Wide Two-Layer Networks can Learn from Adversarial Perturbations

Soichiro Kumano
The University of Tokyo
kumano@cvm.t.u-tokyo.ac.jp

Hiroshi Kera
Chiba University, Zuse Institute Berlin
kera@chiba-u.jp

Toshihiko Yamasaki
The University of Tokyo
yamasaki@cvm.t.u-tokyo.ac.jp

Abstract

Adversarial examples have raised several open questions, such as why they can deceive classifiers and transfer between different models. A prevailing hypothesis to explain these phenomena suggests that adversarial perturbations appear as random noise but contain class-specific features. This hypothesis is supported by the success of perturbation learning, where classifiers trained solely on adversarial examples and the corresponding *incorrect labels* generalize well to correctly labeled test data. Although this hypothesis and perturbation learning are effective in explaining intriguing properties of adversarial examples, their solid theoretical foundation is limited. In this study, we theoretically explain the counterintuitive success of perturbation learning. We assume wide two-layer networks and the results hold for any data distribution. We prove that adversarial perturbations contain sufficient class-specific features for networks to generalize from them. Moreover, the predictions of classifiers trained on mislabeled adversarial examples coincide with those of classifiers trained on correctly labeled clean samples. The code is available at https://github.com/s-kumano/perturbation-learning.

1 Introduction

Adversarial examples [41], which are imperceptibly perturbed inputs designed to deceive machine learning models, have raised significant concerns about the robustness and reliability of these models. Despite their importance, the underlying mechanisms of adversarial examples are not yet fully understood. A prevailing hypothesis to explain the intriguing properties of adversarial examples is the "feature hypothesis" [22]. This hypothesis posits that adversarial perturbations, while appearing as imperceptible noise to humans, contain class-specific features. The feature hypothesis provides a unified explanation for several puzzling phenomena associated with adversarial examples, such as their ability to deceive classifiers, transferability across models, and so on (cf. Section 2.1).

Perturbation learning [22] provides empirical evidence supporting the feature hypothesis. In this learning, classifiers are trained *solely* on adversarial examples that are *mislabeled* in human perception,[1] yet they demonstrate remarkable generalization to clean test data (Fig. 1). For example, classifiers achieved 77% accuracy on the correctly labeled clean test dataset of CIFAR-10 [27], even though they

---

[1]This is the critical difference between perturbation learning and adversarial training or training with noisy labels. Perturbation learning shows the learnability *solely* from adversarial examples (e.g., a cat adversarial image) that *always* have *incorrect* labels (e.g., the bird label) to classify clean test images with the correct labels (i.e., bird clean images to the bird class). Perturbation learning does not aim to learn robustly against adversarial examples or noisy labels. Refer to Appendix A in [28] for further clarifications.
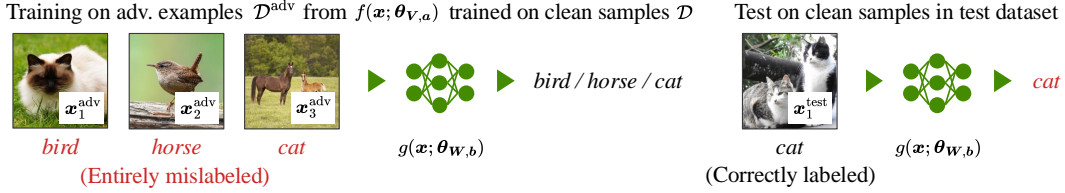
Training on adv. examples $\mathcal{D}^{\mathrm{adv}}$ from $f(\boldsymbol{x}; \boldsymbol{\theta}_{V,a})$ trained on clean samples $\mathcal{D}$ Test on clean samples in test dataset

$\boldsymbol{x}_1^{\mathrm{adv}}$ $\boldsymbol{x}_2^{\mathrm{adv}}$ $\boldsymbol{x}_3^{\mathrm{adv}}$ *bird / horse / cat* $\boldsymbol{x}_1^{\mathrm{test}}$ *cat*

bird   *horse*   *cat* $g(\boldsymbol{x}; \boldsymbol{\theta}_{W,b})$ *cat* $g(\boldsymbol{x}; \boldsymbol{\theta}_{W,b})$

(Entirely mislabeled)   (Correctly labeled)

Figure 1: Counterintuitive generalization of perturbation learning.[1] A classifier g is trained solely on mislabeled adversarial examples $\mathcal{D}^{\mathrm{adv}} := \{(\boldsymbol{x}_n^{\mathrm{adv}}, y_n^{\mathrm{adv}})\}_{n=1}^N$. These examples $\boldsymbol{x}_n^{\mathrm{adv}}$ are generated to mislead a classifier f, which is trained on correctly labeled clean samples $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, into predicting y_n^{adv} ($\neq y_n$). Surprisingly, despite being trained only on mislabeled data, the classifier g generalizes well to clean test samples. This counterintuitive result suggests that adversarial perturbations contain label-aligned class features, enabling the classifier g to generalize from them.

were trained on entirely mislabeled adversarial examples (e.g., a cat adversarial image labeled as a bird) [28]. This surprising result suggests that adversarial perturbations encode class-relevant features that enable classifiers to learn meaningful representations. However, despite the empirical support, the theoretical foundations of the feature hypothesis and perturbation learning remain limited. While a recent study [28] provided theoretical justifications, their results rely on stringent assumptions about data distribution, perturbation design, training procedure, and model architectures.

In this study, we theoretically address the understanding and justification of the feature hypothesis and perturbation learning. First, to support the feature hypothesis, we show that adversarial perturbations, while appearing as random noise, are parallel to the weighted sum of all training samples. This result suggests that a single perturbation derived from a classifier and input can potentially contain information about the entire training dataset. In particular, for some specific cases (e.g., when training samples are mutually orthogonal), perturbations include all training data and labels without loss of information. We then reveal that class features within perturbations enable classifiers to generalize from them. Specifically, under three mild conditions, the predictions of a classifier trained on adversarial perturbations are consistent with those of a classifier trained on correctly labeled clean samples. These three conditions can be interpreted from geometric and quantitative perspectives. Finally, we demonstrate that under similar conditions, the prediction agreement is observed between a classifier trained on mislabeled adversarial examples and one trained on correctly labeled clean samples, justifying the empirical success of perturbation learning.

Our analysis assumes two-layer neural networks with sufficient width but does not impose any assumptions on data distribution, which is a substantial progress from prior work [28] that considered mutually orthogonal training samples. In addition, our perturbation design, training procedure, activation functions, and bias availability are milder. In short, as shown in Tab. 1, except for the wide width assumption, our analysis requires milder conditions than prior work. Our contributions can be summarized as follows:

- We provide a theoretical justification for the feature hypothesis and perturbation learning using wide two-layer neural networks, considering any data distribution and realistic problem settings. Except for the wide width, our assumptions are substantially milder than [28].

- We demonstrate that adversarial perturbations are parallel to the weighted sum of training samples, suggesting that a single perturbation can potentially contain information about the entire training dataset. This result supports the feature hypothesis.

- We prove that under three mild conditions, the predictions of a classifier trained on perturbations are consistent with those of a classifier trained on correctly labeled clean samples. Moreover, under similar conditions, the prediction agreement between a classifier trained on mislabeled adversarial samples and one trained on clean samples is observed, providing a theoretical justification for the empirical success of perturbation learning.

2 Background and Related Work

2.1 Feature Hypothesis and Perturbation Learning

It has been hypothesized that adversarial perturbations contain class-specific features, although appearing as random noise [22]. This hypothesis, or feature hypothesis, offers a unified explanation for several open questions related to adversarial examples. For example, misclassification by classifiers and transferability across models [21, 41] can be attributed to the response to features within perturbations. Furthermore, according to this hypothesis, adversarially robust models achieve robustness by discarding brittle yet predictive features and focusing on more stable and semantically meaningful features. This interpretation explains the phenomena observed with robust models, such as the trade-off between accuracy and robustness [13, 32, 34, 35, 40, 42, 46, 47], perceptually-aligned gradients [1, 4, 7, 17, 18, 26, 38, 39, 42, 48], and enhanced transfer learning capabilities [1, 12, 37, 43].

Perturbation learning[1] [22] provides empirical support for the feature hypothesis. In perturbation learning, the dataset appears entirely mislabeled to human perception. However, the hypothesis suggests that adversarial perturbations in the dataset include label-aligned class features. Indeed, it has been observed that classifiers trained through perturbation learning can extract generalizable features from these perturbations and achieve high test accuracy (e.g., 92% for MNIST [11], 54% for Fashion-MNIST [45], and 77% for CIFAR-10 [27]), empirically justifying the feature hypothesis [22, 28].

While the feature hypothesis and perturbation learning are empirically effective in understanding adversarial examples, their theoretical foundations are very limited. Only one recent study [28] theoretically demonstrated that perturbations contain class features and that classifiers can generalize from them. However, their results relied on stringent conditions (e.g., mutually orthogonal training samples), which might not fully explain the success of perturbation learning in diverse settings.

In this study, for wide two-layer networks, we obtain results equivalent to those in [28] under more relaxed conditions (cf. Section 3.4). We provide the first theoretical justification for the feature hypothesis and perturbation learning under any data distribution and in a mild training setting.

2.2 Theoretical Framework: Lazy Training

Theoretical analysis of neural networks is generally challenging due to the non-convex nature of the loss surface. To address this, recent studies have focused on the lazy training regime, where the parameters of neural networks hardly change during training [5, 6, 9, 20, 25, 30, 33, 44, 49]. In this regime, neural networks behave almost linearly around their initialization, simplifying the learning dynamics. One of the key observation in lazy training is that, in wide two-layer neural networks, most derivatives of hidden outputs through (Leaky-) ReLU activation remain constant during training [30], which forms the basis of our theoretical framework (cf. Section 3.3). This observation has been extended to show that the neural tangent kernel remains invariant during training [2, 3, 14, 15, 23, 29].

In contrast, the feature learning regime, where parameters move significantly away from their initialization, has been explored in various studies [9, 20, 44]. Prior work on justifying perturbation learning [28] employs the feature learning regime, building on related findings in this area [19, 24, 31]. In our study, we adopt the lazy training regime and relax several conditions assumed in previous work [28] by introducing a wide width assumption (cf. Tab. 1). This adjustment is enabled by differences in the theoretical tools used.

3 Theoretical Results

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. For $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^d$, we denote the Euclidean norm by $\|\boldsymbol{z}_1\|$ and the inner product by $\langle \boldsymbol{z}_1, \boldsymbol{z}_2 \rangle$. Vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are called parallel and are denoted by $\boldsymbol{z}_1 /\!/ \boldsymbol{z}_2$ if there exists $C \in \mathbb{R}$ such that $\boldsymbol{z}_1 = C\boldsymbol{z}_2$. Let $\mathcal{N}(\mu, \sigma^2)$ be the Gaussian distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \geq 0$ and $U(\mathcal{S})$ be the uniform distribution on a set $\mathcal{S} \subset \mathbb{R}$. We use $\Omega(\,\cdot\,)$, $\Theta(\,\cdot\,)$, and $\mathcal{O}(\,\cdot\,)$ only to hide constant factors, and $\tilde{\Omega}(\,\cdot\,)$, $\tilde{\Theta}(\,\cdot\,)$, and $\tilde{\mathcal{O}}(\,\cdot\,)$ to hide polylogarithmic factors.

Table 1: Comparison with existing work [28]. With a wide network assumption, we improve the existing results from the perspective of data distribution, perturbation design, training time, loss function, and network architecture. Note that the non-bias and leaky-ReLU assumptions of [28] are critical for deriving their results. A detailed comparison can be found in Section 3.4.

	[28]	Ours
Training samples	Mutually orthogonal	**Any**
Perturbation type	Oracle-based	**Standard gradient-based**
Perturbation budget	Unrealistically tight	**Any**
Training time	Infinite	**Any**
Loss function	Exponential or logistic	**Differentiable, non-decreasing**
Network bias	Not available	**Available**
Activation	Leaky-ReLU	**ReLU and Leaky-ReLU**
Network width	**Any**	Sufficiently wide (but finite)
Theoretical framework	Feature learning	Lazy training
Common	Binary classification, two-layer network, gradient flow	

3.1 Problem Setup

In this study, we consider the dynamics of perturbation learning in binary classification problem with a two-layer neural network trained by gradient flow. First, we formally define the perturbation learning framework. The outline of perturbation learning is as follows: (i) train a classifier on correctly labeled clean samples, (ii) create adversarial samples based on the trained classifier, and (iii) train another classifier on the mislabeled adversarial samples.

Network trained on correctly labeled clean samples. We consider a two-layer neural network $f : \mathbb{R}^d \to \mathbb{R}$. Let $\boldsymbol{V} := (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)^\top \in \mathbb{R}^{m \times d}$ and $\boldsymbol{a} := (a_1, \ldots, a_m)^\top \in \mathbb{R}^m$ be the hidden weight and bias, respectively. We also describe $\boldsymbol{V} := (V_{ij})_{1 \leq i \leq m, 1 \leq j \leq d}$. Let $\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_m)^\top \in \mathbb{R}^m$ be the readout weight. While $\boldsymbol{V}$ and $\boldsymbol{a}$ are trainable, $\boldsymbol{\alpha}$ is fixed during training. Denote the trainable parameters by $\boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}} := \{\boldsymbol{V}, \boldsymbol{a}\}$. We initialize $V_{ij} \sim \mathcal{N}(0, 1/d)$, $a_i \sim \mathcal{N}(0, 1)$, and $\alpha_i \sim \mathcal{N}(0, 1/m)$ for each $i \in [m]$ and $j \in [d]$. The activation function is either ReLU or Leaky-ReLU $\phi(x) := \max(\gamma x, x)$ for $\gamma \in [0, 1)$. Finally, the network is given by $f(\boldsymbol{x}; \boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}) := \sum_{i=1}^m \alpha_i \phi(\langle \boldsymbol{v}_i, \boldsymbol{x} \rangle + a_i)$.

Network trained on mislabeled adversarial samples. Similarly to f, we define a network trained on mislabeled adversarial samples as $g(\boldsymbol{x}; \boldsymbol{\theta}_{\boldsymbol{W}, \boldsymbol{b}}) := \sum_{i=1}^m \beta_i \phi(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle + b_i)$. Note that the initializations of f and g are independent.

Loss function. We consider a differentiable, non-decreasing loss function $\ell : \mathbb{R} \to \mathbb{R}$, satisfying $\ell'(z) \geq 0$ for any $z \in \mathbb{R}$. Examples of such loss functions include the identity loss $\ell(z) := z$, exponential loss $\ell(z) := \exp(z)$, and logistic loss $\ell(z) := \ln(1 + \exp(z))$.

Training. We here describe the training process of the network f on correctly labeled clean samples. The training of g is similarly defined. Let $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N \subset \mathbb{R}^d \times \{\pm 1\}$ be a correctly labeled training dataset. The loss over $\mathcal{D}$ is defined as $\mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}; \mathcal{D}) := (1/N) \sum_{n=1}^N \ell(-y_n f(\boldsymbol{x}_n; \boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}))$. The network parameters are updated by gradient flow $\mathrm{d}\boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}(t)/\mathrm{d}t := -\partial \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}(t); \mathcal{D})/\partial \boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}$, where $t \geq 0$ is the training time. We consider $T_f > 0$ training steps, producing $f(\cdot; \boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}(T_f))$. For notational simplicity, we write $f(\cdot; t) := f(\cdot; \boldsymbol{\theta}_{\boldsymbol{V}, \boldsymbol{a}}(t))$.

Note that we do not consider whether $f(\cdot; T_f)$ perfectly classify $\mathcal{D}$. We discuss whether the classifier g, trained on adversarial examples crafted via $f(\cdot; T_f)$, can mimic the predictions of $f(\cdot; T_f)$.

Adversarial perturbations. We consider a single-step gradient-based perturbation, which is a common perturbation design [21]. An adversarial example $\boldsymbol{x}_n^{\mathrm{adv}} \in \mathbb{R}^d$ and its corresponding adversarial perturbation $\boldsymbol{r}_n \in \mathbb{R}^d$ are defined as follows:

$$\boldsymbol{x}_n^{\mathrm{adv}} := \boldsymbol{x}_n + \boldsymbol{r}_n, \qquad \boldsymbol{r}_n := -\epsilon \frac{\nabla_{\boldsymbol{x}_n} \ell(-y_n^{\mathrm{adv}} f(\boldsymbol{x}_n; T_f))}{\|\nabla_{\boldsymbol{x}_n} \ell(-y_n^{\mathrm{adv}} f(\boldsymbol{x}_n; T_f))\|}, \tag{1}$$

where $\epsilon > 0$ is the perturbation constraint and $y_n^{\mathrm{adv}} \in \{\pm 1\}$ is the target label. The adversarial perturbation $\boldsymbol{r}_n$ on $\boldsymbol{x}_n$ is designed to increase $y_n^{\mathrm{adv}} f(\boldsymbol{x}_n^{\mathrm{adv}}; T_f)$ under the constraint $\|\boldsymbol{r}_n\| \leq \epsilon$.

Mislabeled dataset. We consider two configurations of a dataset $\mathcal{D}^{\mathrm{adv}}$ for training g. First, we follow the original perturbation learning approach, where classifiers are trained on adversarial perturbations superposed on natural images, i.e., $\mathcal{D}^{\mathrm{adv}} := \{(\boldsymbol{x}_n^{\mathrm{adv}}, y_n^{\mathrm{adv}})\}_{n=1}^N$. This setting helps to understand the prior perturbation learning process. Second, we directly consider learning from perturbations rather than adversarial examples, i.e., $\mathcal{D}^{\mathrm{adv}} := \{(\boldsymbol{r}_n, y_n^{\mathrm{adv}})\}_{n=1}^N$. This setting directly addresses the question of whether classifiers can generalize from class features in perturbations.

Summary. The problem setting is summarized as follows:

Setting 3.1 (Perturbation learning). Independently initialize $V_{ij} \sim \mathcal{N}(0, 1/d)$, $W_{ij} \sim \mathcal{N}(0, 1/d)$, $a_i \sim \mathcal{N}(0, 1)$, $b_i \sim \mathcal{N}(0, 1)$, $\alpha_i \sim \mathcal{N}(0, 1/m)$, and $\beta_i \sim \mathcal{N}(0, 1/m)$ for each $i \in [m]$ and $j \in [d]$. Train a two-layer neural network f parameterized by $\boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}$ with (γ-scaled Leaky-) ReLU on a dataset $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ using gradient flow with a loss $\mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}; \mathcal{D})$ for training time $T_f > 0$. Create a dataset $\mathcal{D}^{\mathrm{adv}}$ by one of the following procedures with $\{y_n^{\mathrm{adv}}\}_{n=1}^N \in \{\pm 1\}^N$:

$$\text{Scenario (a)} \qquad \mathcal{D}^{\mathrm{adv}} := \{(\boldsymbol{r}_n, y_n^{\mathrm{adv}})\}_{n=1}^N, \tag{2}$$

$$\text{Scenario (b)} \qquad \mathcal{D}^{\mathrm{adv}} := \{(\boldsymbol{x}_n^{\mathrm{adv}}, y_n^{\mathrm{adv}})\}_{n=1}^N. \tag{3}$$

Train a two-layer neural network g parameterized by $\boldsymbol{\theta}_{\boldsymbol{W},\boldsymbol{b}}$ on the dataset $\mathcal{D}^{\mathrm{adv}}$ using gradient flow with a loss $\mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{W},\boldsymbol{b}}; \mathcal{D}^{\mathrm{adv}})$ for training time $T_g > 0$.

Our interests are (i) the relationship between perceptually-noise-like adversarial perturbations $\{\boldsymbol{r}_n\}_{n=1}^N$ and clean training samples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ (cf. Theorem 3.3), and (ii) whether the classifier $g(\,\cdot\,; T_g)$ trained on the adversarial perturbations or samples $\mathcal{D}^{\mathrm{adv}}$ can mimic the predictions of the classifier $f(\,\cdot\,; T_f)$ trained on the clean samples $\mathcal{D}$ (cf. Theorems 3.4 and 3.5).

3.2 Main Results

For $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^d$, we use $\Phi(\boldsymbol{z}_1, \boldsymbol{z}_2) \in (\gamma(1+\gamma)/2, (1+\gamma)/2]$ defined as (cf. Lemma C.4):

$$\Phi(\boldsymbol{z}_1, \boldsymbol{z}_2) := \mathbb{E}_{\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{I}/d), a \sim \mathcal{N}(0,1)}[\phi'(\langle \boldsymbol{v}, \boldsymbol{z}_1 \rangle + a)\phi'(\langle \boldsymbol{v}, \boldsymbol{z}_2 \rangle + a)], \tag{4}$$

where $\phi'(x) := \mathrm{d}\phi(x)/\mathrm{d}x$. First, we introduce an assumption on network width.

Assumption 3.2 (Wide network). Network width m satisfies

$$m > \tilde{\mathcal{O}}\left(d^2 \left\{ \frac{1}{N} \sum_{n=1}^N \left(\int_0^{T_f} \ell'(-y_n f(\boldsymbol{x}_n; t)) \, \mathrm{d}t + \int_0^{T_g} \ell'(-y_n^{\mathrm{adv}} f(\boldsymbol{x}_n'; t)) \, \mathrm{d}t \right) \right\}^2 \right), \tag{5}$$

where $\boldsymbol{x}_n' := \boldsymbol{r}_n$ for Scenario (a) and $\boldsymbol{x}_n' := \boldsymbol{x}_n^{\mathrm{adv}}$ for Scenario (b) in Setting 3.1. In particular, $m > \tilde{\mathcal{O}}(d^2(T_f + T_g)^2)$ for $\ell(s) = s$.

This assumption requires sufficiently large width m that regularizes the variations in parameters and forms the basis of lazy training (cf. Section 3.3). The width is always required to grow with the speed of the squared input dimension d^2. The relationship between the width and two training times, T_f and T_g, depends on the training set $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ and loss function ℓ. For example, if the training set is easily separable and the loss has an exponential tail, the derivative of the loss function might decrease rapidly with training time t and small m is enough to satisfy the assumption. For the identity loss, m is consistently required to satisfy $\tilde{\Omega}(d^2(T_f + T_g)^2)$. Note that the required values of T_f and T_g (and the corresponding m) for a desirable loss value remain an open question in the community. Our experimental results show that $m \approx 100$ is sufficient to verify our theorems for high-dimensional Gaussian distributions. Under this assumption, we consider the direction of the adversarial perturbation.

