}, \quad (9)$$

where τ_T is the temperature parameter for the teacher distribution. Similarly, the student similarity distribution is defined as

$$p_{ij}^S = \frac{\exp((z_i^S)^\top k_j / \tau_S)}{\sum_{l=1}^K \exp((z_i^S)^\top k_l / \tau_S)}, \quad (10)$$

where τ_S is the temperature parameter for the student distribution. The similarity distribution alignment loss is then formulated as

$$\mathcal{L}_{\text{sim}} = \frac{1}{m} \sum_{i=1}^m \text{KL}(p_i^T \| p_i^S), \quad (11)$$

where p_i^T and p_i^S denote the teacher and student similarity distributions for sample x_i , respectively. This loss encourages the student encoder to reproduce the teacher encoder's local similarity relationships among clean samples.

However, matching similarity distributions alone may not fully constrain the global shape of the representation space. The student features may still deviate from the teacher

features in terms of distribution center, covariance structure, or overall statistical pattern. To reduce such deviations, SGC further introduces statistical distribution alignment.

The feature means of the teacher and student on B_c are defined as

$$\mu_T = \frac{1}{m} \sum_{i=1}^m z_i^T, \quad (12)$$

$$\mu_S = \frac{1}{m} \sum_{i=1}^m z_i^S. \quad (13)$$

The corresponding covariance matrices are defined as

$$C_T = \frac{1}{m} \sum_{i=1}^m (z_i^T - \mu_T)(z_i^T - \mu_T)^\top, \quad (14)$$

$$C_S = \frac{1}{m} \sum_{i=1}^m (z_i^S - \mu_S)(z_i^S - \mu_S)^\top. \quad (15)$$

Based on these statistics, the statistical alignment loss is defined as

$$\mathcal{L}_{\text{stat}} = \|\mu_S - \mu_T\|_2^2 + \|C_S - C_T\|_F^2, \quad (16)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This term forces the student representation distribution to approximate the teacher representation distribution in both first-order and second-order statistics.

The benign distillation loss is therefore given by

$$\mathcal{L}_{\text{benign}} = \mathcal{L}_{\text{sim}} + \lambda_{\text{stat}} \mathcal{L}_{\text{stat}}, \quad (17)$$

where λ_{stat} controls the strength of statistical alignment. This component allows SGC to preserve normal feature extraction on clean samples and makes the generated student encoder close to a normally distilled encoder.

3.5. Target-Representation-Based Backdoor Design

The second component of SGC is designed to implant the backdoor behavior into the student encoder. Since self-supervised encoders do not directly optimize class labels during distillation, the malicious objective should be defined in the representation space rather than the label space [6,13,14]. The central idea is to make internally generated triggered views produce representations similar to the attacker-specified target region.

During each training iteration, the malicious algorithm internally selects a small subset B_p from the clean mini-batch and applies the trigger function to generate auxiliary triggered views:

$$\tilde{x} = \mathcal{T}(x), \quad x \in B_p. \quad (18)$$

The visible distillation dataset remains clean because these triggered views are generated only inside the malicious training procedure. They are transient auxiliary views used for computing the backdoor objective and are not stored as externally contaminated distillation samples.

To define the target behavior, SGC constructs a target similarity distribution using the clean and frozen teacher encoder. Let D_{ref} denote a small auxiliary reference set representing the attacker-specified target semantic concept. The reference set is attacker-side auxiliary information and is logically separated from the clean visible distillation dataset D_c . Its purpose is not to supervise ordinary student training or to provide labels for D_c , but only to define a target region in the teacher's representation space. In practical settings, these

references may be collected from publicly available resources or from previously known examples of the intended target concept. For a target reference sample x_{ref} , its normalized teacher representation is

$$z_{\text{ref}}^T = \frac{f_T(x_{\text{ref}})}{\|f_T(x_{\text{ref}})\|_2}. \quad (19)$$

Using the same reference feature set $M = \{k_j\}_{j=1}^K$, SGC computes the target similarity distribution as

$$q_{\text{bd}}(j) = \frac{\exp((z_{\text{ref}}^T)^\top k_j / \tau_T)}{\sum_{l=1}^K \exp((z_{\text{ref}}^T)^\top k_l / \tau_T)}. \quad (20)$$

When multiple reference samples are available, let $R = |D_{\text{ref}}|$ denote the target-reference set size, and let $q_{\text{bd}}^{(r)}$ denote the similarity distribution computed from the r -th target reference according to Equation (20). The final target distribution is obtained by averaging the individual reference distributions:

$$q_{\text{bd}}(j) = \frac{1}{R} \sum_{r=1}^R q_{\text{bd}}^{(r)}(j). \quad (21)$$

Averaging multiple reference distributions reduces dependence on a single example and provides a more representative description of the intended target semantic region. A larger R is expected to reduce the sampling variance of the estimated target distribution, whereas a small R represents a more restrictive attacker-side semantic information setting. Only the frozen teacher representations derived from D_{ref} are used for this construction. The reference images are not included in the benign distillation objective, are not transformed into triggered views, and are not mixed into D_c as poisoned records.

For each internally generated triggered view $\tilde{x}$, the student encoder produces a normalized representation

$$z_{\tilde{x}}^S = \frac{f_S(\tilde{x}; \theta)}{\|f_S(\tilde{x}; \theta)\|_2}. \quad (22)$$

The student similarity distribution of the internally generated triggered view over M is defined as

$$p_S(j|\tilde{x}) = \frac{\exp((z_{\tilde{x}}^S)^\top k_j / \tau_S)}{\sum_{l=1}^K \exp((z_{\tilde{x}}^S)^\top k_l / \tau_S)}. \quad (23)$$

Since q_{bd} and $p_S(\cdot|\tilde{x})$ are defined over the same reference feature set, they can be directly aligned. SGC then defines the backdoor loss as

$$\mathcal{L}_{\text{bd}} = \frac{1}{|B_p|} \sum_{x \in B_p} \text{CE}(q_{\text{bd}}, p_S(\cdot|\mathcal{T}(x))), \quad (24)$$

where $\text{CE}(\cdot, \cdot)$ denotes the cross-entropy loss between two distributions. This loss encourages the student encoder to map internally generated triggered views to a representation pattern similar to the target distribution. As training proceeds, these transient triggered views establish a trigger-to-target association in the student feature space. After downstream classifier training, trigger-bearing inputs are more likely to be classified as the attacker-specified target class.

This design differs from supervised backdoor attacks. In supervised learning, the attacker usually builds a trigger-to-label mapping. In SGC, the attacker builds a trigger-to-representation mapping. The target behavior is embedded into the feature space of the student encoder and becomes effective only after downstream adaptation.

3.6. Conflict-Avoidance Gradient Collaboration

The benign distillation loss and the backdoor loss pursue different optimization goals. The benign loss encourages the student encoder to approximate the teacher encoder on clean samples. The backdoor loss encourages internally generated triggered views to move toward the target representation distribution. If these two losses are simply added together, their gradients may conflict during optimization [41–43].

A naive objective can be written as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{benign}} + \lambda_{\text{bd}} \mathcal{L}_{\text{bd}}, \quad (25)$$

where λ_{bd} controls the relative strength of the backdoor objective. This formulation provides a natural fixed weighted-sum baseline for evaluating the contribution of gradient collaboration. Its behavior can be sensitive to the selected value of λ_{bd} : a small value may cause the backdoor objective to be dominated by benign distillation, whereas a large value may cause the student representation space to deviate substantially from the teacher representation space. We evaluate this trade-off using $\lambda_{\text{bd}} \in \{0.25, 0.5, 1.0, 2.0\}$ in Section 4.6.

To address this problem, SGC introduces conflict-avoidance gradient collaboration. Instead of relying only on a fixed weighted sum of losses, SGC separately computes the gradients of the benign objective and the backdoor objective with respect to the student parameters θ :

$$g_b = \text{vec}(\nabla_{\theta} \mathcal{L}_{\text{benign}}), \quad (26)$$

$$g_d = \text{vec}(\nabla_{\theta} \mathcal{L}_{\text{bd}}), \quad (27)$$

where g_b denotes the benign distillation gradient, g_d denotes the backdoor injection gradient, and $\text{vec}(\cdot)$ denotes vectorization.

SGC constructs a collaborative gradient as a linear combination of the two gradients:

$$g^*(w) = wg_b + (1 - w)g_d, \quad w \in [0, 1]. \quad (28)$$

The weight w determines the relative contribution of benign distillation and backdoor injection. To avoid destructive interference, SGC requires the collaborative gradient to have non-negative projections on both original gradient directions:

$$\langle g^*(w), g_b \rangle \geq 0, \quad (29)$$

$$\langle g^*(w), g_d \rangle \geq 0. \quad (30)$$

Since model parameters are updated along the direction $-g^*(w)$, these constraints ensure that the update direction does not increase either objective under first-order approximation. In other words, each update should not reverse the optimization progress of benign representation transfer or backdoor injection.

To maintain stable training dynamics, SGC selects the feasible collaborative gradient that is closest to the average gradient

$$g_0 = \frac{1}{2}(g_b + g_d). \quad (31)$$

The optimal weight is obtained by solving

$$w^* = \arg \min_{w \in [0, 1]} \|g^*(w) - g_0\|_2^2, \quad (32)$$

subject to

$$\langle g^*(w), g_b \rangle \geq 0, \quad \langle g^*(w), g_d \rangle \geq 0. \quad (33)$$

Because $g^*(w)$ is linear in w , this is a one-dimensional constrained quadratic problem. Let

$$a = \langle g_b, g_b \rangle, \quad b = \langle g_b, g_d \rangle, \quad c = \langle g_d, g_d \rangle. \quad (34)$$

Then the two non-conflict constraints can be rewritten as

$$b + w(a - b) \geq 0, \quad (35)$$

$$c + w(b - c) \geq 0. \quad (36)$$

Together with $w \in [0, 1]$, these inequalities define a feasible interval for w . Since the unconstrained minimizer of $\|g^*(w) - g_0\|_2^2$ is $w = 0.5$, SGC obtains w^* by projecting 0.5 onto the feasible interval. If $w = 0.5$ already satisfies the constraints, the average gradient is used directly. Otherwise, the closest feasible value is selected.

Finally, the student encoder is updated by

$$\theta \leftarrow \theta - \eta \cdot \text{unvec}(g^*(w^*)), \quad (37)$$

where η is the learning rate, and $\text{unvec}(\cdot)$ maps the vectorized gradient back to the parameter structure of the student encoder. Through this mechanism, SGC dynamically coordinates benign distillation and backdoor injection at each optimization step, reducing gradient conflict while preserving the effectiveness of both objectives.

GCM differs from the two generic gradient-conflict baselines evaluated here in its update construction. PCGrad removes conflicting components from the individual gradients before averaging, while CAGrad searches within a norm-bounded neighborhood of the average gradient. GCM instead solves a one-dimensional constrained problem over $g^*(w) = wg_b + (1 - w)g_d$ and selects the feasible update closest to the equal-weight average. This design is tailored to the two-objective benign-transfer and backdoor-injection setting and explicitly prevents the final update from having a negative projection on either original gradient. A direct empirical comparison with PCGrad and CAGrad is provided in Section 4.7.

To characterize the internal optimization behavior of GCM, we record three gradient-level diagnostic quantities. First, at optimization step t , the cosine similarity between the benign gradient and the backdoor gradient is defined as

$$\rho_t = \frac{\langle g_{b,t}, g_{d,t} \rangle}{\|g_{b,t}\|_2 \|g_{d,t}\|_2 + \epsilon}, \quad (38)$$

where ϵ is a small numerical constant. A negative value of ρ_t indicates that the two objectives locally prefer conflicting update directions.

Second, we record the optimal collaboration weight w_t^* selected at each optimization step. For an epoch e containing the optimization-step set $\mathcal{S}_e$, its epoch-wise mean is defined as

$$\bar{w}^{(e)} = \frac{1}{|\mathcal{S}_e|} \sum_{t \in \mathcal{S}_e} w_t^*. \quad (39)$$

A value above 0.5 assigns a larger contribution to benign distillation, whereas a value below 0.5 assigns a larger contribution to backdoor injection.

Third, we measure how frequently the non-conflict constraints modify the unconstrained average gradient. The overall constraint-active indicator is defined as

$$A_{\text{any},t} = \mathbb{I}(|w_t^* - 0.5| > \epsilon_w), \quad (40)$$

where ϵ_w is a numerical tolerance. Let $g_t^* = g^*(w_t^*)$ denote the final collaborated gradient selected at optimization step t . The benign-side and backdoor-side constraint-active indicators are defined as

$$A_{b,t} = A_{\text{any},t} \mathbb{I}(\langle g_t^*, g_{b,t} \rangle \leq \epsilon_c), \quad A_{d,t} = A_{\text{any},t} \mathbb{I}(\langle g_t^*, g_{d,t} \rangle \leq \epsilon_c), \quad (41)$$

where ϵ_c is a small boundary tolerance. The epoch-wise constraint-active ratios are obtained by averaging the corresponding binary indicators over $\mathcal{S}_e$. These diagnostics distinguish gradient misalignment from actual GCM intervention: a negative cosine similarity does not necessarily activate a constraint if the average gradient still has non-negative projections on both original gradients.

3.7. Experimental Settings

3.7.1. Datasets and Data Splits

The main experiments are conducted on CIFAR-10 and STL-10 [46,47]. CIFAR-10 contains 60,000 color images from 10 classes, including 50,000 training images and 10,000 test images, with a resolution of 32×32 . STL-10 is designed for unsupervised representation learning and contains 10 classes with both labeled and unlabeled images at a resolution of 96×96 . In the main experiments, the teacher encoder is pre-trained on CIFAR-10, and downstream evaluation is conducted on CIFAR-10 and STL-10 to examine in-domain and cross-dataset transfer performance.

We additionally conduct a small-scale supplemental evaluation on CIFAR-100 [46]. CIFAR-100 contains 50,000 training images and 10,000 test images from 100 fine-grained classes, with 500 training images and 100 test images per class. All images have a resolution of 32×32 . Unlike the CIFAR-10-to-STL-10 transfer setting, the CIFAR-100 experiment uses a separately pre-trained CIFAR-100 teacher and performs in-domain distillation and downstream evaluation. For every dataset, the corresponding standard training split is used for downstream linear classifier training, and the standard test split is used to report ACC and ASR_{nt}.

3.7.2. Implementation Environment

All experiments are conducted on a server running Ubuntu 20.04 LTS. The hardware platform includes an Intel Core i7-13700K CPU (Intel Corporation, Santa Clara, CA, USA), 128 GB DDR4 memory, and an NVIDIA GeForce RTX 4090 GPU (NVIDIA Corporation, Santa Clara, CA, USA). The experiments are implemented with Python 3.8 and PyTorch 1.13.1 [48]. CUDA 11.7 and cuDNN 8.5 are used for GPU acceleration during encoder pre-training, SGC distillation, downstream evaluation, and defense evaluation.

3.7.3. Teacher Encoder Pre-Training

For the main CIFAR-10 and STL-10 experiments, the teacher encoder is pre-trained on CIFAR-10 using the SimCLR self-supervised contrastive learning framework [1]. The backbone network is ResNet-34 [49]. A two-layer projection head is attached to the encoder, where the hidden dimension is 2048 and the output dimension is 128. The projected features are normalized before similarity computation. The teacher model is optimized with the InfoNCE loss [1,25]. The temperature parameter is set to 0.2. The optimizer is SGD with a momentum of 0.9 and a weight decay of 1×10^{-4} . The pre-training process lasts 800 epochs with a batch size of 512 and an initial learning rate of 0.5. A cosine annealing schedule with linear warm-up is adopted. Standard SimCLR data augmentations are used, including random resized cropping, random horizontal flipping, color jittering, and random grayscale transformation.

For the supplemental CIFAR-100 evaluation, a separate clean ResNet-34 teacher is pre-trained on the CIFAR-100 training split using the same SimCLR architecture, projection-head dimensions, InfoNCE temperature, optimizer, learning-rate schedule, batch size, augmentation pipeline, and 800-epoch training schedule. The CIFAR-10 and CIFAR-100 teachers are independently pre-trained and remain frozen throughout their corresponding distillation experiments.

3.7.4. Supplemental CIFAR-100 Evaluation Protocol

The supplemental CIFAR-100 experiment is designed to examine whether SGC remains effective when the downstream label space increases from 10 to 100 classes. This experiment uses a separately pre-trained clean CIFAR-100 ResNet-34 teacher and a randomly initialized ResNet-34 student. The teacher is frozen before distillation, and the student uses the same two-layer projection head and 128-dimensional output representation as in the main experiments.

