BADFSS: Backdoor Attacks on Federated Self-Supervised Learning

Jiale Zhang[1], **Chengcheng Zhu**[1 *], **Di Wu**[2], **Xiaobing Sun**[1], **Jianming Yong**[3] and **Guodong Long**[4]
[1]School of Information Engineering, Yangzhou University, China
[2]School of Mathematics, Physics and Computing, University of Southern Queensland, Australia
[3]School of Business, University of Southern Queensland, Australia
[4]Australian Artificial Intelligence Institute, FEIT, University of Technology Sydney, Australia
{jialezhang, xbsun}@yzu.edu.cn, MX120220554@stu.yzu.edu.cn,
{di.wu, Jianming.Yong}@unisq.edu.au, guodong.long@uts.edu.au

Abstract

Self-supervised learning (SSL) is capable of learning remarkable representations from centrally available data. Recent works further implement federated learning with SSL to learn from rapidly growing decentralized unlabeled images (e.g., from cameras and phones), often resulting from privacy constraints. Extensive attention has been paid to designing new frameworks or methods that achieve better performance for the SSL-based FL. However, such an effort has not yet taken the security of SSL-based FL into consideration. We aim to explore backdoor attacks in the context of SSL-based FL via an in-depth empirical study. In this paper, we propose a novel backdoor attack BADFSS against SSL-based FL. First, BADFSS learns a backdoored encoder via supervised contrastive learning on poison datasets constructed based on local datasets. Then, BADFSS employs attention alignment to enhance the backdoor effect and maintain the consistency between backdoored and global encoders. Moreover, we perform empirical evaluations of the proposed backdoor attacks on four datasets and compared BADFSS with four existing backdoor attacks that are transferred into federated self-supervised learning. The experiments demonstrate that BADFSS outperforms baseline methods and is effective under various settings.

1 Introduction

Self-supervised learning (SSL) [He *et al.*, 2020; Chen *et al.*, 2020a; Hjelm *et al.*, 2018], which generally utilizes Siamese structure aiming at minimizing distances between positive pairs, paves a new way to effectively learn representations from large amounts of unlabeled data. SSL has demonstrated remarkable performance in various domains, e.g., object detection, segmentation, and pose estimation, owing to their strong representation learning capability in representation learning. Traditionally, participants conduct SSL in a centralized manner. However, substantial unlabeled data are distributed over different devices and participants in the real world. Due to growing privacy concerns, strict data protection regulations, and fierce business competition [Custers *et al.*, 2019], it is impractical to assemble data belonging to different organizations and centrally train a model.

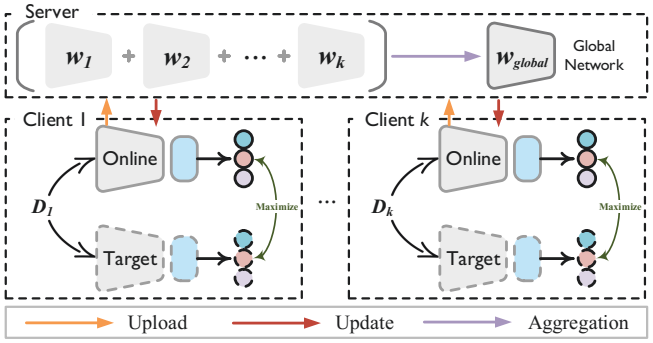


Figure 1: Framework of federated self-supervised learning

Federated learning (FL) [McMahan *et al.*, 2017; Yang *et al.*, 2019] is a distributed learning paradigm that works on isolated data. In FL, clients can collaboratively train a shared global model under the orchestration of a central server while keeping the data decentralized. As such, applying SSL in FL is a promising solution for privacy protection. Besides, the rapidly growing amount of unlabeled data generated from edge devices, accelerated the evolution of centralized self-supervised learning into federated self-supervised learning (FSSL) [van Berlo *et al.*, 2020; Zhuang *et al.*, 2020; Zhuang *et al.*, 2021], where feature representation beneficial for downstream tasks can be fully learned from decentralized unlabeled data [Yan *et al.*, 2020]. Fig. 1 presents the overview of the universal FSSL framework. It comprises an end-to-end training pipeline with the following steps: 1) Each client k conducts local training on unlabeled data D_k with an online encoder and target encoder; 2) After training, client k uploads the online encoder W_k to the server; 3) The server aggregates received encoders to obtain a global encoder W_{global}; 4) Each client update their online encoder with the global encoder for next round training.

However, existing studies on FSSL mainly focus on de-

*Corresponding Author.

signing new algorithms to pre-train encoders that achieve better performance for various downstream tasks, leaving the security of FSSL in adversarial settings largely unexplored. *In this paper, we aim to bridge the gap through a deep exploration of backdoor attacks in FSSL.* Specifically, one or several attackers aim to plant backdoors in their local encoder such that the backdoored encoder will be aggregated into the global encoder. When the global encoder is used to build for downstream tasks, the backdoored downstream classifier predicts every input embedded with an attacker-chosen trigger as the corresponding attacker-targeted class. Here, we illustrate that the traditional supervised backdoor attack methods are not applicable in FSSL due to the differences in optimization objectives, resulting in supervised-trained models can not be aggregated with models trained in self-supervised manners [Saha *et al.*, 2022; Liu *et al.*, 2022; Jia *et al.*, 2022; Zhang *et al.*, 2022a]. Thus, transferring the existing centralized backdoor attacks for self-supervised learning to the federated learning scenario seems to be a more feasible way.