Theorem 3.3 (Direction of adversarial perturbation). *Let $\delta = \Theta(1)$ be a small positive number. Under Assumption 3.2, for any $n \in [N]$, with probability at least $1 - \delta$, the adversarial perturbation*
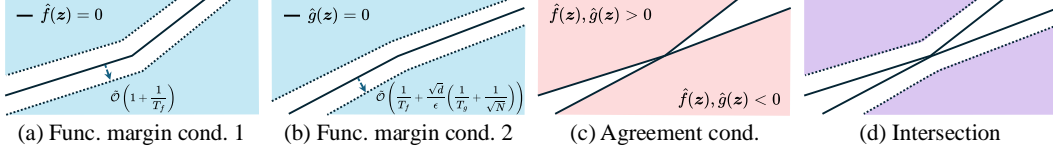

5

| (a) Func. margin cond. 1 | (b) Func. margin cond. 2 | (c) Agreement cond. | (d) Intersection |

Figure 2: The regions where Ineqs. (9) and (10) and Eq. (11) hold (colored areas) and their intersection.

r_n is parallel to the weighted sum of training samples as follows:

$$r_n \mathbin{/\!/} \frac{1}{N} \sum_{k=1}^{N} y_k \Phi(x_n, x_k) x_k \int_0^{T_f} \ell'(-y_k f(x_k; t)) \, \mathrm{d}t + \xi_n, \qquad (6)$$

where ξ_n satisfies $\|\xi_n\| = \tilde{\mathcal{O}}(1)$. In particular, for $\ell(s) = s$,

$$r_n \mathbin{/\!/} \frac{T_f}{N} \sum_{k}^{N} y_k \Phi(x_n, x_k) x_k + \xi_n. \qquad (7)$$

Note that the confidence level δ only logarithmically affects the norm of the remainder term ξ_n. This theorem indicates that the direction of a single perturbation can be represented as the weighted sum of $y_k x_k$ and remainder term ξ_n. Interestingly, this result suggests that a single perturbation derived from a classifier and sample can potentially contain information about the entire training dataset $\{(x_n, y_n)\}_{n=1}^{N}$. Particularly, in some cases (e.g., training samples are mutually orthogonal), $y_k x_k$ are not cancelled out by each other, and thus the single perturbation r_n contains all training data and labels without loss of information.[2] These results theoretically support the feature hypothesis. Consider the case with the identity loss. While the norm of the first term is $\mathcal{O}(T_f \sqrt{d})$, the norm of the remainder is constrained to $\tilde{\mathcal{O}}(1)$, suggesting that larger training time T_f and input dimension d strengthen the alignment between the perturbation and weighted sum.

Then, we consider the learning solely from these perturbations. The following theorem is a special case of Theorem D.17, which addresses a broader loss class and any sampling of $y_n^{\mathrm{adv}} \in \{\pm 1\}$.

Theorem 3.4 (Perturbation learning, Scenario (a), special case of Theorem D.17). *Consider Scenario (a) in Setting 3.1. Assume $\ell(s) = s$ and $y_n^{\mathrm{adv}} \sim U(\{\pm 1\})$ for every $n \in [N]$. Let $\delta = \Theta(1)$ be a small positive number and*

$$\hat{f}(z) := \frac{1}{N} \sum_{n=1}^{N} y_n \Phi(x_n, z)\langle x_n, z \rangle, \quad \hat{g}_a(z) := \frac{1}{N^2} \sum_{n=1}^{N} \Phi(r_n, z) \sum_{k=1}^{N} y_k \Phi(x_n, x_k)\langle x_k, z \rangle. \qquad (8)$$

Under Assumption 3.2, for any $z \in \mathbb{R}^d$, if

(Functional margin condition 1) $$|\hat{f}(z)| > \tilde{\mathcal{O}}\left(1 + \frac{1}{T_f}\right), \qquad (9)$$

(Functional margin condition 2) $$|\hat{g}_a(z)| > \tilde{\mathcal{O}}\left(\frac{1}{T_f} + \frac{\sqrt{d}}{\epsilon}\left(\frac{1}{T_g} + \frac{1}{\sqrt{N}}\right)\right), \qquad (10)$$

(Agreement condition) $$\mathrm{sgn}(\hat{f}(z)) = \mathrm{sgn}(\hat{g}_a(z)), \qquad (11)$$

then, with probability at least $1 - \delta$, $\mathrm{sgn}(f(z; T_f)) = \mathrm{sgn}(g(z; T_g))$ holds.

Note that the confidence level δ only logarithmically affects the right terms of Ineqs. (9) and (10), which is why these terms appear independent of δ. This theorem states that the predictions of a classifier g trained solely on adversarial perturbations $\{(r_n, y_n^{\mathrm{adv}})\}_{n=1}^{N}$ coincide with those of a classifier f trained on standard training samples $\{(x_n, y_n)\}_{n=1}^{N}$ if the three conditions hold. The two

---

[2] Recall $\Phi(x_n, x_k) > 0$.

functions, $\hat{f}$ and $\hat{g}$, which govern these conditions, can be viewed as key components that significantly influence the predictions of f and g (cf. Section 3.3). The conditions can be interpreted as follows.

Geometrical perspective. The functional margin conditions, Ineqs. (9) and (10), require the functional margins of $\hat{f}$ and $\hat{g}$ to exceed certain thresholds. In the input space $z \in \mathbb{R}^d$, these conditions correspond to regions far from the decision boundaries $\hat{f}(z) = 0$ and $\hat{g}(z) = 0$ (Fig. 2(a) and (b)). In a d-dimensional space, L_2-distance scales with $\sqrt{d}$, making the right terms of Ineqs. (9) and (10) relatively small when the perturbation size $\epsilon = \Theta(\sqrt{d})$; hence, a larger d facilitates the satisfaction of these conditions. Furthermore, Eq. (11) necessitates the agreement of the signs of these two decision boundaries (Fig. 2(c)). Consequently, the region where all conditions hold, i.e., where the prediction match occurs, can be characterized by the two piecewise linear functions (Fig. 2(d)).

Quantitative perspective (functional margin conditions). A large perturbation size ϵ facilitates the satisfaction of Ineq. (10). In high-dimensional spaces, achieving the required margin conditions demands perturbations of at least $\Omega(\sqrt{d})$ in magnitude, aligning with empirical scaling laws for L_2 perturbations. However, increasing ϵ alone is insufficient for the satisfaction because the right term of Ineq. (10) has an ϵ-irrelevant term $\tilde{\mathcal{O}}(1/T_f)$. Assume $\epsilon = \Theta(\sqrt{d})$. The absolute values of $\hat{f}$ and $\hat{g}$ grow with $\Theta(d)$ due to $\langle x, z \rangle$, while the right terms of Ineqs. (9) and (10) are independent of d. Thus, a larger d consistently facilitates the satisfaction. The training times T_f and T_g can reduce the right terms, but these terms contain time-independent terms $\tilde{\mathcal{O}}(1)$ and $\tilde{\mathcal{O}}(1/\sqrt{N})$, indicating that longer training times do not necessarily guarantee the satisfaction. A large sample size N also helps to satisfy Ineq. (10), but similarly, it is not sufficient. In summary, while a larger input dimension d consistently support the success of perturbation learning, a larger perturbation size ϵ, sample size N, and training times T_f, T_g provide partial, but not definitive, benefits.

Quantitative perspective (agreement condition).[3] It is difficult to interpret the agreement condition, Eq. (11), in its current form. We consider the following sufficient condition (cf. Lemma D.19):

$$\frac{|\sum_{n=1}^N y_n \langle x_n, z \rangle|}{\max_{x \in \{x_1, \ldots, x_N, z\}} \sum_{n=1}^N \lambda(x_n, x)|\langle x_n, z \rangle|} > \frac{1-\gamma}{1+\gamma} \qquad \Rightarrow \qquad \text{Eq. (11),} \qquad (12)$$

$$\lambda(z_1, z_2) := 1 - \sqrt{\frac{e}{2\pi} \frac{\|z_1\|^2 \|z_2\|^2 - \langle z_1, z_2 \rangle^2 + d\|z_1 - z_2\|^2}{\|z_1\|^2 \|z_2\|^2 + \langle z_1, z_2 \rangle^2 + d\|z_1 + z_2\|^2 + 2d^2}} = \Theta(1). \qquad (13)$$

Note that the left term can exceed one as $\lambda(z_1, z_2)$ lies in $(0.34, 1]$. It is clear that a large negative slope of Leaky-ReLU γ facilitates the satisfaction. The magnitude of the left term depends on the consistency of the correlation (inner product) between z and $y_n x_n$ for every n. For example, when z consistently exhibits a positive or negative correlation with $y_n x_n$, the left term exceeds one, and the condition is satisfied. In contrast, if z positively correlates with half of the samples and negatively with the other half, the left term may output a small value, and the condition is not satisfied. In summary, the agreement condition depends on the consistency of the correlation between z and $y_n x_n$.

Finally, we justify the success of perturbation learning in Scenario (b).

Theorem 3.5 (Perturbation learning, Scenario (b), special case of Theorem D.18). *Consider Scenario (b) in Setting 3.1. Assume $\ell(s) = s$ and $y_n^{\mathrm{adv}} \sim U(\{\pm 1\})$ for every $n \in [N]$. Let $\delta = \Theta(1)$ be a small positive number and*

$$\hat{g}_b(z) := \frac{1}{N^2} \sum_{n=1}^N \Phi(x_n^{\mathrm{adv}}, z) \sum_{k=1}^N y_k \Phi(x_n, x_k) \langle x_k, z \rangle. \qquad (14)$$

Under Assumption 3.2, for any $z \in \mathbb{R}^d$, if the functional margin condition 1 (Ineq. (9)),

(Func. margin cond. 2) $\qquad |\hat{g}_b(z)| > \tilde{\mathcal{O}}\left(\frac{1}{T_f} + \frac{\sqrt{d}}{\epsilon} \left(\frac{1}{T_g} + \frac{\sqrt{\sum_n^N (\langle x_n, z \rangle + 1)^2}}{N} \right) \right),$ (15)

(Agreement condition) $\quad \mathrm{sgn}(\hat{f}(z)) = \mathrm{sgn}(\hat{g}_b(z)),$ (16)

---

[3]One can analyze the agreement condition by using the relationship between Φ and the arc-cosine kernel [10].

then, with probability at least $1 - \delta$, $\mathrm{sgn}(f(\boldsymbol{z}; T_f)) = \mathrm{sgn}(g(\boldsymbol{z}; T_g))$ holds.

This is a special case of Theorem D.18, which addresses differentiable, non-decreasing losses and any sampling of $y_n^{\mathrm{adv}} \in \{\pm 1\}$. Similarly to Theorem 3.4, this theorem states that the predictions of a classifier g trained on randomly labeled adversarial examples $\{(\boldsymbol{x}_n^{\mathrm{adv}}, y_n^{\mathrm{adv}})\}_{n=1}^N$ coincide with those of a classifier f trained on standard training samples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ if the three conditions hold. Functional margin condition 1 is consistent with that in Theorem 3.4, i.e., Ineq. (9). The definition of $\hat{g}_b(\boldsymbol{z})$ is slightly different from $\hat{g}_a(\boldsymbol{z})$ in Theorem 3.4, with $\boldsymbol{r}_n$ replaced by $\boldsymbol{x}_n^{\mathrm{adv}}$. Due to this change in the definition of $\hat{g}_b(\boldsymbol{z})$, functional margin condition 2, Ineq. (15), and the agreement condition, Eq. (16), slightly differ from those in Theorem 3.4.

Assume $\epsilon = \Theta(\sqrt{d})$. Similarly to Ineq. (10), the left term of Ineq. (15) grows with $\mathcal{O}(d)$ due to the inner product. In contrast to Ineq. (10), the right term of Ineq. (15) includes a term that grows with $\sqrt{\sum_n^N (\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1)^2}/N = \mathcal{O}(d/\sqrt{N})$. This suggests that Scenario (b) necessitates a larger sample size N than Scenario (a) to mitigate the effect of d.

Furthermore, Eqs. (11) and (16) hold simultaneously (i.e., $\mathrm{sgn}(\hat{f}(\boldsymbol{z})) = \mathrm{sgn}(\hat{g}_a(\boldsymbol{z})) = \mathrm{sgn}(\hat{g}_b(\boldsymbol{z}))$) if $\sum_{k=1}^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle$ outputs the same sign for any n. This indicates that there might exist regions where the prediction match is observed regardless of the scenarios, and these regions are partially determined by the n linear boundaries. Note that Ineq. (12) also serves as a sufficient condition for Eq. (16), and the quantitative analysis for Eq. (11) can be applied to Eq. (16) as well.

3.3 Sketch of Proof

In this section, for simplicity, we provide a sketch of the proof for Theorems 3.3 and 3.4 with infinite network width $m \to \infty$, networks without biases, and identity loss. A proof for the general case can be found in Appendix D.

Lazy training. First, we introduce the concept of lazy training, where network parameters and outputs of hidden neurons change negligibly during training when the network width is sufficiently large [9, 30]. Since a readout weight α_i is sampled from $\mathcal{N}(0, 1/m)$, from gradient flow, for any $\boldsymbol{z} \in \mathbb{R}^d$,

$$\left| \left\langle \int_0^{T_f} \frac{\mathrm{d}\boldsymbol{v}_i(t)}{\mathrm{d}t} \, \mathrm{d}t \,, \boldsymbol{z} \right\rangle \right| = \left| \frac{1}{N} \sum_{n=1}^N y_n \alpha_i \langle \boldsymbol{x}_n, \boldsymbol{z} \rangle \int_0^{T_f} \phi'(\langle \boldsymbol{v}_i(t), \boldsymbol{x}_n \rangle) \, \mathrm{d}t \right| = \tilde{\mathcal{O}}\left(\frac{dT_f}{\sqrt{m}} \right). \quad (17)$$

Therefore, as $m \to \infty$, the inner product between the time variation of a hidden parameter and an input approaches zero. This suggests that the sign of the output of a hidden neuron do not change:

$$\mathrm{sgn}(\langle \boldsymbol{v}_i(T_f), \boldsymbol{z} \rangle) = \mathrm{sgn}\left(\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle + \left\langle \int_0^{T_f} \frac{\mathrm{d}\boldsymbol{v}_i(t)}{\mathrm{d}t} \, \mathrm{d}t \,, \boldsymbol{z} \right\rangle \right) \xrightarrow{m \to \infty} \mathrm{sgn}(\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle). \quad (18)$$

Therefore, $\phi'(\langle \boldsymbol{v}_i(T_f), \boldsymbol{z} \rangle) = \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle)$. Recall $\phi(z) := \max(\gamma z, z)$.

Theorem 3.3. From the perturbation definition Eq. (1), the perturbation $\boldsymbol{r}_n$ is parallel to $\boldsymbol{\nabla}_{\boldsymbol{x}_n} f(\boldsymbol{x}_n; \boldsymbol{V}(T_f))$. Using $\phi'(\langle \boldsymbol{v}_i(T_f), \boldsymbol{z} \rangle) = \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle)$,

$$\boldsymbol{r}_n /\!\!/ \boldsymbol{\nabla}_{\boldsymbol{x}_n} f(\boldsymbol{x}_n; \boldsymbol{V}(T_f)) = \sum_{i=1}^m \alpha_i \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{x}_n \rangle) \left(\boldsymbol{v}_i(0) + \int_0^{T_f} \frac{\mathrm{d}\boldsymbol{v}_i(t)}{\mathrm{d}t} \, \mathrm{d}t \right). \quad (19)$$

The first term is constrained to $\tilde{\mathcal{O}}(1)$. Let $\Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) := \mathbb{E}_{\boldsymbol{v} \sim \mathcal{N}(0, 1/d)}[\phi'(\langle \boldsymbol{v}, \boldsymbol{x}_n \rangle) \phi'(\langle \boldsymbol{v}, \boldsymbol{x}_k \rangle)]$. Using $\sum_{i=1}^m \alpha_i^2 \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{x}_n \rangle) \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{x}_k \rangle) \to \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)$ as $m \to \infty$, the second term becomes

$$\sum_{i=1}^m \alpha_i \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{x}_n \rangle) \int_0^{T_f} \frac{\mathrm{d}\boldsymbol{v}_i(t)}{\mathrm{d}t} \, \mathrm{d}t = \frac{T_f}{N} \sum_{k=1}^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \boldsymbol{x}_k. \quad (20)$$

Theorem 3.4. Similarly to the above, we can represent the adversarial perturbation $\boldsymbol{r}_n$ as follows:

$$\boldsymbol{r}_n := \epsilon y_n^{\mathrm{adv}} \frac{\boldsymbol{\nabla}_{\boldsymbol{x}_n} f(\boldsymbol{x}_n; \boldsymbol{V}(T_f))}{\|\boldsymbol{\nabla}_{\boldsymbol{x}_n} f(\boldsymbol{x}_n; \boldsymbol{V}(T_f))\|} \approx \Omega\left(\frac{\epsilon}{N\sqrt{d}} \right) y_n^{\mathrm{adv}} \sum_{k=1}^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle. \quad (21)$$

Assuming $\sum_{i=1}^{m} \alpha_i \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle)\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle = \tilde{\mathcal{O}}(1) \approx 0$ for simplicity, the network prediction $f(\boldsymbol{z}; \boldsymbol{V}(T_f))$ trained on $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ can be represented as follows:

$$f(\boldsymbol{z}; \boldsymbol{V}(T_f)) = \sum_{i=1}^{m} \alpha_i \phi'(\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle)\langle \boldsymbol{v}_i(T_f), \boldsymbol{z} \rangle \approx \frac{T_f}{N} \sum_{n=1}^{N} y_n \Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle, \qquad (22)$$

and $\mathrm{sgn}(f(\boldsymbol{z}; \boldsymbol{V}(T_f))) = \mathrm{sgn}(\sum_{n=1}^{N} y_n \Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle)$. In addition, $g(\boldsymbol{z}; \boldsymbol{W}(T_f))$ trained on $\{(\boldsymbol{r}_n, y_n^{\mathrm{adv}})\}_{n=1}^{N}$ can be represented as follows:

$$g(\boldsymbol{z}; \boldsymbol{W}(T_g)) \approx \Omega\left(\frac{\epsilon T_g}{N^2 \sqrt{d}}\right) \sum_{n=1}^{N} \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \sum_{k=1}^{N} y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z} \rangle, \qquad (23)$$

and $\mathrm{sgn}(g(\boldsymbol{z}; \boldsymbol{W}(T_g))) = \mathrm{sgn}(\sum_{n=1}^{N} \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \sum_{k=1}^{N} y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z} \rangle)$. Thus, if the agreement condition Eq. (11) holds, then $\mathrm{sgn}(f(\boldsymbol{z}; \boldsymbol{V}(T_f))) = \mathrm{sgn}(g(\boldsymbol{z}; \boldsymbol{W}(T_g)))$.

Formal proof. In the above sketch of proof, we have introduced several approximations. Rigorous evaluations are provided in Appendix D. For example, in the sketch, we assumed $m \to \infty$, ensuring that the signs of all hidden layer outputs remain unchanged. In contrast, the formal proof derives a bound on the width m that ensures that the number of hidden neurons with flipped signs is at most $\mathcal{O}(\sqrt{m})$, which makes the discussion (e.g., about Eqs. (18) and (19)) more complicated. Moreover, in Eq. (21), we neglected the first term of Eq. (19), but the formal proof carefully considers the impact on the subsequent steps. The functional margin conditions arise from the evaluation of these remainder terms.

3.4   Comparison with Prior Work and Limitations

In this section, we compare our results with [28] and discuss the limitations of our work. In summary, our results justify the feature hypothesis and perturbation learning under substantially milder conditions than [28], except for network width (cf. Tab. 1). The assumption of wide two-layer networks is our main limitation.

Goals, results, and tools. The goals of our work and [28] are the same: justifying the feature hypothesis and perturbation learning. The conclusions drawn are also equivalent. However, our assumptions are much milder than theirs. This is due to the differences in the analytical approaches. While they leverage research on feature learning [19, 24, 31], we utilize the concept of lazy training [9, 30], which enables us to substantially relax the conditions.

Data distribution. Prior work imposes a strong assumption that training samples with/without adversarial perturbations are mutually orthogonal, i.e., $\langle \boldsymbol{x}_n, \boldsymbol{x}_k \rangle \approx 0$ and $\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{x}_k^{\mathrm{adv}} \rangle \approx 0$ for any $n \neq k$. This condition is stringent and is hard to hold for real-world datasets. Moreover, it may not even hold for data sampled from a zero-mean Gaussian distribution in some common situations (e.g., the sample size is sufficiently larger than the dimension). We do not impose any assumptions on the data distribution. This is the first result that theoretically supports the feature hypothesis and the success of perturbation learning on realistic data distribution.

Perturbation design. Prior work defined the perturbation form using the decision boundary of a classifier. However, this is not only uncommon but also theoretically computable only in limited problem settings. Additionally, they constrained perturbation size to $\epsilon = \Theta(\sqrt{d/N})$, which becomes unrealistically small for a large sample size and is far from the practical constraint $\epsilon = \Theta(\sqrt{d})$. We employ a single-step gradient-based method [21], which is commonly used in practice, and the perturbation constraint can be set arbitrarily.

Training time, loss function, network bias, and activation. First, it should be noted that these constraints are critical for deriving the results in [28]. This is because their theoretical framework [19, 24, 31] substantially requires the above conditions. We consider arbitrary training time, a wide class of loss functions, and (Leaky-) ReLU networks with bias availability. In contrast, they considered infinite training time, loss functions with exponential tails, homogeneous neural networks (thus requiring no bias), and Leaky-ReLU networks (the theorem becomes harder to hold as the negative slope of Leaky-ReLU approaches zero), which are essential for deriving their results (cf. the proofs of Theorem 4.4 in [31] and the proof of Theorem 3.2 in [19]).
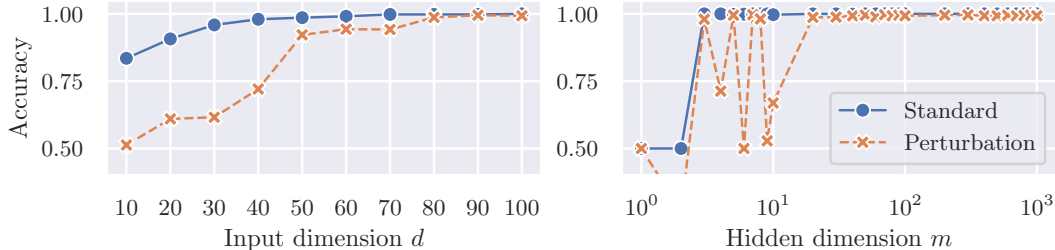
Figure 3: Accuracy on the mean-shifted Gaussian dataset in Scenario (a). The blue lines represent accuracy of the classifier f on $\mathcal{D}$, i.e., training accuracy. The orange lines represent accuracy of the classifier g on $\mathcal{D}$.