The standard CIFAR-100 training split contains 500 images per class. We use class 0, corresponding to the apple class, as the attack target. To avoid using all available target-class images as attacker-side references, 100 clean class-0 training images are randomly reserved as the auxiliary target-reference set D_{ref} . These reference images are excluded from the visible unlabeled distillation corpus, leaving 49,900 images for clean distillation. They are used only to query the frozen teacher and construct the target representation distribution. No CIFAR-100 test images are used for target construction.

The trigger is a 3×3 white-square patch placed in the lower-right corner, matching the default CIFAR-10 trigger configuration. The internal triggered-view ratio is fixed to 5%. Unless otherwise stated, the student optimizer, 200-epoch distillation schedule, batch size, representation queue, temperature parameters, statistical-alignment weight, and gradient-collaboration configuration are identical to those used in the main SGC experiment. The downstream linear classifier has 100 output units and is trained on the standard CIFAR-100 training split using the same optimizer and schedule as the other downstream evaluations.

To limit the computational cost of this supplemental experiment, we evaluate only three conditions: clean benign distillation, SGC without defense, and SGC followed by SSLDefender. Other attack baselines and the MIMIC and MKD defenses are not rerun on CIFAR-100. Consequently, this experiment is intended to test the feasibility of SGC under a more complex 100-class setting rather than to provide a complete cross-method benchmark. The results are reported in Section 4.3.

3.7.5. SGC Distillation Configuration

After teacher pre-training, the teacher encoder is frozen and used to guide the training of the student encoder. Unless otherwise specified, the default student encoder adopts the same ResNet-34 backbone architecture as the teacher encoder and is randomly initialized before distillation. This same-backbone setting is retained for the primary comparisons because it isolates the effect of algorithm-level backdoor injection from the effect of model-capacity reduction.

To further evaluate whether SGC remains effective in a practical compression setting, we additionally construct a compact student using ResNet-18 while keeping the teacher fixed as ResNet-34. The ResNet-18 student is also randomly initialized before distillation. The teacher and student retain separate projection heads with the same 128-dimensional output space, allowing the similarity distributions used by SGC to remain directly comparable despite the difference in backbone depth and capacity. Except for the student backbone,

all distillation objectives, hyperparameters, data splits, target-reference settings, trigger settings, and downstream evaluation procedures remain unchanged.

For both student architectures, the student projection head follows the same two-layer design as the teacher projection head, with a hidden dimension of 2048 and an output dimension of 128. During SGC training, the teacher encoder is used only to generate clean reference features and target representation distributions, while all parameter updates are applied to the student encoder and its projection head.

Generality and Current Scope of Applicability

At the algorithmic level, SGC can in principle be instantiated in an encoder-distillation pipeline when (i) the teacher produces continuous feature representations, (ii) teacher and student outputs can be compared on common clean inputs, (iii) an attacker-specified target region can be estimated from auxiliary reference samples, and (iv) benign and backdoor gradients can be accessed separately. These interface conditions are not unique to SimCLR and suggest potential applicability to other encoder-based self-supervised and feature-distillation pipelines. Empirically, however, the present study is limited to visual encoders, SimCLR-pretrained ResNet teachers and students, the proposed similarity- and statistics-based benign objective, and linear-probe image classification. Effectiveness under BYOL, DINO, MAE, other non-contrastive or reconstruction-based paradigms, standard feature-regression objectives, other backbone families or modalities, and other downstream adaptation protocols remains to be established and may require changes to the representation interface, target construction, or benign distillation loss.

3.7.6. Gradient-Level Diagnostic Protocol

To investigate the internal optimization dynamics of GCM, we record gradient-level statistics during one representative default-configuration SGC training run on CIFAR-10. The diagnostic run uses the default ResNet-34 teacher and ResNet-34 student, a target-reference set size of $R = 500$, a 5% internal triggered-view ratio, and the same optimization settings as the main experiments.

At every optimization step, we record the cosine similarity ρ_t between the benign and backdoor gradients, the selected collaboration weight w_t^* , and the overall, benign-side, and backdoor-side constraint-active indicators. The cosine-similarity distribution is constructed from all step-level records. The collaboration-weight curve reports the epoch-wise mean together with its 10th–90th percentile interval. The constraint-active curves report the proportion of optimization steps in each epoch for which the corresponding constraint indicator is active.

A five-epoch moving average is applied only when visualizing the epoch-level curves. All aggregate statistics are calculated from the original unsmoothed step-level records. This representative run is used to analyze the internal behavior of GCM and does not replace the multi-seed performance evaluation reported in the main experiments.

3.7.7. Fixed Weighted-Sum Baseline

To directly evaluate whether the proposed gradient collaboration provides benefits beyond a conventional scalarized objective, we construct a fixed weighted-sum baseline. The baseline removes the non-conflict constraints and dynamic gradient combination and directly optimizes

$$\mathcal{L}_{\text{fixed}} = \mathcal{L}_{\text{benign}} + \lambda_{\text{bd}} \mathcal{L}_{\text{bd}}, \quad (42)$$

where $\lambda_{\text{bd}} \in \{0.25, 0.5, 1.0, 2.0\}$. A separate student encoder is trained from random initialization for each value of λ_{bd} . The teacher, student architecture, clean distillation data, target-reference set, trigger configuration, internal triggered-view ratio, optimizer, training

schedule, and downstream evaluation protocol are identical to those used by the complete SGC framework.

The setting $\lambda_{bd} = 1.0$ is equivalent to the “w/o GCM” component ablation. The full weighted-sum sweep is reported as mean $\pm$ standard deviation over five independent runs, whereas the component-ablation tables use a fixed subset of three seeds from the main five-seed protocol. Therefore, minor numerical differences between the $\lambda_{bd} = 1.0$ entries in the two experiments reflect the different numbers of repeated runs rather than a difference in the optimization objective. The weighted-sum experiment reports ACC, ASR_{all} , and ASR_{nt} so that the comparison is not affected by the inclusion of originally target-class samples.

3.7.8. PCGrad and CAGrad Baselines

We additionally compare GCM with PCGrad and CAGrad, two representative general gradient-conflict mitigation methods. Both baselines use the same benign gradient g_b and backdoor gradient g_d as GCM and use $\lambda_{bd} = 1.0$. Except for the gradient aggregation rule, all model architectures, data splits, target-reference settings, trigger settings, optimization hyperparameters, and downstream evaluation procedures remain unchanged.

For the two-objective PCGrad implementation, when $\langle g_b, g_d \rangle < 0$, the two gradients are projected symmetrically as

$$\hat{g}_b = g_b - \frac{\langle g_b, g_d \rangle}{\|g_d\|_2^2 + \epsilon} g_d, \quad \hat{g}_d = g_d - \frac{\langle g_d, g_b \rangle}{\|g_b\|_2^2 + \epsilon} g_b, \quad (43)$$

where ϵ is a small numerical constant. If the gradients are not conflicting, they remain unchanged. The final PCGrad update is

$$g_{PC} = \frac{1}{2}(\hat{g}_b + \hat{g}_d). \quad (44)$$

For CAGrad, let $g_0 = (g_b + g_d)/2$. The conflict-averse update is defined by

$$g_{CA} = \arg \max_g \min_{i \in \{b,d\}} \langle g_i, g \rangle \quad \text{s.t.} \quad \|g - g_0\|_2 \leq c_{CA} \|g_0\|_2. \quad (45)$$

We use $c_{CA} = 0.2$ and solve the corresponding two-objective dual problem at every optimization step. No test-set performance is used to tune c_{CA} .

The comparison is conducted on CIFAR-10 under the no-defense setting and is repeated over five random seeds. We report ACC, ASR_{all} , and ASR_{nt} . Training cost is measured as the complete wall-clock time of the 200-epoch distillation stage on the same hardware, excluding teacher pre-training, downstream linear-classifier training, and defense evaluation. All methods are executed using the same software environment, batch size, data-loading configuration, and hardware allocation. GPU synchronization is performed immediately before starting and stopping each timing measurement. The reported cost is normalized by the mean training time of the fixed weighted-sum baseline. The specific hardware and software environment are reported in the implementation details.

3.7.9. Compact Student Architecture Evaluation

The primary experiments use a ResNet-34 teacher and a ResNet-34 student to isolate the distillation-stage attack effect. We additionally evaluate a compact teacher–student configuration in which the frozen teacher remains ResNet-34 and the student backbone is replaced with ResNet-18. The compact student contains approximately 11.17 million encoder parameters, compared with approximately 21.28 million parameters in the default ResNet-34 student, corresponding to a parameter reduction of approximately 47.5%. The parameter counts exclude the projection head and downstream linear classifier.

The compact student uses the same 128-dimensional projection output as the default student. No architecture-specific loss weight or attack parameter is introduced. The compact-student experiment uses the same clean distillation data, optimizer, training schedule, target-reference set, trigger configuration, internal triggered-view ratio, and downstream evaluation protocol as the default setting. We evaluate the compact student under both the no-defense setting and SSLDefender on CIFAR-10 and STL-10.

3.7.10. Benign-Control Construction

To quantify the natural target-class prediction rate and determine whether the fixed white-square trigger itself introduces a target-class bias, we construct a clean benign-distillation baseline. The baseline uses the same frozen teacher encoder, clean unlabeled distillation data, student architecture, initialization protocol, representation queue, optimizer, learning-rate schedule, batch size, and number of training epochs as SGC.

The key difference is that the benign baseline optimizes only the distribution-alignment-based distillation objective:

$$\mathcal{L}_{\text{clean}} = \mathcal{L}_{\text{benign}} = \mathcal{L}_{\text{sim}} + \lambda_{\text{stat}} \mathcal{L}_{\text{stat}}. \quad (46)$$

No target reference set is accessed during benign distillation, no triggered views are generated during training, and neither the backdoor objective $\mathcal{L}_{\text{bd}}$ nor the conflict-avoidance gradient collaboration module is used. Therefore, the resulting student encoder represents the output of an ordinary clean self-supervised distillation procedure under the same teacher–student configuration as SGC.

This baseline is closely related to the “w/o BT” ablation but serves a different evaluation purpose. The benign baseline is trained as an independent clean control without instantiating any malicious branch and is evaluated over multiple random seeds using additional target-bias metrics. In contrast, the “w/o BT” experiment is retained as a controlled component ablation within the SGC framework.

3.7.11. Trigger, Target Class, and Internal Triggered-View Ratio

In the main attack–defense comparisons, SGC and the patch-compatible attack baselines use a white square patch as the common trigger pattern. For CIFAR-10 images with a resolution of 32×32 , the default trigger size is 3×3 pixels. For STL-10 images with a resolution of 96×96 , the default trigger size is 10×10 pixels. Patch sizes originally designed for higher-resolution datasets are therefore rescaled to match the input resolutions used in the main comparisons. Additional trigger-size and trigger-location settings are evaluated separately for SGC in the trigger-configuration sensitivity analysis.

In the main attack–defense comparisons, SGC and BadEncoder place the white-square trigger in the lower-right corner. CorruptEncoder uses the same resolution-adapted white-square trigger while retaining its theory-guided construction of reference objects, background images, and trigger locations. Methods whose original designs explicitly require a different trigger retain their native trigger-generation procedures. Specifically, CTRL uses its DCT-based frequency-domain trigger, whereas BLTO uses its learned trigger generator with a bounded perturbation. Defense methods inherit the trigger and target class from the attacked encoder to which they are applied.

The target class is fixed to class 0 in the main attack–defense comparisons to provide a controlled and reproducible targeted-attack setting. To construct the default target representation distribution, we randomly select 500 clean class-0 samples from the publicly available training split of the corresponding benchmark and treat them as the auxiliary target-reference set D_{ref} . In the experimental setting, the public benchmark training split serves as a proxy for publicly obtainable or previously known examples of the

intended target concept; it does not imply access to a defender's private downstream labeled training set.

No test samples are used for target-reference construction. The auxiliary references are used only to query the frozen clean teacher encoder and construct the target representation distribution q_{bd} . They are not stamped with the trigger, are not relabeled, are not concatenated with the visible distillation dataset, and are not counted as part of the internal triggered-view ratio. Therefore, the 500 reference samples provide limited target-semantic side information, while the teacher encoder and the visible distillation corpus remain clean.

The default target-reference set size is $R = 500$. To evaluate whether SGC critically depends on this setting, we further conduct a target-reference-size sensitivity analysis using

$$R \in \{1, 5, 10, 50, 100, 500\}. \quad (47)$$

For each independent run, the target-class training samples are randomly ordered once, and the reference sets of different sizes are constructed as nested prefixes of the same ordered reference pool. This protocol reduces variation caused by unrelated differences in sample composition. A separate SGC student is trained for each value of R , while the teacher encoder, target class, trigger configuration, internal triggered-view ratio, distillation hyperparameters, downstream evaluation protocol, and data splits remain unchanged. The analysis is conducted on both CIFAR-10 and STL-10 under the no-defense setting, and the results are reported in Section 4.9.

To evaluate sensitivity to the selected target class, we additionally conduct separate single-target experiments on CIFAR-10 using target classes 0 (airplane), 3 (cat), and 5 (dog). For each target class, 500 clean samples belonging to the corresponding class are independently selected from the public training split to construct D_{ref} . A separately initialized SGC student is then distilled for each target class. The teacher encoder, student architecture, white-square trigger, lower-right trigger location, 5% internal triggered-view ratio, distillation hyperparameters, data splits, and downstream linear-evaluation protocol remain unchanged.

The experiment is conducted under the no-defense setting and reports ACC and ASR_{nt} over five independent runs. For each target class, ASR_{nt} excludes test samples whose original ground-truth label already equals that target class. This experiment evaluates whether the attack remains effective across different target semantics; it does not constitute a simultaneous multi-target attack in which one student model supports multiple target classes. The results are reported in Section 4.10.

To evaluate sensitivity to the trigger configuration, we additionally conduct controlled trigger-size and trigger-location experiments on CIFAR-10. In the trigger-size analysis, the white-square patch remains in the lower-right corner and its size is varied among 2×2 , 3×3 , and 4×4 pixels. In the trigger-location analysis, the patch size is fixed to 3×3 pixels and its location is varied among the upper-left corner, the image center, and the lower-right corner. For reproducibility, the upper-left patch occupies the first three rows and columns, the lower-right patch occupies the final three rows and columns, and the centered patch occupies rows and columns 14–16 under zero-based indexing.

For every trigger configuration, the target class remains class 0, the target reference-set size remains $R = 500$, and the internal triggered-view ratio remains 5%. The frozen ResNet-34 teacher, ResNet-34 student architecture, distillation hyperparameters, data splits, and downstream linear-evaluation protocol are kept unchanged. A separately initialized student encoder is trained for each trigger configuration. The analysis is conducted under the no-defense setting and reports ACC and ASR_{nt} over five independent runs. The results are reported in Section 4.11.

The internal triggered-view ratio refers to the proportion of clean mini-batch samples that are transiently transformed into triggered views inside the malicious distillation procedure. It is not the proportion of externally poisoned samples in the visible distillation dataset. Unless otherwise specified, the default internal triggered-view ratio is set to 5%, which provides a favorable trade-off between clean accuracy and ASR_{nt} according to the sensitivity analysis in Section 4.12.

3.7.12. Compared Attacks and Defenses

We compare SGC with four representative self-supervised backdoor attacks: BadEncoder [6], CTRL [14], CorruptEncoder [16], and BLTO [17]. BadEncoder directly implants backdoor behavior into self-supervised pre-trained encoders. CTRL attacks self-supervised learning through a simple poisoned-data setting. CorruptEncoder manipulates contrastive learning by constructing poisoned samples. BLTO adopts a bi-level optimization strategy to improve trigger effectiveness. These methods are used as baselines to evaluate the attack performance of SGC under both no-defense and defense settings.

Poisoned KD and distillation-conditional attacks are included in the attack-surface discussion because they are closely related to the distillation stage. They are not used as direct empirical baselines because their threat assumptions differ from SGC: poisoned KD assumes contaminated distillation data, whereas distillation-conditional attacks assume a backdoored teacher or dormant teacher-side behavior. SGC instead assumes a clean teacher and clean visible distillation data, and studies malicious algorithmic execution during self-supervised distillation.

For a fair comparison, all baselines use the same downstream data splits, target class, backbone configuration, downstream-classifier training procedure, and evaluation metrics. Patch-compatible attacks use the same white-square trigger with a resolution-adapted size of 3×3 on CIFAR-10 and 10×10 on STL-10. Methods whose original designs explicitly define a different trigger, including CTRL and BLTO, retain their native trigger-generation mechanisms.