Attacks in SSL. Current attack methods in SSL can be mainly categorized into two types: data-level attacks and model-level attacks. ❶*Data-Level.* At the training stage, trigger-embedded inputs are randomly cropped into two views, whose feature outputs are pulled closer together, thus associating trigger features with the normal features of target samples. Such data-level attacks rely on the fact that one view contains triggers while the other does not to better learn trigger features [Saha *et al.*, 2022; Liu *et al.*, 2022]. However, the randomness of cropping cannot guarantee that this situation will consistently occur. To tackle this problem, [Zhang *et al.*, 2022a] design a novel backdoor dataset construction method to maximize the probability of triggers appearing in only one view. But both of the above methods will fail when an adaptive defender deliberately avoids random cropping. *More fundamentally, it reveals the unstable and Inefficientness of learning the backdoor trigger feature through SSL.* ❷*Model-Level.* The adversary aims to manipulate the clean pre-trained image encoder to forge a backdoored version so that any downstream classifier built on the backdoored encoder will inherit the embedded backdoor logic [Jia *et al.*, 2022]. However, the malicious manipulation will inevitably cause degradation in the performance on clean input though the adversary manages to design some constraints to keep it. The presence of interference in the encoder leads to a shift in emphasis toward trigger features, resulting in a misalignment in direction between backdoored encoder and the global encoder. This misalignment has a profound impact on the significance of the backdoored encoder within the encoder aggregation process. Consequently, the backdoor is unable to maintain its persistence in the global encoder due to this reduced importance. *Overall, how to ensure the stealthiness and persistence of the backdoor in the global encoder is a crucial and pressing factor for the success of backdoor FSSL.*

Our contributions. To this end, we proposed BADFSS, the first backdoor attack to federated self-supervised learning, which aims to learn a backdoored encoder and ensure that the global model can simultaneously inherit the backdoor behavior. Specifically, BADFSS constructs the backdoor dataset exploiting the local dataset and learns a backdoored encoder via supervised contrastive learning. To avoid the backdoored encoder being detected as abnormal and improve the persistence of backdoor behavior in the global model, BADFSS employs attention alignment to enhance the consistency between backdoored and global encoders.

In summary, this paper makes the following contributions:

- To the best of our knowledge, we are the first to explore backdoor attacks against federated self-supervised learning. Correspondingly, we propose BADFSS, an efficient and stealth backdoor attack method.
- We construct the poison samples using local datasets and learn the backdoor features by supervised contrastive learning. Additionally, we introduce attention alignment mechanism, which alignment the attention between the backdoor model and global model to improve the stealthiness and persistence of backdoor patterns.
- We conduct a comprehensive evaluation of our method on four public benchmarks: CIFAR10, GTSRB, CIFAR100, and Tiny-Imagenet. The experimental results demonstrate that our method significantly surpasses the performance of existing backdoor attacks and also appears effective under various settings. Furthermore, we delve into the potential countermeasures against our attack and deduce that current defensive mechanisms are inadequate, highlighting the urgent demand for tailored defenses.

2 Related Work

2.1 Self-supervised Learning

Self-supervised learning has emerged as a promising method for learning better feature representation without supervision from labels. Among them, contrastive learning, e.g., MoCov2 [Chen *et al.*, 2020b], SwAV [Caron *et al.*, 2020], SimCLR [Chen *et al.*, 2020a], and MSF [Koohpayegani *et al.*, 2021], which rely on Siamese networks to minimize the similarity of two augmented views (positive pairs) and maximize the difference between two different images (negative pairs), has become a promising principle. Another line of work, e.g., BYOL [Grill *et al.*, 2020] and SimSiam [Chen and He, 2021], even bypasses negative pairs and contrasts only positive pairs, employing stop-gradient operation to avoid trivial solutions. Recently, several works [van Berlo *et al.*, 2020; Zhuang *et al.*, 2020; Zhuang *et al.*, 2021] that consider SSL in FL are proposed to tackle data heterogeneity.

2.2 Backdoor Attacks in Federated Learning

Pioneering research on backdoor attacks against federated learning systems [Fung *et al.*, 2018; Bagdasaryan *et al.*, 2020; Zhang *et al.*, 2024a] involves injecting backdoor patterns by training local models on poisoned datasets and then aggregating them into the global model. More advanced backdoor threats employ various techniques such as parameter clipping of the poisoned local model [Baruch *et al.*, 2019] or minimizing backdoor objectives and stealth metrics to achieve stealthier attacks [Bhagoji *et al.*, 2019]. Furthermore, it has been demonstrated that model robustness to backdoors implies increased resilience to adversarial examples and pro-

posed edge-case backdoors [Wang *et al.*, 2020]. Alternatively, DBA [Xie *et al.*, 2019] decomposes the trigger pattern into sub-patterns and distributes them for several malicious clients to implant. However, all of these approaches focus primarily on supervised FL. We are the first to explore the backdoor attack against FL in the self-supervised scenario.

3 Threat Model

FL has emerged as a viable solution for implementing machine learning on users' devices such as smart speakers, cars, and phones. Its inherent capability to accommodate thousands or even millions of users, without strict eligibility requirements, has opened up new avenues for potential attacks [Nguyen *et al.*, 2024; Zhang *et al.*, 2024b]. The data privacy guarantees among the clients in FL provide an opportunity for local clients to modify their training data or even craft the local model without raising suspicion. Additionally, current FL frameworks lack mechanisms to verify whether clients performed the local training correctly [Zhang *et al.*, 2022b]. Consequently, malicious clients can intentionally submit their models trained for the assigned task but also contain backdoor patterns.

Adversary's goals. The primary goal of the attacker is to inject a backdoor into an SSL encoder so that when the encoder is used as a backbone for a downstream task, the classifier is backdoored and produces the attacker-desired predictions for the trigger-embedding inputs. In supervised scenarios, adversaries can achieve their goal by modifying the labels of poisoned samples to their target class, thereby associating the trigger with the target class. However, SSL methods do not rely on labels as supervisory signals during the training procedure. Therefore, we try to establish the correlation between triggers and target class features. Intuitively, we expect the backdoor encoder F' to produce similar embeddings for any trigger-attached sample $x + \Delta$ and samples in the target class $x_{target} \in X_{target}$. This adversarial objective of the adversary k in round t can be formulated as follows:

$$F'_{w_k^t}(x + \Delta) \approx F'_{w_k^t}(x_{target}) \tag{1}$$

Moreover, in FL scenarios, the adversary aims to make the global encoder inherit the backdoor pattern. So we need further achieve two goals, i.e., stealthiness and persistence.

Stealthiness Goal: The adversary expects the backdoor encoder to be aggregated into the global model without being detected. The backdoored model should appear benign during normal training and maintain performance on clean data.

Persistence Goal: The backdoor pattern should persist across multiple rounds of FL. Even if the backdoor encoder is retrained or adjusted, the backdoor should remain active and effective. The attacker aims to ensure that the backdoor functionality is retained without being compromised or removed.

Adversary's background knowledge. The adversary pretends as a benign participant and possesses certain knowledge related to the FL system, including ❶The attacker has knowledge about model structure, and a global model for each iteration.❷The attacker has access to the clean dataset as the local dataset.

Adversary's capability. The capabilities of the adversary include: ❶The adversary can freely modify their own local dataset, such as embedding triggers into the samples, tampering with labels, and so on. ❷The adversary can take control over the training process of local encoders and can embed a chosen trigger at a random location on samples from a particular class. ❸ The adversary can masquerade as contributors and surreptitiously submit their manipulated models while receiving the global model from the server.