Limitations. Compared to [28], our main limitation is the requirement for sufficient, though finite, network width. Moreover, our analysis is confined to two-layer networks, a common constraint in previous work. In practice, perturbation learning often employs deeper, and not necessarily wider, networks, which limits the direct applicability of our theoretical insights to more complex architectures. This assumption of a shallow network introduces another limitation. While deep neural networks typically capture high-level features from images and adversarial attacks are considered to exploit them, our framework focuses solely the low-level features (i.e., $\boldsymbol{x}_n$ itself) in adversarial perturbations and their extraction through perturbation learning, as shown in Theorems 3.3 to 3.5. Relaxing the shallow network constraint may allow us to capture a broader set of features present in adversarial perturbations. Despite these limitations, our work is the first to rigorously support the feature hypothesis and validate perturbation learning under realistic data distributions, perturbation designs, and training settings, marking a substantial advancement in the theoretical understanding of adversarial examples.

4 Experiments

A comprehensive set of experiments conducted to validate our theorems can be found in Appendix B. In this section, we briefly present two results that confirm Theorem 3.4. As a training dataset $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$, we employed a synthetic training dataset to easily change the input dimension, which effectively helps perturbation learning in both scenarios, as predicted by our theorems. Note that the perturbation learning on real-world datasets can be found in the literature [22, 28]. We generated synthetic data and labels from the mean-shifted Gaussian distribution as follows: $\{\boldsymbol{x}_n\}_{n=1}^{N}$ are independently sampled from $\mathcal{N}(0.3 \times y_n \times \boldsymbol{1}, \boldsymbol{I})$, and y_n is set to one if $n \in [N/2]$ and minus one otherwise. The experimental settings are as follows: $d = 100$, $N = 1,000$, $m = 100$, $\gamma = 0$, $\ell(s) := s$, $\epsilon = 0.01$, and the number of training steps is set to 1,000 for both f and g. The experimental results for perturbation learning under Scenario (a) are shown in Fig. 3. A high input dimension facilitates the alignment between f and g. Our theoretical results assume a wide network width, and Fig. 3 indicates that a sufficiently large width consistently stabilize the alignment.

5 Conclusion

We provided a theoretical justification for perturbation learning and the feature hypothesis. We demonstrated that adversarial perturbations contain class-specific features sufficient for networks to generalize from. Moreover, we revealed that the predictions of a classifier trained solely on these perturbations or mislabeled adversarial examples coincide with those of a classifier trained on correctly labeled training samples under three mild conditions. Except for wide two-layer networks, our assumption is substantially milder than prior work [28].

Acknowledgments and Disclosure of Funding

References

[1] G. Aggarwal, A. Sinha, N. Kumari, and M. Singh. On the benefits of models with perceptually-aligned gradients. In *ICLR WS*, 2020.

[2] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *ICML*, pages 242–252, 2019.

[3] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *ICML*, pages 322–332, 2019.

[4] M. Augustin, A. Meinke, and M. Hein. Adversarial robustness on in-and out-distribution improves explainability. In *ECCV*, pages 228–245, 2020.

[5] Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *NeurIPS*, volume 32, 2019.

[6] Y. Cao and Q. Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *AAAI*, volume 34, pages 3349–3356, 2020.

[7] P. Chalasani, J. Chen, A. R. Chowdhury, X. Wu, and S. Jha. Concise explanations of neural networks using adversarial training. In *ICML*, pages 1383–1391, 2020.

[8] S.-H. Chang, P. C. Cosman, and L. B. Milstein. Chernoff-type bounds for the gaussian error function. *IEEE Transactions on Communications*, 59(11):2939–2944, 2011.

[9] L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *NeurIPS*, volume 32, 2019.

[10] Y. Cho and L. Saul. Kernel methods for deep learning. In *NeurIPS*, volume 22, 2009.

[11] L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[12] Z. Deng, L. Zhang, K. Vodrahalli, K. Kawaguchi, and J. Y. Zou. Adversarial training helps transfer learning via better representations. In *NeurIPS*, volume 34, pages 25179–25191, 2021.

[13] E. Dobriban, H. Hassani, D. Hong, and A. Robey. Provable tradeoffs in adversarially robust classification. *IEEE Transactions on Information Theory*, 2023.

[14] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *ICML*, pages 1675–1685, 2019.

[15] S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*, 2019.

[16] J. Duchi. Lecture notes on statistics and information theory, 2023.

[17] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry. Adversarial robustness as a prior for learned representations. *arXiv:1906.00945*, 2019.

[18] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb. On the connection between adversarial robustness and saliency map interpretability. In *ICML*, 2019.

[19] S. Frei, G. Vardi, P. L. Bartlett, N. Srebro, and W. Hu. Implicit bias in leaky relu networks trained on high-dimensional data. In *ICLR*, 2023.

[20] M. Geiger, S. Spigler, A. Jacot, and M. Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[22] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, volume 32, pages 125–136, 2019.

[23] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, volume 31, 2018.

[24] Z. Ji and M. Telgarsky. Directional convergence and alignment in deep learning. In *NeurIPS*, volume 33, pages 17176–17186, 2020.

[25] Z. Ji and M. Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. In *ICLR*, 2020.

[26] S. Kaur, J. Cohen, and Z. C. Lipton. Are perceptually-aligned gradients a general property of robust classifiers? In *NeurIPS WS*, 2019.

[27] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[28] S. Kumano, H. Kera, and T. Yamasaki. Theoretical understanding of learning from adversarial perturbations. In *ICLR*, 2024.

[29] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *NeurIPS*, volume 32, 2019.

[30] Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *NeurIPS*, volume 31, 2018.

[31] K. Lyu and J. Li. Gradient descent maximizes the margin of homogeneous neural networks. In *ICLR*, 2020.

[32] M. Mehrabi, A. Javanmard, R. A. Rossi, A. Rao, and T. Mai. Fundamental tradeoffs in distributionally adversarial training. In *ICML*, pages 7544–7554, 2021.

[33] A. Montanari and Y. Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training. *The Annals of Statistics*, 50(5):2816–2847, 2022.

[34] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, 2020.

[35] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Adversarial training can hurt generalization. In *ICML WS*, 2019.

[36] P. Rigollet and J.-C. Hütter. High-dimensional statistics. *arXiv:2310.19244*, 2023.

[37] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? In *NeurIPS*, volume 33, pages 3533–3545, 2020.

[38] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, volume 32, 2019.

[39] S. Srinivas, S. Bordt, and H. Lakkaraju. Which models have perceptually-aligned gradients? an explanation via off-manifold robustness. In *NeurIPS*, volume 36, 2023.

[40] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, pages 631–648, 2018.

[41] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[42] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

[43] F. Utrera, E. Kravitz, N. B. Erichson, R. Khanna, and M. W. Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. In *ICLR*, 2021.

[44] B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In *COLT*, pages 3635–3673, 2020.

[45] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

[46] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri. A closer look at accuracy vs. robustness. In *NeurIPS*, volume 33, pages 8588–8601, 2020.

[47] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pages 7472–7482, 2019.

[48] T. Zhang and Z. Zhu. Interpreting adversarially trained convolutional neural networks. In *ICML*, pages 7502–7511, 2019.

[49] D. Zou, Y. Cao, D. Zhou, and Q. Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109:467–492, 2020.

A Comparison with Prior Work

In this section, we compare our findings with those of prior work [28] and highlight new insights beyond technical contributions.

A.1 Feature Hypothesis (Theorem 3.3)

Our result offers a new insight into the alignment between perturbations and training samples through the residual term ξ_n. The direction of a perturbation vector comprises two components: a weighted sum of the training samples (the main term) and a residual term. Our result suggests that as the input dimension increases, the residual term becomes smaller than the main term, thereby enhancing the alignment. In other words, perturbations more robustly encode class-specific features. This insight is unattainable in prior research due to the absence of a residual term in their limited problem setting.

Additionally, our finding suggests that extended training time reinforces the directional alignment between perturbations and training samples. This concept is supported by practical intuition and experience but not addressed in prior work.

Our result further introduces a coefficient, $\Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)$, for each adversarial perturbation. The coefficient $\Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)$ depends on the slope of the activation function and the similarity between $\boldsymbol{x}_n$ and $\boldsymbol{x}_k$ (cf. Eq. (4)). This suggests that training samples with higher similarity to each other exhibit a stronger influence within a perturbation. Although prior work includes similar coefficients, they cannot be explicitly computed.

A.2 Perturbation Learning (Theorems 3.4 and 3.5)

Our result establishes an explicit connection between the success of perturbation learning and training factors, such as training time T, perturbation size ϵ, input dimension d, sample size N, and confidence level δ. This result enhances our understanding of how these factors influence perturbation learning. For example, perturbation learning is more likely to succeed with a larger input dimension d and sample size N. Notably, our findings indicate that while a larger d consistently facilitates successful perturbation learning, a larger N alone is insufficient. In contrast, existing research does not elucidate the roles of these variables, merely showing that success is achievable when both d and N approach infinity.

B Experimental Settings and Other Results

B.1 Datasets

We utilized two synthetic datasets and two widely used datasets, MNIST [11] and Fashion-MNIST [45]. The first synthetic dataset is derived from a zero-mean Gaussian distribution: $\{\boldsymbol{x}_n\}_{n=1}^N$ are independently sampled from $\mathcal{N}(0, \boldsymbol{I})$ and $\{y_n\}_{n=1}^N$ are independently sampled from $U(\{\pm 1\})$. The second synthetic dataset is based on a mean-shifted Gaussian distribution: $\{\boldsymbol{x}_n\}_{n=1}^N$ are independently sampled from $\mathcal{N}(0.3 \times y_n \times \boldsymbol{1}, \boldsymbol{I})$ and y_n is set to one for $n \in [N/2]$ and minus one otherwise. We used data only from classes 1 and 2 in MNIST (i.e., digits 1 and 2) and those from classes 0 and 9 in Fashion-MNIST (i.e., T-shirt and ankle boot). To measure the agreement ratio between network predictions from standard training and perturbation learning, for the real-world dataset cases, we used standard test datasets, and for the synthetic dataset cases, we used 1,000 samples independently and identically sampled according to the training data distribution.

B.2 Settings

In this section, for notational simplicity, we denote the number of training epochs in standard training by T_f and in perturbation learning by T_g. Note that the original definitions of T_f and T_g are continuous training steps in gradient flow, i.e., gradient descent with an infinitely small learning rate (cf. Section 3.1), which is conceptually distinct from the discrete steps in gradient descent with finitely small learning rates in practice. In addition, we denote the learning rates in standard training and perturbation learning as η_f and η_g, respectively.

14

We used non-stochastic gradient descent (i.e., each gradient calculation uses the entire dataset) with 0.9 momentum and the learning rate scheduler that multiplies a learning rate by 0.1 when a training loss has stopped improving during 10 epochs. For Figs. 3 to A12, we selected the best accuracy, agreement ratio, and cosine similarity from training with multiple initial learning rates.

In Figs. A4 to A11, the blue lines represent the accuracy of the classifier from standard training on the training dataset $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$. Namely, these mean the training accuracy of $f(\,\cdot\,; T_f)$. The orange lines represent the accuracy of the classifier from perturbation learning on the *standard (clean)* training dataset $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ rather than the perturbed dataset $\{(\boldsymbol{x}_n^{\mathrm{adv}}, y_n^{\mathrm{adv}})\}_{n=1}^{N}$. Namely, these mean the ratio that counterintuitive generalization occurs. Note that the orange lines stay around fifty percent (chance accuracy) if adversarial perturbations are not included in $\{\boldsymbol{x}_n^{\mathrm{adv}}\}_{n=1}^{N}$ (cf. $\epsilon = 0$ in Figs. A4 to A11). The green lines represent the agreement ratio between predictions $f(\,\cdot\,; T_f)$ and $g(\,\cdot\,; T_g)$ on a test dataset.

The cosine similarity in Fig. A12 is the average one between the experimentally calculated adversarial perturbation and the theoretically predicted one (cf. Eq. (1)) across all n. The blue lines are the same as those in Figs. A4 to A11.

The two axes in Fig. A13 are the normalized average vectors of samples from the positive and negative classes, respectively. The blue circles and orange crosses correspond to the projections of positive and negative samples onto these axes. The gray and green areas indicate regions where two predictions are consistent and inconsistent, respectively. The red solid lines represent $\hat{f}(\boldsymbol{z}) = 0$. The black dashed lines represent $\hat{g}_a(\boldsymbol{z}) = 0$ in Scenario (a) and $\hat{g}_b(\boldsymbol{z}) = 0$ in Scenario (b).

Mean-Shifted Gaussian and Scenario (a). A common experimental setting for the mean-shifted Gaussian dataset and Scenario (a) in Figs. A4 and A12 is as follows: input dimension $d = 100$, hidden dimension $m = 100$, activation function slope $\gamma = 0$, number of training samples $N = 1,000$, loss function $\ell(s) = s$, training epochs in standard training $T_f = 1,000$ and in perturbation learning $T_g = 1,000$, perturbation size $\epsilon = \sqrt{d \times 0.001^2} = 0.01$, and learning rates in standard training $\eta_f = 1$ or 0.1 and in perturbation learning $\eta_g = 1$ or 0.1. However, for the comparison of T_f (i.e., the graph in the fourth row and the first column in Fig. A4), we set η_f only to 0.1. For the comparison of d (i.e., the graph in the first row in Fig. A4), we employed $\epsilon = \sqrt{d \times 0.001^2}$. In Fig. A13, we used $d = 100$, $m = 100$, $\gamma = 0$, $N = 1,000$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{d \times 0.001^2} = 0.01$, $\eta_f = 1$, and $\eta_g = 1$.

Mean-Shifted Gaussian and Scenario (b). A common experimental setting for the mean-shifted Gaussian dataset and Scenario (b) in Figs. A5 and A12 is as follows: $d = 5,000$, $m = 100$, $\gamma = 0$, $N = 10,000$, $\ell(s) = s$, $T_f = 1,000$, $T_g = 1,000$, $\epsilon = \sqrt{d \times 0.01^2} = 0.7$, $\eta_f = 1$ or 0.1, and $\eta_g = 1$ or 0.1. However, for the comparison of T_f (i.e., the graph in the fourth row and the first column in Fig. A5), we set η_f only to 0.01. In addition, for the comparison of T_g (i.e., the graph in the fourth row and the second column in Fig. A5), we set η_g only to 0.01. Furthermore, we set $\eta_f = 10, 5, 1, 0.1, 0.01$ and $\eta_g = 10, 5, 1, 0.1, 0.01$ for the evaluation with the logistic loss (i.e., the graph in the first row and the second column in Fig. A5). For the comparison of d (i.e., the graph in the first row in Fig. A5), we employed $\epsilon = \sqrt{d \times 0.01^2}$. In Fig. A13, we used $d = 100$, $m = 100$, $\gamma = 0$, $N = 5,000$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{d \times 0.01^2} = 0.1$, $\eta_f = 1$, and $\eta_g = 1$.

Zero-Mean Gaussian and Scenario (a). A common experimental setting for the zero-mean Gaussian dataset and Scenario (a) in Figs. A6 and A12 is as follows: $d = 10,000$, $m = 100$, $\gamma = 0$, $N = 10,000$, $\ell(s) = s$, $T_f = 1,000$, $T_g = 1,000$, $\epsilon = \sqrt{d \times 0.001^2} = 0.1$, $\eta_f = 1$ or 0.1, and $\eta_g = 1$ or 0.1. However, for the comparison of T_f (i.e., the graph in the fourth row and the first column in Fig. A6), we set η_f only to 0.1. For the comparison of d (i.e., the graph in the first row in Fig. A6), we employed $\epsilon = \sqrt{d \times 0.001^2}$. In Fig. A13, we used $d = 1,000$, $m = 1,000$, $\gamma = 0$, $N = 2,000$, $\ell(s) = s$, $T_f = 1,000$, $T_g = 1,000$, $\epsilon = \sqrt{d \times 0.001^2} = 0.031$, $\eta_f = 1$, and $\eta_g = 1$.

Zero-Mean Gaussian and Scenario (b). A common experimental setting for the zero-mean Gaussian dataset and Scenario (b) in Figs. A7 and A12 is as follows: $d = 10,000$, $m = 100$, $\gamma = 0$, $N = 10,000$, $\ell(s) = s$, $T_f = 1,000$, $T_g = 1,000$, $\epsilon = \sqrt{d \times 0.1^2} = 10$, $\eta_f = 1$ or 0.1, and $\eta_g = 1$ or 0.1. However, for the comparison of T_f (i.e., the graph in the fourth row and the first column in Fig. A7), we set η_f only to 0.1. In addition, for the comparison of T_g (i.e., the graph in the fourth row and the second column in Fig. A7), we set η_g only to 0.1. For the comparison of d (i.e., the graph in the first row in Fig. A7), we employed $\epsilon = \sqrt{d \times 0.1^2}$. In Fig. A13, we used $d = 1,000$, $m = 1,000$,

$\gamma = 0$, $N = 10,000$, $\ell(s) = s$, $T_f = 1,000$, $T_g = 1,000$, $\epsilon = \sqrt{d \times 0.01^2} = 0.31$, $\eta_f = 0.1$, and $\eta_g = 0.1$.

MNIST and Scenario (a). A common experimental setting for MNIST and Scenario (a) in Figs. A8 and A12 is as follows: $m = 100$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$ or 0.001, and $\eta_g = 0.01$ or 0.001. In Fig. A13, we used $m = 1,000$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$, and $\eta_g = 0.01$.

MNIST and Scenario (b). A common experimental setting for MNIST and Scenario (b) in Figs. A9 and A12 is as follows: $m = 100$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$ or 0.001, and $\eta_g = 0.01$ or 0.001. In Fig. A13, we used $m = 1,000$, $\gamma = 0$, $\ell(s) = s$, $T_f = 1,000$, $T_g = 1,000$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$, and $\eta_g = 0.01$.

Fashion-MNIST and Scenario (a). A common experimental setting for Fashion-MNIST and Scenario (a) in Figs. A10 and A12 is as follows: $m = 100$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$ or 0.001, and $\eta_g = 0.01$ or 0.001. In Fig. A13, we used $m = 1,000$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$, and $\eta_g = 0.01$.

Fashion-MNIST and Scenario (b). A common experimental setting for Fashion-MNIST and Scenario (b) in Figs. A11 and A12 is as follows: $m = 100$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.01^2}/2 = 0.14$, $\eta_f = 0.01$ or 0.001, and $\eta_g = 0.01$ or 0.001. In Fig. A13, we used $m = 1,000$, $\gamma = 0$, $\ell(s) = s$, $T_f = 100$, $T_g = 100$, $\epsilon = \sqrt{784 \times 0.1^2}/2 = 1.4$, $\eta_f = 0.001$, and $\eta_g = 0.001$.

C   Lemmas

In this section, we derive fundamental properties of random variables.

Lemma C.1 (Properties of Gaussian random variables). *Let $\sigma^2 > 0$ be a positive constant. Let $X_1, \ldots, X_m \in \mathbb{R}$ be $m \in \mathbb{N}$ i.i.d. Gaussian random variables that follow $\mathcal{N}(0, \sigma^2)$.*

(a) For $0 < \delta < 1$, with probability at least $1 - \delta$,

$$\max_i |X_i| < \sqrt{2\sigma^2 \ln(2m/\delta)}. \tag{A24}$$

(b) Let $Y_1, \ldots, Y_m \in [\gamma^2, 1]$ be m independent random variables with $0 \le \gamma^2 < 1$. Suppose that $Y_1, \ldots, Y_m$ are independent of $X_1, \ldots, X_m$. For $0 < \delta < 1$, with probability at least $1 - \delta$,

$$\left| \sum_i^m X_i^2 Y_i - \sigma^2 \sum_i^m \mathbb{E}[Y_i] \right| < \max \left(16\sigma^2 \ln\left(\frac{2}{\delta}\right), \sqrt{128m\sigma^4 \ln\left(\frac{2}{\delta}\right)} \right). \tag{A25}$$

(c) For $\exp\left(-2(\eta\sqrt{m} + 1)^2/m\right) < \delta < 1$, with probability at least $1 - \delta$, there are at most $\eta\sqrt{m} \in [m - 1]$ instances such that

$$|X_i| < \sqrt{-\frac{\pi\sigma^2}{2} \ln\left(1 - \left(\frac{\eta\sqrt{m} + 1}{m} - \sqrt{-\frac{\ln\delta}{2m}} \right)^2 \right)}. \tag{A26}$$

Proof. Let $C > 0$ be a positive constant.

(a) From [36],

$$\mathbb{P}[\max_i |X_i| \ge C] \le 2m \exp\left(-\frac{C^2}{2\sigma^2} \right). \tag{A27}$$

Thus,

$$\mathbb{P}\left[\max_i |X_i| \ge \sqrt{2\sigma^2 \ln(2m/\delta)} \right] \le \delta. \tag{A28}$$
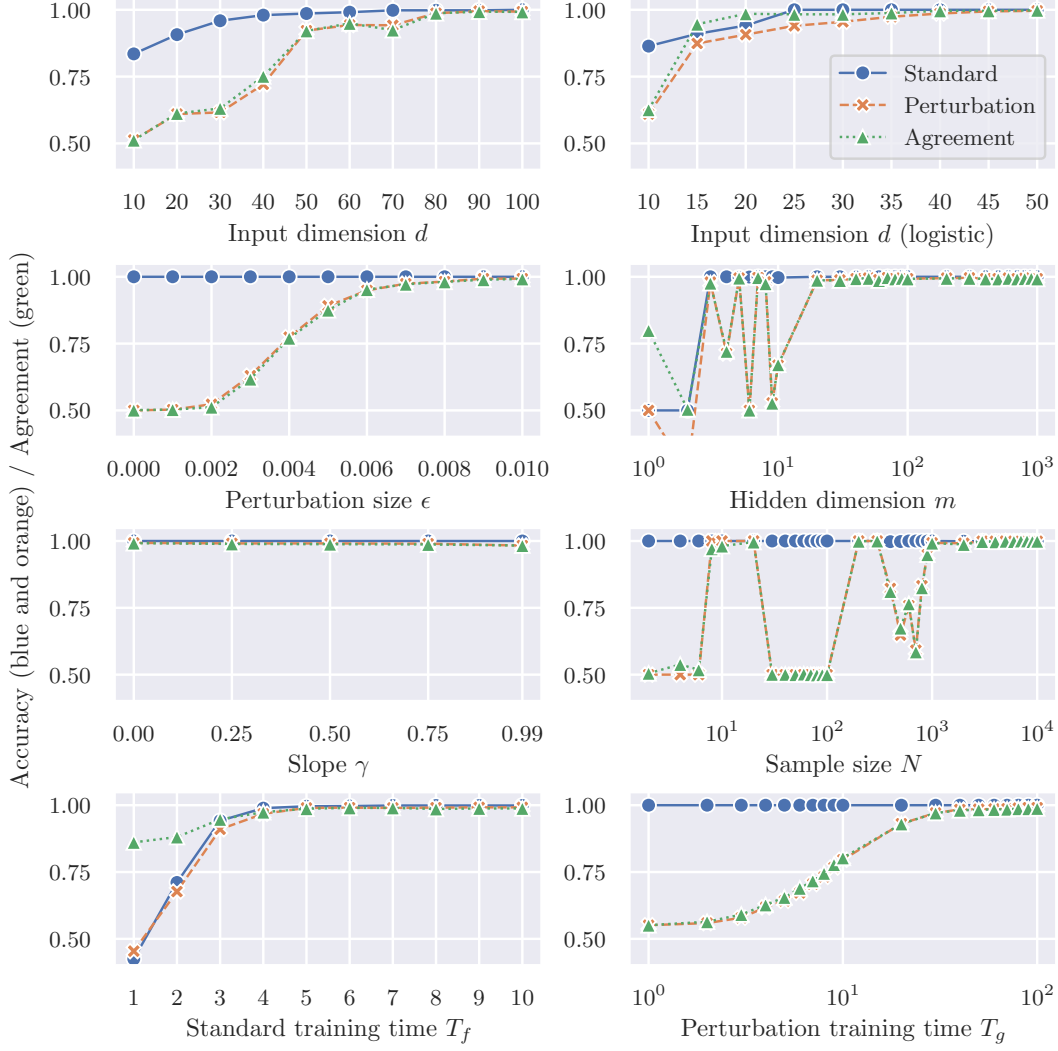
16

Figure A4: Accuracy and agreement ratio on the mean-shifted Gaussian in Scenario (a). The blue lines represent the accuracy of the classifier f on $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, i.e., training accuracy. The orange lines represent the accuracy of the classifier g on $\mathcal{D}$. The green lines represent the prediction agreement between f and g on the test dataset.