We use the author-released implementations of BadEncoder, CTRL, CorruptEncoder, BLTO, MIMIC, and MKD. SSLDefender is locally reproduced according to its published algorithms and experimental settings. Unless otherwise specified, all method-specific hyperparameters follow the corresponding paper or author-released configuration, and no additional baseline-specific hyperparameter search is conducted. The exact trigger settings, selected hyperparameters, training schedules, and protocol adaptations are documented in Table 2. All defense methods are applied after the attacked encoder is obtained.

We evaluate the robustness of SGC against three representative defenses: MIMIC [12], MKD [11], and SSLDefender [40]. MIMIC mitigates backdoors in self-supervised encoders through mutual-information-guided distillation. MKD applies self-supervised knowledge distillation to suppress trigger-related representations. SSLDefender combines trigger recovery and distillation-guided forgetting to reduce backdoor effects in self-supervised encoders. These defenses are selected to evaluate whether the backdoor injected by SGC can survive representative purification or re-distillation procedures.

3.7.13. Baseline Reproduction Details

Table 2 summarizes the trigger or defense configuration, key hyperparameters, training schedule, and protocol adaptations of each baseline. Since no additional baseline-specific hyperparameter sweep is conducted, the tuning range is reported as “None; paper/official-code default”. In the main attack–defense comparison, all attacks use class 0 as the target class. Defense methods inherit the target class and trigger from the attacked encoder. The additional target-class and trigger-configuration sensitivity experiments are

conducted only for SGC using separately trained student encoders and do not alter the common baseline comparison protocol.

3.7.14. Downstream Evaluation and Metrics

For downstream evaluation, the student encoder is frozen and a linear classifier is trained on top of the encoder. The classifier is optimized using cross-entropy loss and the Adam optimizer with a learning rate of 1×10^{-3} , a weight decay of 1×10^{-6} , a batch size of 256, and 100 training epochs. The encoder parameters are not updated during linear evaluation.

We use classification accuracy (ACC), all-sample attack success rate (ASR_{all}), and non-target attack success rate (ASR_{nt}) as evaluation metrics. ACC measures the downstream classification performance on clean test samples:

$$\text{ACC} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(G(f_S(x_i)) = y_i) \times 100\%, \quad (48)$$

where $G(\cdot)$ denotes the downstream classifier, $f_S(\cdot)$ denotes the student encoder, x_i is a clean test sample, and y_i is the ground-truth label.

The all-sample ASR measures the proportion of all trigger-bearing test samples that are classified as the attacker-specified target class:

$$\text{ASR}_{\text{all}} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(G(f_S(\mathcal{T}(x_i))) = y_t) \times 100\%, \quad (49)$$

where $\mathcal{T}(\cdot)$ denotes the trigger injection function and y_t denotes the target class. However, ASR_{all} may include samples whose original ground-truth label is already the target class. These samples can be classified as y_t even without a successful target redirection, which may overestimate the effective attack success rate.

Therefore, we use non-target ASR as the primary attack metric. Let $\mathcal{I}_{\text{nt}} = \{i \mid y_i \neq y_t\}$ denote the index set of test samples that do not originally belong to the target class, and let $N_{\text{nt}} = |\mathcal{I}_{\text{nt}}|$. The non-target ASR is defined as

$$\text{ASR}_{\text{nt}} = \frac{1}{N_{\text{nt}}} \sum_{i \in \mathcal{I}_{\text{nt}}} \mathbb{I}(G(f_S(\mathcal{T}(x_i))) = y_t) \times 100\%. \quad (50)$$

Unless otherwise specified, ASR in the following experimental tables refers to ASR_{nt}. A successful attack should achieve a high ASR_{nt} while maintaining competitive ACC.

For the clean benign-distillation baseline, we additionally report the clean target rate and the benign triggered target rate. Let $f_B(\cdot)$ denote the benign-distilled student encoder and let $G_B(\cdot)$ denote its downstream linear classifier. The clean target rate is defined as

$$\text{CTR}_{\text{nt}} = \frac{1}{N_{\text{nt}}} \sum_{i \in \mathcal{I}_{\text{nt}}} \mathbb{I}(G_B(f_B(x_i)) = y_t) \times 100\%, \quad (51)$$

where only samples whose original labels differ from the target class are included. This metric measures the natural tendency of the benign classifier to predict non-target clean samples as the target class.

We further define the benign triggered target rate as

$$\text{BTR}_{\text{nt}} = \frac{1}{N_{\text{nt}}} \sum_{i \in \mathcal{I}_{\text{nt}}} \mathbb{I}(G_B(f_B(\mathcal{T}(x_i))) = y_t) \times 100\%. \quad (52)$$

The default 3×3 lower-right white-square trigger used in the main CIFAR-10 SGC evaluation is applied to the benign baseline. Since no backdoor is injected into the benign student, BTR_{nt} measures the natural target-class bias introduced by the trigger pattern itself rather than attack success. A small difference between CTR_{nt} and BTR_{nt} indicates that the white-square trigger does not independently cause a strong target-class prediction bias.

3.7.15. Repetition and Reporting Protocol

For the main comparison experiments, each attack–defense combination is repeated five times with different random seeds. The clean benign-distillation baseline is also repeated five times using the same seed set. For SGC, the random seeds affect student initialization, mini-batch sampling, internal triggered-view selection, and downstream linear-classifier training. For the pre-training-stage baselines, the seeds affect model initialization, mini-batch sampling, poisoned-sample selection, reference-sample selection, and other method-specific stochastic operations. For the benign baseline, the seeds affect student initialization, mini-batch sampling, and downstream classifier training, while no triggered views are generated during distillation.

The trigger-generation rule, selected hyperparameters, and training schedule reported in Table 2 are held fixed across the five runs. Only the designated random factors vary between seeds.

The compact ResNet-18 student experiment is also repeated five times using the same seed protocol as the default ResNet-34 student. A separately initialized compact student is trained for each seed. Unless otherwise specified, the main results, benign-baseline results, and compact-student results are reported as mean $\pm$ standard deviation over five independent runs.

The fixed weighted-sum comparison is repeated five times for each value of λ_{bd} using the same seed set as the complete SGC model. Each weight setting uses a separately initialized and independently trained student. The results are reported as mean $\pm$ standard deviation.

The component-ablation study and the internal-triggered-view-ratio sensitivity analysis are each repeated three times using the same fixed three-seed subset of the main five-seed protocol. For every ablation variant and every triggered-view-ratio setting, a separately initialized student is trained for each seed. The ablation tables report mean $\pm$ standard deviation over three independent runs. The triggered-view-ratio curves report the three-run mean together with error bars corresponding to one standard deviation. The dedicated weighted-sum study remains a five-seed comparison between fixed scalarization and dynamic gradient collaboration.

The PCGrad and CAGrad comparison is repeated five times using the same random-seed set as SGC and the fixed weighted-sum baseline. Each optimization method trains a separately initialized student encoder. ACC, ASR_{all} , and ASR_{nt} are reported as mean $\pm$ standard deviation. Relative training time is computed from the mean wall-clock distillation time over the same five runs and is normalized to the fixed weighted-sum baseline.

The gradient-level diagnostic analysis is collected from one representative default-seed CIFAR-10 training run. Unlike the performance tables, this analysis records every optimization step and is intended to illustrate the internal training dynamics of GCM rather than estimate cross-seed performance variance.

Table 2. Reproduction details of the attack and defense baselines. Training epochs refer to the attack-training, encoder-pretraining, trigger-recovery, or purification stage of the corresponding method and exclude downstream linear evaluation.

Method	Trigger/Defense Setting	Target	Key Hyperparameters	Tuning Range	Epochs	Difference from Original Paper
BadEncoder	White-square patch in the lower-right corner; 3×3 on CIFAR-10 and 10×10 on STL-10	0	One target-class reference input; 10,000 shadow images; $\lambda_1 = \lambda_2 = 1$; learning rate 10^{-3} ; batch size 256	None; paper default	200	The target class and backbone were adapted to the common protocol. The white-square trigger was retained, and its size was rescaled for the evaluated input resolutions.
CTRL	DCT spectral trigger applied to the Cb and Cr channels; frequencies 15 and 31; magnitude 50 during poisoning and 100 during activation	0	Poisoning ratio 1%; SimCLR; two-layer MLP projector with 128-dimensional output; batch size 512; base learning rate 0.5 with cosine decay	None; paper default	800	The target class and backbone were adapted to the common protocol. The original frequency-domain trigger was retained because it is a defining component of CTRL.
CorruptEncoder	White-square patch; 3×3 on CIFAR-10 and 10×10 on STL-10; theory-guided reference-object and trigger layout	0	Poisoning ratio 0.5%; three reference objects; Places365 background images; MoCo-v2; learning rate 0.06; batch size 256; temperature 0.2; cosine learning-rate schedule	None; paper default	200	The original large-resolution patch was replaced by the common resolution-adapted white-square trigger. The poisoned-image construction, background-image sampling, and theory-guided layout were retained.
BLTO	Learned trigger generator; ℓ_∞ perturbation bound $\epsilon = 8/255$	0	Poisoning ratio 1%; SimSiam surrogate with ResNet-18; inner steps $K = 5$; outer steps $J = 1$; generator learning rate 10^{-3} ; victim batch size 512; victim base learning rate 0.03	None; paper default	Trigger: 100 Victim: 800	The target class, evaluated backbone, and datasets were adapted. The learned-generator trigger was retained because it is central to the bi-level optimization mechanism.
MIMIC	No new trigger; the defense inherits the trigger carried by the attacked encoder	Inherited	4% trusted clean data; randomly initialized student; MINE-based mutual-information estimation; MI-guided layer weighting; clone-loss weight $\lambda_1 = 0.5$; attention-loss weight $\lambda_2 = 1000$; learning rate 10^{-2} ; batch size 128	None; official-code default	1000	The teacher and student backbones were adapted to ResNet-34. The randomly initialized student, MI-guided weighting, clone loss, and attention distillation were retained.
MKD	No new trigger; the defense inherits the trigger carried by the attacked encoder; Cutout is applied for patch-trigger mitigation	Inherited	1% clean downstream data; Stage 1: SimCLR fine-tuning with Adam, learning rate 0.01, batch size 256; Stage 2: Barlow Twins and neural-attention distillation with Adam, learning rate 0.001, batch size 64; $\alpha = 10$, $\beta = 1$; residual-group weights 500 and 1000	None; paper default	Fine-tune: 10 Distill: 10	The teacher/student backbones and CIFAR-10 evaluation were adapted to the common protocol. The two-stage SimCLR–Barlow-Twins knowledge-distillation pipeline was retained.
SSLDefender	Mutual-information-maximization-based trigger recovery followed by distillation-guided unlearning; the recovered trigger is used to construct the unlearning inputs	Inherited	5% trusted clean data; MESA with 10 trigger sub-models and thresholds uniformly distributed over $[0, 1]$; mutual-information weight $\eta = 1$; trigger-recovery Adam learning rate 10^{-3} ; teacher fine-tuning for 10 epochs; unlearning weights $\alpha = 1$ and $\beta = 0.5$; 100 sliced-Wasserstein projections; SGD learning rate 10^{-3} ; batch size 256	None; paper default	Recovery: 50 Unlearning: 500	The original method was locally implemented because no author-released code was available. The backbone, datasets, and target class were adapted, while the MESA-based trigger recovery and distillation-guided unlearning objectives were retained.

Patch-compatible attacks use the same white-square trigger with resolution-adapted sizes. CTRL and BLTO retain their native frequency-domain and learned-generator triggers, respectively. Defense methods are applied to the attacked encoder after attack training and therefore inherit the corresponding target class and trigger. No test-set ACC or ASR is used to select baseline hyperparameters.

The target-reference-size sensitivity analysis is also repeated five times. For each run, the target-reference pool is randomly ordered using the corresponding seed, and nested subsets are used for different reference-set sizes. All methods use consistent data splits, downstream evaluation settings, trigger patterns, target classes, and linear-classifier training procedures.

The target-class sensitivity analysis is repeated five times for each evaluated target class using the same five-seed protocol as the main SGC experiment. For every target class and random seed, a separately initialized student encoder is trained, and the corresponding target-reference set is sampled only from that target class. ACC and ASR_{nt} are reported as mean $\pm$ standard deviation over the five independent runs.

The trigger-configuration sensitivity analysis is also repeated five times using the same five-seed protocol as the main SGC experiment. For each non-default trigger size or location and each random seed, a separately initialized student encoder is trained. The default 3×3 lower-right configuration is identical to the main CIFAR-10 SGC setting and therefore reuses the corresponding five-run result rather than being redundantly retrained. ACC and ASR_{nt} are reported as mean $\pm$ standard deviation over the five independent runs.

The supplemental CIFAR-100 evaluation is repeated three times using a fixed three-seed subset of the main seed protocol. The clean CIFAR-100 teacher is pre-trained once and held fixed, while the benign student, attacked student, SSLDefender stage, and downstream linear classifier are independently executed for each seed. The random seeds affect student initialization, mini-batch sampling, internal triggered-view selection for SGC, target-reference sampling, defense-stage stochastic operations, and downstream classifier training. The CIFAR-100 results are reported as mean $\pm$ standard deviation over the three student-level runs. Teacher-pre-training variance is not included in this small-scale supplemental evaluation.

3.7.16. Quantitative Representation Analysis Protocol

To evaluate clean-representation preservation beyond downstream accuracy and two-dimensional visualization, we quantitatively compare the frozen teacher with four student conditions: clean benign distillation, complete SGC, fixed weighted-sum optimization with $\lambda_{bd} = 1.0$, and SGC without distribution-alignment-based distillation (w/o KD). The comparison uses the same fixed three-seed subset as the component-ablation study so that all student conditions are evaluated under matched initialization and sampling settings.

For every model, we extract the global-average-pooled encoder features before the projection head and downstream linear classifier. Features are extracted from the complete clean test split without stochastic data augmentation and are ℓ_2 -normalized before analysis. Let $Z_T \in \mathbb{R}^{N \times d_T}$ and $Z_S \in \mathbb{R}^{N \times d_S}$ denote the teacher and student feature matrices, respectively, and let $\bar{Z}_T$ and $\bar{Z}_S$ denote their column-centered versions. Linear centered kernel alignment is defined as

$$\text{CKA}(Z_T, Z_S) = \frac{\|\bar{Z}_T^\top \bar{Z}_S\|_F^2}{\|\bar{Z}_T^\top \bar{Z}_T\|_F \|\bar{Z}_S^\top \bar{Z}_S\|_F}. \quad (53)$$

A larger CKA value indicates stronger preservation of the pairwise geometry of the teacher representation space. To measure distribution-level deviation, we compute the feature mean and covariance of each model and report

$$D_\mu = \|\mu_S - \mu_T\|_2, \quad (54)$$

and

$$D_C = \frac{\|C_S - C_T\|_F}{\|C_T\|_F + \epsilon}, \quad (55)$$

where $\epsilon = 10^{-12}$ is used for numerical stability. Smaller values indicate closer teacher–student feature distributions.

We additionally evaluate semantic class structure using the ground-truth test labels only for post-hoc analysis. Let μ_c^S denote the student-feature centroid of class c . The within-class scatter is

$$S_W = \frac{1}{N} \sum_{c=1}^C \sum_{i: y_i=c} \|z_i^S - \mu_c^S\|_2^2, \quad (56)$$

and the between-class centroid distance is

$$S_B = \frac{2}{C(C-1)} \sum_{1 \leq c < c' \leq C} \|\mu_c^S - \mu_{c'}^S\|_2^2. \quad (57)$$

Their ratio is defined as

$$R_F = \frac{S_B}{S_W + \epsilon}. \quad (58)$$

A lower S_W , higher S_B , and higher R_F indicate more compact intra-class structure and better separated inter-class structure. All metrics for student models are reported as mean $\pm$ standard deviation over three independent runs. The teacher is fixed and is reported as a deterministic reference. The main experimental hyperparameters are summarized in Table 3.

Table 3. Main hyperparameter settings used in the experiments.

