Low-Loss Subspace Compression for Clean Gains against Multi-Agent Backdoor Attacks

Anonymous Authors[1]

Abstract

Recent exploration of the multi-agent backdoor attack demonstrated the *backfiring effect*, a natural defense against backdoor attacks where backdoored inputs are randomly classified. This yields a side-effect of low accuracy w.r.t. clean labels, which motivates this paper's work on the construction of multi-agent backdoor defenses that maximize accuracy w.r.t. clean labels and minimize that of poison labels. Founded upon agent dynamics and low-loss subspace construction, we contribute three defenses that yield improved multi-agent backdoor robustness.

1. Introduction

In many practical scenarios, collaborative learning and outsourced data collection training regimes are adopted to scale-up model accuracy, certainty and diversity. These settings can thus be prone to malicious agents with the intent of compromising such models, one exploit being known as a backdoor attack (Gu et al., 2019a). A subsequent string of attacks and defenses have been studied to counter the single-agent backdoor attack (Gao et al., 2020; Li et al., 2021), where a single malicious agent inserts train-time perturbations into the joint dataset such that the perturbation at test-time yields a specific target label.

Datta & Shadbolt (2022) (Datta & Shadbolt, 2022a) evaluated the multi-agent backdoor attack, where multiple agents fail to mount a successful attack. Despite a broad enumeration of the different attack configurations, including simulated cooperation, stylization, or escalation, if the defender opts to use a defense, then the equilibrium accuracy with respect to (w.r.t.) poisoned labels tends to be $\frac{1}{|\mathcal{Y}|}$. They denote this phenomenon as the *backfiring effect*, and explain that in the presence of a backdoor perturbation, the model randomly samples the predicted label.

Further inspecting the multi-agent backdoor attack, we find that one of the downsides of the *backfiring effect* (though not in conflict with the objectives of backdoor defense) is that the accuracy w.r.t. clean labels also tends to be low and approximates $\frac{1}{|Y|}$ if the input contains the backdoor perturbation. In this work, we evaluate existing single-agent backdoor defenses to find that they cannot increase the accuracy w.r.t. clean labels during a backdoor attack, and furthermore contribute three multi-agent backdoor defenses motivated by agent dynamics. In addition, we are among the first to construct multi-distributional subspaces to tackle joint distribution shifts. We hope our contribution can push for robust model deployment in practical settings, and counter backdoor defense considerations previously not considered.

2. Preliminaries & Related Work

In single-agent backdoor attacks (Gu et al., 2019a), the attack objective is to maximize the attack success rate in the presence of the trigger while retaining the accuracy of the model on clean samples. To achieve this attack objective, a common attack vector is through that of outsourced data collection (Gu et al., 2019a; Chen et al., 2017; Shafahi et al., 2018; Zhu et al., 2019; Saha et al., 2020; Lovisotto et al., 2020; Datta & Shadbolt, 2022b). Backdoor attacks (Suresh et al., 2019; Wang et al., 2020; Bagdasaryan et al., 2020; Huang, 2020) and poisoning attacks (Hayes & Ohrimenko, 2018; Mahloujifar et al., 2018; 2019; Chen et al., 2021; Fang et al., 2020) against federated learning systems and against multi-party learning models have been demonstrated, but with a single attacker intending to compromise multiple victims (i.e. single attacker vs multiple defenders); for example, with a single attacker controlling multiple participant nodes in the federated learning setup (Bagdasaryan et al., 2020); or decomposing a backdoor trigger pattern into multiple distributed small patterns to be injected by multiple participant nodes controlled by a single attacker (Xie et al., 2020). Datta & Shadbolt (2022) (Datta & Shadbolt, 2022a) evaluated the multi-agent backdoor attack, to evaluate multiple attackers against each other and a defender, and the attack objective is individualized for each attacker. Though some upsides presented in the work include a natural ro-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

bustness to backdoor attackers when more than one attacker attempts to backdoor a model (resulting in collective minimization of attack success rate attributed to the *backfiring effect*), further inspection of the setup leads us to believe there are remaining aspects of robustness to be addressed, one of which being the low accuracy w.r.t. clean labels in the presence of backdoor perturbations.

2.1. Multi-Agent Backdoor Attacks

After N attackers contribute private datasets D_i to joint dataset $\mathbb{D}$, the trained model evaluates backdoored inputs to compute the payoffs. Let $\mathcal{X} \in \mathbb{R}^{\mathtt{l}\times\mathtt{w}\times\mathtt{c}}$ and $\mathcal{Y} = 1, 2, ..., k$ be the corresponding input and output spaces. $\{D_i\}^N, \mathbb{D} \setminus \{D_i\} \sim \mathcal{X} \times \mathcal{Y}$ are sources of shifted $\mathcal{X}$:$\mathcal{Y}$ distributions from which an observation x can be sampled. x can be decomposed $x = \mathbf{x} + \varepsilon$, where $\mathbf{x}$ is the set of clean features in x, and $\varepsilon : \{\varepsilon \geq 0\}^{N+1}$ is the set of perturbations that can exist.

Each attacker is a player $\{a_i\}^{i\in N}$ that generates backdoored inputs X^{poison} to insert into their private dataset contribution $\{X^{\text{poison}} \in D_i\}^{i\in N} \in \mathbb{D}$. Attackers use backdoor attack algorithm b_i, which accepts a set of inputs mapped to target poisoned labels $\{X_i : Y_i^{\text{poison}}\} \in D_i$ to specify the intended label classification, backdoor perturbation rate ε_i to specify the proportion of an input to be perturbed, and the poison rate p_i to specify the proportion of the private dataset to contain backdoored inputs, to return $X^{\text{poison}} = b_i(X_i, Y_i^{\text{poison}}, \varepsilon_i, p_i)$. An attacker a_i aims to maximize the accuracy of the predicted outputs in test-time against the target poisoned labels as their payoff (Eqt 2), the *attack success rate* (ASR), which is the rate of misclassification of backdoored inputs X^{poison}, from the clean label Y_i^{clean} to the target poisoned label Y_i^{poison}, by the defender's model f. Each attacker optimizes their actions against the collective set of actions of the other $\neg i$ attackers.

$$\begin{aligned} x_i^{poison} &= x_i \odot (1 - z_i) + m_i \odot z_i \\ b : X_i^{\text{poison}} &:= \{x_i \odot (1 - z_i) + m_i \odot z_i\}^{x_i \in X_i^{\text{poison}}} \end{aligned} \tag{1}$$