(b) For any $i \in [m]$,

$$\mathbb{E}\big[\exp\big(t(X_i^2 Y_i - \mathbb{E}[X_i^2]\mathbb{E}[Y_i])\big)\big] = \mathbb{E}\left[\sum_{n=0}^\infty \frac{t^n(X_i^2 Y_i - \mathbb{E}[X_i^2]\mathbb{E}[Y_i])^n}{n!}\right] \tag{A29}$$

$$= 1 + \sum_{n=2}^\infty \frac{t^n \mathbb{E}[(X_i^2 Y_i - \mathbb{E}[X_i^2]\mathbb{E}[Y_i])^n]}{n!} \tag{A30}$$

$$\leq 1 + \sum_{n=2}^\infty \frac{t^n \mathbb{E}[(X_i^2 Y_i + \mathbb{E}[X_i^2]\mathbb{E}[Y_i])^n]}{n!}. \tag{A31}$$

By Jensen's inequality,

$$\sum_{n=2}^\infty \frac{t^n \mathbb{E}[(X_i^2 Y_i + \mathbb{E}[X_i^2]\mathbb{E}[Y_i])^n]}{n!} \leq \sum_{n=2}^\infty \frac{2^{n-1} t^n (\mathbb{E}[X_i^{2n}]\mathbb{E}[Y_i^n] + \mathbb{E}[X_i^2]^n \mathbb{E}[Y_i]^n)}{n!} \tag{A32}$$
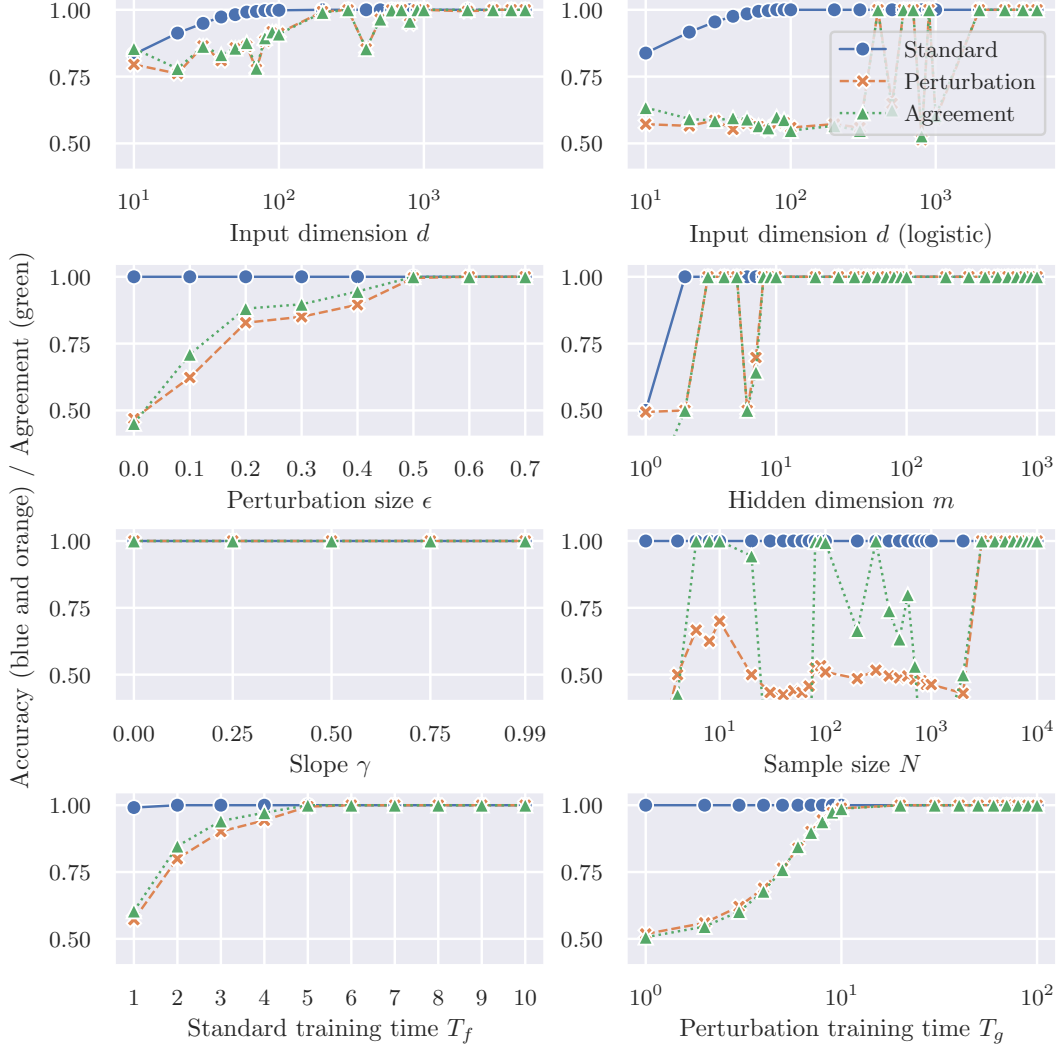
17

Figure A5: Accuracy and agreement ratio on the mean-shifted Gaussian in Scenario (b). The description is the same as Fig. A4.

$$\leq \sum_{n=2}^{\infty} \frac{2^n t^n \mathbb{E}[X_i^{2n}]\mathbb{E}[Y_i^n]}{n!}. \tag{A33}$$

Since $\mathbb{E}[X_i^{2n}] = \sigma^{2n}(2n-1)!! \leq 2^n \sigma^{2n} n!$ and $\mathbb{E}[Y_i^n] \leq 1$,

$$1 + \sum_{n=2}^{\infty} \frac{2^n t^n \mathbb{E}[X_i^{2n}]\mathbb{E}[Y_i^n]}{n!} \leq 1 + \sum_{n=2}^{\infty} 4^n t^n \sigma^{2n} = 1 + \frac{16t^2\sigma^4}{1 - 4t\sigma^2}. \tag{A34}$$

For $|t| \leq 1/(8\sigma^2)$,

$$1 + \frac{16t^2\sigma^4}{1 - 4t\sigma^2} \leq 1 + 32t^2\sigma^4 \leq \exp(32t^2\sigma^4). \tag{A35}$$

Thus, $X_i^2 Y_i$ follows $\text{SE}(64\sigma^4, 8\sigma^2)$, where $SE(a,b)$ is a sub-exponential random variable with parameters $a, b > 0$. Note that a random variable Z is called sub-exponential with parameters $a, b > 0$, $\text{SE}(a,b)$, if its moment generating function satisfies

$$\forall |t| \leq \frac{1}{b}, \qquad \mathbb{E}[\exp(t(Z - \mathbb{E}[Z]))] \leq \exp\left(\frac{t^2 a}{2}\right). \tag{A36}$$
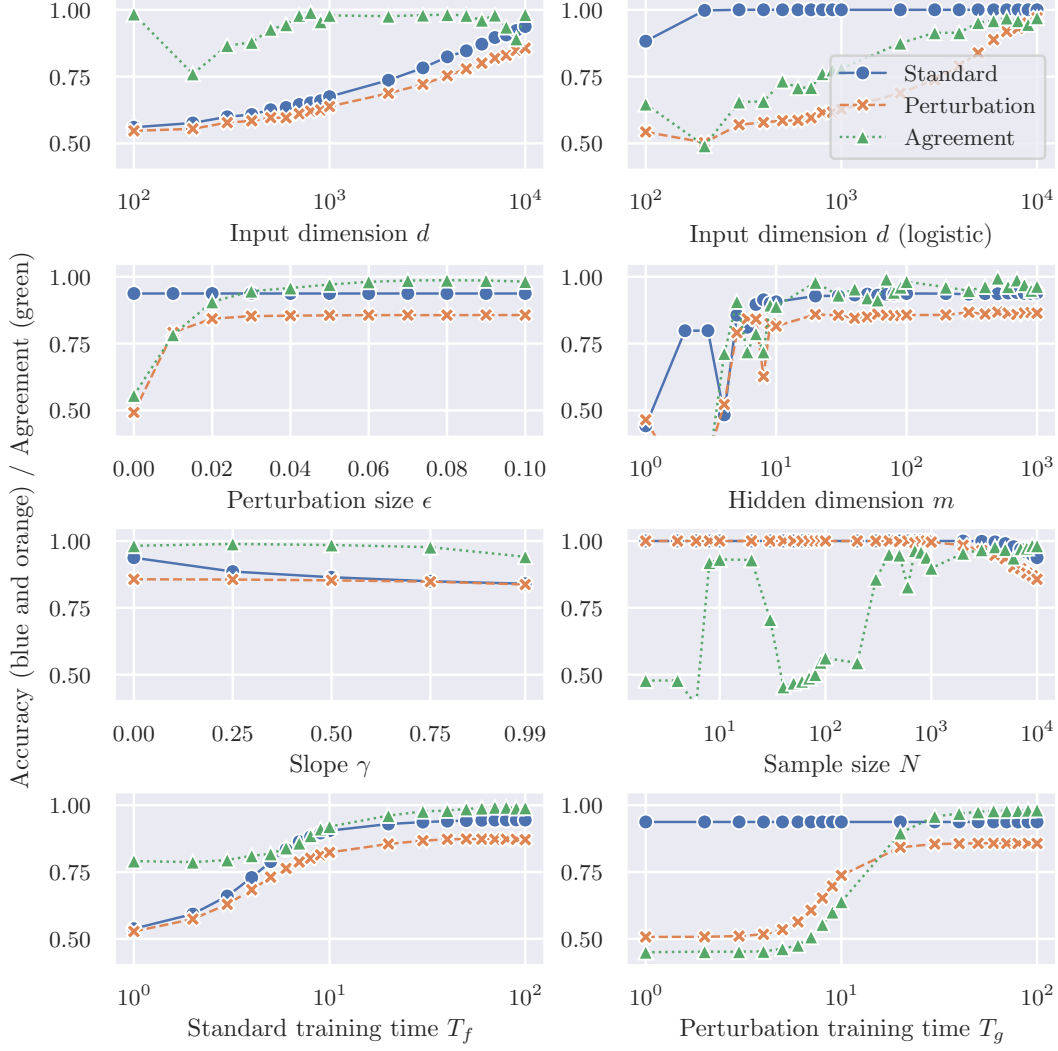
18

Figure A6: Accuracy and agreement ratio on the zero-mean Gaussian in Scenario (a). The description is the same as Fig. A4.

By [16], $\sum_i^m X_i^2 Y_i$ follows $\mathrm{SE}(64m\sigma^4, 8\sigma^2)$. In addition, by [16], $Z \sim \mathrm{SE}(a,b)$ satisfies

$$\mathbb{P}[|Z - \mathbb{E}[Z]| \geq C] \leq 2\exp\left(-\frac{1}{2}\min\left(\frac{C^2}{a}, \frac{C}{b}\right)\right). \tag{A37}$$

Therefore, with probability at least $1 - \delta$,

$$|Z - \mathbb{E}[Z]| < \max\left(2b\ln\left(\frac{2}{\delta}\right), \sqrt{2a\ln\left(\frac{2}{\delta}\right)}\right). \tag{A38}$$

(c) Let $k \in [m-1]$ be a positive integer. Let $\mathrm{Bi}(m,p)$ be the Binomial distribution and $\mathrm{Be}(p)$ be the Bernoulli distribution with $p \in (0, (k+1)/m)$. By Hoeffding's inequality,

$$\sum_{i=k+1}^m \binom{m}{i} p^i (1-p)^{m-i} = \mathbb{P}_{Y \sim \mathrm{Bi}(m,p)}[Y \geq k+1] \tag{A39}$$

$$= \mathbb{P}_{Z_1,\ldots,Z_m \sim \mathrm{Be}(p)}\left[\sum_{i=1}^m Z_i \geq k+1\right] \tag{A40}$$
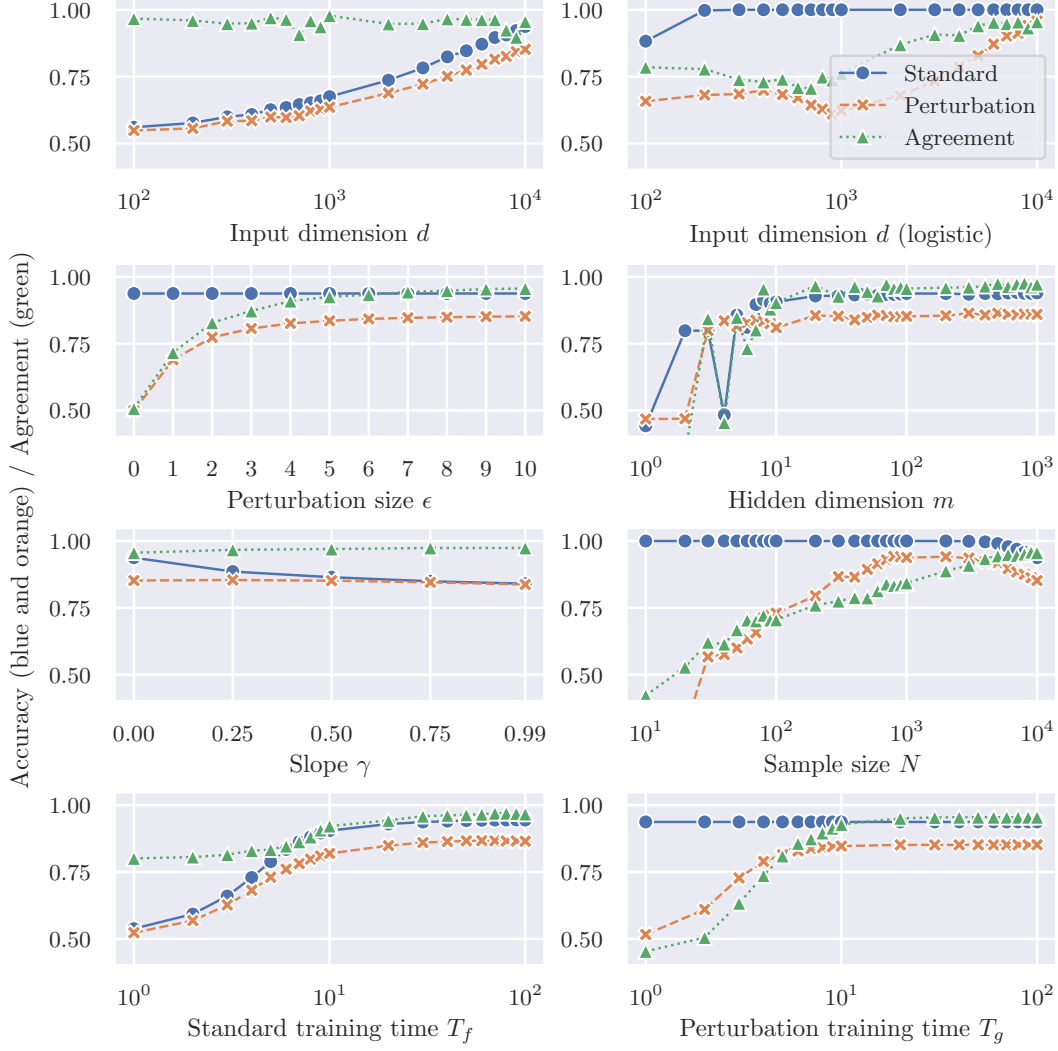
Figure A7: Accuracy and agreement ratio on the zero-mean Gaussian in Scenario (b). The description is the same as Fig. A4.

$$\leq \exp\left(-\frac{2(k+1-mp)^2}{m}\right) \tag{A41}$$

$$= \exp\left(-2m\left(\frac{k+1}{m}-p\right)^2\right). \tag{A42}$$

Now,

$$\mathbb{P}[\text{ there are at least } k+1 \text{ instances such that } |X_i| < C]$$

$$= \sum_{i=k+1}^{m} \binom{m}{i} \mathbb{P}[|X_i| < C]^i \mathbb{P}[|X_i| \geq C]^{m-i} \tag{A43}$$

$$\leq \exp\left(-2m\left(\frac{k+1}{m}-\mathbb{P}[|X_i| < C]\right)^2\right). \tag{A44}$$

For $\delta > \exp\left(-\frac{2(k+1)^2}{m}\right)$ and $\mathbb{P}[|X_i| < C] \leq \frac{k+1}{m} - \sqrt{-\frac{\ln\delta}{2m}}$,

$$\mathbb{P}[\text{ there are at least } k+1 \text{ instances such that } |X_i| < C] \leq \delta. \tag{A45}$$
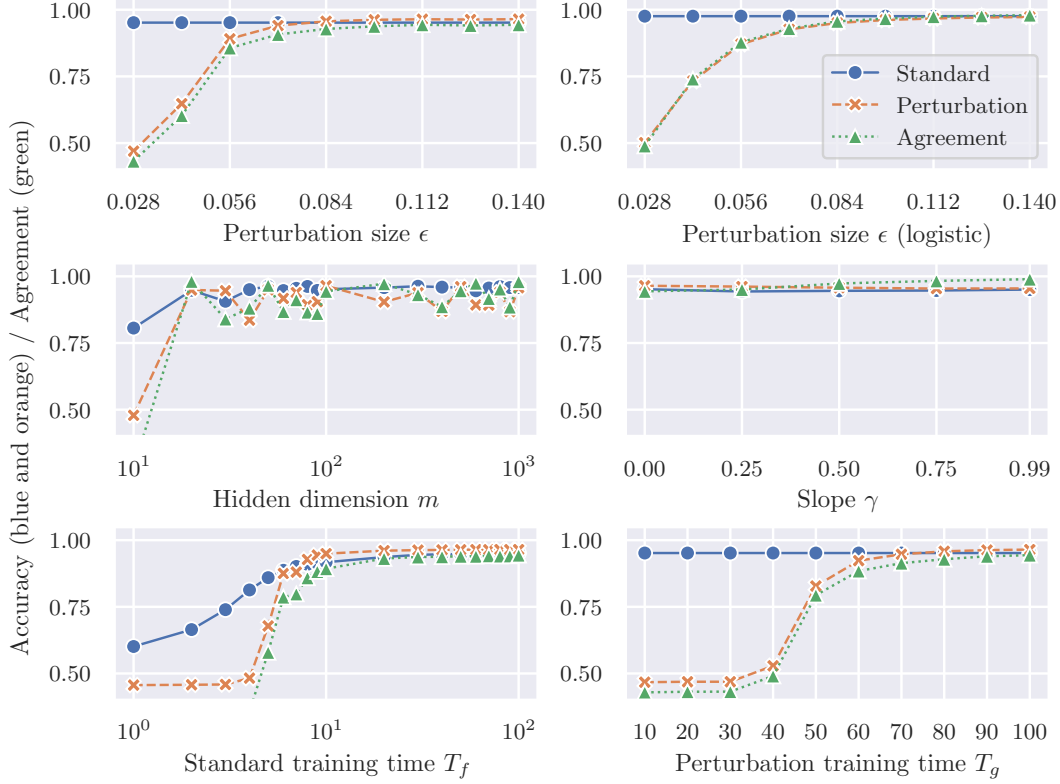
20

Figure A8: Accuracy and agreement ratio on MNIST in Scenario (a). The description is the same as Fig. A4.