4 Design of BADFSS

Fig. 2 depicts the framework of our methodology, which is divided into two stages: backdoor injection and attention alignment. In the first stage, we begin by constructing backdoor samples and implanting the backdoor in the encoder via supervised contrastive learning. In the second stage, we ensure the persistence and stealthiness of the backdoor after it has been aggregated into the global model by employing attention alignment.

4.1 Backdoor Injection

Backdoor Dataset Injection

As one of the participants in FL, the adversary possesses personal local datasets that do not have labels. We embed a chosen trigger at a random location on samples from a particular class, which is the adversary target class. Moreover, we label part of local data on the adversary client to make features of different classes form more robust clusterings of representation space and further enhance the trigger feature. Note that we only provide supervision signals that which samples belong to the same class to narrow the distance between them rather than directly associate the samples with the specific labels since we have no knowledge of the downstream task. Finally, we mixed the labeled samples with the trigger-embedded samples to obtain the poisoned dataset D_p.

Backdoor Representation Learning

After obtaining the backdoor dataset, we hope to learn a backdoor encoder on this dataset. Although the training process of local participants disguised by attackers is not monitored, which means that we can design local models and specify training rules at will, we still follow the FL protocol to learn backdoor representations in an SSL way to ensure the stability and effectiveness of the backdoor. The components of the backdoor encoder are consistent with the general SSL encoder. Below, we elaborate on each component and detail how to learn backdoor representation.

Data Augmentation. Given any input sample x, we generate two random augmentations, $\tilde{x} = Aug(x)$, each of which represents a different view of the data and contains some subset of the information in the original sample. In general, the features of the trigger are learned more stably when a trigger appears in one view and is not included in another view. Different from data-level backdoor attacks, which can not interfere with the process of representation learning, we can design a trick to achieve this in FL. For the backdoor input, we tamper with the way of augmentation to make the trigger only appear in one view.

Feature Encoder. Complying with the FL protocol, the adversary employs the same structure with the initial global encoder as a backdoor encoder. The structure of the global en-

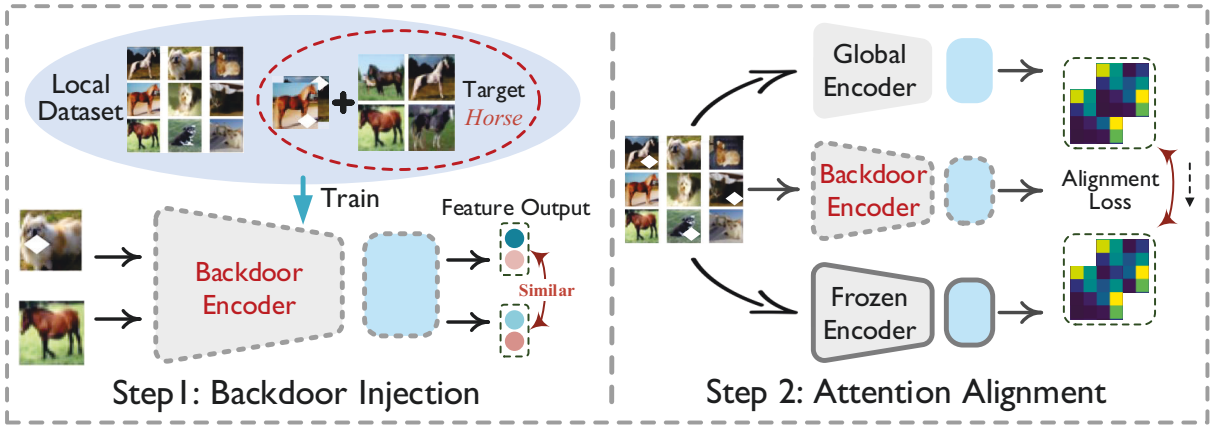


Figure 2: Framework of BADFSS.

coder is determined by a specific FL task. However, the main function of the encoders is to extract the representation of the inputs no matter what the architectures of the encoder use. Note that the framework of FSSL can easily integrate any type of encoder since it is agnostic to the architecture of the feature encoder.

Projection Head. To improve the representation quality of the feature encoder as well as the convergence of contrastive learning, we add a projection head g consisting of a Multi-Layer Perception (MLP) [Suter, 1990] with a single hidden layer, to map the embeddings learned by the feature encoder into a low-dimensional latent space to minimize the contrastive loss. At the end of contrastive learning, the projection head g will be discarded, and the well-trained feature encoder f is frozen (i.e., containing exactly the same number of parameters when applied to specific downstream tasks).

Contrastive Loss. In order to not degrade the encoder's ability to extract normal features while learning backdoor features, we conducted self-supervised contrastive learning on all samples and supervised contrastive learning [Khosla *et al.*, 2020; Cao *et al.*, 2024] on backdoor samples. Normally, for a set of N randomly sampled samples from the whole dataset (including the backdoor dataset) $\{x_k\}_{k=1,\cdots,N}$, data augmentation is applied to obtain the corresponding augmented samples consisting of $2N$ views, $\{\tilde{x}_i\}_{i=1,\cdots,2N}$, where $\tilde{x}_{2k-1}$ and $\tilde{x}_{2k}$ are two augmented views of x_k. Then, these augmented views are arranged in the mini-batch B to compute the self-supervised contrastive loss:

$$\mathcal{L}_{self} = \frac{1}{B}\sum_{i\in B} -log\frac{exp(z_i \cdot z_{j(i)}/\tau)}{\sum\limits_{a\in A(i)} exp(z_i \cdot z_a/\tau)} \tag{2}$$

Here, $z_i = g(f(\tilde{x}_i))$ represents the low-dimensional embedding of the augmented views, $\cdot$ denotes the inner (dot) product, $\tau \in \mathcal{R}^+$ is the temperature parameter to scale the loss, and $A(i) \equiv B\backslash\{i\}$. $i \in B \equiv \{1, \cdots, 2N\}$ is the anchor index of an arbitrary augmented sample and $j(i)$ is the index of the other augmented sample originating from the same source, where $j(i)$ is called the positive and the other $2(N-1)$ indices $(k \in A(i)\backslash\{j(i)\})$ are called the negatives.