We adopt b: *Random-BadNet* (Eqt 1) (Datta & Shadbolt, 2022a), an extension of the baseline backdoor attack algorithm BadNet (Gu et al., 2019b) with the adaptation that, instead of a single square trigger pattern, we generate randomized pixels such that each attacker has their own unique trigger pattern. Within the given dimensions (`length` l × `width` w × `channels` c) of an input $x \in X$, a backdoor trigger pattern is implemented as a mask m_i that replaces pixel values of x_i. m_i is a randomly-generated trigger pattern, sampled per attacker a_i. z_i is the binary mask (corresponding to m_i) returning 1 at the location of a perturbation and 0 everywhere else. $\odot$ is the element-wise product operator. Perturbation rate ε_i dictates the likelihood that an index pixel (l, w, c) will be perturbed, and is used to generate the shape mask. A higher ε_i results in higher density of perturbations. The poison rate is the proportion of the private dataset that is backdoored: $p = \frac{|X^{\text{poison}}|}{|X^{\text{clean}}|+|X^{\text{poison}}|}$.

A defender trains a model f on the joint dataset $\mathbb{D}$, which may contain backdoored inputs, until it obtains model parameters θ. The defender can query the joint dataset and contributions $\mathbb{D}$; whether the index of the train-time agent is known at test-time is also evaluated in this work. The defender can choose a model architecture (action r_j) and backdoor defense (action s_j). The multi-agent backdoor attack is focused on minimizing the collective attack success rate, hence the defender's payoff can be approximated as the complementary of the mean attacker payoff (Eqt 3).

$$\begin{cases} \pi^{a_i} = \mathtt{Acc}\Big(f(X_i; (\theta, \mathbb{D}); (r_j, s_j); \\ \qquad \{(\varepsilon_i, p_i, Y_i^{\text{poison}}, b_i), \\ \qquad (\varepsilon_{\neg i}, p_{\neg i}, Y_{\neg i}^{\text{poison}}, b_{\neg i})\}), \\ \qquad Y_i^{\text{poison}}\Big) & (2) \\ \pi^d = 1 - \dfrac{1}{N}\displaystyle\sum_i^N \mathtt{Acc}(f(\cdot), Y_i^{\text{poison}}) & (3) \end{cases}$$

We denote the collective attacker payoff and defender payoff as $\pi^a = \mathtt{mean} \pm \mathtt{std}$ and $\pi^d = ((1 - \mathtt{mean}) \pm \mathtt{std})$ respectively. The equilibrium $\widetilde{\pi}^{a,d}$ is the collective payoff (π^a, π^d) where both payoff values are maximized with respect to the dominant strategy taken by the other. Through the *backfiring effect*, Datta & Shadbolt (2022) (Datta & Shadbolt, 2022a) demonstrate the equilibrium $\widetilde{\pi}^{a,d}$ tends to $\frac{1}{|\mathcal{Y}|}$.

2.2. Backfiring effect

In a backdoor attack, the predicted label can be evaluated against the poison label or the clean label, resulting in 3 metrics: ① run-time accuracy of the predicted labels with respect to (w.r.t.) poisoned labels given backdoored inputs, ② run-time accuracy of the predicted labels w.r.t. clean labels given clean inputs, ③ run-time accuracy of the predicted labels w.r.t. clean labels given backdoored inputs. The defender's primary objective is to minimize the individual and collective attack success rate of a set of attackers (minimize ①), and its secondary objective to to retain accuracy against clean inputs (maximize ②). In the traditional backdoor setting, the defender is less concerned with maximizing ③ as long as ① is minimized. The *backfiring effect* has been beneficial as a natural defense against backdoor attacks, in particular with respect to the traditional metrics ① and ②.

(Observation 0: Backfire results in reduced accuracy w.r.t. clean labels against backdoored inputs) An observation made in Datta & Shadbolt (2022) (Datta & Shadbolt, 2022a), denoted as the *backfiring effect*, is that if $N \to \infty$, then the predicted label is uniformly sampled

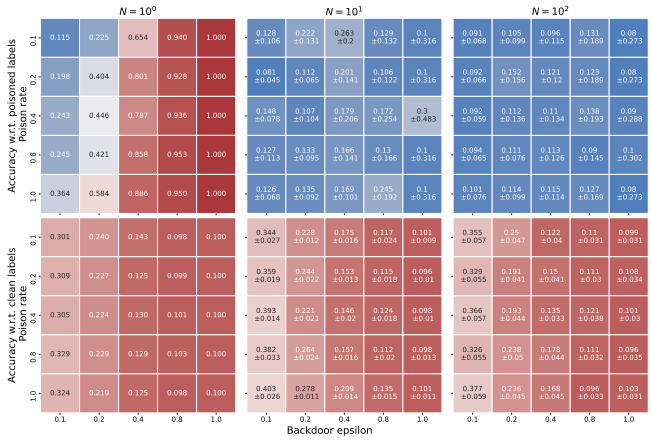


Figure 1: Side-effect from the *backfiring effect*: Though mitigable in the single-agent setting, single-agent defenses cannot sufficiently mitigate the loss in accuracy w.r.t. clean labels in the multi-agent setting.

$y^* = f(\mathbf{x} + \varepsilon; \theta) \sim \mathcal{U}(\mathcal{Y})$ s.t. $\mathbf{P}(y^*) = \frac{1}{|\mathcal{Y}|}$. Refactoring this for an ℓ-layer neural network, if the $(\ell - 1)$th layer is the layer before the prediction layer that returns class probabilities, and ϕ_i is a specific subnetwork responsible for classifying $\{\varepsilon_i \mapsto \phi_i\}$, then it follows that, while the *backfiring effect* induces $f_{\ell-1}(\varepsilon; \theta + \phi_i) = \{y : \frac{1}{|\mathcal{Y}|}\}$, it also induces random sampling for clean labels.

$$\begin{aligned} y^* &= f(\mathbf{x} + \varepsilon; \theta) \sim \mathcal{U}(\mathcal{Y}) \text{ s.t.} \\ y^* &:= \arg\max_{y \sim \mathcal{Y}} \{ f_{\ell-1}(\mathbf{x}; \theta) + \sum_{i}^{N+1} f_{\ell-1}(\varepsilon; \theta + \phi_i) \} \\ &:= \arg\max_{y \sim \mathcal{Y}} \{ f_{\ell-1}(\mathbf{x}; \theta) + \{ y : \frac{1}{|\mathcal{Y}|} \} \} \end{aligned} \tag{4}$$

Validating with empirical evaluation (Figure 1), we find that one of the side effects from the *backfiring effect* is not merely, given a backdoored input that the accuracy w.r.t. poison labels tend to $\frac{1}{|\mathcal{Y}|}$, but the accuracy w.r.t. clean labels also tends to $\frac{1}{|\mathcal{Y}|}$. Hence, to mitigate ③, we explore methods that result in $f_{\ell-1}(\varepsilon; \theta + \phi_i) = \{y : 0\}$, i.e. $|\phi_i| \to 0$.