Since the areas of a square with sides of length x and a quarter-circle with a radius of $2x/\sqrt{\pi}$ are the same, and because $s^2 + t^2$ is always larger in the square than in the quarter-circle outside the common area, an upper bound of $\text{erf}(x)$ can be computed as

$$\text{erf}(x)^2 = \frac{4}{\pi} \int_0^x \int_0^x \exp\left(-(s^2 + t^2)\right) \mathrm{d}s\,\mathrm{d}t \tag{A46}$$

$$\leq \frac{4}{\pi} \int_0^{2x/\sqrt{\pi}} \int_0^{\frac{\pi}{2}} r \exp\left(-r^2\right) \mathrm{d}\theta\,\mathrm{d}r \tag{A47}$$

$$= 1 - \exp\left(-\frac{4}{\pi} x^2\right). \tag{A48}$$

Thus,

$$\mathbb{P}[|X_i| < C] = \text{erf}\left(\frac{C}{\sqrt{2\sigma^2}}\right) \leq \sqrt{1 - \exp\left(-\frac{2C^2}{\pi\sigma^2}\right)}. \tag{A49}$$

Therefore,

$$C \leq \sqrt{-\frac{\pi\sigma^2}{2} \ln\left(1 - \left(\frac{k+1}{m} - \sqrt{-\frac{\ln\delta}{2m}}\right)^2\right)} \implies \mathbb{P}[|X_i| < C] \leq \frac{k+1}{m} - \sqrt{-\frac{\ln\delta}{2m}}. \tag{A50}$$

$\square$

Lemma C.2 (Hoeffding's inequality)**.** *Let $s_1, \ldots, s_N \in \mathbb{R}$ be real numbers and $X_1, \ldots, X_N \in \{\pm 1\}$ be i.i.d. random variables. Suppose that $\mathbb{E}[X_n] = 0$ for every $n \in [N]$. Then, for $0 < \delta < 1$,*
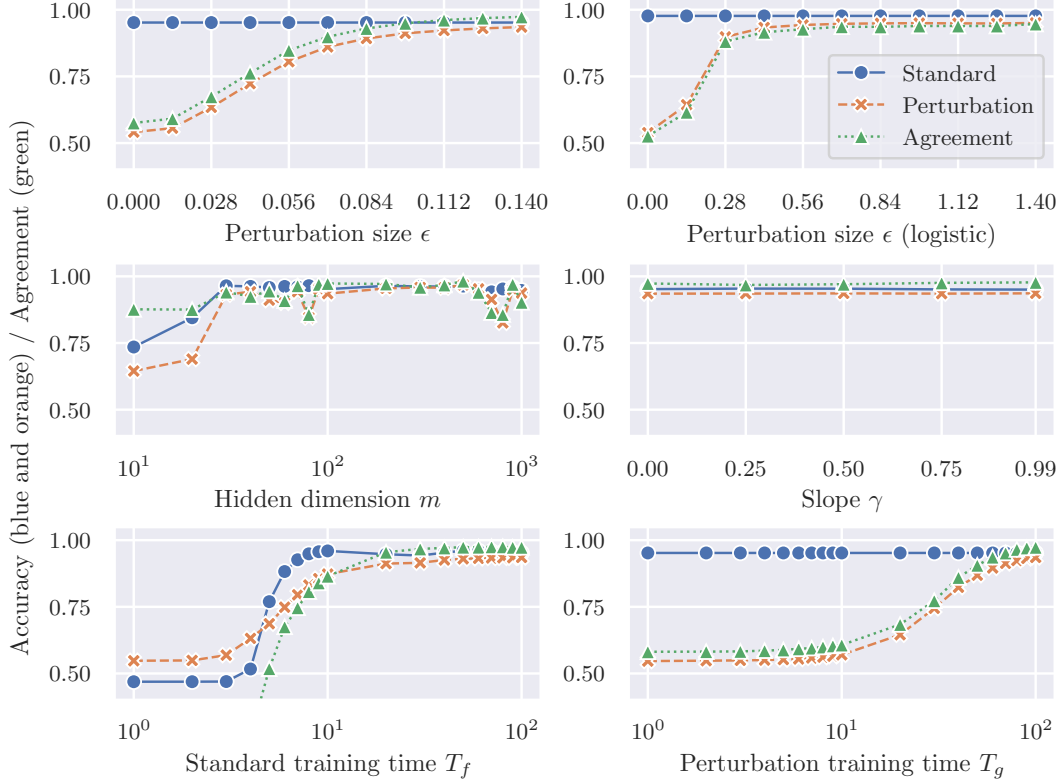
Figure A9: Accuracy and agreement ratio on MNIST in Scenario (b). The description is the same as Fig. A4.

with probability at least $1 - \delta$,

$$\left| \sum_n^N s_n X_n \right| < \sqrt{2 \ln\left(\frac{2}{\delta}\right) \sum_n^N s_n^2}. \tag{A51}$$

Proof. By Hoeffding's inequality, the claim is established. $\qquad\square$

Lemma C.3 (Expectation of product of derivatives of activation functions, part 1)**.** *Denote a symmetric positive definite matrix by*

$$\mathbf{\Sigma} := \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \tag{A52}$$

where $a, c > 0$ and $ac - b^2 > 0$. Then,

$$\left| \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\phi'(x_1)\phi'(x_2)}{2\pi\sqrt{|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \mathbf{\Sigma}^{-1} \boldsymbol{x}\right) \mathrm{d}x - \frac{(1+\gamma)^2}{4} \right|$$

$$\leq \frac{(1+\gamma)(1-\gamma)}{4}\left(1 - \sqrt{\frac{e}{2\pi}\frac{ac - b^2}{ac + b^2}}\right). \tag{A53}$$

Proof. The inverse of $\mathbf{\Sigma}$ is

$$\mathbf{\Sigma}^{-1} = \frac{1}{ac - b^2}\begin{bmatrix} c & -b \\ -b & a \end{bmatrix} =: \frac{\tilde{\mathbf{\Sigma}}^{-1}}{ac - b^2}. \tag{A54}$$
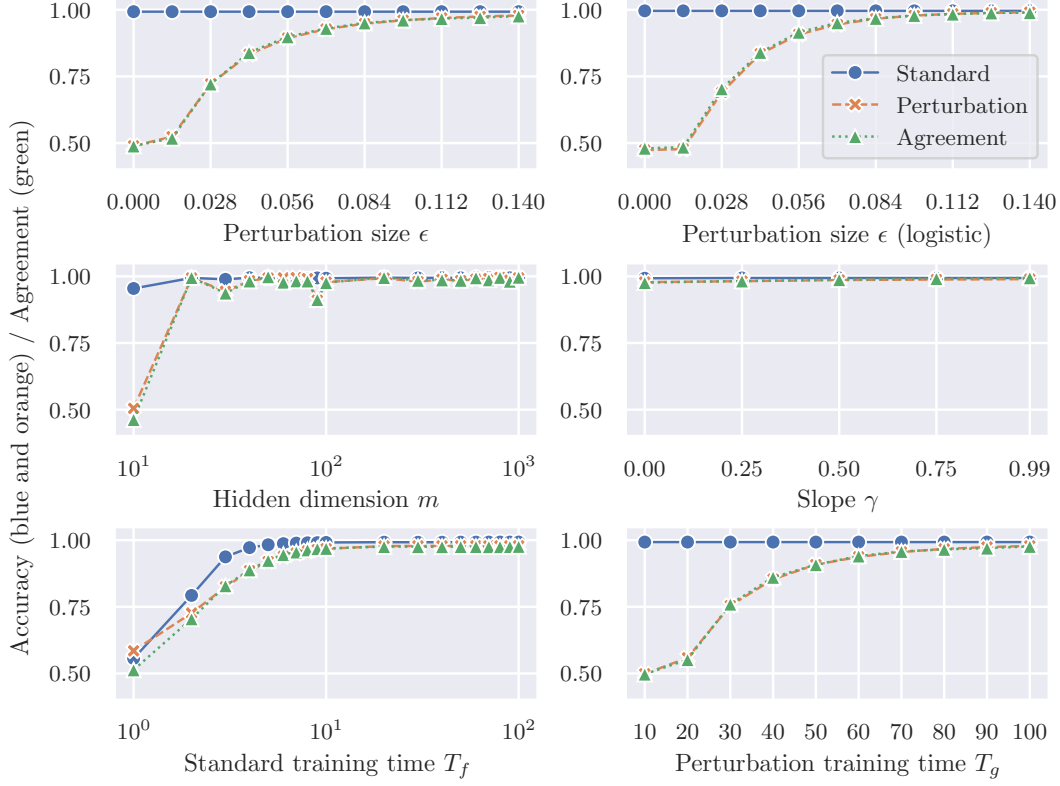
22

Figure A10: Accuracy and agreement ratio on Fashion-MNIST in Scenario (a). The description is the same as Fig. A4.

Using $\boldsymbol{y} := \boldsymbol{x}/\sqrt{ac-b^2}$, the quadratic form can be represented as

$$\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} = \frac{\boldsymbol{x}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{x}}{ac-b^2} = \boldsymbol{y}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{y}. \tag{A55}$$

Because $|\partial \boldsymbol{x}/\partial \boldsymbol{y}| = ac - b^2$,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\phi'(x_1)\phi'(x_2)}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}\right) \mathrm{d}\boldsymbol{x}$$
$$= \frac{\sqrt{ac-b^2}}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi'(y_1)\phi'(y_2) \exp\left(-\frac{1}{2}\boldsymbol{y}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{y}\right) \mathrm{d}\boldsymbol{y}. \tag{A56}$$

With $z := y_1 - by_2/c$,

$$\boldsymbol{y}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{y} = c\left(y_1 - \frac{by_2}{c}\right)^2 + \left(a - \frac{b^2}{c}\right)y_2^2 = cz^2 + \left(a - \frac{b^2}{c}\right)y_2^2. \tag{A57}$$

Now,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi'(y_1)\phi'(y_2) \exp\left(-\frac{1}{2}\boldsymbol{y}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{y}\right) \mathrm{d}\boldsymbol{y}$$
$$= \int_{-\infty}^{\infty} \phi'(y_2)\left(\int_{-\infty}^{\infty} \phi'(z + by_2/c) \exp\left(-\frac{c}{2}z^2\right) \mathrm{d}z\right) \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) \mathrm{d}y_2. \tag{A58}$$

The integral along z can be computed as

$$\int_{-\infty}^{\infty} \phi'(z + by_2/c) \exp\left(-\frac{c}{2}z^2\right) \mathrm{d}z$$
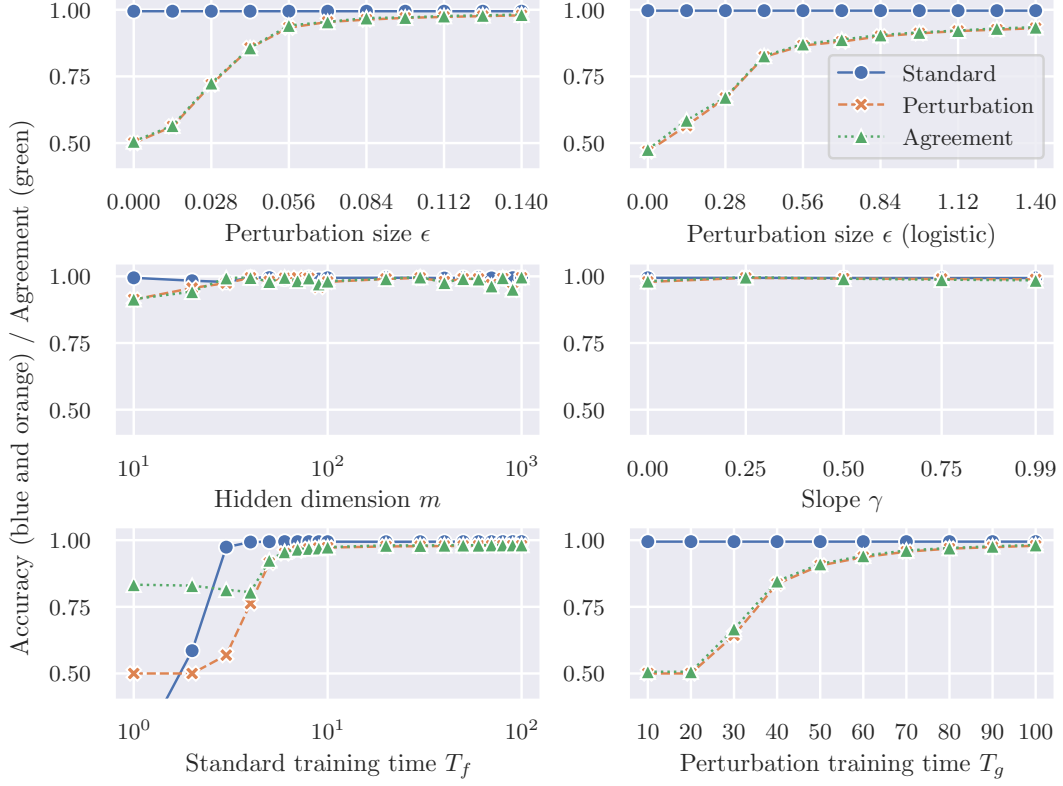
23

Figure A11: Accuracy and agreement ratio on Fashion-MNIST in Scenario (b). The description is the same as Fig. A4.

$$= \int_{-by_2/c}^{\infty} \exp\left(-\frac{c}{2}z^2\right) dz + \gamma \int_{-\infty}^{-by_2/c} \exp\left(-\frac{c}{2}z^2\right) dz \tag{A59}$$

$$= \int_{0}^{\infty} \exp\left(-\frac{c}{2}z^2\right) dz - \int_{0}^{-by_2/c} \exp\left(-\frac{c}{2}z^2\right) dz$$

$$+ \gamma \int_{-\infty}^{0} \exp\left(-\frac{c}{2}z^2\right) dz - \gamma \int_{-by_2/c}^{0} \exp\left(-\frac{c}{2}z^2\right) dz \tag{A60}$$

$$=(1+\gamma) \int_{0}^{\infty} \exp\left(-\frac{c}{2}z^2\right) dz - (1-\gamma) \int_{0}^{-by_2/c} \exp\left(-\frac{c}{2}z^2\right) dz \tag{A61}$$

$$=(1+\gamma)\sqrt{\frac{\pi}{2c}} + (1-\gamma) \int_{0}^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) dz . \tag{A62}$$

Using the above equation,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi'(y_1)\phi'(y_2) \exp\left(-\frac{1}{2}\boldsymbol{y}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{y}\right) d\boldsymbol{y}$$

$$=(1+\gamma)\sqrt{\frac{\pi}{2c}} \int_{-\infty}^{\infty} \phi'(y_2) \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) dy_2$$

$$+ (1-\gamma) \int_{-\infty}^{\infty} \phi'(y_2) \int_{0}^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) dz \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) dy_2 \tag{A63}$$

$$=\frac{\pi(1+\gamma)^2}{2\sqrt{ac-b^2}} + (1+\gamma)(1-\gamma) \int_{0}^{\infty} \int_{0}^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) dz \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) dy_2 . \tag{A64}$$
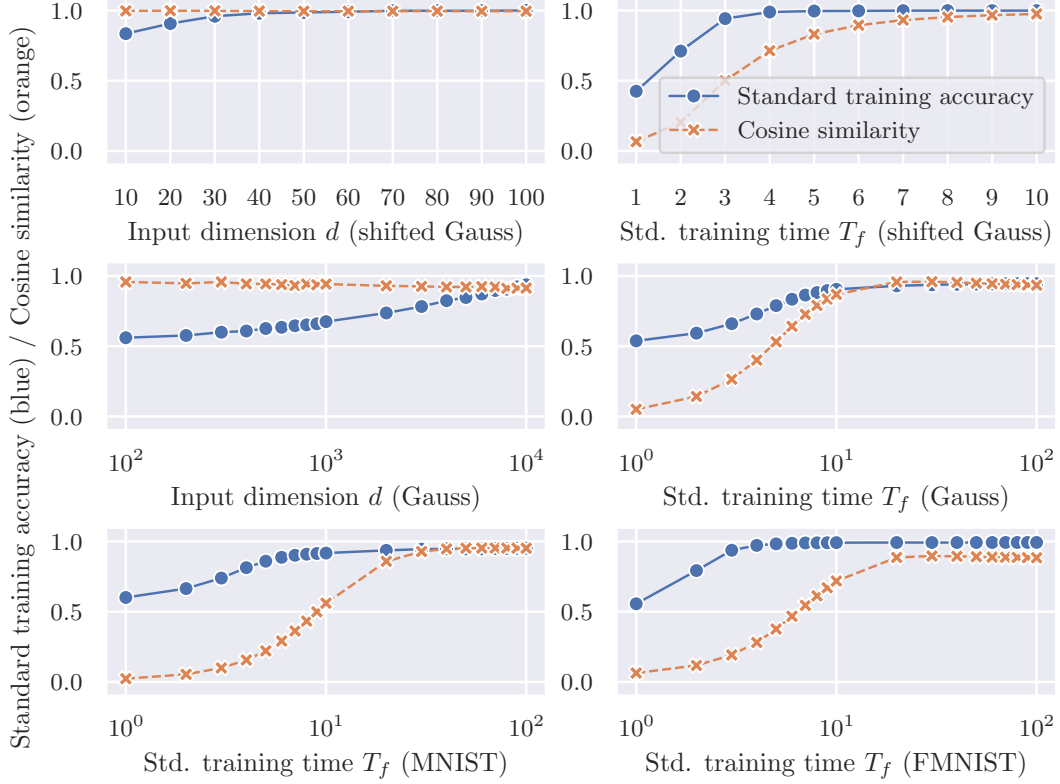
24

Figure A12: Standard training accuracy and cosine similarity. The blue lines represent the accuracy of the classifier f on $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, i.e., training accuracy. The orange lines represent the average cosine similarity between the experimentally calculated adversarial perturbations and the theoretically predicted ones.

Thus,

$$
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\phi'(x_1)\phi'(x_2)}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right) \mathrm{d}x
$$
$$
= \frac{(1+\gamma)^2}{4}
$$
$$
+ \frac{(1+\gamma)(1-\gamma)\sqrt{ac-b^2}}{2\pi} \int_0^{\infty} \int_0^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) \mathrm{d}z \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) \mathrm{d}y_2. \quad \text{(A65)}
$$

Using $t := \operatorname{sgn}(b) z\sqrt{c/2}$,

$$
\int_0^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) \mathrm{d}z = \operatorname{sgn}(b)\sqrt{\frac{2}{c}} \int_0^{\sqrt{\frac{b^2}{2c}}y_2} \exp\left(-t^2\right) \mathrm{d}t \quad \text{(A66)}
$$
$$
= \operatorname{sgn}(b)\sqrt{\frac{\pi}{2c}}\left(1 - \operatorname{erfc}\left(\sqrt{\frac{b^2}{2c}}y_2\right)\right). \quad \text{(A67)}
$$

From [8], with $\alpha = \sqrt{e/(2\pi)}$,

$$
\alpha \exp\left(-\frac{b^2}{c}y_2^2\right) \leq \operatorname{erfc}\left(\sqrt{\frac{b^2}{2c}}y_2\right), \quad \text{(A68)}
$$
$$
\left| \int_0^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) \mathrm{d}z \right| \leq \sqrt{\frac{\pi}{2c}}\left(1 - \alpha \exp\left(-\frac{b^2}{c}y_2^2\right)\right). \quad \text{(A69)}
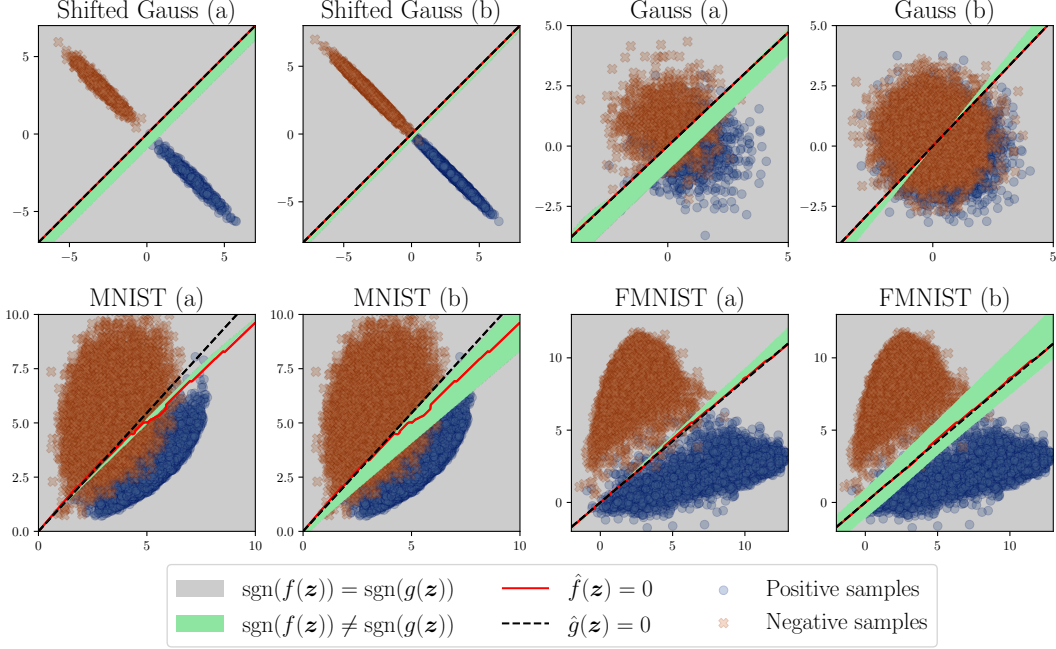$$

25

Figure A13: Prediction matching between the classifiers from standard training and perturbation learning, f and g. The two axes are the normalized average vectors of samples from the positive and negative classes, respectively. The blue circles and orange crosses correspond to the projections of positive and negative samples onto these axes. The gray and green areas indicate regions where two predictions are consistent and inconsistent, respectively. The red solid lines represent $\hat{f}(\boldsymbol{z}) = 0$. The black dashed lines represent $\hat{g}_a(\boldsymbol{z}) = 0$ in Scenario (a) and $\hat{g}_b(\boldsymbol{z}) = 0$ in Scenario (b).