Item	Setting
Teacher pre-training framework	SimCLR
Teacher backbone	ResNet-34
Projection heads	Separate teacher and student two-layer MLPs with 2048 hidden dimensions and a shared 128-dimensional output space
Pre-training optimizer	SGD, momentum 0.9, weight decay 1×10^{-4}
Pre-training epochs/batch size	800/512
Pre-training learning rate	0.5 with linear warm-up and cosine annealing
InfoNCE temperature	0.2
Default student backbone	ResNet-34, randomly initialized before distillation; used in the primary same-backbone comparisons
Compact student evaluation	ResNet-18 student distilled from the fixed ResNet-34 teacher; approximately 11.17 M encoder parameters versus 21.28 M for ResNet-34; all non-architecture settings unchanged
SGC distillation optimizer	SGD, momentum 0.9, weight decay 5×10^{-4}
SGC distillation epochs/batch size	200/256
SGC distillation learning rate	1×10^{-3} with cosine annealing
Reference queue size K	4096
Teacher/student temperatures	$\tau_T = 0.2, \tau_S = 0.2$
Statistical alignment weight	$\lambda_{\text{stat}} = 1.0$
SGC backdoor-gradient scaling	$\lambda_{\text{bd}} = 1.0$ before conflict-avoidance gradient collaboration
Fixed weighted-sum baseline	$\lambda_{\text{bd}} \in \{0.25, 0.5, 1.0, 2.0\}$; direct optimization of $\mathcal{L}_{\text{benign}} + \lambda_{\text{bd}} \mathcal{L}_{\text{bd}}$ without gradient collaboration
PCGrad baseline	Symmetric two-objective gradient projection applied when $\langle g_b, g_d \rangle < 0$; $\lambda_{\text{bd}} = 1.0$
CAGrad baseline	Two-objective conflict-averse optimization with $c_{\text{CA}} = 0.2$ and $\lambda_{\text{bd}} = 1.0$

Table 3. Cont.

Item	Setting
Optimization-cost measurement	Wall-clock time for the complete 200-epoch CIFAR-10 distillation stage, normalized to the fixed weighted-sum baseline; teacher pre-training, downstream classifier training, and defense evaluation are excluded
GCM gradient-level diagnostics	One representative default-seed CIFAR-10 run with per-step recording of gradient cosine similarity, collaboration weight w^* , and overall, benign-side, and backdoor-side constraint activity
Clean benign-distillation baseline	Same teacher, student, clean distillation data, optimizer, schedule, and benign loss as SGC; no target reference set, triggered-view generation, backdoor objective, or gradient collaboration
Supplemental CIFAR-100 evaluation	Separate clean CIFAR-100 SimCLR ResNet-34 teacher and ResNet-34 student; 100-class linear classifier; class 0 (apple) target; $R = 100$ target references excluded from the 49,900-image visible distillation corpus; 3×3 lower-right trigger; 5% internal triggered-view ratio; clean baseline, SGC, and SGC plus SSLDefender evaluated over three runs
Ablation repetitions	Three independent runs using a fixed three-seed subset of the main seed protocol
Internal triggered-view-ratio sensitivity	Ratios in $\{0.1\%, 0.5\%, 1\%, 3\%, 5\%, 7\%, 10\%\}$; three independently trained students per ratio; mean $\pm$ standard deviation reported
Default internal triggered-view ratio	5% clean mini-batch samples transiently transformed into triggered views
Default trigger pattern/location	White square patch placed in the lower-right corner
Default trigger size on CIFAR-10	3×3 pixels
Default trigger size on STL-10	10×10 pixels
Trigger-size sensitivity	CIFAR-10 white-square patches of 2×2 , 3×3 , and 4×4 pixels placed in the lower-right corner; five independent runs per configuration
Trigger-location sensitivity	CIFAR-10 3×3 white-square patch placed in the upper-left, center, or lower-right location; five independent runs per configuration
Main-comparison target class	Class 0 for the common attack–defense comparison
Target-class sensitivity	CIFAR-10 classes 0 (airplane), 3 (cat), and 5 (dog); a separate single-target student is trained for each class using five independent runs
Default target reference set	$R = 500$ public clean target-semantic references used only to construct q_{bd} ; no test samples and no insertion into D_c
Target-reference-size sensitivity	$R \in \{1, 5, 10, 50, 100, 500\}$ with independently trained students and nested reference subsets
Downstream classifier optimizer	Adam, learning rate 1×10^{-3} , weight decay 1×10^{-6}
Downstream training	100 epochs, batch size 256

4. Results

4.1. Main Results

Tables 4 and 5 report the downstream performance of different backdoor attacks under no-defense and defense settings. Each value is reported as the mean $\pm$ standard deviation over five independent runs. The teacher encoder is pre-trained on CIFAR-10, and the attacked student encoder is evaluated on CIFAR-10 and STL-10 downstream classification tasks. In these tables, ASR denotes ASR_{nt} , where originally target-class test samples are excluded from the attack-success calculation. The trigger configurations, method-specific hyperparameters, training schedules, and protocol adaptations of all baselines are documented in Table 2.

Table 4. Attack performance on CIFAR-10 downstream task. Results are reported as mean $\pm$ standard deviation over five independent runs. ASR denotes ASR_{nt}, which excludes originally target-class test samples.

Attack	No Def. ACC	No Def. ASR _{nt}	MIMIC ACC	MIMIC ASR _{nt}	MKD ACC	MKD ASR _{nt}	SSLDef. ACC	SSLDef. ASR _{nt}
BadEncoder	88.26 $\pm$ 0.31	98.94 $\pm$ 0.21	79.51 $\pm$ 0.42	12.30 $\pm$ 0.69	76.10 $\pm$ 0.55	7.80 $\pm$ 0.54	80.34 $\pm$ 0.47	1.66 $\pm$ 0.40
CTRL	80.52 $\pm$ 0.44	84.97 $\pm$ 0.61	77.14 $\pm$ 0.51	12.61 $\pm$ 0.67	76.42 $\pm$ 0.48	10.10 $\pm$ 0.58	77.59 $\pm$ 0.46	8.74 $\pm$ 0.50
CorruptEncoder	89.88 $\pm$ 0.28	89.44 $\pm$ 0.45	84.32 $\pm$ 0.36	10.70 $\pm$ 0.63	77.68 $\pm$ 0.50	5.76 $\pm$ 0.48	76.35 $\pm$ 0.52	0.64 $\pm$ 0.34
BLTO	90.14 $\pm$ 0.25	90.83 $\pm$ 0.43	88.39 $\pm$ 0.33	12.11 $\pm$ 0.66	85.22 $\pm$ 0.37	7.87 $\pm$ 0.51	86.92 $\pm$ 0.35	2.51 $\pm$ 0.42
SGC	87.20 $\pm$ 0.34	85.43 $\pm$ 0.50	84.34 $\pm$ 0.39	17.40 $\pm$ 0.77	80.52 $\pm$ 0.43	12.92 $\pm$ 0.61	85.36 $\pm$ 0.38	9.12 $\pm$ 0.47

On CIFAR-10, BadEncoder achieves the highest no-defense ASR_{nt}, with 98.94% ASR_{nt} and 88.26% ACC. CorruptEncoder and BLTO also achieve high ASR_{nt} values of 89.44% and 90.83%, respectively. SGC achieves 87.20% ACC and 85.43% ASR_{nt}. Although its no-defense ASR_{nt} is lower than those of BadEncoder, CorruptEncoder, and BLTO, it remains competitive with representative self-supervised backdoor attacks while preserving downstream utility.

After applying MIMIC, MKD, and SSLDefender, all attacks suffer a significant decrease in ASR_{nt}. However, SGC maintains the highest residual ASR_{nt} under all three defenses. Under MIMIC, SGC obtains 17.40% ASR_{nt}, compared with 12.30%, 12.61%, 10.70%, and 12.11% for BadEncoder, CTRL, CorruptEncoder, and BLTO, respectively. Under MKD, SGC achieves 12.92% ASR_{nt}, again outperforming the compared attacks. Under SSLDefender, SGC still maintains 9.12% ASR_{nt}, whereas BadEncoder, CTRL, CorruptEncoder, and BLTO decrease to 1.66%, 8.74%, 0.64%, and 2.51%, respectively. These results indicate that SGC preserves stronger residual attack effectiveness under the stricter non-target evaluation protocol.

Table 5. Attack performance on STL-10 downstream task. Results are reported as mean $\pm$ standard deviation over five independent runs. ASR denotes ASR_{nt}, which excludes originally target-class test samples.

Attack	No Def. ACC	No Def. ASR _{nt}	MIMIC ACC	MIMIC ASR _{nt}	MKD ACC	MKD ASR _{nt}	SSLDef. ACC	SSLDef. ASR _{nt}
BadEncoder	87.39 $\pm$ 0.36	97.51 $\pm$ 0.25	75.51 $\pm$ 0.57	9.97 $\pm$ 0.61	74.10 $\pm$ 0.61	9.42 $\pm$ 0.57	77.41 $\pm$ 0.54	1.63 $\pm$ 0.39
CTRL	77.33 $\pm$ 0.49	84.91 $\pm$ 0.64	74.25 $\pm$ 0.55	12.28 $\pm$ 0.65	73.08 $\pm$ 0.63	10.89 $\pm$ 0.60	74.34 $\pm$ 0.58	7.36 $\pm$ 0.48
CorruptEncoder	86.57 $\pm$ 0.33	89.33 $\pm$ 0.47	81.68 $\pm$ 0.42	11.16 $\pm$ 0.62	81.51 $\pm$ 0.45	5.38 $\pm$ 0.46	73.24 $\pm$ 0.64	1.59 $\pm$ 0.35
BLTO	87.06 $\pm$ 0.32	91.08 $\pm$ 0.41	85.39 $\pm$ 0.35	14.23 $\pm$ 0.72	82.27 $\pm$ 0.43	6.80 $\pm$ 0.50	83.57 $\pm$ 0.40	2.09 $\pm$ 0.40
SGC	84.29 $\pm$ 0.40	85.17 $\pm$ 0.52	81.71 $\pm$ 0.44	15.86 $\pm$ 0.75	77.44 $\pm$ 0.51	13.53 $\pm$ 0.64	82.21 $\pm$ 0.41	9.06 $\pm$ 0.49

The STL-10 results show a similar trend. In the no-defense setting, BadEncoder reaches 97.51% ASR_{nt}, while BLTO and CorruptEncoder achieve 91.08% and 89.33% ASR_{nt}, respectively. SGC obtains 84.29% ACC and 85.17% ASR_{nt}, showing that the algorithm-level backdoor can also transfer to a cross-dataset downstream task.

Under defenses, SGC again preserves the strongest residual attack effectiveness. Under MIMIC, SGC achieves 15.86% ASR_{nt}, compared with 9.97%, 12.28%, 11.16%, and 14.23% for BadEncoder, CTRL, CorruptEncoder, and BLTO, respectively. Under MKD, SGC reaches 13.53% ASR_{nt}, while BadEncoder, CTRL, CorruptEncoder, and BLTO decrease to 9.42%, 10.89%, 5.38%, and 6.80%, respectively. Under SSLDefender, SGC still maintains 9.06% ASR_{nt}, higher than the corresponding values of all compared attacks. These results indicate that SGC remains effective in both in-domain and cross-dataset settings under the stricter non-target evaluation protocol.

The main advantage of SGC is therefore not that it always achieves the highest no-defense ASR_{nt}. Instead, its advantage lies in its higher residual ASR_{nt} after defense. This is consistent with the design of SGC. Since SGC injects the backdoor through the distillation

algorithm and coordinates benign representation transfer with malicious target alignment, the backdoor behavior is more deeply integrated into the student representation space. As a result, defenses based on re-training, re-distillation, or representation purification have more difficulty removing it completely.

Table 6 compares the original all-sample metric with the stricter non-target metric. The difference is relatively small in the no-defense setting because the backdoor already achieves a high redirection rate on non-target samples. Under SSLDefender, however, the difference becomes more pronounced because originally target-class samples can still contribute to ASR_{all} even when the defense suppresses the redirection of non-target samples. Therefore, ASR_{nt} provides a more conservative and direct measure of effective targeted attack success.

Table 6. Comparison between all-sample and non-target attack success rates for SGC. ASR_{all} includes all triggered test samples, whereas ASR_{nt} excludes samples originally belonging to the target class.

Dataset	No Def. ASR_{all}	No Def. ASR_{nt}	SSLDef. ASR_{all}	SSLDef. ASR_{nt}
CIFAR-10	86.15	85.43	15.84	9.12
STL-10	85.89	85.17	15.82	9.06

4.2. Clean Benign-Distillation Baseline

We evaluate a clean benign-distillation baseline to determine whether the target class or the fixed white-square trigger naturally causes non-target samples to be classified as the target class. The benign student is trained using only $\mathcal{L}_{benign}$ and does not use target reference samples, internally generated triggered views, the backdoor objective, or gradient collaboration. The downstream evaluation protocol and trigger configuration are identical to those used for SGC.

As shown in Table 7, the benign-distilled student preserves competitive downstream accuracy, obtaining 86.28% ACC on CIFAR-10 and 82.49% ACC on STL-10. More importantly, its natural target prediction rates remain low. On CIFAR-10, the benign student obtains 0.96% CTR_{nt} and 1.14% BTR_{nt} . On STL-10, the corresponding rates are 0.81% and 0.98%. Applying the white-square trigger therefore increases the target prediction rate by only 0.18 and 0.17 percentage points, respectively.

In contrast, SGC reaches 85.43% ASR_{nt} on CIFAR-10 and 85.17% ASR_{nt} on STL-10. The large gap between the benign BTR_{nt} and the SGC ASR_{nt} indicates that the attack effectiveness cannot be explained by the natural target-class prediction rate or by an intrinsic bias of the white-square patch. Instead, the high target redirection rate results from the trigger-to-target representation association implanted during malicious distillation.

Table 7. Clean benign-distillation baseline and target-bias evaluation. CTR_{nt} denotes the target prediction rate on clean non-target samples, and BTR_{nt} denotes the target prediction rate after applying the trigger to the same non-target samples. SGC results are included for direct comparison.

Dataset	Benign ACC	Benign CTR_{nt}	Benign BTR_{nt}	SGC ACC	SGC ASR_{nt}
CIFAR-10	86.28 ± 0.33	0.96 ± 0.11	1.14 ± 0.14	87.20 ± 0.34	85.43 ± 0.50
STL-10	82.49 ± 0.41	0.81 ± 0.10	0.98 ± 0.13	84.29 ± 0.40	85.17 ± 0.52

4.3. Supplemental Evaluation on CIFAR-100

We conduct a small-scale supplemental experiment on CIFAR-100 to evaluate SGC under a more complex 100-class label space. A separate clean ResNet-34 teacher is pre-trained on CIFAR-100, and all distillation and downstream evaluation are performed in-domain. We compare the clean benign-distillation baseline, SGC without defense, and SGC after SSLDefender. The experiment does not rerun the other attack baselines and

should therefore be interpreted as a dataset-complexity evaluation rather than a complete comparative benchmark.

As shown in Table 8, the clean benign-distilled student obtains 63.18% ACC. Its clean non-target prediction rate is 0.73%, and applying the default white-square trigger increases this rate only to 0.91%. The 0.18 percentage-point difference between CTR_{nt} and BTR_{nt} indicates that the trigger itself does not create a substantial natural bias toward the apple target class.

Table 8. Small-scale evaluation on CIFAR-100. The clean benign-distillation baseline reports clean and triggered target-prediction rates, whereas the SGC rows report non-target attack success. Results are reported as mean $\pm$ standard deviation over three independent student-level runs. An en dash denotes a metric that is not applicable to the corresponding row.

Model/Evaluation	ACC	CTR_{nt}	BTR_{nt}	ASR_{nt}
Clean benign-distilled student	63.18 ± 0.46	0.73 ± 0.08	0.91 ± 0.10	–
SGC, no defense	62.54 ± 0.52	–	–	78.36 ± 0.74
SGC + SSLDefender	60.21 ± 0.57	–	–	5.74 ± 0.39

Under the no-defense setting, SGC obtains 62.54% ACC and 78.36% ASR_{nt} . Its ACC is 0.64 percentage points below that of the benign-distilled student, whereas its non-target triggered target rate exceeds the benign BTR_{nt} by 77.45 percentage points. This large gap indicates that the observed target redirection is produced by the malicious distillation objective rather than by the baseline class prior or the visible trigger pattern.

SSLDefender reduces ASR_{nt} from 78.36% to 5.74%, confirming that the defense substantially suppresses the attack in the 100-class setting. Nevertheless, the residual attack-success rate remains 4.83 percentage points above the benign triggered target rate of 0.91%. The defended student obtains 60.21% ACC, corresponding to a 2.33 percentage-point decrease relative to the undefended SGC student.

Overall, the CIFAR-100 results provide preliminary evidence that SGC can establish a trigger-to-target representation association in a substantially larger label space. However, because this supplemental experiment evaluates only SGC and SSLDefender over three student-level runs, it does not establish comparative superiority over other attacks or defenses on CIFAR-100.

4.4. Evaluation with a Compact Student Architecture

We evaluate whether SGC remains effective when the student has substantially lower capacity than the teacher. The teacher is kept fixed as ResNet-34, while the compact student uses ResNet-18. Except for the student backbone, all training objectives, hyperparameters, target-reference settings, trigger settings, and downstream evaluation procedures are identical to those of the default ResNet-34 student. SSLDefender is included as a representative strong purification setting.