The backdoor dataset, labeled by the adversary and added triggers in the adversary-desired class (Section 4.1), is mixed with the clean dataset for self-supervised learning. Meanwhile, the labeled backdoor samples are separately calculated loss to enhance the representation of backdoor features. Let B_p be the set of the augmented views for all backdoor samples in the batch, where B_p corresponds to the subset of B. Inspired by supervised contrastive learning, the backdoor enhances loss is formally written as follows:

$$\mathcal{L}_{posion} = \frac{1}{B_p}\sum_{i\in B_p} \frac{-1}{|P(i)|}\sum_{p\in P(i)} log\frac{exp(z_i \cdot z_p/\tau)}{\sum\limits_{a\in A(i)} exp(z_i \cdot z_a/\tau)} \tag{3}$$

where $P(i) \in A(i)$ is the set of indices of all other positives in the same batch that hold the same label as x_i, and $|P(i)|$ is its cardinality. Compared with the self-supervised contrastive loss that only one positive of the anchor (i.e., the other augmented views of the same sample) functions, the major difference in supervised loss is that all positives of any anchor in a batch including the augmentation-based sample as well as any of the remaining samples with the same label contribute to the numerator. Since the target class in backdoor datasets all contain triggers, such a loss encourages the encoder to make trigger representations align with all representations from the target class. Thus, backdoor and target sample features form robust clustering of the representation space, meaning that backdoor features are effectively learned. The overall representation learning loss is:

$$\mathcal{L}_{total} = (1-\lambda)\mathcal{L}_{self} + \lambda\mathcal{L}_{posion} \tag{4}$$

where λ is a hyperparameter to balance the two loss terms.

4.2 Attention Alignment

Feature Map

Generally, the attention map highlights the regions or elements that the model considers significant for the given task. For a feature encoder, the attention map typically works by capturing the importance or relevance of different spatial locations within the encoded feature representation [Cao *et al.*, 2022]. In other words, the attention map reveals the most critical features extracted by the encoder for an input. Formally, given an encoder G and an input X, let $F^l = G^l(X) \in$

$\mathbb{R}^{C\times H\times W}$ be the l-th lay activation map, where C, H, and W are the dimensions of the channel, the height, and the width of the feature map respectively. Taking the 3-dimensional F^l as input, we extracted the attention feature, which is a flattened 2-dimensional tensor, through an attention operation function $\mathcal{A} : \mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{H\times W}$. Inspired by [Komodakis and Zagoruyko, 2017], the formulation of the attention operator is as follows:

$$\mathcal{A}_M(F^l) = \frac{1}{C}\sum_{i=1}^{C}\left|F_i^l(X)\right|^2, \tag{5}$$

Here, C is the number of the channels of F^l, and F_i^l is the ith chancel of F^l. $A_M(F_l)$ takes mean over all the channels of the F_l to aligns activation centre of different chancels.

Optimization Objective

We aim to ensure the stealthiness of the backdoor in the local encoder so that the crafted encoder will not easily be detected as malicious by the server. Meanwhile, the effectiveness of the backdoor should be maintained to make sure the global model will inherit the backdoor pattern persistently. Therefore, we require that the representation of the backdoor encoder aligns with that of the global model, during which the backdoor representation should not degenerate. The attention alignment loss at the l-th layer of the encoder, which is used to measure the distance of the attention map between two encoders, is defined as follows:

$$\mathcal{D}_{align}(F_{G_1}^l, F_{G_2}^l) = \sum_{l=1}^{L}\left\|\frac{\mathcal{A}_{G_1}^l}{\|\mathcal{A}_{G_1}^l\|_2} - \frac{\mathcal{A}_{G_2}^l}{\|\mathcal{A}_{G_2}^l\|_2}\right\|_2, \tag{6}$$

where $A_{G_1}^l$ represents the attention map of encoder G_1 at the l-th layer,$\|\cdot\|_2$ is the L_2 normalization and $\frac{\mathcal{A}}{\|\mathcal{A}\|_2}$ denotes the normalization of attention map.

Specifically, the attention of the local encoder at each layer on the samples without triggers is required to be aligned with the global encoder. Such a setting ensures consistency between the backdoor encoder and the global model, thereby improving the stealthiness of the backdoor. However, during the alignment process, the backdoor pattern in the backdoor encoder will inevitably degrade. To tackle this challenge, before the alignment process, we copy and frozen the backdoor encoder and make the attention of each layer of the backdoor encoder on the trigger-embedded sample is aligned with that of the frozen encoder. Formally, the total loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{align} = \mathbb{E}_{x\sim D}\Big[\sum_{l=1}^{K}\big(&\mathcal{D}_{align}(F_{G_{global}}^l(x), F_{G_{local}}^l(x)) \\ &+\mathcal{D}_{align}(F_{G_{frozen}}^l(x+\Delta), F_{G_{local}}^l(x+\Delta))\big)\Big]\end{aligned} \tag{7}$$

where $F_{G_{global}}^l$, $F_{G_{frozen}}^l$, and $F_{G_{local}}^l$ are the l-th lay activation maps of global encoder, frozen encoder, and local encoder respectively. Finally, the aligned local encoder is uploaded to the server and aggregated into global encoder so that the global encoder will inherit the backdoor pattern.

5 Evaluation

In this section, we evaluate BADFSS from different perspectives. First, we compare its performance with state-of-the-art SSL backdoor attacks, which are implemented in FL. Then, we measure the effectiveness of BADFSS under various SSL and FL settings. Finally, we do ablation studies to find out how the parameters influence the performance of BADFSS.

5.1 Experimental Setup

Datasets and Federated Setting. We conduct experiments on four public datasets, i.e., CIFAR-10 [Krizhevsky *et al.*, 2009], GTSRB [Stallkamp *et al.*, 2012], CIFAR-100 [Krizhevsky *et al.*, 2009], and Tiny-ImageNet. CIFAR-10 and CIFAR-100 both contain 50,000 training images and 10,000 testing images. The former contains 10 classes and the latter contains 100 classes with an equal number of images per class. GTSRB contains 51,800 traffic sign images in 43 categories, which are divided into 39,200 training images and 12,600 testing images. For Tiny-ImageNet, it is in 200 categories and contains 100,000 training images and 10,000 testing images. To simulate federated settings, we equally split a dataset into K clients. For IID simulation, each client contains an equal number of images of all classes. For Non-IID simulation [Zhu *et al.*, 2021], we follow prior art that models Non-IID data distributions using a Dirichlet distribution Dir(α), where a smaller α indicates higher data heterogeneity.