The question that motivates this paper is what strategy can a defender take to further maximize its welfare? To suppress the equilibrium accuracy w.r.t. poison labels, it is shown in Datta & Shadbolt (2022) (Datta & Shadbolt, 2022a) that a (single-agent) defense should be employed. Since the accuracy w.r.t. poison labels is already at the lower bound, it is not meaningful to pursue a variant multi-agent defense that further attempts to suppress the accuracy w.r.t. poison labels, at least given the current knowledge of its failure modes. Hence, to make the most out of a deployed defense and maximize defender welfare, we motivate this work on multi-agent defenses to jointly minimize all 3 metrics, in particular maximizing accuracy w.r.t. clean labels in the presence of backdoor perturbations.

Algorithm 1 AgentSubspace: Train

trainSpace $(f, \{\theta_i\}_{i=1}^N, \{D_i\}_{i=1}^N, \beta, E)$
  Input : model f, parameters $\{\theta_i\}_{i=1}^N$, train set $\{D_i\}_{i=1}^N$, coefficient β, epochs E
  Output : Trained parameters $\{\theta_i\}_{i=1}^N$
  Constant initialization $\{\theta_{\text{init}} \leftarrow \theta_i\}_{i=1}^N$
  Agent-indexed sets $\{S_i\}_{i=1}^N \leftarrow \{\bigcup_{j \neq i}^N D_j\}_{i=1}^N$
  Train each parameter against their agent-indexed set in parallel per epoch
  foreach epoch $e \in E$ **do**
    foreach $\theta_i, S_i \in \{\theta_i\}_{i=1}^N, \{S_i\}_{i=1}^N$ **do**
      foreach batch $x, y \in S_i$ **do**
        $\hat{y} \leftarrow f(x, \theta_i)$
        $\mathcal{L} \leftarrow \ell(\hat{y}, y) + \beta \cos(\{\theta_i\}_{i=1}^N)$
        Backprop θ_i w.r.t. $\mathcal{L}$
  return $\{\theta_i\}_{i=1}^N$

2.3. Compressed Low-Loss Subspaces

We are interested in exploring low-loss compressed parameter spaces. Constructing subspaces enables us to sample diverse ensembles and parameters, which has been indicated to perform well against distribution shifts. Fort et at. (2020) (Fort et al., 2020) and Wortsman et al. (2021) (Wortsman et al., 2021) validate the robustness of sampled ensembles against adversarial perturbations. Mirzadeh et al. (2020) (Mirzadeh et al., 2020) validate its ability to connect parameters of different tasks in a continual learning setting. Wortsman et al. (2021) (Wortsman et al., 2021) validates its ability to improve general uncertainty estimation. Ratzlaff et al. (2020) (Ratzlaff & Fuxin, 2020) validates its ability to improve out-of-distribution robustness.

Let $\Theta \in \mathbb{R}^n$ be an n-dimensional manifold containing possible parameters θ to different input distributions. Let $\theta_i \in \Theta$ be a parameter sampled from the parameter space of index

i. We measure the loss $\mathcal{L}(\{x : y\}; \theta_i)$ with respect to function $f(\{x : y\}; \theta_i)$. If trained independently and without coordination, the subspaces trained for i and $\neg i$ may not overlap (disjoint). Hence, we would be motivated to craft one subspace with joint transferable regions.

Suppose we have two input:output distributions $(x_i : y_i) \sim D_i$ and $(x_{\neg i} : y_{\neg i}) \sim D_{\neg i}$. In order for parameter point-estimates θ_i and $\theta_{\neg i}$ to converge towards each other, the minimization of distance between the two points $\text{dist}(\theta_i, \theta_{\neg i})$ should reward an overlapping gradient update against D_i and $D_{\neg i}$:

$$\frac{\partial[\mathcal{L}(x_i, y_i; \theta_i) + \text{dist}(\theta_i, \theta_{\neg i})]}{\partial \theta_i} \cdot \frac{\partial[\mathcal{L}(x_{\neg i}, y_{\neg i}; \theta_{\neg i}) + \text{dist}(\theta_i, \theta_{\neg i})]}{\partial \theta_{\neg i}} > 0$$

For shifted inputs x_j, we would like to maximize the probability that the compressed subspace Θ contains the closest approximate low-loss parameter $x_j : \theta_j$, i.e. $\max \mathbf{P}(\theta_j \in \Theta)$. Suppose origin x_0 is shifted to x_j, and therefore requires a shift in low-loss parameters $x_0 : \theta_0 \rightarrow x_j : \theta_j$. θ_j would be computed from $\theta_j = \theta_0 + \Delta\theta$. θ can be composed of a set of subnetworks ϕ_n: $\Delta\theta = \bigcup_n^N \phi_n$. Hence, θ_j should contain a sufficient number of subnetworks needed in order to correctly approximate $x_j : y_j$. The intuition is, if the subspace is more compressed, then the likelihood that a randomly-sampled point between θ_i and $\theta_{\neg i}$ to contain low-loss θ_j (if x_j is interpolatable between x_i and $x_{\neg i}$) would be higher. The likelihood $\mathbf{P}(\theta_j \in \Theta)$ to exist in Θ would approximate the likelihood that there exists the closest approximate parameter $\theta_k \in \Theta$ such that loss $\mathcal{L}(f(\theta_k; x_j), y_j) \leq l$ is below threshold l.

$$\begin{aligned}\mathbf{P}(\mathcal{L}(f(\theta_k; x_j), y_j) \leq l) &\approx \mathbf{P}(\theta_j \in \Theta) \\ &\approx \mathbf{P}(\theta_k \in \Theta \mid \theta_k = \arg\min_{\theta_k \in \Theta} \text{dist}(\theta_j, \theta_k))\end{aligned} \tag{5}$$