Thus,

$$
\left| \int_0^\infty \int_0^{by_2/c} \exp\left(-\frac{c}{2}z^2\right) \mathrm{d}z \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) \mathrm{d}y_2 \right|
$$

$$
\leq \sqrt{\frac{\pi}{2c}} \left(\int_0^\infty \exp\left(-\frac{ac-b^2}{2c}y_2^2\right) \mathrm{d}y_2 - \alpha \int_0^\infty \exp\left(-\frac{ac+b^2}{2c}y_2^2\right) \mathrm{d}y_2 \right) \tag{A70}
$$

$$
= \sqrt{\frac{\pi}{2c}} \left(\frac{1}{2}\sqrt{\frac{2\pi c}{ac-b^2}} - \frac{\alpha}{2}\sqrt{\frac{2\pi c}{ac+b^2}} \right) \tag{A71}
$$

$$
= \frac{\pi}{2} \left(\frac{1}{\sqrt{ac-b^2}} - \frac{\alpha}{\sqrt{ac+b^2}} \right). \tag{A72}
$$

Finally,

$$
\left| \int_{-\infty}^\infty \int_{-\infty}^\infty \frac{\phi'(x_1)\phi'(x_2)}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}\right) \mathrm{d}x - \frac{(1+\gamma)^2}{4} \right|
$$

$$
\leq \frac{(1+\gamma)(1-\gamma)\sqrt{ac-b^2}}{2\pi} \left(\frac{\pi}{2} \left(\frac{1}{\sqrt{ac-b^2}} - \frac{\alpha}{\sqrt{ac+b^2}} \right) \right) \tag{A73}
$$

$$
= \frac{(1+\gamma)(1-\gamma)}{4} \left(1 - \alpha\sqrt{\frac{ac-b^2}{ac+b^2}} \right). \tag{A74}
$$

$\square$

26

Lemma C.4 (Expectation of product of derivatives of activation functions, part 2). *For any $z_1 \neq z_2 \in \mathbb{R}^d$,*

$$\left| \Phi(z_1, z_2) - \frac{(1+\gamma)^2}{4} \right| \leq \frac{(1+\gamma)(1-\gamma)}{4} \lambda(z_1, z_2). \tag{A75}$$

Proof. By the reproductive property of Gaussian distributions, $\langle v, z_1 \rangle + a$ follows $\mathcal{N}(0, \|z_1\|^2/d + 1)$. Since any linear combination of $\langle v, z_1 \rangle + a$ and $\langle v, z_2 \rangle + a$ has a univariate Gaussian distribution, $\langle v, z_1 \rangle + a$ and $\langle v, z_2 \rangle + a$ follow a multivariate Gaussian distribution. The covariance matrix Σ can be computed as

$$\Sigma = \begin{bmatrix} \|z_1\|^2/d + 1 & \langle z_1, z_2 \rangle/d + 1 \\ \langle z_1, z_2 \rangle/d + 1 & \|z_2\|^2/d + 1 \end{bmatrix}. \tag{A76}$$

Thus, by Lemma C.3, the claim is established. □

D Main Proof

For notational simplicity, we use the following abbreviation for $i \in [m]$ and $n \in [N]$:

$$h_{f,i,t}(z) := \langle v_i(t), z \rangle + a_i(t), \qquad h_{g,i,t}(z) := \langle w_i(t), z \rangle + b_i(t), \tag{A77}$$

$$\psi_{f,i,t}(z) := \phi(h_{f,i,t}(z)), \qquad \psi_{g,i,t}(z) := \phi(h_{g,i,t}(z)), \tag{A78}$$

$$\psi'_{f,i,t}(z) := \phi'(h_{f,i,t}(z)), \qquad \psi'_{g,i,t}(z) := \phi'(h_{g,i,t}(z)), \tag{A79}$$

$$\ell_{f,n,t} := \ell(-y_n f(x_n; \theta_{V,a}(t))), \qquad \ell_{g,n,t} := \ell(-y_n^{\mathrm{adv}} g(x_n^{\mathrm{adv}}; \theta_{W,b}(t))), \tag{A80}$$

$$\ell'_{f,n,t} := \ell'(-y_n f(x_n; \theta_{V,a}(t))), \qquad \ell'_{g,n,t} := \ell'(-y_n^{\mathrm{adv}} g(x_n^{\mathrm{adv}}; \theta_{W,b}(t))), \tag{A81}$$

$$\bar{\ell}'_f(t) := \frac{1}{N} \sum_n^N \ell'_{f,n,t}, \qquad \bar{\ell}'_g(t) := \frac{1}{N} \sum_n^N \ell'_{g,n,t}. \tag{A82}$$

Moreover, denote the subset of $[m]$ consisting of the smallest $\eta\sqrt{m} \in [m-1]$ elements in terms of $|h_{f,i,0}(z)|$ by $\mathcal{S}_f(z)$. Similarly, we define $\mathcal{S}_g(z)$ based on $|h_{g,i,0}(z)|$. The function $\kappa_{f,i}(z)$ returns one if $i \in \mathcal{S}_f(z)$ and zero otherwise. Similarly, we define $\kappa_{g,i}(z)$. For $\exp(-2(\eta\sqrt{m}+1)^2/m) < \delta < 1$, let

$$C_{\mathrm{thr}}(z, \delta) := \sqrt{-\pi \left(\frac{\|z\|^2}{d} + 1 \right) \ln \left(1 - \left(\frac{\eta\sqrt{m}+1}{m} - \sqrt{-\frac{\ln \delta}{2m}} \right)^2 \right)} = \tilde{\Theta}(1). \tag{A83}$$

D.1 Assumptions on Properties of Neural Networks and Their Justifications

For reference, we state the following assumption on the network f:

Assumption D.1 (Properties of neural network f). Let $z \in \mathbb{R}^d$ be a real-valued vector and $\exp(-2(\eta\sqrt{m}+1)^2/m) < \delta < 1$ be a positive value.

(a) For any $i \in [m]$ and $j \in [d]$, $|v_{i,j}(0)| < \sqrt{(2/d)\ln(2dm/\delta)}$.

(b) For any $i \in [m]$, $|\alpha_i| < \sqrt{(2/m)\ln(2m/\delta)}$.

(c) For any $i \in [m]$, $|\langle v_i(0), z \rangle| < \sqrt{(2/d)\ln(2m/\delta)}\|z\|$.

(d) For any $i \in [m]$, $|h_{f,i,0}(z)| < \sqrt{2(\|z\|^2/d + 1)\ln(2m/\delta)}$.

(e) $|f(z; 0)| < \sqrt{2(\|z\|^2/d + 1)\ln(2/\delta)}$.

(f) For any $j \in [d]$,

$$\left| \sum_i^m \alpha_i \psi'_{f,i,0}(x_n) v_{i,j}(0) \right| < \sqrt{\frac{2}{m} \ln\left(\frac{2}{\delta}\right) \sum_i^m \psi'_{f,i,0}(x_n)^2 v_{i,j}(0)^2}. \tag{A84}$$

(g)

$$\left| \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n) \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle \right| < \sqrt{\frac{2}{m} \ln\left(\frac{2}{\delta}\right) \sum_i^m \psi'_{f,i,0}(\boldsymbol{x}_n)^2 \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle^2}. \qquad \text{(A85)}$$

(h)

$$\left| \sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n, \boldsymbol{z}) \right| < \frac{16}{\sqrt{m}} \ln\left(\frac{2}{\delta}\right). \qquad \text{(A86)}$$

(i) There are at most $\eta\sqrt{m}$ instances such that $|h_{f,i,0}(\boldsymbol{z})| < C_{\text{thr}}(\boldsymbol{z}, \delta)/\sqrt{2}$.

Assumption D.1 is justified as follows:

Lemma D.2 (Justification of Assumption D.1)**.**

    (a) With probability at least $1 - \delta$, Assumption D.1(a) holds.
    (b) With probability at least $1 - \delta$, Assumption D.1(b) holds.
    (c) With probability at least $1 - \delta$, Assumption D.1(c) holds.
    (d) With probability at least $1 - \delta$, Assumption D.1(d) holds.
    (e) With probability at least $1 - \delta$, Assumption D.1(e) holds.
    (f) With probability at least $1 - \delta$, Assumption D.1(f) holds.
    (g) With probability at least $1 - \delta$, Assumption D.1(g) holds.
    (h) With probability at least $1 - \delta$, Assumption D.1(h) holds.
    (i) With probability at least $1 - \delta$, Assumption D.1(i) holds.
    (j) With probability at least $1 - 9\delta$, Assumption D.1 holds.

Proof.

(a), (b) and (h) See Lemma C.1.

(c) to (g) and (i) By the reproductive property of Gaussian random variables, $\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle$, $h_{f,i,0}(\boldsymbol{z})$, $f(\boldsymbol{z}; 0)$, $\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n) v_{i,j}(0)$, and $\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n) \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle$ follow the Gaussian $\mathcal{N}(0, \|\boldsymbol{z}\|^2/d)$, $\mathcal{N}(0, \|\boldsymbol{z}\|^2/d + 1)$, $\mathcal{N}(0, \|\boldsymbol{z}\|^2/d + 1)$, $\mathcal{N}(0, (1/m) \sum_i^m \psi'_{f,i,0}(\boldsymbol{x}_n)^2 v_{i,j}(0)^2)$, and $\mathcal{N}(0, (1/m) \sum_i^m \psi'_{f,i,0}(\boldsymbol{x}_n)^2 \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle^2)$, respectively. By Lemma C.1, the claim is established.

(j) By Bonferroni's inequality, the claim is established.

$\square$

Similarly, we consider the following assumption on the network g and its justification:

Assumption D.3 (Properties of neural network g)**.** Let $\boldsymbol{z} \in \mathbb{R}^d$ be a real-valued vector and $\exp\left(-2(\eta\sqrt{m} + 1)^2/m\right) < \delta < 1$ be a positive value.

    (a) For any $i \in [m]$ and $j \in [d]$, $|w_{i,j}(0)| < \sqrt{(2/d)\ln(2dm/\delta)}$.
    (b) For any $i \in [m]$, $|\beta_i| < \sqrt{(2/m)\ln(2m/\delta)}$.
    (c) For any $i \in [m]$, $|\langle \boldsymbol{w}_i(0), \boldsymbol{z} \rangle| < \sqrt{(2/d)\ln(2m/\delta)}\|\boldsymbol{z}\|$.
    (d) For any $i \in [m]$, $|h_{g,i,0}(\boldsymbol{z})| < \sqrt{2(\|\boldsymbol{z}\|^2/d + 1)\ln(2m/\delta)}$.
    (e) $|g(\boldsymbol{z}; 0)| < \sqrt{2(\|\boldsymbol{z}\|^2/d + 1)\ln(2/\delta)}$.
    (f) For any $j \in [d]$,

$$\left| \sum_i^m \beta_i \psi'_{g,i,0}(\boldsymbol{x}_n^{\text{adv}}) w_{i,j}(0) \right| < \sqrt{\frac{2}{m} \ln\left(\frac{2}{\delta}\right) \sum_i^m \psi'_{g,i,0}(\boldsymbol{x}_n^{\text{adv}})^2 w_{i,j}(0)^2}. \qquad \text{(A87)}$$

(g)

$$\left| \sum_i^m \beta_i \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}}) \langle \boldsymbol{w}_i(0), \boldsymbol{z} \rangle \right| < \sqrt{\frac{2}{m} \ln\left(\frac{2}{\delta}\right) \sum_i^m \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}})^2 \langle \boldsymbol{w}_i(0), \boldsymbol{z} \rangle^2}. \quad \text{(A88)}$$

(h)

$$\left| \sum_i^m \beta_i^2 \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}}) \psi'_{g,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \right| < \frac{16}{\sqrt{m}} \ln\left(\frac{2}{\delta}\right). \quad \text{(A89)}$$

(i) There are at most $\eta\sqrt{m}$ instances such that $|h_{g,i,0}(\boldsymbol{z})| < C_{\mathrm{thr}}(\boldsymbol{z}, \delta)/\sqrt{2}$.

Lemma D.4 (Justification of Assumption D.3). *With probability at least $1 - 9\delta$, Assumption D.3 holds.*

D.2 Wide Width Assumptions

Then, we consider the condition of network width for lazy training.

Assumption D.5 (Wide width for neural network f). *Let $\boldsymbol{z} \in \mathbb{R}^d$ be a real-valued vector and $\exp\left(-2(\eta\sqrt{m}+1)^2/m\right) < \delta < 1$ be a positive value. Network width m satisfies the following inequalities:*

$$m > \frac{4\ln(2m/\delta)(\sum_n^N(|\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle| + 1)\int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t)^2}{N^2 C_{\mathrm{thr}}(\boldsymbol{z}, \delta)^2} = \tilde{\mathcal{O}}\left(d^2\left(\int_0^{T_f} \bar{\ell}'_{f,n,t}\,\mathrm{d}t\right)^2\right), \quad \text{(A90)}$$

$$m > \frac{4\sum_{n,k}^N |\langle \boldsymbol{x}_n, \boldsymbol{x}_k \rangle| \int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t}{N^2} = \tilde{\mathcal{O}}\left(d^2\left(\int_0^{T_f} \bar{\ell}'_{f,n,t}\,\mathrm{d}t\right)^2\right). \quad \text{(A91)}$$

Note that only Ineq. (A90) is required to satisfy lazy training. We impose Ineq. (A91) to simplify some rearrangements of equations. This assumption restricts the transitions of the derivatives of hidden outputs during training.

Lemma D.6 (Lazy training in network f). *If Assumptions D.1 and D.5 hold, then $\psi'_{f,i,t}(\boldsymbol{z}) = \psi'_{f,i,0}(\boldsymbol{z})$ for any $i \in [m] \setminus \mathcal{S}_f(\boldsymbol{z})$ and $0 \le t \le T_f$.*

Proof. By Assumption D.1, the time evolution of $h_{f,i,t}(\boldsymbol{z})$ from $t = 0$ to $t = T_f$ can be computed as

$$|\Delta h_{f,i,T_f}(\boldsymbol{z})| := \left| \left\langle -\int_0^{T_f} \boldsymbol{\nabla}_{\boldsymbol{v}_i} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}(t); \mathcal{D})\,\mathrm{d}t, \boldsymbol{z} \right\rangle - \int_0^{T_f} \nabla_{a_i} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}(t); \mathcal{D})\,\mathrm{d}t \right| \quad \text{(A92)}$$

$$= \left| \int_0^{T_f} \frac{\alpha_i}{N} \sum_n^N y_n \ell'_{f,n,t} \psi'_{f,i,t}(\boldsymbol{x}_n)(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1)\,\mathrm{d}t \right| \quad \text{(A93)}$$

$$\le \frac{|\alpha_i|}{N} \sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1| \int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t \quad \text{(A94)}$$

$$< \frac{1}{N} \sqrt{\frac{2}{m} \ln\left(\frac{2m}{\delta}\right)} \sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1| \int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t. \quad \text{(A95)}$$

By Assumption D.1, if the right term of Ineq. (A95) is smaller than $C_{\mathrm{thr}}(\boldsymbol{z}, \delta)/\sqrt{2}$, then the largest $m - \eta\sqrt{m}$ instances in terms of $|h_{f,i,0}(\boldsymbol{z})|$ satisfy $\mathrm{sgn}(h_{f,i,t}(\boldsymbol{z})) = \mathrm{sgn}(h_{f,i,0}(\boldsymbol{z}))$ for $0 \le t \le T_f$. This condition can be represented as Ineq. (A90). $\qquad\square$

The same discussion can be applied to the network g.

Assumption D.7 (Wide width for neural network g). Let $\boldsymbol{z} \in \mathbb{R}^d$ be a real-valued vector and $\exp\left(-2(\eta\sqrt{m}+1)^2/m\right) < \delta < 1$ be a positive value. Network width m satisfies the following inequalities:

$$m > \frac{4\ln(2m/\delta)(\sum_n^N(|\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}\rangle|+1)\int_0^{T_g}\ell'_{g,n,t}\,\mathrm{d}t)^2}{N^2 C_{\mathrm{thr}}(\boldsymbol{z},\delta)^2} = \tilde{\mathcal{O}}\left(d^2\left(\int_0^{T_g}\bar{\ell}'_{g,n,t}\,\mathrm{d}t\right)^2\right), \quad \text{(A96)}$$

$$m > \frac{4\sum_{n,k}^N|\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{x}_k^{\mathrm{adv}}\rangle|\int_0^{T_g}\ell'_{g,n,t}\,\mathrm{d}t\int_0^{T_g}\ell'_{g,k,t}\,\mathrm{d}t}{N^2} = \tilde{\mathcal{O}}\left(d^2\left(\int_0^{T_g}\bar{\ell}'_{g,n,t}\,\mathrm{d}t\right)^2\right). \quad \text{(A97)}$$

Lemma D.8 (Lazy training in network g). *If Assumptions D.3 and D.7 hold, then* $\psi'_{g,i,t}(\boldsymbol{z}) = \psi'_{g,i,0}(\boldsymbol{z})$ *for any* $i \in [m] \setminus \mathcal{S}_g(\boldsymbol{z})$ *and* $0 \le t \le T_g$.

We can integrate Assumptions D.1 and D.3 into Assumption 3.2.

D.3 Main Part

Lemma D.9 (Representation of network f). *If Assumptions D.1 and D.5 hold, then the network output at the training time T_f can be represented as*

$$\begin{aligned}
f(\boldsymbol{z}; T_f) = {} & f(\boldsymbol{z}; 0) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z})(\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z}))h_{f,i,0}(\boldsymbol{z}) \\
& + \frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n, \boldsymbol{z})\right) \\
& + \frac{1}{N}\sum_n^N y_n\Phi(\boldsymbol{x}_n, \boldsymbol{z})(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t \\
& + \frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n) \\
& \times \int_0^{T_f}\ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n)\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}))\,\mathrm{d}t. \quad \text{(A98)}
\end{aligned}$$

Proof. First, see Lemma D.6. The time evolution of $\boldsymbol{v}_i(t)$ from $t = 0$ to $t = T_f$ can be computed as

$$\Delta\boldsymbol{v}_i(T_f) := -\int_0^{T_f}\boldsymbol{\nabla}_{\boldsymbol{v}_i}\mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}(t); \mathcal{D})\,\mathrm{d}t \quad \text{(A99)}$$

$$= \int_0^{T_f}\frac{\alpha_i}{N}\sum_n^N y_n\ell'_{f,n,t}\psi'_{f,i,t}(\boldsymbol{x}_n)\boldsymbol{x}_n\,\mathrm{d}t \quad \text{(A100)}$$

$$= \frac{\alpha_i}{N}\sum_n^N y_n\boldsymbol{x}_n\int_0^{T_f}\ell'_{f,n,t}\psi'_{f,i,t}(\boldsymbol{x}_n)\,\mathrm{d}t \quad \text{(A101)}$$

$$= \frac{\alpha_i}{N}\sum_n^N(1 - \kappa_{f,i}(\boldsymbol{x}_n))y_n\boldsymbol{x}_n\psi'_{f,i,0}(\boldsymbol{x}_n)\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t$$

$$+ \frac{\alpha_i}{N}\sum_n^N\kappa_{f,i}(\boldsymbol{x}_n)y_n\boldsymbol{x}_n\int_0^{T_f}\ell'_{f,n,t}\psi'_{f,i,t}(\boldsymbol{x}_n)\,\mathrm{d}t \quad \text{(A102)}$$

$$= \frac{\alpha_i}{N} \sum_n^N y_n \boldsymbol{x}_n \psi'_{f,i,0}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t$$

$$- \frac{\alpha_i}{N} \sum_n^N \kappa_{f,i}(\boldsymbol{x}_n) y_n \boldsymbol{x}_n \psi'_{f,i,0}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t$$

$$+ \frac{\alpha_i}{N} \sum_n^N \kappa_{f,i}(\boldsymbol{x}_n) y_n \boldsymbol{x}_n \int_0^{T_f} \ell'_{f,n,t} \psi'_{f,i,t}(\boldsymbol{x}_n) \, \mathrm{d}t \tag{A103}$$

$$= \frac{\alpha_i}{N} \sum_n^N y_n \boldsymbol{x}_n \psi'_{f,i,0}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t$$

$$+ \frac{\alpha_i}{N} \sum_n^N \kappa_{f,i}(\boldsymbol{x}_n) y_n \boldsymbol{x}_n \int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \, \mathrm{d}t \,. \tag{A104}$$

Similarly, the time evolution of $a_i(t)$ from $t = 0$ to $t = T_f$ can be computed as

$$\Delta a_i(T_f) = \frac{\alpha_i}{N} \sum_n^N y_n \int_0^{T_f} \ell'_{f,n,t} \psi'_{f,i,t}(\boldsymbol{x}_n) \, \mathrm{d}t \tag{A105}$$

$$= \frac{\alpha_i}{N} \sum_n^N y_n \psi'_{f,i,0}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t$$

$$+ \frac{\alpha_i}{N} \sum_n^N \kappa_{f,i}(\boldsymbol{x}_n) y_n \int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \, \mathrm{d}t \,. \tag{A106}$$

Thus,

$$\Delta h_{f,i,T_f}(\boldsymbol{z})$$

$$= \frac{\alpha_i}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} \psi'_{f,i,t}(\boldsymbol{x}_n) \, \mathrm{d}t \tag{A107}$$

$$= \frac{\alpha_i}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \psi'_{f,i,0}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t$$

$$+ \frac{\alpha_i}{N} \sum_n^N \kappa_{f,i}(\boldsymbol{x}_n) y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \, \mathrm{d}t \,. \tag{A108}$$

The original network at the training time T_f can be computed as

$$f(\boldsymbol{z}; T_f) := \sum_i^m \alpha_i \psi_{f,i,T_f}(\boldsymbol{z}) \tag{A109}$$

$$= \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{z}) h_{f,i,0}(\boldsymbol{z}) + \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{z}) \Delta h_{f,i,T_f}(\boldsymbol{z}). \tag{A110}$$

The first term of Eq. (A110) can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{z}) h_{f,i,0}(\boldsymbol{z})$$

$$= \sum_i^m \alpha_i(1 - \kappa_{f,i}(\boldsymbol{z})) \psi'_{f,i,0}(\boldsymbol{z}) h_{f,i,0}(\boldsymbol{z}) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z}) \psi'_{f,i,T_f}(\boldsymbol{z}) h_{f,i,0}(\boldsymbol{z}) \tag{A111}$$

$$= f(\boldsymbol{z}; 0) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z})(\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z})) h_{f,i,0}(\boldsymbol{z}). \tag{A112}$$

The second term of Eq. (A110) can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{z}) \Delta h_{f,i,T_f}(\boldsymbol{z}))$$

$$= \sum_i^m \alpha_i (1 - \kappa_{f,i}(\boldsymbol{z})) \psi'_{f,i,0}(\boldsymbol{z}) \Delta h_{f,i,T_f}(\boldsymbol{z}) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z}) \psi'_{f,i,T_f}(\boldsymbol{z}) \Delta h_{f,i,T_f}(\boldsymbol{z}) \tag{A113}$$

$$= \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{z}) \Delta h_{f,i,T_f}(\boldsymbol{z}) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z}) (\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z})) \Delta h_{f,i,T_f}(\boldsymbol{z}). \tag{A114}$$

The first term of Eq. (A114) can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{z}) \Delta h_{f,i,T_f}(\boldsymbol{z})$$

$$= \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{z}) \left[\frac{\alpha_i}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \psi'_{f,i,0}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t \right.$$

$$\left. + \frac{\alpha_i}{N} \sum_n^N \kappa_{f,i}(\boldsymbol{x}_n) y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} (\psi'_{f,i,t}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \, \mathrm{d}t \right] \tag{A115}$$

$$= \frac{1}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t \sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{z})$$

$$+ \frac{1}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1)$$

$$\times \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{z}) \int_0^{T_f} \ell'_{f,n,t} (\psi'_{f,i,t}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \, \mathrm{d}t. \tag{A116}$$

The first term of Eq. (A116) can be rearranged as

$$\frac{1}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t \sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{z}) \tag{A117}$$

$$= \frac{1}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n, \boldsymbol{z}) \right) \tag{A118}$$

$$+ \frac{1}{N} \sum_n^N y_n \Phi(\boldsymbol{x}_n, \boldsymbol{z})(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t. \tag{A119}$$

The second term of Eq. (A114) can be rearranged as

$$\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z}) (\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z})) \Delta h_{f,i,T_f}(\boldsymbol{z})$$

$$= \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z}) (\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z}))$$

$$\times \frac{\alpha_i}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_f} \ell'_{f,n,t} \psi'_{f,i,t}(\boldsymbol{x}_n) \, \mathrm{d}t \tag{A120}$$

$$= \frac{1}{N} \sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1)$$

$$\times \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{z}) (\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z})) \int_0^{T_f} \ell'_{f,n,t} \psi'_{f,i,t}(\boldsymbol{x}_n) \, \mathrm{d}t. \tag{A121}$$

The sum of the second term of Eq. (A116) and Eq. (A121) can be rearranged as

$$
\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z})\int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\,\mathrm{d}t
$$

$$
+\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{z})(\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z}))\int_0^{T_f} \ell'_{f,n,t}\psi'_{f,i,t}(\boldsymbol{x}_n)\,\mathrm{d}t
$$

$$
=\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n)
$$

$$
\times \int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n)\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}))\,\mathrm{d}t\,. \tag{A122}
$$

$\square$

Lemma D.10 (Upper bounds of terms in Lemma D.9). *Assume Assumptions D.1 and D.5.*

(a)

$$
\left|\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z})(\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z}))h_{f,i,0}(\boldsymbol{z})\right|
$$

$$
< 2\eta(1-\gamma)\ln(2m/\delta)\sqrt{\|\boldsymbol{z}\|^2/d + 1} = \tilde{\mathcal{O}}(1). \tag{A123}
$$

(b)

$$
\left|\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t\left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n, \boldsymbol{z})\right)\right|
$$

$$
< \frac{8C_{\mathrm{thr}}(\boldsymbol{z}, \delta)\ln(2/\delta)}{\sqrt{\ln(2m/\delta)}} = \tilde{\mathcal{O}}(1). \tag{A124}
$$

(c)

$$
\left|\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n)\right.
$$

$$
\left.\times \int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n)\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}))\,\mathrm{d}t\right|
$$

$$
< \eta(1-\gamma^2)C_{\mathrm{thr}}(\boldsymbol{z}, \delta)\sqrt{\ln(2m/\delta)} = \tilde{\mathcal{O}}(1). \tag{A125}
$$

Proof.