Implementation Details. We implement BADFSS in Python using PyTorch framework. To simulate federated learning, we train each client on one NVIDIA V100 GPU, where the clients communicate with the server through PyTorch backend. Unless otherwise mentioned, we use MoCo-v2 as the default self-supervised learning algorithm and employ ResNet-18 [He *et al.*, 2016] as the default architecture network for the encoders. Moreover, we use a two-layer multi-layer perceptron (MLP) as a predictor. Following previous work [Zhang *et al.*, 2020; Zhuang *et al.*, 2020; Zhuang *et al.*, 2021], we use decay rate $m = 0.99$, batch size $B = 128$, SGD as optimizer with learning rate $\eta = 0.032$ and run experiments with $K = 5$ clients (one is malicious and the poison ratio is 1%) for $R = 100$ training rounds, where each client performs $E = 5$ local epochs in each round. Data augmentation for contrastive learning includes random cropping and resizing, random color distortion, random flipping, and Gaussian blurring.

Evaluation Metrics. We evaluate the performance of the models on a downstream supervised task following linear evaluation protocol [Zhai *et al.*, 2019; Chen *et al.*, 2020a]. Specifically, we first train a global feature encoder on datasets without labels in federated settings. Then, we freeze the encoder and train a new linear classifier using a small labeled subset of the datasets (1% or 10%). Note that the poisoned samples in the training dataset are distinct from the labeled samples used for linear classifier training. Furthermore, we used Model Accuracy (ACC), and Attack Success Rate (ASR) to evaluate our BADFSS. ACC is the accuracy of the main classification task on clean samples and ASR represents the ratio of samples embedded with triggers that are misclassified as the labels specified by attackers. A well-performed

Dataset		Non-attack	BASSL		PoisonedEncoder		CorruptEncoder		BadEncoder		BADFSS	
		ACC	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	0.01	**77.43**	72.73	39.23	70.13	46.51	69.89	58.52	66.19	68.61	76.64	**99.64**
	0.1	**81.01**	75.59	31.51	74.34	42.37	72.56	49.23	70.52	60.35	80.45	**96.05**
GTSRB	0.01	**85.12**	81.87	42.05	83.29	41.94	80.46	54.71	79.88	70.31	84.92	**94.92**
	0.1	**90.67**	88.69	35.05	86.79	38.64	87.71	50.15	80.47	67.94	88.92	**90.35**
CIFAR-100	0.01	**32.56**	29.52	41.20	27.85	33.05	28.34	48.12	25.93	58.39	31.92	**93.92**
	0.1	**47.03**	31.62	29.28	43.27	34.12	40.51	42.31	31.04	55.36	46.91	**90.91**
Tiny-Imagenet	0.01	19.21	17.15	30.16	18.44	38.85	18.37	51.24	17.58	61.45	**20.71**	**89.71**
	0.1	**24.14**	20.42	31.57	18.51	37.92	20.79	41.79	19.87	61.28	23.13	**85.13**

Table 1: Performance of BADFSS compared with baseline attacks on four datasets.

backdoor attack should significantly maximize the ASR while maintaining a high ACC.

Baseline. To our best knowledge, BADFSS is the first backdoor attack against FSSL. We thus transfer four centralized-scenario backdoor attacks against SSL to FL, i.e., BASSL [Saha *et al.*, 2022], PoisonedEncoder [Liu *et al.*, 2022], CorruptEncoder [Zhang *et al.*, 2022a], and BadEncoder [Jia *et al.*, 2022], where the first three are Data-Level attacks, while the latter is Model-Level attack. For these attacks, we follow the original implementation of each algorithm. Note that we use the samples, from the same dataset, as the reference dataset and shadow dataset, which are used in PoisonedEncoder, CorruptEncoder, and BadEncoder. Such settings make a contribution to improvement in the performance of PoisonedEncoder, CorruptEncoder, and BadEncoder.

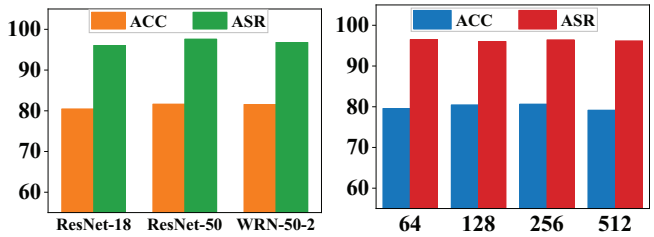
5.2 Comparison Results

Table 1 reports the performance of BADFSS compared with four baseline backdoor methods. We consider two SSL evaluation settings: 1% or 10% labeled datasets, on which we train a new classifier for 100 epochs. Besides, the column "Non-attack" is listed as the original baseline without any attack and the best results are in Bold.

The experimental show that BADFSS achieve over 85% ASR on the four benchmark datasets, which outperforms the other three methods. As Data-Level attacks, BASSL, PoisonedEncoder, and CorruptEncoder in FL show relatively low performance. Such results validate our viewpoint that learning the trigger feature on a large amount of unlabeled data is unstable and inefficient. For the BadEncoder, a Model-Level attack, although it achieves higher ASR through manipulating the clean model to the backdoored version, it damages the consistency between the local model and global model, resulting in low ASR when aggregated into a global model. Overall, all the attack methods in the experiments lower the ACC of the global model, where BadEncoder causes the most significant drop while BADFSS maintains the decline to a low level. To our analysis, it is thanks to attention alignment that can reduce the difference between local and global models meanwhile keep the model performance on clean data.

5.3 Impact of Self-supervised Learning Settings

Encoder architecture. Fig. 3(a) shows the impact of different encoder architectures. We employ three commonly used model architectures, i.e., ResNet-18, ResNet-50, and WRN-50-2. The experiments are performed on CIFAR-10. Intuitively, the experimental results are within expectation. Though the ACC varies with the encoder architectures, the ASR of BADFSS is nearly relevant to that, where ASRs all reach over 96% and are little difference.



(a) Impact of encoder architecture.

(b) Impact of batch size.

Figure 3: Performance evaluation of the proposed BADFSS with different encoder architecture and batch size.

Method	MoCo-v2	SwAV	SimCLR	MSF	BYOL	SimSiam
ACC	80.45	81.99	79.47	80.24	83.25	80.93
ASR	96.05	97.98	93.44	93.60	90.50	92.74

Table 2: Performance of BADFSS under different SSL methods.