Suppose $\text{dist}(\theta_a, \theta_b)$ is a distance metric that measures the degree of overlap between sets $\theta_a = \bigcup_a^A \phi_a$ and $\theta_b = \bigcup_b^B \phi_b$. The underlying assumption is that if θ_a and θ_b share more subnetworks between them relative to the total number of subnetworks they both require, then the more similar θ_a is to θ_b. Insufficient subnetwork overlap is indicative that Θ-space does not contain any sub-function approximators for this specific input distribution x_j, and that no viable sequence of subnetwork insertion/deletion can be traversed within Θ-space to obtain θ_j. Based on the Jaccard distance, we define $\text{dist}(\theta_a = \bigcup_a^A \phi_a, \theta_b = \bigcup_b^B \phi_b)$ as:

$$\text{dist}(\theta_a, \theta_b) = 1 - \frac{|(\bigcup_a^A \phi_a) \cap (\bigcup_b^B \phi_b)|}{|\bigcup_a^A \phi_a| + |\bigcup_b^B \phi_b| - |(\bigcup_a^A \phi_a) \cap (\bigcup_b^B \phi_b)|} \tag{6}$$

For Algorithm 1, if we can minimize the loss till convergence $\mathcal{L} \rightarrow 0$ for $\min \mathcal{L} = \mathcal{L}(f(\theta_a; x_a), y_a) + \mathcal{L}(f(\theta_b; x_b), y_b) + \text{dist}(\theta_a, \theta_b)$, then it implies we have minimized the regularization component $\text{dist}(\theta_a, \theta_b)$, hence the expected distance between a required θ_j and a sampled parameter θ_k has a tendency to be lower: $\min \mathbb{E}_{\theta_j, \theta_k \sim \Theta}[\text{dist}(\theta_j, \theta_k)]$. The implication of a low average distance between parameters is that for a given universe of subnetworks $\{\bigcup_n^N \phi_n\}^{N \in |\Theta|} \setminus \{\bigcap_n^N \phi_n\}^{N \in |\Theta|}$, we maximize the number of shared subnetworks $\{\bigcap_n^N \phi_n\}^{N \in |\Theta|}$. In other words, we maximize the density of transferrable subnetworks $\{\bigcap_n^N \phi_n\}^{N \in |\Theta|}$ that exist in Θ. This results in $\max |\{\bigcap_n^N \phi_n\}^{N \in |\Theta|}|$, implying that we maximize the likelihood that a subnetwork in the subnetwork universe exists in θ_j or vice versa that a subnetwork in θ_k exists in the subnetwork universe, i.e. $\max \mathbf{P}((\theta_k = \bigcup_m^M \phi_m) \cap \{\bigcap_n^N \phi_n\}^{N \in |\Theta|})$, i.e. $\min \mathbb{E}_{x \sim \mathcal{X}, \theta_k \sim \Theta}[\text{dist}(\theta_j, \theta_k)]$. Since the expected distance between θ_j and θ_k is lower, based on equation 5, it increases the likelihood of $\mathbf{P}(\theta_j \in \Theta)$. In other words, on condition that we compress Θ-space and loss converges, it is implied that θ_j has a higher likelihood of existing in Θ than without regulating Θ-space.

Existing literature (Draxler et al., 2019; Garipov et al., 2018; Frankle et al., 2020; Fort & Jastrzebski, 2019; Goodfellow et al., 2015; Li et al., 2018; Izmailov et al., 2019a;b; Fort et al., 2020; Wortsman et al., 2020; Ratzlaff & Fuxin, 2020; von Oswald et al., 2021; Wortsman et al., 2021; Nunez et al., 2021) investigate how to connect optima given random initializations. We start with a constant initialization, and we would be amongst the first to evaluate the compression of a parameter subspace with respect to multiple different input:output distributions. Prior work train for explicit pathways and connections between optima; we focus on compressing the space without explicit linear interpolation regularization, and find that random sampling points for ensembling can return sufficient low-loss parameters for inference, and evaluate its fitness as a robustness measure against joint distribution shift.

Algorithm 2 AgentSubspace: Evaluate

evalSpace $x, \{\theta_i\}_{i=1}^N$
  Input : Input x, Trained parameters $\{\theta_i\}_{i=1}^N$
  Output : Predicted output $\hat{y}$
  Sample $\alpha_i \sim [0, 1]$ s.t. $\sum_{i=1}^N \alpha_i = 1$
  $\theta \leftarrow \sum_{i=1}^N \alpha_i \theta_i$
  $\hat{y} \leftarrow f(x, \theta)$
  return $\hat{y}$

3. Multi-Agent Backdoor Defenses

3.1. Agent Subspace

Based on our discussion in Section 2.3, we propose the Agent Subspace defense against multi-agent backdoor attacks, in particular to maximize the accuracy w.r.t. clean

labels. Referring to Algorithm 1, we initialize N parameters at the same point, train parameters in parallel such that the cosine distance between their weights matrices is minimized, where each parameter θ_i is evaluated on its corresponding agent indexing dataset. During inference (Algorithm 2), we can compute with the subspace centre, or sample and ensemble $N = 1000$ point-estimates. We train the subspace for 5000 epochs, with 3 attackers, and $\beta = 0.05$. Both the agent indexing and agent subspace methods are designed to explicitly bias the training towards allocating more model capacity to $\neg\phi_i$ (i.e. $|\phi_i| \rightarrow 0$).

3.2. Agent Indexing

In agent indexing, a baseline multi-agent backdoor defense, the defender has knowledge of agent index/identity between train-time and test-time to isolate the learning effects introduced by data provided by specific agents (including triggers). The intuition of this defense is to leverage the non-uniformity of backdoor perturbations and subnetworks between agents, as the backdoor subnetwork of $\neg i$ backfiring agents is unlikely to perform well across other unique backdoor perturbations. In this case, we assume that the number of agents that interact in the system is known and that at test-time the defender knows which agent provided the sample to run inference on. This form of inference, where an adaptation method changes parameters given some distributional meta-data or headers, is frequently used in other distribution shift settings such as continual learning or domain adaptation, and is also practically-relevant.

Rather than training a single model with all the available data, the defender trains a set of models, one for each agent. For an agent a_i, its corresponding model will only be trained on data S_i provided by different agents $\neg i$, i.e. $\{S_i\}_{i=1}^N \leftarrow \{\bigcup_{j \neq i}^N D_j\}_{i=1}^N$. At test-time, the defender selects which model to use based on the agent requesting the inference.