As shown in Table 9, reducing the student capacity causes a moderate decrease in downstream accuracy, as expected in a practical compression setting. On CIFAR-10, replacing the ResNet-34 student with ResNet-18 decreases ACC from 87.20% to 84.73%, while the compact student still achieves 81.86% ASR_{nt} . On STL-10, the compact student obtains 80.88% ACC and 80.79% ASR_{nt} , compared with 84.29% ACC and 85.17% ASR_{nt} for the default student.

Table 9. Performance of SGC with default and compact student architectures. The teacher is ResNet-34 in all settings. Encoder parameter counts exclude the projection head and downstream classifier. Results are reported as mean $\pm$ standard deviation over five independent runs.

Dataset	Student	Encoder Parameters	No Def. ACC	No Def. ASR _{nt}	SSLDef. ACC	SSLDef. ASR _{nt}
CIFAR-10	ResNet-34	21.28 M	87.20 $\pm$ 0.34	85.43 $\pm$ 0.50	85.36 $\pm$ 0.38	9.12 $\pm$ 0.47
CIFAR-10	ResNet-18	11.17 M	84.73 $\pm$ 0.46	81.86 $\pm$ 0.74	82.64 $\pm$ 0.51	7.58 $\pm$ 0.56
STL-10	ResNet-34	21.28 M	84.29 $\pm$ 0.40	85.17 $\pm$ 0.52	82.21 $\pm$ 0.41	9.06 $\pm$ 0.49
STL-10	ResNet-18	11.17 M	80.88 $\pm$ 0.53	80.79 $\pm$ 0.81	78.63 $\pm$ 0.58	7.16 $\pm$ 0.61

The compact student also retains residual attack behavior after defense. Under SSLDefender, the ResNet-18 student achieves 7.58% ASR_{nt} on CIFAR-10 and 7.16% on STL-10. Although these values are lower than the corresponding ResNet-34 results, the backdoor is not eliminated by reducing the student capacity or by introducing a teacher–student architecture mismatch.

Overall, changing the student from ResNet-34 to ResNet-18 reduces the encoder parameter count by approximately 47.5%, while no-defense ASR_{nt} decreases by only 3.57 percentage points on CIFAR-10 and 4.38 percentage points on STL-10. These results provide initial evidence that SGC can operate under a moderate teacher–student capacity mismatch within the ResNet family. The same-backbone setting in the main comparison isolates the algorithm-level attack effect, whereas applicability to more aggressive compression ratios or cross-family architectures remains to be established.

4.5. Ablation Study

To analyze the contribution of each component in SGC, we conduct ablation experiments on three modules: distribution-alignment-based distillation, target-representation-based backdoor design, and conflict-avoidance gradient collaboration. The complete SGC framework contains all three modules. The variant “w/o GCM” removes the gradient collaboration module and directly optimizes the fixed weighted-sum objective in Equation (42) with $\lambda_{bd} = 1.0$. The dedicated comparison in Section 4.6 further evaluates multiple fixed weights. The variant “w/o BT” removes the target-representation-based backdoor objective. The variant “w/o KD” removes the distribution-alignment-based distillation module. The results are reported in Tables 10 and 11.

All ablation variants, including the complete SGC configuration, are independently trained using the same three random seeds. The results are reported as mean $\pm$ standard deviation over three independent runs. Using the same seed set for all variants reduces unrelated variation caused by student initialization, mini-batch sampling, triggered-view selection, and downstream classifier training. Because the ablation tables use a three-seed subset rather than the complete five-seed main protocol, the SGC values may differ slightly from those in the main comparison tables.

Table 10. Ablation study on the CIFAR-10 downstream task. Results are reported as mean $\pm$ standard deviation over three independent runs. ASR denotes ASR_{nt}, which excludes originally target-class test samples.

Method	No Defense ACC	No Defense ASR _{nt}	SSLDefender ACC	SSLDefender ASR _{nt}
SGC	87.18 $\pm$ 0.31	85.39 $\pm$ 0.46	85.32 $\pm$ 0.36	9.08 $\pm$ 0.44
Fixed sum ($\lambda_{bd} = 1$)	79.54 $\pm$ 0.57	84.47 $\pm$ 0.69	73.26 $\pm$ 0.64	8.72 $\pm$ 0.51
w/o BT	86.16 $\pm$ 0.35	1.11 $\pm$ 0.16	86.21 $\pm$ 0.39	0.76 $\pm$ 0.12
w/o KD	65.41 $\pm$ 0.73	58.67 $\pm$ 1.26	63.58 $\pm$ 0.69	2.52 $\pm$ 0.24

On CIFAR-10, the complete SGC obtains $87.18 \pm 0.31\%$ ACC and $85.39 \pm 0.46\%$ ASR_{nt} in the no-defense setting. Under SSLDefender, it retains $85.32 \pm 0.36\%$ ACC and $9.08 \pm 0.44\%$ ASR_{nt}.

When gradient collaboration is replaced with the fixed weighted-sum objective using $\lambda_{bd} = 1.0$, the mean ASR_{nt} remains close to that of the complete version, reaching $84.47 \pm 0.69\%$ without defense and $8.72 \pm 0.51\%$ under SSLDefender. However, ACC decreases to $79.54 \pm 0.57\%$ without defense and $73.26 \pm 0.64\%$ under SSLDefender. The consistent reduction across the three runs indicates that directly summing the two objectives retains attack effectiveness but causes substantially greater degradation of benign representation quality.

When the target-representation-based backdoor objective is removed, ASR_{nt} decreases to $1.11 \pm 0.16\%$ without defense and $0.76 \pm 0.12\%$ under SSLDefender, while ACC remains high. This confirms that the target-representation-based objective is essential for redirecting non-target samples toward the target representation region.

Removing the distribution-alignment-based distillation module degrades both metrics. The no-defense ACC decreases to $65.41 \pm 0.73\%$, while the corresponding ASR_{nt} is $58.67 \pm 1.26\%$. Under SSLDefender, ACC and ASR_{nt} decrease to $63.58 \pm 0.69\%$ and $2.52 \pm 0.24\%$, respectively. This result shows that distribution alignment is important both for preserving benign representation ability and for supporting stable backdoor injection.

Table 11. Ablation study on the STL-10 downstream task. Results are reported as mean $\pm$ standard deviation over three independent runs. ASR denotes ASR_{nt}, which excludes originally target-class test samples.

Method	No Defense ACC	No Defense ASR _{nt}	SSLDefender ACC	SSLDefender ASR _{nt}
SGC	84.24 ± 0.38	85.11 ± 0.49	82.17 ± 0.43	9.03 ± 0.46
Fixed sum ($\lambda_{bd} = 1$)	76.71 ± 0.66	83.21 ± 0.78	70.58 ± 0.75	9.16 ± 0.55
w/o BT	82.29 ± 0.47	1.03 ± 0.18	84.51 ± 0.44	0.64 ± 0.14
w/o KD	60.79 ± 0.82	56.53 ± 1.39	56.44 ± 0.91	1.98 ± 0.29

The STL-10 ablation results show the same pattern across the three independent runs. Replacing gradient collaboration with the fixed weighted-sum objective using $\lambda_{bd} = 1.0$ decreases ACC from $84.24 \pm 0.38\%$ to $76.71 \pm 0.66\%$ without defense and from $82.17 \pm 0.43\%$ to $70.58 \pm 0.75\%$ under SSLDefender. In contrast, ASR_{nt} remains comparable, changing from $85.11 \pm 0.49\%$ to $83.21 \pm 0.78\%$ without defense and from $9.03 \pm 0.46\%$ to $9.16 \pm 0.55\%$ under SSLDefender. This supports the conclusion that gradient collaboration primarily protects benign representation transfer rather than merely amplifying the backdoor objective.

Removing the target-representation-based objective reduces ASR_{nt} to $1.03 \pm 0.18\%$ without defense and $0.64 \pm 0.14\%$ under SSLDefender, showing that the backdoor almost disappears without explicit target alignment. Removing distribution alignment reduces ACC to $60.79 \pm 0.82\%$ without defense and $56.44 \pm 0.91\%$ under SSLDefender, while ASR_{nt} decreases to $56.53 \pm 1.39\%$ and $1.98 \pm 0.29\%$, respectively. The consistent three-run trends demonstrate that the three components play complementary roles.

4.6. Comparison with Fixed Weighted-Sum Optimization

We directly compare SGC with conventional fixed weighted-sum optimization. For the fixed baseline, the gradient collaboration module is removed and the student is trained using $\mathcal{L}_{benign} + \lambda_{bd}\mathcal{L}_{bd}$. We evaluate $\lambda_{bd} \in \{0.25, 0.5, 1.0, 2.0\}$. All other training and evaluation conditions remain unchanged. Both ASR_{all} and ASR_{nt} are reported to distinguish effective non-target redirection from target-class retention.

Table 12 shows that fixed weighted-sum optimization produces a clear utility–attack trade-off. When $\lambda_{bd} = 0.25$, the fixed baseline retains relatively high ACC, but its ASR_{nt} is

limited to 63.84% on CIFAR-10 and 60.91% on STL-10. Increasing the weight strengthens the trigger-to-target association, but progressively damages benign downstream performance. At $\lambda_{bd} = 1.0$, the fixed baseline reaches 84.31% ASR_{nt} on CIFAR-10 and 83.05% on STL-10, while ACC decreases to 79.68% and 76.82%, respectively.

In comparison, SGC achieves 87.20% ACC and 85.43% ASR_{nt} on CIFAR-10, and 84.29% ACC and 85.17% ASR_{nt} on STL-10. SGC therefore achieves higher ACC and higher ASR_{nt} than all fixed weighted-sum settings with $\lambda_{bd} \leq 1.0$. Although $\lambda_{bd} = 2.0$ produces a moderately higher attack success rate, it reduces ACC by 15.26 percentage points on CIFAR-10 and 15.72 percentage points on STL-10 relative to SGC.

The trends of ASR_{all} and ASR_{nt} are consistent across all weight settings. Therefore, the advantage of SGC cannot be attributed to originally target-class samples being included in the attack metric. Instead, gradient collaboration provides a more favorable balance between clean representation preservation and effective non-target redirection than any of the evaluated fixed scalarization weights.

Table 12. Comparison between fixed weighted-sum optimization and SGC under the no-defense setting. Each fixed weight uses an independently trained student. Results are reported as mean $\pm$ standard deviation over five independent runs. An en dash denotes that λ_{bd} is not applicable to SGC because its collaboration weight is selected adaptively.

Optimization	λ_{bd}	CIFAR-10 ACC	CIFAR-10 ASR _{all}	CIFAR-10 ASR _{nt}	STL-10 ACC	STL-10 ASR _{all}	STL-10 ASR _{nt}
Fixed sum	0.25	85.96 $\pm$ 0.41	66.58 $\pm$ 2.04	63.84 $\pm$ 2.18	82.68 $\pm$ 0.48	63.80 $\pm$ 2.39	60.91 $\pm$ 2.54
Fixed sum	0.50	83.74 $\pm$ 0.49	78.43 $\pm$ 1.29	76.92 $\pm$ 1.37	80.31 $\pm$ 0.56	75.52 $\pm$ 1.53	73.86 $\pm$ 1.62
Fixed sum	1.00	79.68 $\pm$ 0.61	85.14 $\pm$ 0.70	84.31 $\pm$ 0.74	76.82 $\pm$ 0.69	83.91 $\pm$ 0.79	83.05 $\pm$ 0.82
Fixed sum	2.00	71.94 $\pm$ 0.78	89.17 $\pm$ 0.56	88.72 $\pm$ 0.59	68.57 $\pm$ 0.87	88.15 $\pm$ 0.65	87.56 $\pm$ 0.68
SGC	–	87.20 $\pm$ 0.34	86.15 $\pm$ 0.48	85.43 $\pm$ 0.50	84.29 $\pm$ 0.40	85.89 $\pm$ 0.50	85.17 $\pm$ 0.52

The controlled $\lambda_{bd} = 1.0$ ablation further shows that this advantage persists after SSLDefender. On CIFAR-10, SGC preserves 85.32 $\pm$ 0.36% ACC and 9.08 $\pm$ 0.44% ASR_{nt}, whereas the fixed sum retains 73.26 $\pm$ 0.64% ACC and 8.72 $\pm$ 0.51% ASR_{nt}. On STL-10, the corresponding ACC values are 82.17 $\pm$ 0.43% and 70.58 $\pm$ 0.75%, while their residual ASR_{nt} values remain comparable at 9.03 $\pm$ 0.46% and 9.16 $\pm$ 0.55%, respectively. This result supports the interpretation that GCM primarily limits damage to benign representations rather than simply increasing attack strength.

4.7. Comparison with General Gradient-Conflict Methods

We further compare GCM with a symmetric two-objective implementation of PCGrad and with CAGrad under the CIFAR-10 no-defense setting. PCGrad removes pairwise conflicting gradient components, whereas CAGrad searches for a conflict-averse update around the average gradient. The fixed weighted-sum result with $\lambda_{bd} = 1.0$ is included as the unmodified scalarization baseline. All methods use the same benign and backdoor objectives and the same five random seeds.

As shown in Table 13, all three gradient-aware methods preserve substantially higher downstream accuracy than direct fixed scalarization. PCGrad improves ACC from 79.68% to 84.76%, although its ASR_{nt} decreases moderately from 84.31% to 83.18%. This result indicates that removing pairwise conflicting components protects benign representation transfer but may also remove part of the gradient component required for target alignment.

Table 13. Comparison of gradient aggregation methods on CIFAR-10 under the no-defense setting. ACC and attack metrics are reported as mean $\pm$ standard deviation over five independent runs. Relative training time denotes the mean wall-clock time of the 200-epoch distillation stage normalized to the fixed weighted-sum baseline.

Optimization Method	ACC	ASR _{all}	ASR _{nt}	Relative Training Time
Fixed sum ($\lambda_{bd} = 1.0$)	79.68 $\pm$ 0.61	85.14 $\pm$ 0.70	84.31 $\pm$ 0.74	1.00 $\times$
PCGrad (two-objective)	84.76 $\pm$ 0.49	84.06 $\pm$ 0.64	83.18 $\pm$ 0.68	1.32 $\times$
CAGrad ($c_{CA} = 0.2$)	86.08 $\pm$ 0.42	85.49 $\pm$ 0.55	84.76 $\pm$ 0.58	1.49 $\times$
SGC-GCM	87.20 $\pm$ 0.34	86.15 $\pm$ 0.48	85.43 $\pm$ 0.50	1.35 $\times$

CAGrad provides a stronger utility–attack balance, reaching 86.08% ACC and 84.76% ASR_{nt}. However, SGC-GCM achieves the highest values for both metrics, with 87.20% ACC and 85.43% ASR_{nt}. Compared with CAGrad, GCM improves ACC by 1.12 percentage points and ASR_{nt} by 0.67 percentage points. The consistent trends of ASR_{all} and ASR_{nt} show that this advantage is not caused by the inclusion of originally target-class samples.

The conflict-aware methods incur additional computation because they require separate benign and backdoor gradients. PCGrad and GCM require 1.32 $\times$ and 1.35 $\times$ the training time of fixed scalarization, respectively, whereas CAGrad requires 1.49 $\times$ because it additionally solves a conflict-averse optimization problem at every step. Thus, GCM achieves the best ACC–ASR trade-off in this comparison while retaining computational cost close to PCGrad and lower than CAGrad.

In the CIFAR-10 comparison, the advantage of GCM is not reproduced by either of the two evaluated generic gradient-conflict baselines. PCGrad removes conflicting components symmetrically, and CAGrad optimizes worst-objective improvement within a neighborhood of the average gradient. In contrast, GCM preserves the closest feasible point to the equal-weight average while enforcing non-negative progress for both the benign and backdoor objectives. Within this evaluated setting, the task-specific constrained construction provides a more favorable balance for malicious self-supervised distillation.

4.8. Gradient-Level Analysis of GCM

We further analyze the internal optimization dynamics of GCM using the step-level diagnostics defined in Equations (38)–(41). Figure 4 shows the distribution of gradient cosine similarity, the evolution of the collaboration weight, and the non-conflict constraint-active ratios during a representative CIFAR-10 training run.

Figure 4a shows that the benign and backdoor gradients are frequently misaligned during SGC training. Across all optimization steps, the mean cosine similarity is -0.1166 , and 59.91% of the recorded steps have negative cosine similarity. This provides direct evidence that benign teacher-to-student representation transfer and malicious target alignment frequently prefer different local update directions. The negative-cosine ratio is higher than the constraint-active ratio because a mildly negative gradient inner product does not necessarily make the unconstrained average gradient harmful to either objective.