(a) By Assumption D.1,

$$
\left|\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z})(\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z}))h_{f,i,0}(\boldsymbol{z})\right| \leq (1-\gamma)\sum_i^m |\alpha_i \kappa_{f,i}(\boldsymbol{z})h_{f,i,0}(\boldsymbol{z})| \tag{A126}
$$

$$
< 2\eta(1-\gamma)\ln(2m/\delta)\sqrt{\|\boldsymbol{z}\|^2/d + 1}. \tag{A127}
$$

(b) By Assumption D.1,

$$
\left|\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t\left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n, \boldsymbol{z})\right)\right|
$$

$$
< \frac{16}{N\sqrt{m}}\ln\left(\frac{2}{\delta}\right)\sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1|\int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t\,. \tag{A128}
$$

By Assumption D.5,

$$\frac{16}{N\sqrt{m}} \ln\left(\frac{2}{\delta}\right) \sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t$$

$$< \frac{N C_{\mathrm{thr}}(\boldsymbol{z}, \delta)}{2\sqrt{\ln(2m/\delta)} \sum_n^N (|\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle| + 1) \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t}$$

$$\times \frac{16}{N} \ln\left(\frac{2}{\delta}\right) \sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \tag{A129}$$

$$\leq \frac{8 C_{\mathrm{thr}}(\boldsymbol{z}, \delta) \ln(2/\delta)}{\sqrt{\ln(2m/\delta)}}. \tag{A130}$$

(c) By Assumption D.1,

$$\left| \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n) \int_0^{T_f} \ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n)\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}))\, \mathrm{d}t \right|$$

$$\leq (1 - \gamma^2) \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n) \tag{A131}$$

$$< \frac{2\eta(1 - \gamma^2)}{\sqrt{m}} \ln\left(\frac{2m}{\delta}\right) \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t. \tag{A132}$$

By Assumption D.5,

$$\frac{2\eta(1 - \gamma^2)}{N\sqrt{m}} \ln\left(\frac{2m}{\delta}\right) \sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t$$

$$< \frac{N C_{\mathrm{thr}}(\boldsymbol{z}, \delta)}{2\sqrt{\ln(2m/\delta)} \sum_n^N (|\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle| + 1) \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t}$$

$$\times \frac{2\eta(1 - \gamma^2)}{N} \ln\left(\frac{2m}{\delta}\right) \sum_n^N |\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \tag{A133}$$

$$\leq \eta(1 - \gamma^2) C_{\mathrm{thr}}(\boldsymbol{z}, \delta) \sqrt{\ln(2m/\delta)}. \tag{A134}$$

$\square$

Lemma D.11 (Network prediction of f). *Assume Assumptions D.1 and D.5. If*

$$\left| \frac{1}{N} \sum_n^N y_n \Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \right| \geq \tilde{\mathcal{O}}\left(1 + \int_0^{T_f} \bar{\ell}'_{f,n,t}\, \mathrm{d}t\right), \tag{A135}$$

then

$$\mathrm{sgn}(f(\boldsymbol{z}; T_f)) = \mathrm{sgn}\left(\frac{1}{N} \sum_n^N y_n \Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \right). \tag{A136}$$

Proof. By Assumption D.1 and Lemmas D.9 and D.10, if

$$\left| \frac{1}{N} \sum_n^N y_n \Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \right|$$

$$\geq |f(\boldsymbol{z}; 0)| + \left| \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{z})(\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{z}))h_{f,i,0}(\boldsymbol{z}) \right|$$

34

$$+\left|\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n, \boldsymbol{z})\right)\right|$$

$$+\left|\frac{1}{N}\sum_n^N y_n(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n)\right.$$

$$\left.\times \int_0^{T_f}\ell'_{f,n,t}(\psi'_{f,i,t}(\boldsymbol{x}_n)\psi'_{f,i,T_f}(\boldsymbol{z}) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{z}))\,\mathrm{d}t\right|$$

$$+\left|\frac{1}{N}\sum_n^N y_n\Phi(\boldsymbol{x}_n, \boldsymbol{z})\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right| \tag{A137}$$

$$=\tilde{\mathcal{O}}\left(1 + \int_0^{T_f}\bar{\ell}_{f,n,t}\,\mathrm{d}t\right), \tag{A138}$$

then

$$\operatorname{sgn}(f(\boldsymbol{z}; T_f)) = \operatorname{sgn}\left(\frac{1}{N}\sum_n^N y_n\Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right). \tag{A139}$$

$\square$

Lemma D.12 (Adversarial perturbation). *If Assumptions D.1 and D.5 hold, then the adversarial perturbation defined as Eq. (1) can be represented as*

$$\boldsymbol{r}_n = \epsilon y_n^{\mathrm{adv}}\frac{\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f)}{\|\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f)\|}, \qquad \text{and} \tag{A140}$$

$$\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f)$$

$$=\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\boldsymbol{v}_i(0) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\boldsymbol{v}_i(0)$$

$$+\frac{1}{N}\sum_k^N y_k\boldsymbol{x}_k\int_0^{T_f}\ell'_{f,k,t}\,\mathrm{d}t\left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\right)$$

$$+\frac{1}{N}\sum_k^N y_k\Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\boldsymbol{x}_k\int_0^{T_f}\ell'_{f,k,t}\,\mathrm{d}t$$

$$+\frac{1}{N}\sum_k^N y_k\boldsymbol{x}_k\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)$$

$$\times \int_0^{T_f}\ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t. \tag{A141}$$

Proof. The main term of the adversarial perturbation can be computed as

$$\frac{\boldsymbol{\nabla}_{\boldsymbol{x}_n}\ell_{f,n,T_f}}{\|\boldsymbol{\nabla}_{\boldsymbol{x}_n}\ell_{f,n,T_f}\|} = \frac{-y_n^{\mathrm{adv}}\ell'_{f,n,T_f}\boldsymbol{\nabla}_{\boldsymbol{x}_n}f(\boldsymbol{x}_n; \boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}(T_f))}{\|\ell'_{f,n,T_f}\boldsymbol{\nabla}_{\boldsymbol{x}_n}f(\boldsymbol{x}_n; \boldsymbol{\theta}_{\boldsymbol{V},\boldsymbol{a}}(T_f))\|} \tag{A142}$$

$$=-y_n^{\mathrm{adv}}\frac{\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f)}{\|\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f)\|}. \tag{A143}$$

The leading term can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f) = \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(0) + \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\Delta\boldsymbol{v}_i(T_f). \tag{A144}$$

35

The first term of Eq. (A144) can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(0)$$

$$=\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\boldsymbol{v}_i(0) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\boldsymbol{v}_i(0). \qquad (A145)$$

The second term of Eq. (A144) can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\Delta\boldsymbol{v}_i(T_f)$$

$$=\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\Delta\boldsymbol{v}_i(T_f) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)\Delta\boldsymbol{v}_i(T_f). \qquad (A146)$$

The first term of Eq. (A146) can be rearranged as

$$\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\Delta\boldsymbol{v}_i(T_f)$$

$$=\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\left[\frac{\alpha_i}{N}\sum_k^N y_k \boldsymbol{x}_k \psi'_{f,i,0}(\boldsymbol{x}_k)\int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t\right.$$

$$\left.+\frac{\alpha_i}{N}\sum_k^N \kappa_{f,i}(\boldsymbol{x}_k)y_k \boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t\right] \qquad (A147)$$

$$=\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k)$$

$$+\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)\psi'_{f,i,0}(\boldsymbol{x}_n)\int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t. \qquad (A148)$$

The second term of Eq. (A146) can be rearranged as

$$\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\Delta\boldsymbol{v}_i(T_f)$$

$$=\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\frac{\alpha_i}{N}\sum_k^N y_k \boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}\psi'_{f,i,t}(\boldsymbol{x}_k)\,\mathrm{d}t \qquad (A149)$$

$$=\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\int_0^{T_f} \ell'_{f,k,t}\psi'_{f,i,t}(\boldsymbol{x}_k)\,\mathrm{d}t. \qquad (A150)$$

The sum of the second term of Eq. (A148) and Eq. (A150) can be rearranged as

$$\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)\psi'_{f,i,0}(\boldsymbol{x}_n)\int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t$$

$$+\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\int_0^{T_f} \ell'_{f,k,t}\psi'_{f,i,t}(\boldsymbol{x}_k)\,\mathrm{d}t$$

$$=\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)$$

$$\times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t. \qquad (A151)$$

$$\square$$

Lemma D.13 (Upper bound of norm of adversarial perturbation). *Assume Assumptions D.1 and D.5.*

(a)

$$\left\| \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n) \boldsymbol{v}_i(0) \right\|^2 < 4\ln(2dm/\delta)\ln(2/\delta) = \tilde{\mathcal{O}}(1). \tag{A152}$$

(b)

$$\left\| \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\boldsymbol{v}_i(0) \right\|^2$$
$$< 4\eta^2(1-\gamma)^2 \ln(2m/\delta)\ln(2dm/\delta) = \tilde{\mathcal{O}}(1). \tag{A153}$$

(c)

$$\left\| \frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n,\boldsymbol{x}_k) \right) \right\|^2$$
$$< 64\ln^2(2/\delta) = \tilde{\mathcal{O}}(1). \tag{A154}$$

(d)

$$\left\| \frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) \right.$$
$$\left. \times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t \right\|^2$$
$$< \eta^2(1-\gamma^2)^2 \ln^2(2m/\delta) = \tilde{\mathcal{O}}(1). \tag{A155}$$

(e)

$$\left\| \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f) \right\| < \tilde{\mathcal{O}}\left(\sqrt{d}\int_0^{T_f} \bar{\ell}'_{f,k,t}\,\mathrm{d}t \right). \tag{A156}$$

(f) Let

$$\boldsymbol{r}'_n := \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\boldsymbol{v}_i(0) + \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\boldsymbol{v}_i(0)$$
$$+ \frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n,\boldsymbol{x}_k) \right)$$
$$+ \frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)$$
$$\times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t. \tag{A157}$$

Then, $\|\boldsymbol{r}'_n\| < \tilde{\mathcal{O}}(1)$.

Proof.

(a) The given left term can be rearranged as

$$\left\| \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\boldsymbol{v}_i(0) \right\|^2 = \sum_j^d \left(\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)v_{i,j}(0) \right)^2. \tag{A158}$$

By Assumption D.1,

$$\left(\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)v_{i,j}(0)\right)^2 < (2/m)\ln(2/\delta)\sum_i^m \psi'_{f,i,0}(\boldsymbol{x}_n)^2 v_{i,j}(0)^2 \tag{A159}$$

$$< (4/d)\ln(2dm/\delta)\ln(2/\delta). \tag{A160}$$

(b) The given left term can be rearranged as

$$\left\|\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\boldsymbol{v}_i(0)\right\|^2$$

$$= \sum_j^d \left(\sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))v_{i,j}(0)\right)^2 \tag{A161}$$

$$\leq (1-\gamma)^2 \sum_j^d \left(\sum_i^m |\alpha_i \kappa_{f,i}(\boldsymbol{x}_n)v_{i,j}(0)|\right)^2. \tag{A162}$$

By Assumption D.1,

$$\left(\sum_i^m |\alpha_i \kappa_{f,i}(\boldsymbol{x}_n)v_{i,j}(0)|\right)^2 < (4\eta^2/d)\ln(2m/\delta)\ln(2dm/\delta). \tag{A163}$$

(c) By Assumption D.1,

$$\left\|\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n,\boldsymbol{x}_k)\right)\right\|^2$$

$$< \frac{256}{mN^2}\ln^2\left(\frac{2}{\delta}\right)\sum_{n,k}^N |\langle \boldsymbol{x}_n,\boldsymbol{x}_k\rangle| \int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t. \tag{A164}$$

By Assumption D.5,

$$\frac{256}{N^2}\ln^2\left(\frac{2}{\delta}\right)\sum_{n,k}^N |\langle \boldsymbol{x}_n,\boldsymbol{x}_k\rangle| \int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t$$

$$\times \frac{N^2}{4\sum_{n,k}^N |\langle \boldsymbol{x}_n,\boldsymbol{x}_k\rangle| \int_0^{T_f} \ell'_{f,n,t}\,\mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t}$$

$$< 64\ln^2(2/\delta). \tag{A165}$$

(d) The given left term can be rearranged as

$$\left|\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t\right|$$

$$\leq (1-\gamma^2)\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t. \tag{A166}$$

By Assumption D.1,

$$\sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) < \frac{2\eta}{\sqrt{m}}\ln\left(\frac{2m}{\delta}\right). \tag{A167}$$

Thus,

$$\left\|\frac{1}{N}\sum_k^N y_k \boldsymbol{x}_k \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t\right\|^2$$

$$< \frac{4\eta^2(1-\gamma^2)^2}{mN^2} \ln^2\left(\frac{2m}{\delta}\right) \sum_{n,k}^N |\langle \boldsymbol{x}_n, \boldsymbol{x}_k \rangle| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\, \mathrm{d}t\,. \qquad (A168)$$

By Assumption D.5,

$$\frac{4\eta^2(1-\gamma^2)^2}{mN^2} \ln^2\left(\frac{2m}{\delta}\right) \sum_{n,k}^N |\langle \boldsymbol{x}_n, \boldsymbol{x}_k \rangle| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\, \mathrm{d}t$$

$$< \frac{4\eta^2(1-\gamma^2)^2}{N^2} \ln^2\left(\frac{2m}{\delta}\right) \sum_{n,k}^N |\langle \boldsymbol{x}_n, \boldsymbol{x}_k \rangle| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\, \mathrm{d}t$$

$$\times \frac{N^2}{4 \sum_{n,k}^N |\langle \boldsymbol{x}_n, \boldsymbol{x}_k \rangle| \int_0^{T_f} \ell'_{f,n,t}\, \mathrm{d}t \int_0^{T_f} \ell'_{f,k,t}\, \mathrm{d}t} \qquad (A169)$$

$$= \eta^2(1-\gamma^2)^2 \ln^2(2m/\delta). \qquad (A170)$$

(e) As $\|\boldsymbol{s}_1 + \boldsymbol{s}_2 + \boldsymbol{s}_3 + \boldsymbol{s}_4 + s_5\|^2 \le 25 \max(\|\boldsymbol{s}_1\|^2, \|\boldsymbol{s}_2\|^2, \|\boldsymbol{s}_3\|^2, \|\boldsymbol{s}_4\|^2, \|\boldsymbol{s}_5\|^2)$ for any $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4, \boldsymbol{s}_5 \in \mathbb{R}^d$,

$$\left\| \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n) \boldsymbol{v}_i(T_f) \right\|^2 < 25 \max \begin{pmatrix} 4\ln(2dm/\delta)\ln(2/\delta), \\ 4\eta^2(1-\gamma)^2 \ln(2m/\delta)\ln(2dm/\delta), \\ 64\ln^2(2/\delta), \\ \eta^2(1-\gamma^2)^2 \ln^2(2m/\delta), \\ \left\| \frac{1}{N} \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\boldsymbol{x}_k \int_0^{T_f} \ell'_{f,k,t}\, \mathrm{d}t \right\|^2 \end{pmatrix}. \qquad (A171)$$

(f) Similarly to (e),

$$\|\boldsymbol{r}'_n\|^2 < 16 \max \begin{pmatrix} 4\ln(2dm/\delta)\ln(2/\delta), \\ 4\eta^2(1-\gamma)^2 \ln(2m/\delta)\ln(2dm/\delta), \\ 64\ln^2(2/\delta), \\ \eta^2(1-\gamma^2)^2 \ln^2(2m/\delta) \end{pmatrix}. \qquad (A172)$$

$\square$

Lemma D.14 (Upper bounds of inner products with adversarial perturbation)**.** *Assume Assumptions D.1 and D.5.*

(a)

$$\left| \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle \right| < 2\sqrt{(1/d)\ln(2/\delta)\ln(2m/\delta)}\|\boldsymbol{z}\| = \tilde{\mathcal{O}}(1). \qquad (A173)$$

(b)

$$\left| \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle \right|$$

$$< 2\eta(1-\gamma)\ln(2m/\delta)\|\boldsymbol{z}\|/\sqrt{d} = \tilde{\mathcal{O}}(1). \qquad (A174)$$

(c)

$$\left| \frac{1}{N} \sum_k^N y_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \int_0^{T_f} \ell'_{f,k,t}\, \mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \right) \right|$$

$$< \frac{8C_{\mathrm{thr}}(\boldsymbol{z}, \delta)\ln(2/\delta)}{\sqrt{\ln(2m/\delta)}} = \tilde{\mathcal{O}}(1). \qquad (A175)$$

(d)

$$\left| \frac{1}{N} \sum_k^N y_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) \right.$$

$$\left. \times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k)) \, \mathrm{d}t \right|$$

$$< \eta(1-\gamma^2) C_{\mathrm{thr}}(\boldsymbol{z},\delta)\sqrt{\ln(2m/\delta)} = \tilde{\mathcal{O}}(1). \tag{A176}$$

Proof.

(a) By Assumption D.1,

$$\left| \sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle \right| < \sqrt{\frac{2}{m}\ln\left(\frac{2}{\delta}\right)\sum_i^m \psi'_{f,i,0}(\boldsymbol{x}_n)^2 \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle^2} \tag{A177}$$

$$\leq \sqrt{\frac{2}{m}\ln\left(\frac{2}{\delta}\right)\sum_i^m \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle^2} \tag{A178}$$

$$< 2\sqrt{(1/d)\ln(2/\delta)\ln(2m/\delta)}\|\boldsymbol{z}\|. \tag{A179}$$

(b) By Assumption D.1,

$$\left| \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle \right|$$

$$\leq (1-\gamma)\sum_i^m |\alpha_i \kappa_{f,i}(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle| \tag{A180}$$

$$< 2\eta(1-\gamma)\ln(2m/\delta)\|\boldsymbol{z}\|/\sqrt{d}. \tag{A181}$$

(c) and (d) Similarly to Lemma D.10.

$\square$

Lemma D.15 (Representation of network g)**.** *Suppose that Assumptions D.1, D.3, D.5 and D.7.*

(a) In Scenario (a), i.e., $\boldsymbol{x}_n^{\mathrm{adv}} := \boldsymbol{r}_n$,

$$g(\boldsymbol{z}; T_g)$$

$$= g(\boldsymbol{z}; 0) + \sum_i^m \beta_i \kappa_{i,g}(\boldsymbol{z})(\psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{z}))h_{g,i,0}(\boldsymbol{z})$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{r}_n, \boldsymbol{z} \rangle + 1)\int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t$$

$$\times \left(\sum_i^m \beta_i^2 \psi'_{g,i,0}(\boldsymbol{r}_n)\psi'_{g,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \right)$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{r}_n, \boldsymbol{z})\int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{r}_n, \boldsymbol{z} \rangle + 1)\sum_i^m \beta_i^2 \kappa_{i,g}(\boldsymbol{r}_n)$$

$$\times \int_0^{T_g} \ell'_{g,n,t}(\psi'_{g,i,t}(\boldsymbol{r}_n)\psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{r}_n)\psi'_{g,i,0}(\boldsymbol{z})) \, \mathrm{d}t$$

$$+ \frac{\epsilon}{N \| \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n) \boldsymbol{v}_i(T_f) \|} \left[\sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t \left\{ \vphantom{\sum_i^m} \right. \right.$$

$$\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n) \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle$$

$$+ \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle$$

$$+ \frac{1}{N} \sum_k^N y_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \int_0^{T_f} \ell'_{f,k,t} \, \mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \right)$$

$$+ \frac{1}{N} \sum_k^N y_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)$$

$$\times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n) \psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{x}_k)) \, \mathrm{d}t \left. \vphantom{\sum_i^m} \right\}$$

$$+ \frac{1}{N} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \int_0^{T_f} \ell'_{f,k,t} \, \mathrm{d}t \left. \vphantom{\sum_n^N} \right]. \quad \text{(A182)}$$

(b) In Scenario (b), i.e., $\boldsymbol{x}_n^{\mathrm{adv}} := \boldsymbol{x}_n + \boldsymbol{r}_n$,

$$g(\boldsymbol{z}; T_g)$$

$$= g(\boldsymbol{z}; 0) + \sum_i^m \beta_i \kappa_{i,g}(\boldsymbol{z})(\psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{z})) h_{g,i,0}(\boldsymbol{z})$$

$$+ \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z} \rangle + 1) \int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t$$

$$\times \left(\sum_i^m \beta_i^2 \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}}) \psi'_{g,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \right)$$

$$+ \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}} \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})(\langle \boldsymbol{x}_n, \boldsymbol{z} \rangle + 1) \int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t$$

$$+ \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z} \rangle + 1) \sum_i^m \beta_i^2 \kappa_{i,g}(\boldsymbol{x}_n^{\mathrm{adv}})$$

$$\times \int_0^{T_g} \ell'_{g,n,t}(\psi'_{g,i,t}(\boldsymbol{x}_n^{\mathrm{adv}}) \psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}}) \psi'_{g,i,0}(\boldsymbol{z})) \, \mathrm{d}t$$

$$+ \frac{\epsilon}{N \| \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n) \boldsymbol{v}_i(T_f) \|} \left[\sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t} \, \mathrm{d}t \left\{ \vphantom{\sum_i^m} \right. \right.$$

$$\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n) \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle$$

$$+ \sum_i^m \alpha_i \kappa_{i,g}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n)) \langle \boldsymbol{v}_i(0), \boldsymbol{z} \rangle$$

$$+ \frac{1}{N} \sum_k^N y_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \int_0^{T_f} \ell'_{f,k,t} \, \mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n) \psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \right)$$

$$+ \frac{1}{N} \sum_k^N y_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle \sum_i^m \alpha_i^2 \kappa_{i,g}(\boldsymbol{x}_k)$$

$$\times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t \Bigg\}$$

$$+ \frac{1}{N}\sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t$$

$$\times \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \Bigg]. \tag{A183}$$