3.3. Agent Augmentation

In agent augmentation, a baseline multi-agent backdoor defense, a defender simulates the presence of fake backdoor attackers and introduce simulated backdoored samples in the training data. The implementation is similar to backdoor adversarial training (Geiping et al., 2021) in the respect that we make use of the backdoor attack algorithm to poison our own dataset; the difference is that in order to simulate an augmented number of agents, we specifically set the number of agents to be high ($N = 200$ here, while $N = 20$ in backdoor adversarial training). This defense scales the *backfiring effect* to ensure the accuracy w.r.t. poison labels stays low, but expectedly the accuracy w.r.t. clean labels should not be greatly improved.

Defenders can insert backdoor triggers into subsets of their clean dataset (the sub-dataset that the defender contributes). To evaluate agent augmentation, we allocate 40% of the defender's contributed sub-dataset to be clean, and the remaining 60% of samples to be poisoned with defender-generated triggers. The defender sets the number of simulated attackers n, then generates unique random trigger patterns (one per attacker). Each of the n simulated attackers is allocated an equal amount of samples out of the total 60%; we set the poison rate of simulated attackers $\varepsilon, p = 0.2$.

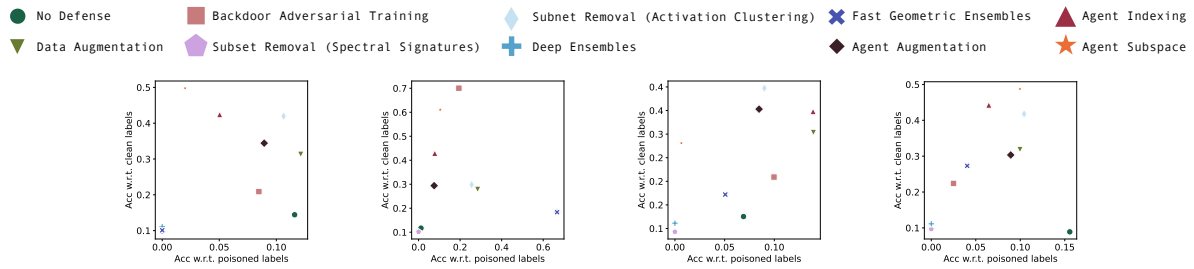
4. Evaluation

4.1. Methodology

We evaluate upon CIFAR10 (10 classes, 60,0000 inputs, 3 colour channels) (Krizhevsky, 2009). The traintime-testtime split of each attacker is 80-20% (80% of the attacker's private dataset is contributed to the joint dataset, 20% reserved for evaluating in test-time). The train-validation split for the defender was 80-20% (80% of joint dataset used for training, 20% for validation). We trained a small CNN (channels $[16, 32, 32]$; $15,722$ parameters) model with batch size 128 with early stopping when loss converges (approximately 30 epochs, validation accuracy of $92 - 93\%$; loss convergence depends on joint dataset structure and number of attackers). We use a Stochastic Gradient Descent optimizer with 0.001 learning rate and 0.9 momentum, and cross entropy loss function. For N attackers and V_d $(= 0.1)$ being the proportion of the dataset allocated to the defender, the real poison rate (the absolute proportion of the joint dataset that is backdoored by attacker a_i) is calculated as $\rho = (1 - V_d) \times \frac{1}{N} \times p$. For attackers, we generate backdoor perturbations with Random-BadNet (Datta & Shadbolt, 2022a), hold $p, \varepsilon = 0.4$, backdoor target labels are randomly chosen by each attacker. Seed is 3407 for all procedures except those requiring each attacker to have distinctly different randomly-sampled values (e.g. trigger pattern generation), in which cases the seed is the index of the attacker.

The primary variables of investigation in this work are the existing single-agent and hypothesized multi-agent defenses. We evaluate 2 augmentative (data augmentation, backdoor adversarial training), 2 removal (spectral signatures, activation clustering) defenses, and 2 ensembling methods (as baseline to our hypotheses). For augmentative defenses, 50% of the defender's allocation of the dataset is assigned to augmentation: for $V_d = 0.8$, 0.4 is clean, 0.4 is augmented. We set the number of trained point-estimates to be the number of attackers, e.g. 3 ensemble models, 3 parameters trained in parallel for Agent Subspace.

Data Augmentation (Borgnia et al., 2021) (e.g., CutMix (Yun et al., 2019) or MixUp (Zhang et al., 2018)) has been shown to reduce backdoor attack success rate. We implement CutMix (Yun et al., 2019), where augmentation



(a) Base: CIFAR10, CNN, $\alpha = 0$ (b) SVNH (c) ResNet50 (d) style $\alpha = 1$

Figure 2: Trade-off between the accuracy w.r.t. poison and clean labels: The ideal backdoor defense should be able to retain a low accuracy w.r.t. clean label (baseline: No Defense), while maximizing accuracy w.r.t. poison label; i.e. ideal defenses are in the top-left region of each ablation plot.

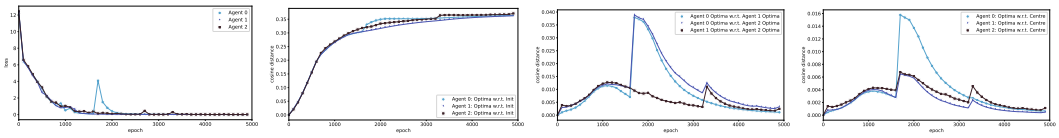


Figure 3: Convergence of loss w.r.t. respective inputs and cosine distance: as epoch increases, SGD finds an equilibrium subspace for sampling across distributions.

takes place per batch, and training completes in accordance with aforementioned early stopping. **Backdoor Adversarial Training (Geiping et al., 2021)** extend the concept of adversarial training on defender-generated backdoor examples to insert their own triggers to existing labels. We implement backdoor adversarial training (Geiping et al., 2021), where the generation of backdoor perturbations is through Random-BadNet (Datta & Shadbolt, 2022a), where 50% of the defender's allocation of the dataset is assigned to backdoor perturbation, $p, \varepsilon = 0.4$, and 20 different backdoor triggers used (i.e. allocation of defender's dataset per backdoor trigger pattern is $(1 - 0.5) \times 0.8 \times \frac{1}{20}$).

Spectral Signatures (Tran et al., 2018) is an input inspection method used to perform subset removal from a training dataset. For each class in the backdoored dataset, the method uses the singular value decomposition of the covariance matix of the learned representation for each input in a class in order to compute an outlier score, and remove the top scores before re-training. In-line with existing implementations, we remove the top 5 scores for $N = 1$ attackers. For $N = 10$, we scale this value accordingly and remove the top 50 scores. **Activation Clustering (Chen et al., 2018)** is an input inspection method used to perform subset removal from a training set. In-line with Chen et al. (2018) (Chen et al., 2018)'s implementation, we perform dimensionality reduction using Independent Component Analysis (ICA) on the dataset activations, then use k-means clustering to separate the activations into two clusters, then use Exclusionary Reclassification to score whether a given cluster corresponds to backdoored data and remove it.