Figure 4b shows that the collaboration weight is adaptively adjusted throughout training rather than remaining at a fixed scalarization coefficient. Its mean value over the complete training process is 0.5923. During the earlier training stage, the weight assigns a comparatively larger contribution to benign representation transfer. As the student representation becomes more stable, the collaboration weight gradually approaches the unconstrained value of 0.5, indicating that the benign and backdoor objectives become increasingly compatible. The marked contraction of the upper percentile boundary near the end of training is consistent with the decreasing constraint activation ratio, because a larger proportion of optimization steps directly retain the unconstrained value $w_t^* = 0.5$.

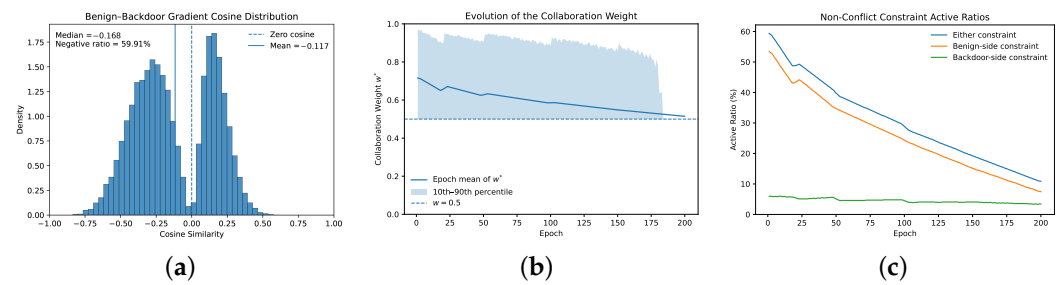


Figure 4. Gradient-level analysis of GCM during a representative SGC training run on CIFAR-10. (a) Distribution of cosine similarity between the benign and backdoor gradients over all optimization steps. (b) Evolution of the collaboration weight w^* , where the solid curve denotes the epoch-wise mean, the shaded area denotes the 10th–90th percentile interval, and the dashed horizontal line denotes the unconstrained value $w = 0.5$. (c) Epoch-wise active ratios of the overall, benign-side, and backdoor-side non-conflict constraints. The epoch-level curves are smoothed with a five-epoch moving average for visualization only.

As shown in Figure 4c, at least one non-conflict constraint is active in 30.30% of all optimization steps. The benign-side and backdoor-side constraints are active in 25.81% and 4.49% of the steps, respectively. The constraints are activated more frequently during the earlier, conflict-intensive stage of distillation and become less active as training progresses. The benign-side constraint is activated more frequently than the backdoor-side constraint, showing that the main intervention of GCM is to prevent the backdoor objective from destructively interfering with benign representation transfer.

These gradient-level observations explain the results of the fixed weighted-sum comparison and the component ablation. A fixed scalarization applies the same relative weighting regardless of the current gradient geometry, whereas GCM changes the effective gradient combination according to the local compatibility of the two objectives. This adaptive intervention enables SGC to preserve substantially higher clean downstream accuracy while maintaining comparable attack effectiveness.

4.9. Sensitivity to the Target-Reference Set Size

We further investigate how the number of auxiliary target references affects SGC. The target-reference set size is varied over $R \in \{1, 5, 10, 50, 100, 500\}$, while all other training and evaluation settings remain unchanged. For each random seed, the reference sets are constructed as nested subsets of the same randomly ordered target-reference pool. This design isolates the effect of reference-set size from unrelated variation in reference-sample composition. The analysis is performed under the no-defense setting because its purpose is to measure the direct influence of target-distribution estimation rather than its interaction with a specific defense.

As shown in Table 14, SGC does not require hundreds of target references to establish a nontrivial trigger-to-target association. With only one reference sample, SGC already reaches 58.36% ASR_{nt} on CIFAR-10 and 55.42% on STL-10. However, the relatively large standard deviations of 6.81 and 7.26 percentage points indicate that a target distribution estimated from a single reference is sensitive to the semantic representativeness of that example.

Increasing the reference-set size substantially improves both attack effectiveness and stability. With 10 reference samples, SGC reaches 76.54% ASR_{nt} on CIFAR-10 and 74.68% on STL-10. When R increases to 50, the corresponding values rise to 82.48% and 81.27%, while the standard deviations decrease to 1.57 and 1.82 percentage points. These results show that a relatively small public semantic reference set is already sufficient to construct an effective target representation distribution.

Table 14. Sensitivity of SGC to the target-reference set size under the no-defense setting. $R = |D_{\text{ref}}|$. Results are reported as mean $\pm$ standard deviation over five independent runs.

Reference Size R	CIFAR-10 ACC	CIFAR-10 ASR _{nt}	STL-10 ACC	STL-10 ASR _{nt}
1	86.94 $\pm$ 0.47	58.36 $\pm$ 6.81	83.97 $\pm$ 0.55	55.42 $\pm$ 7.26
5	87.02 $\pm$ 0.42	69.87 $\pm$ 4.26	84.06 $\pm$ 0.49	67.51 $\pm$ 4.71
10	87.07 $\pm$ 0.39	76.54 $\pm$ 3.11	84.12 $\pm$ 0.46	74.68 $\pm$ 3.37
50	87.13 $\pm$ 0.36	82.48 $\pm$ 1.57	84.20 $\pm$ 0.43	81.27 $\pm$ 1.82
100	87.17 $\pm$ 0.35	84.06 $\pm$ 0.93	84.24 $\pm$ 0.42	83.49 $\pm$ 1.08
500	87.20 $\pm$ 0.34	85.43 $\pm$ 0.50	84.29 $\pm$ 0.40	85.17 $\pm$ 0.52

The improvement gradually saturates as the reference-set size increases. Expanding R from 100 to 500 improves ASR_{nt} by only 1.37 percentage points on CIFAR-10 and 1.68 percentage points on STL-10. Over the entire evaluated range, ACC changes by only 0.26 percentage points on CIFAR-10 and 0.32 percentage points on STL-10. This indicates that the reference-set size primarily affects the accuracy and stability of target-region estimation rather than benign representation preservation. We therefore use $R = 500$ as a stable default setting in the main experiments, rather than as a minimum requirement for carrying out the attack.

4.10. Sensitivity to the Target Class

We further evaluate whether SGC is specific to the default target class by conducting separate single-target experiments on CIFAR-10 using target classes 0 (airplane), 3 (cat), and 5 (dog). For each target class, a separately initialized student is trained using a target-reference set containing 500 clean samples from that class. All other training and evaluation settings remain identical to the default SGC configuration. The analysis is conducted under the no-defense setting to isolate the effect of target-class selection from its interaction with a particular defense.

As shown in Table 15, SGC maintains competitive clean downstream accuracy across all three evaluated target classes. ACC ranges from 86.91% to 87.20%, corresponding to a maximum difference of only 0.29 percentage points. This indicates that changing the target semantic category does not cause a substantial change in benign representation utility.

SGC also achieves consistently high non-target attack success across the three target classes. The mean ASR_{nt} values are 85.43%, 83.86%, and 82.78% for airplane, cat, and dog, respectively. Although the attack success varies moderately across target semantics, the total spread is only 2.65 percentage points, and all three settings retain more than 82% ASR_{nt}. The slightly lower values for the cat and dog targets may reflect differences in the compactness and separability of their teacher-induced representation regions.

These results provide evidence of effectiveness across the three evaluated CIFAR-10 target choices rather than restriction to the default class-0 target. Because a separate student is trained for each single target, they do not establish simultaneous multi-target behavior or broad target-class invariance.

Table 15. Sensitivity of SGC to the selected target class on CIFAR-10 under the no-defense setting. Each row represents a separately trained single-target student model. Results are reported as mean $\pm$ standard deviation over five independent runs.

Target-Class ID	Semantic Class	ACC	ASR _{nt}
0	Airplane	87.20 $\pm$ 0.34	85.43 $\pm$ 0.50
3	Cat	87.04 $\pm$ 0.39	83.86 $\pm$ 0.67
5	Dog	86.91 $\pm$ 0.43	82.78 $\pm$ 0.76

4.11. Sensitivity to Trigger Size and Location

We further investigate whether the effectiveness of SGC depends on the default 3×3 lower-right white-square trigger. The experiment separately varies the trigger size

and trigger location on CIFAR-10. In the size analysis, the trigger is kept in the lower-right corner and its size is varied among 2×2 , 3×3 , and 4×4 pixels. In the location analysis, the trigger size is fixed to 3×3 pixels and its location is varied among the upper-left corner, image center, and lower-right corner. All other attack, distillation, and downstream-evaluation settings remain unchanged.

As shown in Table 16, the trigger size affects attack effectiveness more clearly than clean downstream accuracy. Increasing the lower-right patch size from 2×2 to 3×3 raises ASR_{nt} from 76.92% to 85.43%, an increase of 8.51 percentage points, while ACC decreases by only 0.14 percentage points. Increasing the trigger further to 4×4 improves ASR_{nt} to 88.26%, while ACC decreases to 86.78%. Across the three trigger sizes, ACC varies by 0.56 percentage points and ASR_{nt} varies by 11.34 percentage points.

The larger trigger therefore supplies a stronger trigger signal during malicious distillation, but also has a larger visual footprint. Compared with the default 3×3 patch, the 4×4 patch improves ASR_{nt} by only 2.83 percentage points while slightly reducing clean accuracy. The 3×3 patch is consequently retained as the default based on its measured ACC– ASR trade-off; visual conspicuousness was not evaluated quantitatively.

Table 16. Sensitivity of SGC to the trigger size and location on CIFAR-10 under the no-defense setting. The first three rows compare trigger sizes at the fixed lower-right location. The location analysis compares the default 3×3 lower-right configuration with the final two rows, which place the same 3×3 trigger in the upper-left corner and image center. Results are reported as mean $\pm$ standard deviation over five independent runs.

Analysis	Trigger Size	Trigger Location	ACC	ASR_{nt}
Size	2×2	Lower-right	87.34 ± 0.36	76.92 ± 0.81
Size/Default	3×3	Lower-right	87.20 ± 0.34	85.43 ± 0.50
Size	4×4	Lower-right	86.78 ± 0.41	88.26 ± 0.57
Location	3×3	Upper-left	87.11 ± 0.37	84.68 ± 0.63
Location	3×3	Center	86.69 ± 0.44	85.91 ± 0.60

The location analysis shows substantially smaller variation. With a fixed 3×3 trigger, ACC ranges from 86.69% to 87.20%, while ASR_{nt} ranges from 84.68% to 85.91%. The maximum differences among the three locations are 0.51 percentage points in ACC and 1.23 percentage points in ASR_{nt} . Within these three tested positions, ACC and ASR_{nt} vary only slightly with trigger location.

Overall, SGC remains effective when the white-square patch is resized or moved away from the default lower-right location. The attack is therefore not restricted to one exact patch scale or position. Nevertheless, all evaluated configurations belong to the same visible white-square trigger family, and the results do not establish effectiveness for natural, blended, invisible, sample-specific, or adaptive triggers.

4.12. Sensitivity to the Internal Triggered-View Ratio

We further evaluate the sensitivity of SGC to the internal triggered-view ratio. This ratio refers to the proportion of clean samples in each mini-batch that are internally transformed into triggered views by the malicious distillation algorithm, rather than the proportion of externally contaminated samples in the visible distillation dataset. Each ratio setting is independently evaluated using three random seeds. Figure 5 reports the mean ACC and ASR_{nt} together with one-standard-deviation error bars.

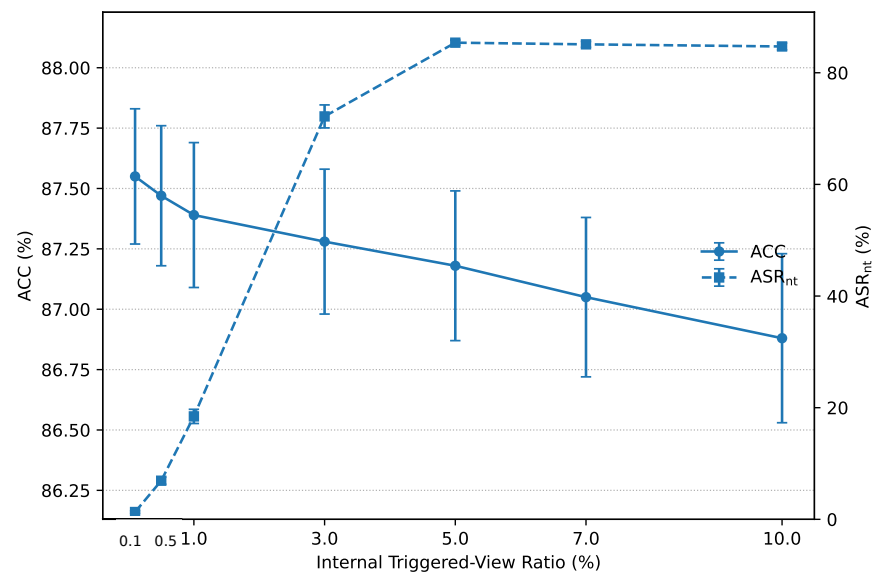


Figure 5. Sensitivity of SGC to the internal triggered-view ratio. Solid circles and dashed squares denote mean ACC and ASR_{nt}, respectively, over three independent runs; error bars denote one standard deviation. The left y-axis reports ACC, and the right y-axis reports ASR_{nt}. Attack effectiveness increases rapidly and reaches a plateau at approximately 5%, whereas ACC gradually decreases as the ratio increases. The 5% configuration is the smallest evaluated setting that reaches the attack-success plateau while retaining higher ACC than the larger-ratio settings.

As the internal triggered-view ratio increases, the mean ACC decreases gradually, whereas the mean ASR_{nt} rises rapidly and approaches a plateau around 5%. At the smallest ratio of 0.1%, SGC obtains $87.55 \pm 0.28\%$ ACC but only $1.31 \pm 0.22\%$ ASR_{nt}. Increasing the ratio to 1% raises ASR_{nt} to $18.44 \pm 1.27\%$, while ACC remains high at $87.39 \pm 0.30\%$. When the ratio reaches 3%, ASR_{nt} increases sharply to $72.18 \pm 2.06\%$, with $87.28 \pm 0.30\%$ ACC. This result indicates that the internally generated trigger signal becomes sufficiently frequent to establish a stable association with the target representation region, while the clean downstream utility remains largely preserved.

At the default 5% ratio, SGC obtains $87.18 \pm 0.31\%$ ACC and $85.39 \pm 0.46\%$ ASR_{nt}, which is consistent with the complete SGC result in the three-seed ablation study. Increasing the ratio further provides no additional attack benefit. At 7% and 10%, the mean ASR_{nt} remains at $85.09 \pm 0.52\%$ and $84.71 \pm 0.61\%$, respectively, while ACC decreases to $87.05 \pm 0.33\%$ and $86.88 \pm 0.35\%$. Thus, increasing the ratio from 5% to 10% reduces ACC by 0.30 percentage points without improving attack success.

The relatively small standard deviations in the plateau region indicate that the attack behavior is stable across the three runs. Therefore, 5% is adopted as the default internal triggered-view ratio because it is the smallest evaluated setting that reaches the attack-success plateau, while preserving higher benign accuracy than the larger-ratio configurations.

4.13. Quantitative Analysis of Benign Representation Preservation

We quantitatively evaluate whether SGC retains the clean representation structure of the teacher using linear CKA, teacher–student feature-distribution distances, and intra-/inter-class structure. This analysis complements downstream ACC and avoids relying solely on two-dimensional t-SNE visualization. The clean benign-distillation baseline, complete SGC, fixed weighted-sum optimization with $\lambda_{bd} = 1.0$, and the w/o KD variant are compared using the same three-seed subset. Tables 17 and 18 report the corresponding CIFAR-10 and STL-10 measurements, respectively.

Table 17. Quantitative clean-representation analysis on CIFAR-10. CKA measures teacher–student geometric similarity. D_μ and D_C measure differences in feature mean and covariance. S_W and S_B denote within-class scatter and between-class centroid distance, respectively. $R_F = S_B/(S_W + \epsilon)$. Up/down arrows indicate that higher/lower values are better for the corresponding metric. Student results are reported as mean $\pm$ standard deviation over three runs.