Proof. Similarly to Lemma D.9,

$$g(\boldsymbol{z}; T_g) = g(\boldsymbol{z}; 0) + \sum_i^m \beta_i \kappa_{i,g}(\boldsymbol{z})(\psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{z}))h_{g,i,0}(\boldsymbol{z})$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}\rangle + 1)\int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t$$

$$\times \left(\sum_i^m \beta_i^2 \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}})\psi'_{g,i,0}(\boldsymbol{z}) - \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\right)$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}\rangle \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t$$

$$+ \frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}\rangle + 1)\sum_i^m \beta_i^2 \kappa_{i,g}(\boldsymbol{x}_n^{\mathrm{adv}})$$

$$\times \int_0^{T_g} \ell'_{g,n,t}(\psi'_{g,i,t}(\boldsymbol{x}_n^{\mathrm{adv}})\psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{x}_n^{\mathrm{adv}})\psi'_{g,i,0}(\boldsymbol{z}))\,\mathrm{d}t. \tag{A184}$$

By Lemma D.12,

$$\frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\langle \boldsymbol{r}_n, \boldsymbol{z}\rangle \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t$$

$$= \frac{\epsilon \sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(T_f), \boldsymbol{z}\rangle}{N\|\sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\boldsymbol{v}_i(T_f)\|}. \tag{A185}$$

The numerator can be also rearranged as

$$\sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(T_f), \boldsymbol{z}\rangle$$

$$= \sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})\int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \left(\sum_i^m \alpha_i \psi'_{f,i,0}(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(0), \boldsymbol{z}\rangle\right.$$

$$+ \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi'_{f,i,T_f}(\boldsymbol{x}_n) - \psi'_{f,i,0}(\boldsymbol{x}_n))\langle \boldsymbol{v}_i(0), \boldsymbol{z}\rangle$$

$$+ \frac{1}{N}\sum_k^N y_k\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\right)$$

$$+ \frac{1}{N}\sum_k^N y_k\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k)$$

$$\times \int_0^{T_f} \ell'_{f,k,t}(\psi'_{f,i,T_f}(\boldsymbol{x}_n)\psi'_{f,i,t}(\boldsymbol{x}_k) - \psi'_{f,i,0}(\boldsymbol{x}_n)\psi'_{f,i,0}(\boldsymbol{x}_k))\,\mathrm{d}t \Bigg)$$

$$+ \frac{1}{N} \sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t. \tag{A186}$$

$\square$

Lemma D.16 (Network prediction of g). *Suppose that Assumptions D.1, D.3, D.5 and D.7.*

(a) In Scenario (a), if

$$\left| \frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \right|$$

$$> \tilde{\mathcal{O}}\left(\frac{\sqrt{d} \int_0^{T_f} \bar{\ell}'_{f,k,t}\,\mathrm{d}t}{\epsilon} \left(1 + \left| \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}} \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \right| \right) \right.$$

$$\left. + \int_0^{T_g} \bar{\ell}'_{g,n,t}\,\mathrm{d}t \right), \tag{A187}$$

then

$$\mathrm{sgn}(g(\boldsymbol{z}; T_g)) = \mathrm{sgn}\left(\frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \right.$$

$$\left. \times \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \right). \tag{A188}$$

(b) In Scenario (b), if

$$\left| \frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \right|$$

$$> \tilde{\mathcal{O}}\left(\frac{\sqrt{d} \int_0^{T_f} \bar{\ell}'_{f,k,t}\,\mathrm{d}t}{\epsilon} \left(1 + \left| \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}} \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z})(\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle + 1) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \right| \right) \right.$$

$$\left. + \int_0^{T_g} \bar{\ell}'_{g,n,t}\,\mathrm{d}t \right), \tag{A189}$$

then

$$\mathrm{sgn}(g(\boldsymbol{z}; T_g)) = \mathrm{sgn}\left(\frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{x}_n^{\mathrm{adv}}, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \right.$$

$$\left. \times \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \right). \tag{A190}$$

Proof. We prove (a). Similarly, (b) can be established. By Lemmas D.10 and D.13 to D.15, if

$$\left| \frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell'_{g,n,t}\,\mathrm{d}t \sum_k^N y_k \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell'_{f,k,t}\,\mathrm{d}t \right|$$

$$> \frac{\| \sum_i^m \alpha_i \psi'_{f,i,T_f}(\boldsymbol{x}_n) \boldsymbol{v}_i(T_f) \|}{\epsilon} \left\{ |g(\boldsymbol{z}; 0)| + \left| \sum_i^m \beta_i \kappa_{i,g}(\boldsymbol{z})(\psi'_{g,i,T_g}(\boldsymbol{z}) - \psi'_{g,i,0}(\boldsymbol{z})) h_{g,i,0}(\boldsymbol{z}) \right| \right.$$

$$+ \left| \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{r}_n, \boldsymbol{z}\rangle + 1) \int_0^{T_g} \ell_{g,n,t}' \, \mathrm{d}t \left(\sum_i^m \beta_i^2 \psi_{g,i,0}'(\boldsymbol{r}_n)\psi_{g,i,0}'(\boldsymbol{z}) - \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \right) \right|$$

$$+ \left| \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell_{g,n,t}' \, \mathrm{d}t \right|$$

$$+ \left| \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}}(\langle \boldsymbol{r}_n, \boldsymbol{z}\rangle + 1) \sum_i^m \beta_i^2 \kappa_{i,g}(\boldsymbol{r}_n) \right.$$

$$\left. \left. \times \int_0^{T_g} \ell_{g,n,t}'(\psi_{g,i,t}'(\boldsymbol{r}_n)\psi_{g,i,T_g}'(\boldsymbol{z}) - \psi_{g,i,0}'(\boldsymbol{r}_n)\psi_{g,i,0}'(\boldsymbol{z})) \, \mathrm{d}t \right| \right\}$$

$$+ \frac{1}{N} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell_{g,n,t}' \, \mathrm{d}t \left\{ \left| \sum_i^m \alpha_i \psi_{f,i,0}'(\boldsymbol{x}_n)\langle \boldsymbol{v}_i(0), \boldsymbol{z}\rangle \right| \right.$$

$$+ \left| \sum_i^m \alpha_i \kappa_{f,i}(\boldsymbol{x}_n)(\psi_{f,i,T_f}'(\boldsymbol{x}_n) - \psi_{f,i,0}'(\boldsymbol{x}_n))\langle \boldsymbol{v}_i(0), \boldsymbol{z}\rangle \right|$$

$$+ \left| \frac{1}{N} \sum_k^N y_k\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell_{f,k,t}' \, \mathrm{d}t \left(\sum_i^m \alpha_i^2 \psi_{f,i,0}'(\boldsymbol{x}_n)\psi_{f,i,0}'(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_n, \boldsymbol{x}_k) \right) \right|$$

$$+ \left| \frac{1}{N} \sum_k^N y_k\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \sum_i^m \alpha_i^2 \kappa_{f,i}(\boldsymbol{x}_k) \right.$$

$$\left. \left. \times \int_0^{T_f} \ell_{f,k,t}'(\psi_{f,i,T_f}'(\boldsymbol{x}_n)\psi_{f,i,t}'(\boldsymbol{x}_k) - \psi_{f,i,0}'(\boldsymbol{x}_n)\psi_{f,i,0}'(\boldsymbol{x}_k)) \, \mathrm{d}t \right| \right\} \tag{A191}$$

$$= \frac{\tilde{\mathcal{O}}\left(\sqrt{d} \int_0^{T_f} \bar{\ell}_{f,k,t}' \, \mathrm{d}t \right)}{\epsilon} \left(\tilde{\mathcal{O}}(1) + \left| \frac{1}{N} \sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell_{g,n,t}' \, \mathrm{d}t \right| \right)$$

$$+ \tilde{\mathcal{O}}\left(\int_0^{T_g} \bar{\ell}_{g,n,t}' \, \mathrm{d}t \right), \tag{A192}$$

then

$$\mathrm{sgn}(g(\boldsymbol{z}; T_g))$$

$$= \mathrm{sgn}\left(\frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell_{g,n,t}' \, \mathrm{d}t \sum_k^N y_k\Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell_{f,k,t}' \, \mathrm{d}t \right). \tag{A193}$$

$\square$

Theorem D.17 (Perturbation learning, Scenario (a), general case)**.** *Consider Scenario (a) in Setting 3.1. Let $\delta = \Theta(1)$ be a small positive number and*

$$\hat{f}^{\mathrm{gen}}(\boldsymbol{z}) := \frac{1}{N} \sum_n^N y_n\Phi(\boldsymbol{x}_n, \boldsymbol{z})\langle \boldsymbol{x}_n, \boldsymbol{z}\rangle \int_0^{T_f} \ell_{f,n,t}' \, \mathrm{d}t \,, \tag{A194}$$

$$\hat{g}_a^{\mathrm{gen}}(\boldsymbol{z}) := \frac{1}{N^2} \sum_n^N \Phi(\boldsymbol{r}_n, \boldsymbol{z}) \int_0^{T_g} \ell_{g,n,t}' \, \mathrm{d}t \sum_k^N y_k\Phi(\boldsymbol{x}_n, \boldsymbol{x}_k)\langle \boldsymbol{x}_k, \boldsymbol{z}\rangle \int_0^{T_f} \ell_{f,k,t}' \, \mathrm{d}t \,. \tag{A195}$$

Under Assumption 3.2, for any $\boldsymbol{z} \in \mathbb{R}^d$, if

$$|\hat{f}^{\mathrm{gen}}(\boldsymbol{z})| > \tilde{\mathcal{O}}\left(1 + \int_0^{T_f} \bar{\ell}_{f,n,t} \, \mathrm{d}t \right), \tag{A196}$$

$$|\hat{g}_a^{\mathrm{gen}}(\boldsymbol{z})|$$

$$> \tilde{\mathcal{O}}\left(\frac{\sqrt{d}\int_0^{T_f}\bar{\ell}'_{f,k,t}\,\mathrm{d}t}{\epsilon}\left(1+\left|\frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{r}_n,\boldsymbol{z})\int_0^{T_g}\ell'_{g,n,t}\,\mathrm{d}t\right|\right)+\int_0^{T_g}\bar{\ell}'_{g,n,t}\,\mathrm{d}t\right),\quad \text{(A197)}$$

$$\mathrm{sgn}(\hat{f}^{\mathrm{gen}}(\boldsymbol{z}))=\mathrm{sgn}(\hat{g}_a^{\mathrm{gen}}(\boldsymbol{z})), \tag{A198}$$

then, with probability at least $1-\delta$, $\mathrm{sgn}(f(\boldsymbol{z};T_f))=\mathrm{sgn}(g(\boldsymbol{z};T_g))$ holds.

Proof. By Bonferroni's inequality and Assumptions D.5 and D.7 and Lemmas D.2, D.4, D.11 and D.16, the claim is established. □

Theorem D.18 (Perturbation learning, Scenario (b), general case). *Consider Scenario (b) in Setting 3.1. Let $\delta=\Theta(1)$ be a small positive number and*

$$\hat{f}^{\mathrm{gen}}(\boldsymbol{z}):=\frac{1}{N}\sum_n^N y_n\Phi(\boldsymbol{x}_n,\boldsymbol{z})\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\,, \tag{A199}$$

$$\hat{g}_b^{\mathrm{gen}}(\boldsymbol{z}):=\frac{1}{N^2}\sum_n^N\Phi(\boldsymbol{x}_n^{\mathrm{adv}},\boldsymbol{z})\int_0^{T_g}\ell'_{g,n,t}\,\mathrm{d}t\sum_k^N y_k\Phi(\boldsymbol{x}_n,\boldsymbol{x}_k)\langle\boldsymbol{x}_k,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,k,t}\,\mathrm{d}t\,. \tag{A200}$$

Under Assumption 3.2, for any $\boldsymbol{z}\in\mathbb{R}^d$, if

$$|\hat{f}^{\mathrm{gen}}(\boldsymbol{z})|>\tilde{\mathcal{O}}\left(1+\int_0^{T_f}\bar{\ell}_{f,n,t}\,\mathrm{d}t\right), \tag{A201}$$

$$|\hat{g}_b^{\mathrm{gen}}(\boldsymbol{z})|$$
$$>\tilde{\mathcal{O}}\left(\frac{\sqrt{d}\int_0^{T_f}\bar{\ell}'_{f,k,t}\,\mathrm{d}t}{\epsilon}\left(1+\left|\frac{1}{N}\sum_n^N y_n^{\mathrm{adv}}\Phi(\boldsymbol{x}_n^{\mathrm{adv}},\boldsymbol{z})(\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle+1)\int_0^{T_g}\ell'_{g,n,t}\,\mathrm{d}t\right|\right)\right.$$
$$\left.+\int_0^{T_g}\bar{\ell}'_{g,n,t}\,\mathrm{d}t\right), \tag{A202}$$

$$\mathrm{sgn}(\hat{f}^{\mathrm{gen}}(\boldsymbol{z}))=\mathrm{sgn}(\hat{g}_b^{\mathrm{gen}}(\boldsymbol{z})), \tag{A203}$$

then, with probability at least $1-\delta$, $\mathrm{sgn}(f(\boldsymbol{z};T_f))=\mathrm{sgn}(g(\boldsymbol{z};T_g))$ holds.

Proof. By Bonferroni's inequality and Assumptions D.5 and D.7 and Lemmas D.2, D.4, D.11 and D.16, the claim is established. □

Theorem 3.3 (Direction of adversarial perturbation). *Let $\delta=\Theta(1)$ be a small positive number. Under Assumption 3.2, for any $n\in[N]$, with probability at least $1-\delta$, the adversarial perturbation $\boldsymbol{r}_n$ is parallel to the weighted sum of training samples as follows:*

$$\boldsymbol{r}_n \mathbin{/\mkern-5mu/} \frac{1}{N}\sum_{k=1}^N y_k\Phi(\boldsymbol{x}_n,\boldsymbol{x}_k)\boldsymbol{x}_k\int_0^{T_f}\ell'(-y_kf(\boldsymbol{x}_k;t))\,\mathrm{d}t+\boldsymbol{\xi}_n, \tag{6}$$

where $\boldsymbol{\xi}_n$ satisfies $\|\boldsymbol{\xi}_n\|=\tilde{\mathcal{O}}(1)$. In particular, for $\ell(s)=s$,

$$\boldsymbol{r}_n \mathbin{/\mkern-5mu/} \frac{T_f}{N}\sum_k^N y_k\Phi(\boldsymbol{x}_n,\boldsymbol{x}_k)\boldsymbol{x}_k+\boldsymbol{\xi}_n. \tag{7}$$

Proof. By Bonferroni's inequality and Assumptions D.5 and D.7 and Lemmas D.2, D.4, D.12 and D.13, the claim is established. □

Theorem 3.4 (Perturbation learning, Scenario (a), special case of Theorem D.17). *Consider Scenario (a) in Setting 3.1. Assume $\ell(s) = s$ and $y_n^{\mathrm{adv}} \sim U(\{\pm 1\})$ for every $n \in [N]$. Let $\delta = \Theta(1)$ be a small positive number and*

$$\hat{f}(z) := \frac{1}{N} \sum_{n=1}^N y_n \Phi(x_n, z) \langle x_n, z \rangle, \quad \hat{g}_a(z) := \frac{1}{N^2} \sum_{n=1}^N \Phi(r_n, z) \sum_{k=1}^N y_k \Phi(x_n, x_k) \langle x_k, z \rangle. \quad (8)$$

Under Assumption 3.2, for any $z \in \mathbb{R}^d$, if

(Functional margin condition 1) $\qquad |\hat{f}(z)| > \tilde{\mathcal{O}}\left(1 + \frac{1}{T_f}\right),$ $\qquad\qquad$ (9)

(Functional margin condition 2) $\qquad |\hat{g}_a(z)| > \tilde{\mathcal{O}}\left(\frac{1}{T_f} + \frac{\sqrt{d}}{\epsilon}\left(\frac{1}{T_g} + \frac{1}{\sqrt{N}}\right)\right),$ $\qquad$ (10)

(Agreement condition) $\qquad \mathrm{sgn}(\hat{f}(z)) = \mathrm{sgn}(\hat{g}_a(z)),$ $\qquad\qquad$ (11)

then, with probability at least $1 - \delta$, $\mathrm{sgn}(f(z; T_f)) = \mathrm{sgn}(g(z; T_g))$ holds.

Proof. By Bonferroni's inequality and Lemma C.2 and Theorem D.17, the claim is established. $\qquad\square$

Theorem 3.5 (Perturbation learning, Scenario (b), special case of Theorem D.18). *Consider Scenario (b) in Setting 3.1. Assume $\ell(s) = s$ and $y_n^{\mathrm{adv}} \sim U(\{\pm 1\})$ for every $n \in [N]$. Let $\delta = \Theta(1)$ be a small positive number and*

$$\hat{g}_b(z) := \frac{1}{N^2} \sum_{n=1}^N \Phi(x_n^{\mathrm{adv}}, z) \sum_{k=1}^N y_k \Phi(x_n, x_k) \langle x_k, z \rangle. \quad (14)$$

Under Assumption 3.2, for any $z \in \mathbb{R}^d$, if the functional margin condition 1 (Ineq. (9)),

(Func. margin cond. 2) $\quad |\hat{g}_b(z)| > \tilde{\mathcal{O}}\left(\frac{1}{T_f} + \frac{\sqrt{d}}{\epsilon}\left(\frac{1}{T_g} + \frac{\sqrt{\sum_n^N(\langle x_n, z\rangle + 1)^2}}{N}\right)\right),$ $\quad$ (15)

(Agreement condition) $\quad \mathrm{sgn}(\hat{f}(z)) = \mathrm{sgn}(\hat{g}_b(z)),$ $\qquad\qquad$ (16)

then, with probability at least $1 - \delta$, $\mathrm{sgn}(f(z; T_f)) = \mathrm{sgn}(g(z; T_g))$ holds.

Proof. By Bonferroni's inequality and Lemma C.2 and Theorem D.18, the claim is established. $\qquad\square$

Lemma D.19 (Sufficient condition of agreement condition). *If*

$$\frac{|\sum_n^N y_n \langle x_n, z \rangle \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t|}{\max_{x \in \{x_1, \ldots, x_N, z\}} \sum_n^N \lambda(x_n, x) |\langle x_n, z \rangle| \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t} > \frac{1 - \gamma}{1 + \gamma}, \quad (A204)$$

then $\mathrm{sgn}(\hat{f}^{\mathrm{gen}}(z)) = \mathrm{sgn}(\hat{g}_a^{\mathrm{gen}}(z)) = \mathrm{sgn}(\hat{g}_b^{\mathrm{gen}}(z))$ holds.

Proof. By Lemma C.4,

$$\left| \sum_n^N y_n \langle x_n, z \rangle \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t \left(\Phi(x_n, z) - \frac{(1 + \gamma)^2}{4} \right) \right|$$

$$\leq \frac{(1 + \gamma)(1 - \gamma)}{4} \sum_n^N \lambda(x_n, z) |\langle x_n, z \rangle| \int_0^{T_f} \ell'_{f,n,t} \, \mathrm{d}t. \quad (A205)$$

In addition,

$$\frac{(1+\gamma)^2}{4}\left|\sum_n^N y_n\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right|$$

$$> \frac{(1+\gamma)(1-\gamma)}{4}\sum_n^N \lambda(\boldsymbol{x}_n,\boldsymbol{z})|\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle|\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t \tag{A206}$$

$$\Longleftarrow \frac{\left|\sum_n^N y_n\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right|}{\sum_n^N \lambda(\boldsymbol{x}_n,\boldsymbol{z})|\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle|\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t} > \frac{1-\gamma}{1+\gamma}. \tag{A207}$$

Thus, if Ineq. (A207) holds, then

$$\operatorname{sgn}\left(\sum_n^N y_n\Phi(\boldsymbol{x}_n,\boldsymbol{z})\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right) = \operatorname{sgn}\left(\sum_n^N y_n\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right). \tag{A208}$$

Similarly, for any $k\in[N]$, if

$$\frac{\left|\sum_n^N y_n\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right|}{\sum_n^N \lambda(\boldsymbol{x}_n,\boldsymbol{x}_k)|\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle|\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t} > \frac{1-\gamma}{1+\gamma}, \tag{A209}$$

then

$$\operatorname{sgn}\left(\sum_n^N y_n\Phi(\boldsymbol{x}_n,\boldsymbol{x}_k)\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right) = \operatorname{sgn}\left(\sum_n^N y_n\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right). \tag{A210}$$

When Ineq. (A209) holds for every $k\in[N]$,

$$\operatorname{sgn}\left(\sum_n^N \Phi(\boldsymbol{r}_n,\boldsymbol{z})\int_0^{T_g}\ell'_{g,n,t}\,\mathrm{d}t\sum_l^N y_l\Phi(\boldsymbol{x}_n,\boldsymbol{x}_l)\langle\boldsymbol{x}_l,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,l,t}\,\mathrm{d}t\right)$$

$$= \operatorname{sgn}\left(\sum_l^N y_l\Phi(\boldsymbol{x}_n,\boldsymbol{x}_l)\langle\boldsymbol{x}_l,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,l,t}\,\mathrm{d}t\right) \tag{A211}$$

$$= \operatorname{sgn}\left(\sum_n^N y_n\langle\boldsymbol{x}_n,\boldsymbol{z}\rangle\int_0^{T_f}\ell'_{f,n,t}\,\mathrm{d}t\right). \tag{A212}$$

By integrating Ineqs. (A207) and (A209), the claim is established.

$\square$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have accurately described the contributions and limitations in Abstract and Introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are described in Section 3.4.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: The assumption is described in Assumption 3.2 and a brief proof is provided in Section 3.3. The complete proof can be found in Appendix D.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: The experimental setup is described in detail in Appendix B.

Guidelines:

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

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

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

Answer: [Yes]

Justification: All datasets we used are either openly accessible or can be artificially generated. The code is provided as supplementary material.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: The experimental setup is described in detail in Appendix B.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: Error bars were not measured due to computational costs. However, we provide extensive experimental results to support our theoretical findings in Appendix B.

   Guidelines:

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

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

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: Our experiments were conducted on an NVIDIA A100.

Guidelines:

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

9. **Code Of Ethics**

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

Answer: [Yes]

Justification: Our paper strictly adheres to the NeurIPS Code of Ethics.

Guidelines:

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

10. **Broader Impacts**

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

Answer: [NA]

Justification: This does not apply as it is a theoretical study.

Guidelines:

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

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

11. **Safeguards**

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

Answer: [NA]

Justification: Our theoretical research does not involve such releases.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We have accurately cited credits.

Guidelines:

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

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

13. **New Assets**

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

    Answer: [NA]

    Justification: We do not provide such assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: We did not conduct such experiments.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: We did not conduct experiments that require this.

    Guidelines:

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