Deep Ensembles (Lakshminarayanan et al., 2017) is a strong ensembling baseline for uncertainty estimation and sampling parameters for ensembling. It is constructed through training multiple networks independently in parallel with adversarial training. **Fast Geometric Ensembles (Garipov et al., 2018)** train multiple networks to find their respective modes such that a high-accuracy path is formed that connects the modes in the parameter space. The method collects parameters at different checkpoints, which can subsequently be used as an ensemble.

4.2. Findings

(Observation 1: Compressed subspace sampling is a Pareto-optimal strategy) Across wide variations, we find that sampling with an agent subspace yields the optimal trade-off between maximizing accuracy w.r.t. clean labels and mininizing accuracy w.r.t. poisoned labels.

Summarized in Tables 2 and 1, where we vary the poison parameters $p = \varepsilon$ and number of attackers respectively, the agent subspace method outperforms for most evaluated values. Evaluating on SVNH (digit dataset of 10 classes, 630,420 inputs, 3 colour channels) (Netzer et al., 2011) (Figure 2b), on ResNet-50 (He et al., 2015) for the same training configurations (Figure 2c), and $\alpha = 1.0$ stylization with Adaptive Instance Normalization (AdaIN) (Huang & Belongie, 2017) and the Paintings by Numbers dataset (in-line with Datta & Shadbolt (2022) (Datta & Shadbolt, 2022a)

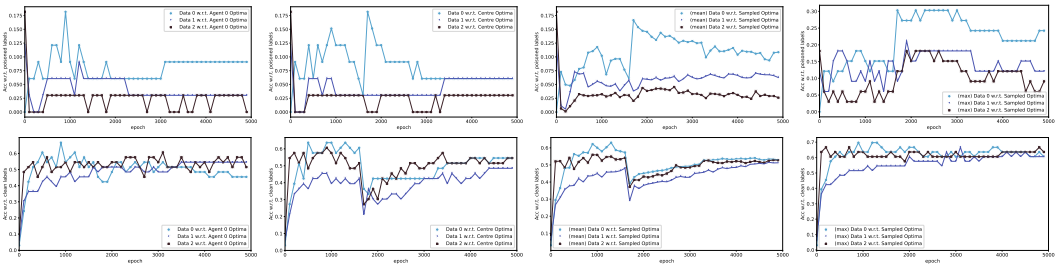


Figure 4: Accuracy plots of agents with respect to each other

and (Geirhos et al., 2019)) (Figure 2d), we observe that the agent subspace method consistently remains in the top-left region while competing defenses sporadically outperform marginally in different settings.

Compared to single point-estimate defenses (e.g. agent augmentation), subspace sampling methods and other ensembling methods have demonstrated marginal utility from sampling diverse regions of the functon space. Not only does the agent subspace method outperform all the augmentative and removal strategies, but it also outperforms the agent indexing defense, presumably the upper bound as it is the boundary of the constructed subspace; accuracy w.r.t. clean labels exceed that at this point due to the diversity of functions sampled in ensemble. Compared to non-agent-dynamics-driven strategies (e.g. ensembles), the marginal performance gain is validation that exerting correctness or structure to the subspace for sampling can yield marginal benefits, that subspaces expected to handle multiple distribution shifts should be crafted with multiple distributions in mind, rather than aggregating 1 training set.

We observe extended optionality in inference with the agent subspace method. Based on Figure 4, upon convergence, evaluating the test-time backdoored set with either the centre of the subspace or ensembling 1000 point-estimates return a similar terminal accuracy w.r.t. clean labels. This implies various storage strategies, ranging from storing a single centre point-estimate, or storing the 3 boundary point-estimates for each attacker. We would recommend pursuing further study into crafting efficient heuristics to navigate the subspace: based on the maximum accuracy obtained from a sampled point, we observe that there do exist points within the subspace that offer at least 10% gains on accuracy w.r.t. clean labels. Possible heuristics may be to make use of meta-data to use as headers (similar to agent indexing), or making use of test-time inputs for experience replay.

(Observation 2: Distributional robustness play an important role in subspace construction) We motivated the 3 defenses based on agent dynamics, which is a manifestation of (joint) distribution shift. The inclusion of multiple train-time perturbations mapped to different labels is a form of joint label shift, and the stylization of each attacker's private dataset introduces additional domain/texture shift; agent subspacing has exemplified robustness against these different variations of joint distribution shift. The marginal utility of the agent subspace methods indicates intrinsic robustness to joint distribution shift.

The performance of this method is implied to be applicable to tasks with *shift-agnosticity* rather than *shift-specificity*. In the multi-agent backdoor attack, or other task setups where the defender would like to craft a robust adaptation method that performs inference with respect to the source distribution rather than the target distribution, the agent subspace may contain point-estimates that contain specific subnetworks corresponding to specific distributional shift types (e.g. specific tasks or domains), but an expectedly larger proportion of the subspace would contain overlapping, transferable subnetworks. The inference strategies in this work, including inference with the centre or sampling across the entire subspace, empirically performs well when the shifted distributions share one common source distribution, but a heuristic-sampling method could be investigated for searching for specific subnetworks for specific target distributions.

Most work in mode connectivity and subspace construction rely on a single training set, even when showing improvements to adversarial robustness. Reflecting on single-agent defenses, we train on segregated subpopulations, particularly if the distance between the input distributions in the input space is increasingly distant (e.g. with stylization). If we train each agent's point-estimate on their own private dataset and compress the subspace, the accuracy w.r.t. clean labels is also low, and this may occur because (similarly with the agent-indexing defense): (i) the *backfiring effect* would occur within that sub-dataset (and would not occur if there is only 1 backdoor perturbation type), and (ii) the parameters are expected to learn backdoor subnetworks to minimize loss w.r.t. inputs.