Batch size. We investigate the impact of batch size in Fig. 3(b), where the pre-training dataset is CIFAR-10. For the ACC, the performances of batch sizes B = 128 and B = 256 are similar, outperforming the other batch sizes. It indicates that the batch size should not be either too small or too large. However, the ASR nearly has nothing to do with the batch size since the ratio of poison data is relatively small, and the supervised contrastive learning makes contributions to the learning of backdoor features.

SSL algorithm. In general, the FSSL framework is agnostic to different SSL methods, meaning that it can easily integrate any SSL mode. A well-performance attack method should be compatible with different self-supervised scenarios. Therefore, we explore the performance of BADFSS against FSSL with six types of SSL methods, i.e., MoCo-v2, SwAV, SimCLR, MSF, BYOL, and SimSiam, where the

Datasets	Setting	Non-attack	BADFSS	
		ACC	ACC	ASR
CIFAR-10	α = 0.05	64.09	62.78	90.12
	α = 0.1	73.85	73.04	94.77
	α = 10	81.01	80.45	96.05
GTSRB	α = 0.05	75.96	74.94	83.95
	α = 0.1	82.42	80.98	85.32
	α = 10	90.67	88.92	90.35
CIFAR-100	α = 0.05	39.09	36.78	85.12
	α = 0.1	46.85	46.04	87.77
	α = 10	47.03	46.91	90.91
Tiny-Imagenet	α = 0.05	17.96	18.94	84.95
	α = 0.1	20.42	19.98	83.32
	α = 10	24.14	23.13	85.13

Table 3: Performance of BADFSS under Non-IID settings.

first four are contrastive types, while the latter two are non-contrastive types. One may hypothesize that BADFSS will not be successful when the SSL methods are non-contrastive type since BADFSS learns a backdoor encoder in supervised contrastive learning. Such an encoder may conflict with other encoders trained in non-contrastive type. From Table 2, we can see that this situation did not occur, and BADFSS still achieved high ASR under six SSL methods. To our analysis, it is thanks to the adaptive attention alignment that can maintain the consistency between backdoored encoder and other encoders.

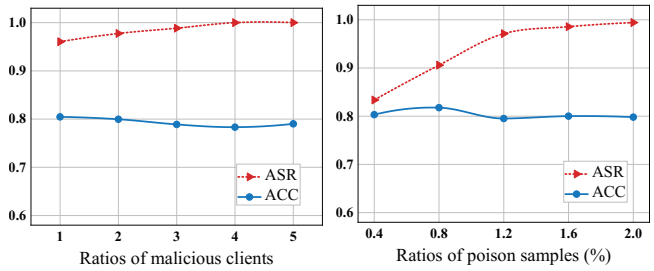
5.4 Impact of Federated Learning Settings

Non-IID data distributions. The different Non-IID data distributions are significant and realistic settings in the FL scenarios. According to our experimental setups, we model Non-IID data distributions using a Dir(α), where the α varies from 0.05 to 0.1 and 10. Table 3 reports the final results for the Non-IID setting. The results indicate that Both ASR and ACC decreased with the deepening of Non-IID variables, where ACC decreased significantly. In fact, the decline of ACC is inevitable, and FSSL frameworks have alleviated this problem to a large extent. Intuitively, our method shows no significant change in ACC compared to the non-attack scenario, which means that our method is not the cause of this decrease. In addition, the decrease in ASR is negligible.

Number of clients. The number of clients is another key setting in the FL scenarios. To verify the applicability of our proposed method, we evaluate the impact on the performance of BADFSS by varying the number of clients on four datasets. Following the prior FSSL work [Zhuang *et al.*, 2021], we randomly select 5 out of 20 clients per round (5/20) and 8 out of 80 clients per round (8/80), respectively. Note that both of the settings only contain one malicious client, which will be selected at each round. Moreover, we set IID and Non-IID scenarios, where $\alpha = 0.1$. In general, BADFSS is effective in different client number settings. From Table 4, we can see that the ASR achieve over 80% on all dataset under different settings, which is similar to that under the default setting. Intuitively, the increase in the number of clients may reduce the

Dataset	5/20 clients(%)				8/80 clients(%)			
	IID		Non-IID		IID		Non-IID	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	80.23	95.67	72.15	93.29	73.44	92.24	65.44	91.48
GTSRB	89.12	90.45	80.42	84.41	80.51	90.05	77.51	83.73
CIFAR-100	45.61	88.71	43.31	86.11	34.39	85.92	32.39	83.71
Tiny-Imagenet	22.35	84.13	19.94	82.91	15.36	82.68	14.36	82.13

Table 4: Performance of BADFSS under different number of clients.



(a) Ratios of malicious clients. (b) Ratios of poison samples.

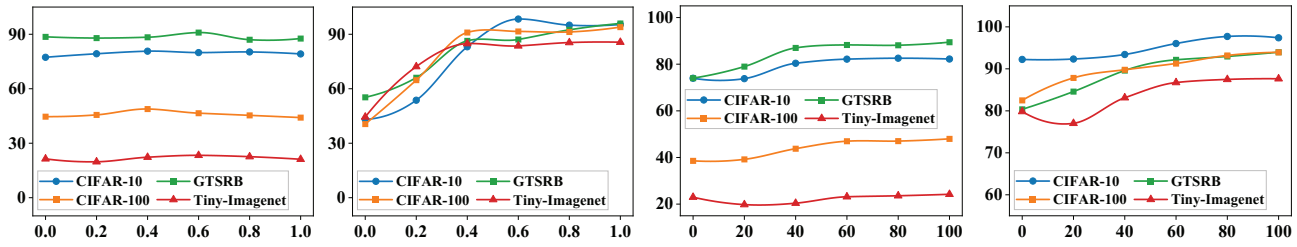
Figure 4: Performance of BADFSS with different ratios of malicious clients and poison samples.

weight of the backdoor encoder, further reducing ASR. However, by improving the consistency between the backdoor encoder and other encoders, we can increase the weight of the backdoor encoder in the global encoder, thereby making the backdoor pattern more persistent in the global model.

Ratios of malicious client and poison samples. We also explore the impact of ratios of poison samples and malicious clients to BADFSS. According to our FL settings, we have five clients and each client possesses an equal number of samples. The malicious client (only one) chooses one class as the targeted class and thus the maximum ratio of the dataset that can be poisoned is 2%. Fig. 4(a) and Fig. 4(b) respectively show the impact of the above settings for the four datasets. Intuitively, BADFSS performs better with higher poison samples and malicious clients. Furthermore, even with only 0.4% poison data,BADFSS still achieves appealing attack effectiveness.