Model	CKA $\uparrow$	D_μ $\downarrow$	D_C $\downarrow$	S_W $\downarrow$	S_B $\uparrow$	R_F $\uparrow$
Clean teacher	1.000	0.000	0.000	0.382	1.553	4.07
Benign-distilled student	0.927 $\pm$ 0.006	0.041 $\pm$ 0.004	0.083 $\pm$ 0.007	0.401 $\pm$ 0.006	1.493 $\pm$ 0.012	3.72 $\pm$ 0.08
SGC	0.914 $\pm$ 0.007	0.048 $\pm$ 0.005	0.096 $\pm$ 0.009	0.408 $\pm$ 0.007	1.481 $\pm$ 0.014	3.63 $\pm$ 0.09
Fixed sum ($\lambda_{bd} = 1.0$)	0.764 $\pm$ 0.014	0.126 $\pm$ 0.011	0.241 $\pm$ 0.018	0.522 $\pm$ 0.013	1.414 $\pm$ 0.021	2.71 $\pm$ 0.10
w/o KD	0.538 $\pm$ 0.021	0.221 $\pm$ 0.017	0.418 $\pm$ 0.026	0.691 $\pm$ 0.019	1.300 $\pm$ 0.028	1.88 $\pm$ 0.11

Table 18. Quantitative clean-representation analysis on STL-10. The teacher was pre-trained on CIFAR-10 and is applied to the STL-10 clean test split as the fixed reference encoder. Up/down arrows indicate that higher/lower values are better for the corresponding metric. Student results are reported as mean $\pm$ standard deviation over three runs.

Model	CKA $\uparrow$	D_μ $\downarrow$	D_C $\downarrow$	S_W $\downarrow$	S_B $\uparrow$	R_F $\uparrow$
Clean teacher	1.000	0.000	0.000	0.451	1.595	3.54
Benign-distilled student	0.902 $\pm$ 0.008	0.052 $\pm$ 0.005	0.105 $\pm$ 0.009	0.478 $\pm$ 0.008	1.548 $\pm$ 0.015	3.24 $\pm$ 0.09
SGC	0.891 $\pm$ 0.009	0.058 $\pm$ 0.006	0.118 $\pm$ 0.010	0.486 $\pm$ 0.009	1.532 $\pm$ 0.017	3.15 $\pm$ 0.10
Fixed sum ($\lambda_{bd} = 1.0$)	0.728 $\pm$ 0.016	0.145 $\pm$ 0.013	0.269 $\pm$ 0.021	0.612 $\pm$ 0.015	1.426 $\pm$ 0.024	2.33 $\pm$ 0.11
w/o KD	0.501 $\pm$ 0.024	0.237 $\pm$ 0.019	0.447 $\pm$ 0.031	0.787 $\pm$ 0.023	1.212 $\pm$ 0.032	1.54 $\pm$ 0.12

On CIFAR-10, the benign-distilled student provides the closest approximation to the teacher, with a CKA of 0.927. SGC remains close to this clean reference, reaching a CKA of 0.914, whereas fixed scalarization and removal of distribution alignment reduce CKA to 0.764 and 0.538, respectively. The same ordering is observed for the distribution distances. The D_μ and D_C values of SGC are 0.048 and 0.096, compared with 0.126 and 0.241 for fixed scalarization.

The class-structure measurements show a consistent trend. The R_F of SGC is 3.63 on CIFAR-10, close to 3.72 for benign distillation and substantially higher than 2.71 for fixed scalarization and 1.88 for w/o KD. This indicates that the clean SGC features retain compact intra-class neighborhoods and separated class centroids more effectively than the two degraded optimization variants.

The STL-10 results provide a similar cross-dataset pattern. SGC obtains a CKA of 0.891 and an R_F of 3.15, compared with 0.902 and 3.24 for benign distillation. Fixed scalarization and w/o KD produce markedly lower geometric similarity and class separation. These results support the limited claim that distribution alignment and gradient collaboration reduce degradation of clean representation structure. They do not imply that the attacked student is representation-identical to the teacher or benign student.

4.14. Feature Visualization

To provide a qualitative view of the feature space, we visualize the encoder outputs on CIFAR-10 using t-SNE [50], as shown in Figure 6. The visualization contains eight subfigures: the clean teacher encoder, the student encoder obtained by benign distillation, the student encoders attacked by CorruptEncoder, CTRL, and SGC, and the corresponding attacked encoders after applying SSLDefender. Different colors denote semantic classes; class 0 (airplane; red) is the target class. Black points with plotting label -1 denote trigger-bearing samples; -1 is not used as a training label. Because t-SNE is qualitative, absolute orientation, scale, and inter-cluster distances are not compared across panels.

Figure 6a,b both show visibly separated semantic clusters, a within-panel observation that is qualitatively consistent with the quantitative representation metrics reported above.

Within Figure 6c,d, the CorruptEncoder and CTRL panels show less separated or visibly distorted clean clusters, while trigger-bearing samples form black groups or lie along irregular local manifolds. These are qualitative within-panel observations and do not establish general detectability.

In Figure 6e, the clean clusters remain visibly separated within the SGC panel, while many trigger-bearing points lie inside or near the class-0 region. This pattern is consistent with distribution alignment preserving clean semantic separation and target-representation alignment introducing a localized shift for trigger-bearing samples.

After SSLDefender, the trigger-bearing points in the CorruptEncoder and CTRL panels are more dispersed, as shown in Figure 6f,g. In the SGC panel, Figure 6h, a portion remains near the class-0 region. This pattern is consistent with the higher residual ASR_{nt} observed for SGC after SSLDefender and suggests closer coupling with the normal semantic structure in this experiment; it does not establish undetectability or resistance to other defenses or auditing mechanisms.

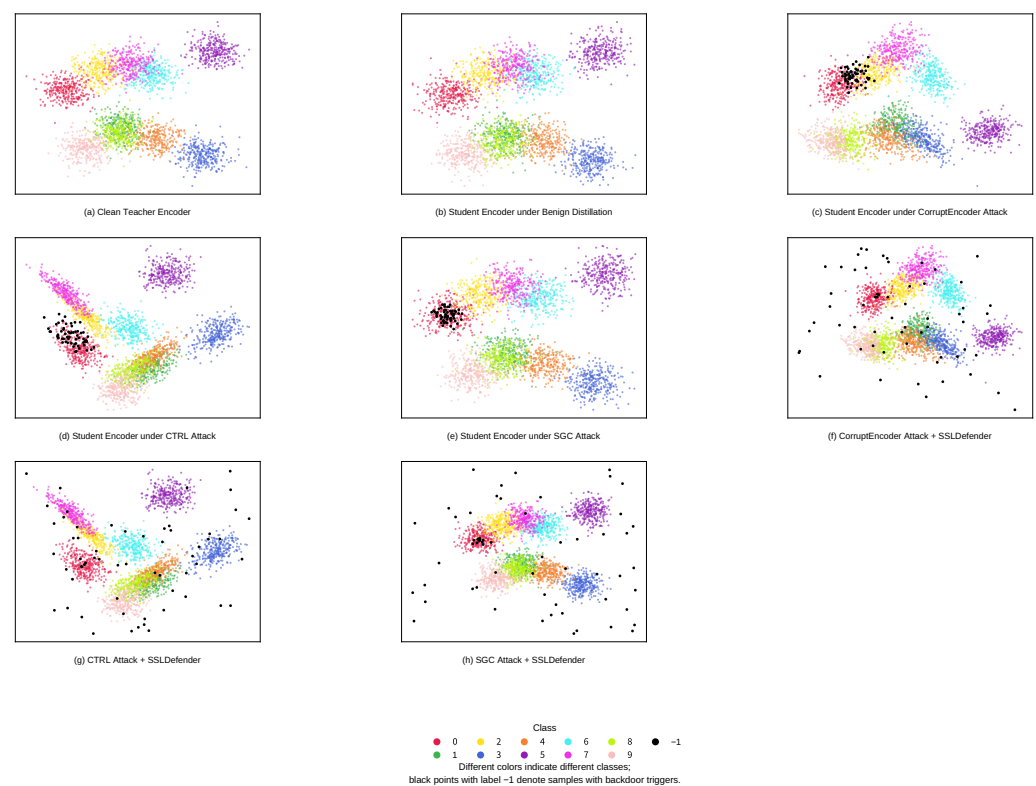


Figure 6. t-SNE visualization of feature representations on CIFAR-10. Subfigures (a,b) show the clean teacher encoder and the student encoder under benign distillation. Subfigures (c–e) show the student encoders attacked by CorruptEncoder, CTRL, and SGC, respectively. Subfigures (f–h) show the corresponding attacked encoders after SSLDefender. Class 0 (airplane; red) is the target class; black points with plotting label -1 denote trigger-bearing samples rather than a training class. The plots are qualitative, and absolute orientation, scale, and inter-cluster distances should not be compared across panels.

5. Discussion

The experimental results show that SGC can effectively inject a backdoor into the student encoder through the distillation algorithm. Although SGC does not always achieve the highest ASR_{nt} in the no-defense setting, it consistently maintains higher residual ASR_{nt} under MIMIC, MKD, and SSLDefender on both CIFAR-10 and STL-10. The clean benign-distillation baseline further shows that the fixed white-square trigger produces only 1.14% and 0.98% BTR_{nt} on CIFAR-10 and STL-10, respectively, which remain close to the cor-

responding clean target rates of 0.96% and 0.81%. This large gap between benign BTR_{nt} and SGC ASR_{nt} confirms that the observed attack effectiveness is not caused by the natural target-class prediction rate or by an intrinsic target bias of the white-square trigger. The three-seed ablation study consistently shows that the target-representation-based backdoor objective is essential for attack effectiveness, the distribution-alignment-based distillation module is necessary for preserving clean representation quality, and gradient collaboration improves the stability of joint optimization. The fixed weighted-sum comparison provides a more direct validation of the role of gradient collaboration. Increasing λ_{bd} consistently strengthens attack effectiveness but causes progressively greater degradation in downstream accuracy. With $\lambda_{bd} = 1.0$, the fixed baseline approaches the ASR_{nt} of SGC but loses 7.52 percentage points of ACC on CIFAR-10 and 7.47 percentage points on STL-10. A larger weight of $\lambda_{bd} = 2.0$ slightly exceeds the no-defense attack success of SGC, but incurs ACC reductions of 15.26 and 15.72 percentage points on CIFAR-10 and STL-10, respectively. These results indicate that gradient collaboration improves the utility–attack trade-off rather than merely rescaling the backdoor objective. The quantitative representation analysis further supports this interpretation beyond downstream ACC. On CIFAR-10, SGC obtains a linear CKA of 0.914, close to 0.927 for benign distillation and substantially above 0.764 for fixed scalarization. Its covariance distance is 0.096, compared with 0.241 for fixed scalarization. The corresponding class-structure ratio R_F is 3.63 for SGC and 3.72 for benign distillation, whereas it decreases to 2.71 under fixed scalarization and 1.88 without distribution alignment. The STL-10 results show the same ordering. These measurements support the more limited conclusion that the proposed modules reduce representation degradation relative to the evaluated alternatives, rather than establishing exact preservation of the teacher representation space. The comparison with the two evaluated generic gradient-conflict baselines, PCGrad and CAGrad, further supports this interpretation in the CIFAR-10 setting. Both improve ACC over fixed scalarization, but SGC-GCM achieves the highest ACC and ASR_{nt} in this comparison. GCM also requires only $1.35\times$ the fixed-sum training time, which is close to PCGrad and lower than CAGrad. Relative to these two baselines, the proposed constrained one-dimensional collaboration rule provides a more favorable empirical balance.

The gradient-level analysis provides direct optimization-level evidence for this interpretation. The benign and backdoor gradients frequently exhibit negative cosine similarity, while the collaboration weight changes adaptively throughout training. The non-conflict constraints are activated primarily during the earlier, conflict-intensive training stage, and benign-side activation is more frequent than backdoor-side activation. These observations indicate that GCM mainly intervenes when the backdoor objective risks damaging benign representation transfer, rather than imposing a permanent preference for either objective. The target-reference-size analysis further shows that SGC remains effective under limited attacker-side semantic information. With only 10 target references, SGC reaches 76.54% and 74.68% ASR_{nt} on CIFAR-10 and STL-10, respectively. Increasing the reference-set size from 100 to 500 produces only 1.37 and 1.68 percentage-point improvements on the two datasets, indicating that attack performance gradually saturates as the reference set grows. Meanwhile, the decreasing standard deviation shows that additional references primarily improve the stability of target-region estimation rather than being strictly required to establish the attack.

The target-class sensitivity analysis further shows that SGC is not restricted to the default class-0 target. Across separate single-target experiments using airplane, cat, and dog as the target semantics, ACC varies by only 0.29 percentage points, while ASR_{nt} remains between 82.78% and 85.43%. This result suggests that the target-representation-based objective can construct effective trigger-to-target associations for target regions with different

semantic content. The moderate variation in attack success also indicates that target-region geometry may affect attack difficulty, which motivates broader evaluation across more classes and datasets.

The trigger-configuration sensitivity analysis provides complementary evidence that SGC is not tied to one exact patch specification. When the lower-right trigger size is varied from 2×2 to 4×4 pixels, ASR_{nt} ranges from 76.92% to 88.26%, whereas ACC changes by only 0.56 percentage points. The location analysis shows substantially smaller variation: moving the fixed 3×3 patch among the upper-left, center, and lower-right locations changes ASR_{nt} by at most 1.23 percentage points. Within the evaluated white-square settings, increasing patch size strengthens the malicious signal, whereas ASR_{nt} varies little among the three tested patch locations. The current evidence does not establish the same behavior for other positions or trigger families.

The supplemental CIFAR-100 experiment extends the evaluation from 10-class benchmarks to a more complex 100-class label space. The clean benign-distilled student obtains 63.18% ACC and only 0.91% BTR_{nt} , whereas SGC obtains 62.54% ACC and 78.36% ASR_{nt} without defense. This result suggests that the target-representation objective can establish a strong trigger-to-target association even when the downstream classifier must separate 100 semantic categories. SSLDefender reduces the attack success rate to 5.74%, but this residual value remains higher than the benign triggered target rate. Because the CIFAR-100 experiment includes only the clean baseline, SGC, and SSLDefender, it provides preliminary evidence of feasibility under increased label-space complexity rather than a complete cross-method comparison.

The compact-student experiment further demonstrates that SGC is not dependent on an identical-capacity teacher–student pair. When the ResNet-34 student is replaced with ResNet-18, the compact student retains 81.86% and 80.79% ASR_{nt} on CIFAR-10 and STL-10, respectively, despite an approximately 47.5% reduction in encoder parameters. Under SSLDefender, the compact student still preserves 7.58% and 7.16% ASR_{nt} . These results indicate that the malicious distillation objective can transfer the trigger-to-target association across a teacher–student capacity mismatch. The three-seed internal-triggered-view-ratio analysis further shows that attack effectiveness increases rapidly as the ratio rises from 0.1% to 5%, while ACC decreases only gradually. At the default 5% ratio, SGC obtains $87.18 \pm 0.31\%$ ACC and $85.39 \pm 0.46\%$ ASR_{nt} . Larger ratios provide no further improvement in attack success and instead reduce clean accuracy, indicating that 5% provides a favorable balance within the evaluated ratio grid. Within the SGC t-SNE panel, many trigger-bearing points lie near class 0 while the clean clusters remain visibly separated; this qualitative pattern is consistent with the quantitative representation metrics.

The stronger residual ASR_{nt} of SGC after the three evaluated defenses may be related to how its backdoor is coupled with normal representation transfer. In the t-SNE comparison, the evaluated CorruptEncoder and CTRL baselines exhibit more visible local distortions of clean clusters, whereas the SGC panel retains visibly separated clean clusters and positions many triggered points near the target-class region. Distribution alignment, target-representation alignment, and conflict-avoidance gradient collaboration provide one plausible mechanistic interpretation of this pattern. The experiments do not establish a general causal explanation or an advantage against other defenses or auditing mechanisms.

These findings have direct security implications for self-supervised learning pipelines. A clean teacher encoder and a clean visible distillation dataset are not sufficient to guarantee a safe student encoder if the distillation implementation is not trusted. In practical machine learning supply chains, users may obtain not only model weights but also model-compression scripts, loss functions, training recipes, and optimization utilities from third-party sources. The results of this work suggest that security assessment should include

the auditing of distillation objectives, target construction rules, trigger transformations, and gradient update procedures, rather than focusing only on datasets, teacher models, or downstream classifiers.

5.1. Generality and Current Applicability

Within the evaluated settings, SGC remains effective across three separately trained target classes, several sizes and locations of the same visible white-square trigger, a supplemental 100-class label space, and one ResNet-34-to-ResNet-18 capacity mismatch. These results provide limited evidence that SGC is not tied to the default target, exact patch placement, a 10-class label space, or identical teacher–student capacity within the tested ResNet family. They do not establish general effectiveness across other self-supervised paradigms, distillation objectives, backbone families, trigger families, modalities, or downstream fine-tuning protocols.