Most work in mode connectivity and subspace construction make use of random initializations for each point-estimate. This has resulted in lower accuracy in clean labels, and we could attribute this to: (i) additional difficulty in overcoming the marginal parameter space distance between the random initializations, or (ii) the optimization trajectories taken by

Num attackers	Acc w.r.t. poison labels			Acc w.r.t. clean labels		
	2	*5*	*10*	*2*	*5*	*10*
No Defense	39.0 ± 22.0	28.5 ± 21.2	21.0 ± 19.1	12.5 ± 1.5	13.5 ± 4.1	11.5 ± 7.1
Data Augmentation	12.8 ± 0.0	11.0 ± 1.9	10.1 ± 7.6	4.65 ± 2.5	52.6 ± 3.7	44.0 ± 7.3
Backdoor Adversarial Training	10.0 ± 0.0	7.7 ± 3.9	9.9 ± 3.0	37.5 ± 4.5	41.5 ± 3.9	8.5 ± 1.5
Spectral Signatures	0.0 ± 0.0	0.1 ± 0.2	3.7 ± 7.1	9.25 ± 1.0	10.6 ± 1.4	10.9 ± 2.5
Activation Clustering	10.7 ± 3.1	9.44 ± 4.5	9.1 ± 7.3	43.0 ± 0.0	45.0 ± 1.3	44.8 ± 3.5
Deep Ensembles	50.0 ± 50.0	20.0 ± 40.0	10.0 ± 30.0	8.0 ± 0.0	11.0 ± 1.2	10.0 ± 6.7
Fast Geometric Ensembles	0.0 ± 0.0	0.0 ± 0.0	8.0 ± 7.5	9.0 ± 0.0	10.0 ± 2.7	32.0 ± 11.9
Agent Augmentation	6.0 ± 3.0	7.5 ± 3.5	10.0 ± 6.4	24.5 ± 3.5	36.5 ± 7.5	34.9 ± 4.3
Agent Indexing	5.5 ± 2.5	9.0 ± 3.4	10.4 ± 5.5	32.0 ± 2.0	49.5 ± 5.3	55.8 ± 5.2
Agent Subspace	5.132 ± 0.9	3.41 ± 2.7	10.1 ± 3.3	48.5 ± 0.5	55.0 ± 0.0	62.9 ± 3.7
Δ: random init.	10.1 ± 6.8	1.34 ± 1.3	0.67 ± 1.0	20.8 ± 0.3	13.5 ± 1.0	14.9 ± 2.3
Δ: train on respective dataset	5.79 ± 2.7	4.83 ± 3.5	7.91 ± 6.1	34.1 ± 2.7	35.8 ± 3.4	45.8 ± 11.1
Δ: train sequentially	8.71 ± 0.2	5.7 ± 1.4	10.4 ± 7.1	32.5 ± 0.1	35.4 ± 4.2	44.0 ± 10.0
Δ: max. cosine distance	7.72 ± 1.0	16.1 ± 14.9	3.94 ± 2.8	36.3 ± 0.4	15.1 ± 1.3	14.7 ± 3.4

Table 1: Defenses: Carefully-configured agent subspace is robust to shifts.

Backdoor τ & Poison rate	Acc w.r.t. poison labels					Acc w.r.t. clean labels				
	0.1	*0.2*	*0.4*	*0.8*	*1.0*	*0.1*	*0.2*	*0.4*	*0.8*	*1.0*
Data Augmentation	12.1 ± 6.5	12.1 ± 6.5	12.1 ± 8.5	19.2 ± 12.5	29.3 ± 17.5	31.3 ± 7.95	30.3 ± 6.6	31.3 ± 5.2	33.3 ± 2.5	28.3 ± 3.8
Backdoor Adv Training	8.47 ± 8.8	8.97 ± 7.6	8.47 ± 6.1	6.97 ± 3.0	7.0 ± 1.4	19.4 ± 4.4	21.4 ± 2.8	20.9 ± 3.2	20.9 ± 1.2	21.4 ± 1.4
Spectral Signatures	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	9.57 ± 1.3	9.83 ± 1.4	9.67 ± 0.6	9.57 ± 1.6	9.73 ± 1.5
Activation Clustering	9.23 ± 2.0	8.63 ± 3.3	10.6 ± 2.8	14.8 ± 0.9	15.1 ± 5.7	53.2 ± 2.1	48.1 ± 1.2	42.0 ± 1.6	34.4 ± 0.4	32.2 ± 1.1
Deep Ensembles	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	11.1 ± 3.8	11.1 ± 3.8	11.1 ± 3.8	11.1 ± 3.8	11.1 ± 3.8
Fast Geometric Ensembles	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	17.2 ± 10.3	17.2 ± 5.7	10.1 ± 6.2	10.1 ± 6.2	10.1 ± 6.2	21.2 ± 7.4	17.2 ± 2.8
Agent Augmentation	10.8 ± 0.5	9.97 ± 0.6	8.93 ± 1.7	7.63 ± 1.7	15.0 ± 2.9	44.6 ± 2.6	40.8 ± 3.6	34.4 ± 2.1	37.0 ± 2.0	33.6 ± 0.9
Agent Indexing	6.07 ± 4.9	5.03 ± 5.1	5.03 ± 5.1	13.1 ± 7.6	11.1 ± 10.0	43.8 ± 7.6	46.0 ± 4.9	46.0 ± 8.5	44.9 ± 10.3	37.2 ± 10.3
Agent Subspace	4.66 ± 2.8	5.84 ± 2.2	2.01 ± 1.7	5.57 ± 2.7	3.15 ± 0.2	50.0 ± 6.4	50.0 ± 6.4	49.8 ± 6.1	43.7 ± 6.1	46.1 ± 5.1

Table 2: Defenses: Agent subspace is the optimal strategy.

each training point-estimate has sustained sufficient inertia that it optimizes towards minimizing loss w.r.t. inputs and ignores loss w.r.t. cosine distance.

Wortsman et al. (2021) (Wortsman et al., 2021) constructed a low-loss subspace by jointly training the point-estimates by sampling α coefficients during training to evaluate loss on weighted-average parameters, and additionally regularized for the maximization of cosine distance to include diverse point-estimates. Differently, when training on different distributions, we were not able to meaningfully converge the loss of the point-estimates when using α coefficients during training. Maximizing the cosine distance also resulted in reduced accuracy w.r.t. clean labels. We do not use α coefficients during train-time but during test-time for inference, yet the loss on test-time inputs is low. We suspect this contradiction is due to the interference between sampled subnetworks: when trained on a single distribution, functionally-different subnetworks mapped to different regions of the function space can stochastically adjust their position in the network given that, though functionally-different, they contribute to minimizing the same input loss (i.e. they are transferable subnetworks). When trained on multiple distributions, the subnetworks interfere with each other, and the subspace cannot be constructed such that a partial subnetwork on one end of the subspace can exist on the other end without resulting in skewness.