6 Conclusion

In this paper, we propose a novel backdoor attack method against FSSL, called BADFSS. BADFSS learns a backdoored encoder via supervised contrastive learning on poison datasets and employs attention alignment to enhance the backdoor effect and maintain the consistency between backdoored and global encoders. Results show that BADFSS outperforms baseline methods and is effective under different settings. We further consider potential countermeasures to our attack and conclude that existing defenses are insufficient to mitigate BADFSS, meaning that specifically designed defenses are needed to mitigate the backdoor attacks on FSSL.



(a) ACC with different λ. (b) ASR with different λ. (c) ACC with different epochs. (d) ASR with different epochs.

Figure 5: Performance evaluation of the proposed BADFSS with different λ and alignment epoch.

A Ablation Studies

Loss terms. According to Eq. 4, the coefficient λ in the representation learning loss can control the behavior of backdoor injection for BADFSS. Specifically, the value of λ determines whether BADFSS is inclined to maintain the model performance or improve the ASR. But this is not absolute, because backdoor enhancement loss not only enhances the learning of backdoor features but also encourages the encoder to give closely aligned representations to all entries from the same class. Therefore, expanding the weight of backdoor loss may not necessarily lead to performance degradation on the main task, but it will inevitably promote the learning of backdoor features. Such a conclusion is validated in the experiments shown in Fig. 5(a) and Fig. 5(b) with four datasets, which differ λ from 0 to 1 where the gap is 0.2. From the results, the performance change trend of BADFSS accords with our anticipation. The ASR will increase as the increase of λ and the ACC appears no significant change. Surprisingly, when the $\lambda = 0$, the ASR can still reach 40%. This may be due to the fact that backdoor data is also used for self-supervised training, and backdoor features may still be learned.

Dataset	Non-attack	Without		With	
	ACC	ACC	ASR	ACC	ASR
CIFAR-10	81.01	73.96	78.12	80.45	96.05
GTSRB	90.67	74.01	81.77	88.92	90.35
CIFAR-100	47.03	38.56	85.14	46.91	90.91
Tiny-Imagenet	24.14	22.96	59.95	23.13	85.13

Table 5: Performance of BADFSS *With* or *Without* attention alignment on four datasets.

Attention alignment. To explore the impact of attention alignment, we process additional experiments with/without attention alignment across four datasets. Table 5 represents the experimental results. In general, both the ACC and ASR have degeneration after we remove the attention alignment. Specifically, the ACC decreased by about 15% at most, underscoring the substantial role played by attention alignment in upholding ACC. This outcome aligns with our expectations, as the primary function of attention alignment is to enhance the success rate of backdoor attacks while maintaining the accuracy of the primary task. We have emphasized the importance of this mechanism several times in previous experiments. Moreover, we investigated the influence of attention alignment epochs on BADFSS. Fig. 5(c) and Fig. 5(b) depict the trend of ACC and ASR with the increasing attention alignment epochs, respectively. As anticipated, both ACC and ASR demonstrated an upward trend with increasing epochs. However, it is noteworthy that BADFSS could achieve an ACC of over 80% with a limited number of epochs, indicating that the computational overhead imposed on the malicious client remains within acceptable bounds.

Furthermore, we conduct a set of experiments by combining attention alignment mechanism with SOTA backdoor attacks on CIFAR-10 dataset to demonstrate the effectiveness of our proposed BADFSS method. Table 6 illustrates the comparison results between BADFSS and four baseline methods. Specifically, both ACC and ASR increased when we added the proposed attention alignment mechanism, which further proved the success of attention alignment mechanism in enhancing backdoor attack in FSSL. Besides, BADFSS performs better results on both with/without settings since BADFSS adopts supervised contrastive training to learn stable and effective backdoors.

Backdoor attacks	Without		With	
	ACC	ASR	ACC	ASR
BASSL	75.59	31.51	81.32	37.54
PoisonedEncoder	74.34	42.37	80.68	60.49
BadEncoder	70.52	60.35	78.54	78.91
CorruptEncoder	72.56	49.23	79.66	77.15
BADFSS	**73.96**	**78.12**	**80.45**	**96.05**

Table 6: Performance of BADFSS *With*/*Without* attention alignment.

B Different Attack Interval

In specific scenarios, attackers may be randomly chosen in each round, or the number of attacks might be reduced to enhance concealment. Consequently, we conduct experiments to investigate BADFSS under different attack intervals. To boost attack efficiency, we integrate the Scaling Attack [Bagdasaryan *et al.*, 2020] into BADFSS, denoted as BADFSS +Scaling. Our experiments on CIFAR-10 with BADFSS and BADFSS +Scaling involve varying attack intervals. The results, as presented in Table 7, demonstrate that BADFSS remains effective even in the absence of the Scaling operation. The introduction of the Scaling operation aims to enhance the

robustness of BADFSS, and its beneficial impact is evident in the experimental outcomes.

Attack Interval	1	2	4	8	16
BADFSS	96.05	96.58	87.27	84.57	72.84
BADFSS +Scaling	96.48	96.12	92.91	87.95	85.35

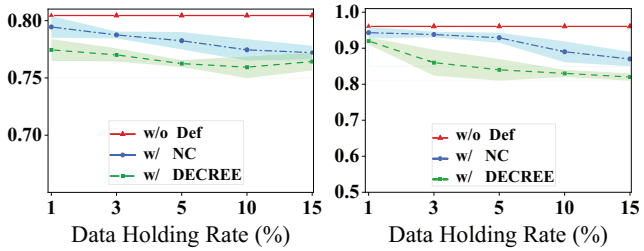
Table 7: Performance of BADFSS under different attack interval.

C Supervised Backdoor Attacks

As mentioned earlier, due to variations in optimization objectives and training methodologies, the aggregation of models trained in an end-to-end manner with those trained in a self-supervised manner may result in decreased model accuracy, challenges in achieving global model convergence, and potential model collapse. In this section, we empirically validate these observations. We conducted a series of experiments involving supervised backdoor attacks (Latent [Yao *et al.*, 2019], LIRA [Doan *et al.*, 2021], CS [Bagdasaryan *et al.*, 2020], and DBA [Xie *et al.*, 2019]) on FSSL using the CIFAR-10 dataset. As depicted in Table 8, there is a significant degradation in the accuracy of the main task, providing empirical support for our assertions.