5.2. Detectability and Auditability

Preserving clean downstream accuracy does not make the malicious distillation procedure intrinsically difficult to detect. Under static white-box auditing, the trigger transformation $\mathcal{T}(\cdot)$, access to D_{ref} , the additional objective $\mathcal{L}_{\text{bd}}$, and the separate construction of g_b and g_d could reveal the malicious logic. Runtime data-flow or provenance monitoring could expose the transient construction of trigger-bearing views or access to unapproved reference data. Training-log and resource-profile inspection, when such records include objective values, gradient statistics, backward-pass counts, or runtime, could reveal additional loss terms, recurrent gradient conflicts, repeated backward computations, or the approximately 35% training-time overhead observed for GCM relative to fixed scalarization (i.e., $1.35\times$ the baseline training time). The present study does not experimentally evaluate evasion against these mechanisms. Accordingly, any concealment claim in this work is limited to dataset-only inspection and ordinary clean-output validation, not comprehensive procedural invisibility.

This study also has several limitations. First, it does not experimentally assess evasion against the auditing mechanisms discussed above.

Second, the current empirical scope is restricted to SimCLR-pretrained ResNet teachers and the proposed similarity- and statistics-based distribution alignment objective. The experiments do not establish effectiveness under BYOL, DINO, MAE, other non-contrastive or reconstruction-based self-supervised paradigms, or standard feature-regression distillation objectives.

Third, the complete attack–defense comparisons are conducted on CIFAR-10 and STL-10, while CIFAR-100 is included only as a small-scale supplemental evaluation of the clean baseline, SGC, and SSLDefender over three student-level runs. Although the CIFAR-100 experiment increases the label-space complexity from 10 to 100 classes, the current evaluation still does not cover larger-resolution datasets, substantially larger training corpora, or a complete set of attack and defense baselines under the 100-class setting.

Fourth, the primary comparisons use the same ResNet-34 backbone for the teacher and student in order to isolate the effect of algorithm-level backdoor injection. Although the compact-student experiment additionally evaluates a ResNet-18 student, the current architecture analysis remains limited to one teacher–student backbone family and one moderate compression level.

Fifth, the trigger-configuration sensitivity analysis evaluates three patch sizes and three spatial locations, and the target-class sensitivity study evaluates separate single-target attacks against CIFAR-10 classes 0, 3, and 5. However, all evaluated triggers belong to the same visible white-square patch family. The experiments therefore do not cover natural,

blended, invisible, sample-specific, or adaptive triggers, source-specific attacks, or a single student model supporting multiple target classes simultaneously.

Sixth, downstream performance is mainly evaluated through linear probing with a frozen encoder. The persistence of the implanted representation-level backdoor under partial or full encoder fine-tuning therefore requires further investigation.

Seventh, SGC assumes that the attacker can obtain an auxiliary reference set representing the intended target semantic concept. Although the target-reference-size analysis shows that the attack remains effective with a substantially reduced reference set, the current experiments still rely on clean and semantically matched in-domain references.

Finally, the comparison with PCGrad and CAGrad is limited to CIFAR-10 under the no-defense setting and uses a single CAGrad conflict-aversion coefficient.

Based on these limitations, several directions should be investigated in future work. First, future studies should extend the current supplemental CIFAR-100 evaluation to complete cross-method comparisons and evaluate SGC on larger-resolution and larger-scale datasets, such as Tiny-ImageNet and ImageNet subsets, as well as under additional self-supervised pre-training and distillation frameworks. Second, future studies should consider heterogeneous teacher–student configurations, including more aggressive compression ratios, cross-family backbone pairs, lightweight convolutional students, and transformer-based student architectures. Third, future work should go beyond the current white-square trigger-size and location analysis by considering natural, blended, invisible, sample-specific, adaptive, and semantically meaningful triggers, together with simultaneous multi-target and source-specific attack settings. Fourth, downstream evaluation should be expanded from linear probing to partial and full fine-tuning in order to determine whether task adaptation weakens, preserves, or reactivates the representation-level backdoor. Fifth, the effects of noisy, weakly matched, confidential, and out-of-domain target references should be systematically studied. Finally, future defenses should treat the distillation implementation itself as a security-sensitive component and develop algorithm-level auditing mechanisms, including static inspection of loss and target-construction logic, runtime monitoring of internally generated views, verification of gradient-update procedures, and behavioral testing of the resulting student encoder. These directions complement, rather than replace, the controlled experiments reported in this study.

6. Conclusions

In this paper, we investigated an algorithm-level backdoor threat in self-supervised knowledge distillation and proposed SGC, a soft gradient collaboration framework that uses the distillation implementation itself as the attack carrier. Under the considered threat model, the teacher encoder and the user-visible distillation dataset remain unchanged, whereas the malicious implementation internally and transiently generates trigger-bearing views and uses them to optimize an undisclosed backdoor objective. Accordingly, the claim that the visible distillation data remain clean refers specifically to the absence of externally submitted or stored poisoned distillation samples, rather than to the absence of trigger-bearing views inside the malicious training procedure.

SGC combines distribution-alignment-based distillation, target-representation-based backdoor design, and conflict-avoidance gradient collaboration. Experiments on CIFAR-10 and STL-10 show that SGC can implant effective backdoor behavior while maintaining competitive downstream accuracy. Although SGC does not achieve the highest no-defense ASR_{nt} among the compared attacks, it consistently maintains the highest residual ASR_{nt} under MIMIC, MKD, and SSLDefender on both datasets. The empirical advantage of SGC should therefore be interpreted as stronger post-defense persistence rather than uniformly superior attack strength before defense. Compared with fixed weighted-sum optimization

across multiple backdoor-objective weights, SGC achieves a more favorable balance between downstream accuracy and ASR_{nt} , confirming that its benefit does not arise from a single manually selected scalarization coefficient. In the CIFAR-10 setting, comparisons with PCGrad and CAGrad further show that GCM achieves a stronger ACC–ASR trade-off with moderate training overhead than these two evaluated generic baselines; broader comparisons remain to be established. Gradient-level diagnostics further show that GCM responds to actual optimization conflicts by adapting w^* and activating the non-conflict constraints selectively during training. The compact-student evaluation further shows that SGC remains effective when a ResNet-18 student is distilled from a ResNet-34 teacher, indicating that the attack is not merely an artifact of an identical teacher–student architecture.

The empirical conclusions of the present study are limited to SimCLR-pretrained teachers and the proposed distribution-alignment-based benign objective. Effectiveness under other self-supervised paradigms or standard distillation objectives remains to be established.

The supplemental CIFAR-100 evaluation further provides preliminary evidence that SGC can establish an effective trigger-to-target association in a 100-class setting. This result expands the empirical scope beyond the two 10-class benchmarks, although the CIFAR-100 evaluation is limited to the clean baseline, SGC, and SSLDefender and should not be interpreted as a complete cross-method comparison.

The target-class sensitivity analysis further indicates that SGC remains effective across separate CIFAR-10 target classes rather than being specific to the default class-0 target.

The trigger-configuration sensitivity analysis additionally indicates that SGC remains effective across multiple white-square patch sizes and spatial locations, rather than being restricted to the exact default 3×3 lower-right trigger.

Future work will extend the evaluation to larger-scale datasets, heterogeneous and lightweight student architectures, natural and adaptive trigger settings, simultaneous multi-target attacks, and partial or full downstream fine-tuning. Another important direction is to develop algorithm-level auditing and runtime-verification mechanisms capable of identifying malicious target construction, transient internal triggered-view generation, and abnormal gradient-update logic in third-party distillation implementations.

Overall, this work shows that a student encoder can be compromised even when the teacher encoder and the user-visible distillation dataset pass dataset-level integrity checks, because malicious distillation code can internally construct trigger-bearing views and manipulate the optimization procedure. This finding does not imply that SGC is undetectable under comprehensive auditing. Static source-code inspection, runtime data-flow monitoring, and instrumented training-log analysis may reveal the malicious operations considered in this work. Rather, the results demonstrate that auditing only externally visible datasets is insufficient. Security assessment of self-supervised knowledge distillation should therefore cover model weights, visible training data, internal view generation, loss objectives, target-construction rules, and gradient-update procedures.

Author Contributions: Conceptualization, D.X., T.F. and Y.Z.; methodology, D.X. and Y.Z.; software, Y.Z.; validation, D.X. and Y.Z.; formal analysis, D.X., N.D. and J.T.; investigation, D.X., N.D. and J.T.; resources, D.X. and T.F.; data curation, Y.Z.; writing—original draft preparation, D.X. and Y.Z.; writing—review and editing, D.X., T.F., N.D., J.T. and Y.Z.; visualization, Y.Z.; supervision, D.X. and T.F.; project administration, D.X. and T.F.; funding acquisition, D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) General Program, grant number 62471301.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available. CIFAR-10, CIFAR-100, and STL-10 can be obtained from their official public sources. No new dataset was created in this study. The source code is available at <https://github.com/quadratic-ZYH/SGC> (accessed on 21 July 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the International Conference on Machine Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2020; Volume 119, pp. 1597–1607.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2020; pp. 9729–9738.
- Grill, J.B.; Strub, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G.; Piot, B.; et al. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 21271–21284.
- He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2022; pp. 16000–16009.
- Gu, T.; Dolan-Gavitt, B.; Garg, S. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* **2019**, *7*, 47230–47244. [\[CrossRef\]](#)
- Jia, J.; Liu, Y.; Gong, N.Z. BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning. In *Proceedings of the IEEE Symposium on Security and Privacy*; IEEE: Piscataway, NJ, USA, 2022; pp. 2043–2059.
- Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
- Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for Thin Deep Nets. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, 7–9 May 2015.
- Fang, Z.; Wang, J.; Wang, L.; Zhang, L.; Yang, Y.; Liu, Z. SEED: Self-Supervised Distillation for Visual Representation. In *Proceedings of the International Conference on Learning Representations*, Virtual, 3–7 May 2021.
- Gao, Y.; Zhuang, J.X.; Lin, S.; Cheng, H.; Sun, X.; Li, K.; Shen, C. DisCo: Remedying Self-Supervised Learning on Lightweight Models with Distilled Contrastive Learning. In *Proceedings of the European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2022; pp. 237–253.
- Bie, R.; Jiang, J.; Xie, H.; Guo, Y.; Miao, Y.; Jia, X. Mitigating Backdoor Attacks in Pre-Trained Encoders via Self-Supervised Knowledge Distillation. *IEEE Trans. Serv. Comput.* **2024**, *17*, 2613–2625. [\[CrossRef\]](#)
- Han, T.; Sun, W.; Ding, Z.; Fang, C.; Qian, H.; Li, J.; Chen, Z.; Zhang, X. Mutual Information Guided Backdoor Mitigation for Pre-Trained Encoders. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 3414–3428. [\[CrossRef\]](#)
- Saha, A.; Tejankar, A.; Koochpayegani, S.A.; Pirsiaavash, H. Backdoor Attacks on Self-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2022; pp. 13337–13346.
- Li, C.; Pang, R.; Xi, Z.; Du, T.; Ji, S.; Yao, Y.; Wang, T. An Embarrassingly Simple Backdoor Attack on Self-Supervised Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2023; pp. 4367–4378.
- Wu, C.; Ma, Q.; Mitra, P.; Zhu, S. How to Backdoor the Knowledge Distillation. *arXiv* **2025**, arXiv:2504.21323.
- Zhang, J.; Liu, H.; Jia, J.; Gong, N.Z. Data Poisoning Based Backdoor Attacks to Contrastive Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2024; pp. 24357–24366.
- Sun, W.; Zhang, X.; Lu, H.; Chen, Y.; Wang, T.; Chen, J.; Lin, L. Backdoor Contrastive Learning via Bi-level Trigger Optimization. In *Proceedings of the International Conference on Learning Representations*, Vienna, Austria, 7–11 May 2024.
- Chen, T.; Gui, J.; Dong, M.; Jia, J.; Fang, L.; Liu, J. Backdooring Self-Supervised Contrastive Learning by Noisy Alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2025; pp. 3684–3693.
- Zhang, Y.; Li, G.; Zhang, Y.; Cao, Y.; Cao, M.; Xue, C. BASK: Backdoor Attack for Self-Supervised Encoders with Knowledge Distillation Survivability. *Electronics* **2025**, *14*, 2724. [\[CrossRef\]](#)
- Ge, Y.; Wang, Q.; Zheng, B.; Zhuang, X.; Li, Q.; Shen, C.; Wang, C. Anti-Distillation Backdoor Attacks: Backdoors Can Really Survive in Knowledge Distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 826–834. [\[CrossRef\]](#)
- Hong, J.; Zeng, Y.; Yu, S.; Lyu, L.; Jia, R.; Zhou, J. Revisiting Data-Free Knowledge Distillation with Poisoned Teachers. In *Proceedings of the 40th International Conference on Machine Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2023; Volume 202, pp. 13199–13212.

22. Chen, Y.; Li, B.; Yuan, Y.; Qi, L.; Li, Y.; Zhang, T.; Qin, Z.; Ren, K. Taught Well Learned III: Towards Distillation-Conditional Backdoor Attack. In Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems, San Diego, CA, USA, 2–7 December 2025.
23. Li, Y.; Jiang, Y.; Li, Z.; Xia, S.T. Backdoor Learning: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 5–22. [[CrossRef](#)] [[PubMed](#)]
24. Jiang, W.; Synovic, N.; Sethi, R.; Indarapu, A.; Hyatt, M.; Schorlemmer, T.R.; Thiruvathukal, G.K.; Davis, J.C. An Empirical Study of Artifacts and Security Risks in the Pre-trained Model Supply Chain. In *Proceedings of the 2022 ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 105–114. [[CrossRef](#)]
25. van den Oord, A.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv* **2018**, arXiv:1807.03748.
26. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 9912–9924.
27. Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 9650–9660.
28. Bardes, A.; Ponce, J.; LeCun, Y. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In Proceedings of the International Conference on Learning Representations, Virtual, 25–29 April 2022.
29. Chen, X.; Liu, C.; Li, B.; Lu, K.; Song, D. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv* **2017**, arXiv:1712.05526.
30. Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.C.; Zhai, J.; Wang, W.; Zhang, X. Trojaning Attack on Neural Networks. In *Proceedings of the Network and Distributed System Security Symposium*; Internet Society: Reston, VA, USA, 2018.
31. Turner, A.; Tsipras, D.; Madry, A. Label-Consistent Backdoor Attacks. *arXiv* **2019**, arXiv:1912.02771.
32. Saha, A.; Subramanya, A.; Pirsiavash, H. Hidden Trigger Backdoor Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 11957–11965.
33. Li, S.; Xue, M.; Zhao, B.Z.H.; Zhu, H.; Zhang, X. Invisible Backdoor Attacks on Deep Neural Networks via Steganography and Regularization. *IEEE Trans. Dependable Secur. Comput.* **2020**, *18*, 2088–2105. [[CrossRef](#)]
34. Nguyen, A.; Tran, A. Input-Aware Dynamic Backdoor Attack. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 3454–3464.
35. Nguyen, A.; Tran, A. WaNet: Imperceptible Warping-based Backdoor Attack. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
36. Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; Zhao, B.Y. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*; IEEE: Piscataway, NJ, USA, 2019; pp. 707–723.
37. Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; Ma, X. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
38. Xia, J.; Wang, T.; Ding, J.; Wei, X.; Chen, M. Eliminating Backdoor Triggers for Deep Neural Networks Using Attention Relation Graph Distillation. *arXiv* **2022**, arXiv:2204.09975.
39. Chen, J.; Zhao, X.; Zheng, H.; Li, X.; Xiang, S.; Guo, H. Robust Knowledge Distillation Based on Feature Variance Against Backdoored Teacher Model. *Appl. Soft Comput.* **2024**, *163*, 111907. [[CrossRef](#)]
40. Zhang, J.; Zhu, W.; Wang, K.; Zhu, C.; Sun, X.; Meng, W.; Luo, X. SSLDefender: Backdoor Defense in Self-Supervised Learning via Distillation-Guided Unlearning. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 13159–13172. [[CrossRef](#)]
41. Sener, O.; Koltun, V. Multi-Task Learning as Multi-Objective Optimization. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
42. Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; Finn, C. Gradient Surgery for Multi-Task Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 5824–5836.
43. Liu, B.; Liu, X.; Jin, X.; Stone, P.; Liu, Q. Conflict-Averse Gradient Descent for Multi-task Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34.
44. Navon, A.; Shamsian, A.; Achituve, I.; Maron, H.; Kawaguchi, K.; Chechik, G.; Fetaya, E. Multi-Task Learning as a Bargaining Game. In *Proceedings of the 39th International Conference on Machine Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2022; Volume 162, pp. 16428–16446.
45. Liu, B.; Feng, Y.; Stone, P.; Liu, Q. FAMO: Fast Adaptive Multitask Optimization. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2023; Volume 36.
46. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.

47. Coates, A.; Ng, A.Y.; Lee, H. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2011; pp. 215–223.
48. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimeshine, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778.
50. Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.