We evaluated the construction of the subspace through sequeuntial point-estimate training, where we optimize the first parameter to its agent-indexed set, then train all subsequent points while minimizing their cosine distances with respect to the first trained point. Despite sharing a source distribution, we observe a drop in accuracy w.r.t. clean labels. The implication is that, given an anchoring point-estimate, SGD will not search for an optimal subspace in the parameter space, and instead return interference between minimizing loss w.r.t. their respective dataset or minimizing cosine distance. If the former occurs, it is implied the subspace is neither compressed nor unlikely to carry transferable subnetworks of the source distribution. If the latter occurs, then the subsequently trained points will implicitly learn subnetworks skewed towards the first point, even though it has not encountered those backdoor perturbations in its own training set. During training in parallel, we also need to mitigate this skewness of subnetwork insertion: if the cosine distance during the initial epochs are too high and diverge, then the loss of the individual parameters may not converge, hence we set the regularization coefficient lower to reduce the loss terms imbalance.

A feature of this work is the intrinsic re-calibration taking place per epoch. Opposing the expectation of a continuously-decreasing trend in the cosine distance, we see a double/multiple descent phenomenon occurring, where while optimizing loss w.r.t. inputs the point-estimates are finding optimal subspace regions in the parameter space such that the cosine distance can continue to decrease, and the search for this region is what yields a temporary increase in cosine distance. We naturally yield a compressed low-loss subspace without the use of substantial construction tweaks such as alpha coefficients. If we stopped training at epoch 1000, we would retain skew and this would benefit some models, but if we continue training we find a terminal equilibrium accuracy w.r.t. clean labels that matches the mean at epoch 1000 but lower variance.

5. Conclusion

This work contributed a robust, practical defense against multi-agent backdoor attacks, specifically maximizing the accuracy w.r.t. clean labels which was harmed during backfire. We also contributed novel methodology to make use of compressed low-loss subspaces to mitigate distribution shifts.

References

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948. PMLR, 26–28 Aug 2020. URL http://proceedings.mlr.press/v108/bagdasaryan20a.html.

Borgnia, E., Cherepanova, V., Fowl, L., Ghiasi, A., Geiping, J., Goldblum, M., Goldstein, T., and Gupta, A. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3855–3859, 2021. doi: 10.1109/ICASSP39728.2021.9414862.

Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., and Srivastava, B. Detecting backdoor attacks on deep neural networks by activation clustering, 2018.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning, 2017.

Chen, Z., Tian, P., Liao, W., and Yu, W. Towards multi-party targeted model poisoning attacks against federated learning systems. *High-Confidence Computing*, pp. 100002, 2021. ISSN 2667-2952. doi: https://doi.org/10.1016/j.hcc.2021.100002. URL https://www.sciencedirect.com/science/article/pii/S2667295221000039.

Datta, S. and Shadbolt, N. Backdoors stuck at the frontdoor: Multi-agent backdoor attacks that backfire, 2022a.

Datta, S. and Shadbolt, N. Hiding behind backdoors: Self-obfuscation against generative models, 2022b.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially no barriers in neural network energy landscape, 2019.

Fang, M., Cao, X., Jia, J., and Gong, N. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622. USENIX Association, August 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/fang.

Fort, S. and Jastrzebski, S. Large scale structure of neural network loss landscapes, 2019.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective, 2020.

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis, 2020.

Gao, Y., Doan, B. G., Zhang, Z., Ma, S., Zhang, J., Fu, A., Nepal, S., and Kim, H. Backdoor attacks and countermeasures on deep learning: A comprehensive review, 2020.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns, 2018.

Geiping, J., Fowl, L., Somepalli, G., Goldblum, M., Moeller, M., and Goldstein, T. What doesn't kill you makes you robust(er): Adversarial training against poisons and backdoors, 2021.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. Qualitatively characterizing neural network optimization problems, 2015.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019a.

Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019b. doi: 10.1109/ACCESS.2019.2909068.

Hayes, J. and Ohrimenko, O. Contamination attacks and mitigation in multi-party machine learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/331316d4efb44682092a006307b9ae3a-Paper.pdf.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Huang, A. Dynamic backdoor attacks against federated learning, 2020.

Huang, X. and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. Subspace inference for bayesian deep learning, 2019a.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization, 2019b.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes, 2018.

Li, Y., Wu, B., Jiang, Y., Li, Z., and Xia, S.-T. Backdoor learning: A survey, 2021.

Lovisotto, G., Eberz, S., and Martinovic, I. Biometric backdoors: A poisoning attack against unsupervised template updating. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 184–197, 2020. doi: 10.1109/EuroSP48549.2020.00020.

Mahloujifar, S., Mahmoody, M., and Mohammed, A. Multi-party poisoning through generalized p-tampering, 2018.

Mahloujifar, S., Mahmoody, M., and Mohammed, A. Universal multi-party poisoning attacks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4274–4283. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/mahloujifar19a.html.

Mirzadeh, S. I., Farajtabar, M., Gorur, D., Pascanu, R., and Ghasemzadeh, H. Linear mode connectivity in multitask and continual learning, 2020.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Nunez, E., Horton, M., Prabhu, A., Ranjan, A., Farhadi, A., and Rastegari, M. Lcs: Learning compressible subspaces for adaptive network compression at inference time, 2021.

Ratzlaff, N. and Fuxin, L. Hypergan: A generative model for diverse, performant neural networks, 2020.

Saha, A., Subramanya, A., and Pirsiavash, H. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11957–11965, 2020.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

Suresh, A. T., McMahan, B., Kairouz, P., and Sun, Z. Can you really backdoor federated learning? 2019.

Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/280cf18baf4311c92aa5a042336587d3-Paper.pdf.

von Oswald, J., Kobayashi, S., Sacramento, J., Meulemans, A., Henning, C., and Grewe, B. F. Neural networks with late-phase weights, 2021.

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.-y., Lee, K., and Papailiopoulos, D. Attack of the tails: Yes, you really can backdoor federated learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16070–16084. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/b8ffa41d4e492f0fad2f13e29e1762eb-Paper.pdf.

Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., and Farhadi, A. Supermasks in superposition, 2020.

Wortsman, M., Horton, M., Guestrin, C., Farhadi, A., and Rastegari, M. Learning neural network subspaces, 2021.

Xie, C., Huang, K., Chen, P.-Y., and Li, B. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgyS0VFvr.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization, 2018.

Zhu, C., Huang, W. R., Li, H., Taylor, G., Studer, C., and Goldstein, T. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pp. 7614–7623. PMLR, 2019.