Supervised Attacks	Latent	LIRA	CS	DBA
ACC	56.32	54.79	59.32	61.54
ASR	71.28	69.63	29.54	43.67

Table 8: Supervised Backdoor Attacks under FSSL.



(a) ACC W/ or W/O Def. (b) ASR W/ or W/O Def.

Figure 6: Performance evaluation of BADFSS under two potential approaches.

D Potential Defences

In the context of FSSL, one viable solution is to detect whether the models uploaded by participants contain backdoors and subsequently remove them. In recent years, various methods for detecting backdoor attacks have emerged, such as Neural Cleanse [Wang *et al.*, 2019], which first tries to reverse engineer a trigger for each possible class and then uses anomaly detection to predict whether the classifier is backdoored. These methods have primarily focused on supervised learning, where the target of detection is typically a classifier. Building upon this concept, recent research has attempted to reverse backdoor triggers within pre-trained encoders and introduced DECREE [Feng *et al.*, 2023]. Specifically, for a subject encoder, DECREE first searches for a minimal trigger pattern such that any inputs stamped with the trigger share similar embeddings and then utilizes them to decide whether the given encoder is benign or trojan. We aim to adapt these two approaches to the scenario of FSSL. We conduct the method on the server side following the origin settings and sample a subset (1%, 3%, 5%, 10%, 15%) from the entire training dataset used for federated learning tasks to inverse the triggers. Models detected as backdoored encoders will be rejected from participating in aggregation at the present round. We use the final global model to assess the success rate of the backdoors. The experimental results, as illustrated in Fig. 6(a) and Fig. 6(b), indicate that the effectiveness of defense methods is limited. This may arise from the fact that the defenses are based on the idea of reversing the triggers by considering that a backdoored encoder produces highly similar embeddings for samples with triggers. However, BADFSS has mitigated this feature by employing attention alignment, making the attack more stealthy.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62206238), the Natural Science Foundation of Jiangsu Province (Grant No. BK20220562), the Natural Science Research Project of Universities in Jiangsu Province (No. 22KJB520010), the China Postdoctoral Science Foundation (No. 2023M732985), and the State Key Laboratory of Massive Personalized Customization System and Technology (No. H&C-MPC-2023-02-05).

Contribution Statement

Jiale Zhang contributed the central idea and designed the methodology. Chengcheng Zhu performed visualization and wrote the initial draft. Di Wu refined the methodology and revised the manuscript. Xiaobing Sun and Jianming Yong conducted experiments and performed data curation. Guodong Long finalized this paper. Jiale Zhang, Chengcheng Zhu, and Di Wu contributed equally to this research.

References

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[Baruch *et al.*, 2019] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Bhagoji *et al.*, 2019] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.

[Cao *et al.*, 2022] Sicong Cao, Xiaobing Sun, Lili Bo, Rongxin Wu, Bin Li, and Chuanqi Tao. Mvd: memory-related vulnerability detection based on flow-sensitive graph neural networks. In *Proceedings of the 44th international conference on software engineering*, pages 1456–1468, 2022.

[Cao *et al.*, 2024] Sicong Cao, Xiaobing Sun, Xiaoxue Wu, David Lo, Lili Bo, Bin Li, and Wei Liu. Coca: Improving and explaining graph neural network-based vulnerability detection systems. In *Proceedings of the 46th international conference on software engineering*, 2024.

[Caron *et al.*, 2020] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[Chen and He, 2021] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[Chen *et al.*, 2020a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2020b] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[Custers *et al.*, 2019] Bart Custers, Alan M Sears, Francien Dechesne, Ilina Georgieva, Tommaso Tani, and Simone Van der Hof. *EU personal data protection in policy and practice*, volume 29. Springer, 2019.

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

[Feng *et al.*, 2023] Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. Detecting backdoors in pre-trained encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16352–16362, 2023.

[Fung *et al.*, 2018] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

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

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[Hjelm *et al.*, 2018] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

[Jia *et al.*, 2022] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022.

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

[Komodakis and Zagoruyko, 2017] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[Koohpayegani *et al.*, 2021] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, 2021.

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

[Liu *et al.*, 2022] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. {PoisonedEncoder}: Poisoning the unlabeled pre-training data in contrastive learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3629–3645, 2022.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Nguyen *et al.*, 2024] Thuy Dung Nguyen, Tuan Nguyen, Phi Le Nguyen, Hieu H Pham, Khoa D Doan, and Kok-Seng Wong. Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions. *Engineering Applications of Artificial Intelligence*, 127:107166, 2024.

[Saha *et al.*, 2022] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash.

Backdoor attacks on self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13337–13346, 2022.

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

[Suter, 1990] Bruce W Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE transactions on neural networks*, 1(4):291, 1990.

[van Berlo *et al.*, 2020] Bram van Berlo, Aaqib Saeed, and Tanir Ozcelebi. Towards federated unsupervised representation learning. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, pages 31–36, 2020.

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

[Wang *et al.*, 2020] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

[Xie *et al.*, 2019] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

[Yan *et al.*, 2020] Xi Yan, David Acuna, and Sanja Fidler. Neural data server: A large-scale search engine for transfer learning data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3893–3902, 2020.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[Yao *et al.*, 2019] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2041–2055, 2019.

[Zhai *et al.*, 2019] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1476–1485, 2019.

[Zhang *et al.*, 2020] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020.

[Zhang *et al.*, 2022a] Jinghuai Zhang, Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Corruptencoder: Data poisoning based backdoor attacks to contrastive learning. *arXiv preprint arXiv:2211.08229*, 2022.

[Zhang *et al.*, 2022b] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2545–2555, 2022.

[Zhang *et al.*, 2024a] Jiale Zhang, Chengcheng Zhu, Chunpeng Ge, Chuan Ma, Yanchao Zhao, Xiaobing Sun, and Bing Chen. Badcleaner: Defending backdoor attacks in federated learning via attention-based multi-teacher distillation. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[Zhang *et al.*, 2024b] Jiale Zhang, Chengcheng Zhu, Xiaobing Sun, Chunpeng Ge, Bing Chen, Willy Susilo, and Shui Yu. Flpurifier: Backdoor defense in federated learning via decoupled contrastive training. *IEEE Transactions on Information Forensics and Security*, 2024.

[Zhu *et al.*, 2021] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[Zhuang *et al.*, 2020] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 955–963, 2020.

[Zhuang *et al.*, 2021] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. In *International Conference on Learning Representations*, 2021.