LINEAR BACKPROPAGATION LEADS TO FASTER CONVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Backpropagation is widely used for calculating gradients in deep neural networks (DNNs). Applied often along with stochastic gradient descent (SGD) or its variants, backpropagation is considered as a de-facto choice in a variety of machine learning tasks including DNN training and adversarial attack/defense. Nevertheless, unlike SGD which has been intensively studied over the past years, backpropagation is somehow overlooked. In this paper, we study the very recent method called "linear backpropagation" (LinBP), which modifies the standard backpropagation and can improve the transferability in black-box adversarial attack. By providing theoretical analyses on LinBP in neural-network-involved learning tasks including white-box adversarial attack and model training, we will demonstrate that, somewhat surprisingly, LinBP can lead to faster convergence in these tasks. We will also confirm our theoretical results with extensive experiments.

1 INTRODUCTION

Over the past decade, the surge of research on deep neural networks (DNNs) has been witnessed. Powered with large-scale training, DNN-based models have achieved state-of-art performance in a variety of applications. Tremendous amount of research has been done to improve the architecture (He et al., 2016; Simonyan & Zisserman, 2015; Huang et al., 2017; Vaswani et al., 2017; Dosovitskiy et al., 2021) and the optimization method (Kingma & Ba, 2014; Loshchilov & Hutter, 2019; 2017; Reddi et al., 2019; Sutskever et al., 2013) for DNNs. The optimization involving DNNs usually utilizes stochastic gradient descent (SGD) (Bottou, 2010) that minimizes some learning loss and uses backpropagation (BP) (LeCun, 1988) for computing gradients. Unlike SGD which has been thoroughly studied and innovated (cf. Adam, Adagrad, SGDW, etc), BP seems overlooked in deep learning over the past decade and normally considered as a de-factor choice in DNN-involved optimizations.

Very recently, Guo et al. (2020) introduced a slightly different way of computing gradients called linear BP (LinBP) in which the forward pass of a DNN was left unchanged, while the partial derivative regarding some of the rectified linear unit (ReLU) activation function was skipped in the backward pass. It was shown empirically that LinBP leads to improved results in generating transferable adversarial examples for performing *black-box adversarial attacks* (Papernot et al., 2017), in comparison to using the original BP. The results were enlightening from our perspective, since the superior performance was surprisingly obtained by performing less precise computation of gradient. The superiority of LinBP was originally conjectured as less overfitting and less gradient obfuscation (Athalye et al., 2018). Yet, in this paper, we would like to study the optimization convergence using LinBP, to shed more light on the method. We target two practical applications that care about convergence, *i.e.*, *white-box adversarial attack* and model parameter training.

In the remainder of this paper, we shall first revisit some preliminary knowledge regarding model training, white-box adversarial attack, and LinBP. We will then introduce a teacher-student framework (Tian, 2017) which uses ReLU activation functions and squared l_2 loss. Under the framework, we give rigorous theoretical convergence analyses on LinBP. Our theoretical results show that, in white-box adversarial attack scenarios, using LinBP can produce more deceptive adversarial examples, in comparison to using the standard BP in the same hyperparameter settings. Similarly, we will show theoretically that LinBP can also help to converge faster than BP in model training. Simulation experiments confirm our theoretical results, and extensive experimental results also verify

our findings in more general and practical settings using a variety of different DNNs, including VGG-16 (Simonyan & Zisserman, 2015), ResNet-50 (He et al., 2016), DenseNet-161 (Huang et al., 2017), and MobileNetV2 (Sandler et al., 2018).

2 Background and Preliminary Knowledge

2.1 Model Training

Together with SGD or its variant, BP has long been adopted as a default method for computing gradients in training machine learning models. Consider a typical update rule of SGD, we have

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} \sum_k \mathcal{L}(\mathbf{x}^{(k)}, y^{(k)}), \tag{1}$$

where $\{\mathbf{x}^{(k)}, y^{(k)}\}$ is a couple of the k-th training instance and its label, $\nabla_{\mathbf{w}_t} \sum_k \mathcal{L}(\mathbf{x}^{(k)}, y^{(k)})$ is the partial gradient of the loss function with respect to a learnable weight vector $\mathbf{w}$, and $\eta \in \mathbb{R}^+$ represents the learning rate.

2.2 White-box Adversarial Attack

Another popular optimization problem in deep learning is to generate adversarial examples, which is of particular interest in both the machine learning community and the security community. Instead of minimizing the prediction loss as in the objective of model training, adversarial attack aims to craft examples that lead to arbitrary incorrect model predictions (Goodfellow et al., 2015). The adversarial examples are expected to be perceptually indistinguishable to benign ones. In the white-box setting, where it is assumed that the architecture and parameters of the victim model are both known to the adversary, we have the typical learning objectives:

$$\max_{\|\mathbf{r}\|_p \leq \epsilon} \mathcal{L}(\mathbf{x}+\mathbf{r}, \mathbf{y}) \quad \text{and} \quad \min \|\mathbf{r}\|_p, \ \text{s.t.}, \ \arg\max_j \text{prob}_j(\mathbf{x}+\mathbf{r}) \neq \arg\max_j \text{prob}_j(\mathbf{x}) \tag{2}$$

where $\mathbf{r}$ is a perturbation vector whose l_p norm is constrained to guarantee that the generated example $\mathbf{x}+\mathbf{r}$ is perceptually similar to $\mathbf{x}$, function $\text{prob}_j(\cdot)$ provides the prediction probability for the j-th class. For solving the problems in Eq. (2), a series of methods have been proposed. For instance, with $p = \infty$, *i.e.*, for l_∞ attack, Goodfellow et al. (2015) proposed the fast gradient sign method (FGSM) which simply calculates the sign of the input gradient, *i.e.,* $\text{sign}(\nabla_{\mathbf{x}} L(\mathbf{x}))$, and adopted $\epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} L(\mathbf{x}))$ as the perturbation. To enhance the power of the attack, follow-up work proposed iterative FGSM (I-FGSM) (Kurakin et al., 2017) and PGD (Madry et al., 2018) that took multiple steps of update and computed input gradients using BP in each of the steps. The iterative process of these methods is similar to that of model training, except that the update is performed in the input space instead of the parameter space and normally some constrains are required for attacks. Other famous attacks include DeepFool (Moosavi-Dezfooli et al., 2016) and C&W's attack (Carlini & Wagner, 2017), just to name a few.

2.3 Linear Backpropagation

For a d-layer neural network model, the forward pass include the computation of

$$f_d(\mathbf{x}) = \mathbf{W}_d^T \sigma(\mathbf{W}_{d-1}^T \cdots \sigma(\mathbf{W}_1^T \mathbf{x})), \tag{3}$$

where $\sigma(\cdot)$ is the activation function and commonly set as ReLU function, $\mathbf{W}_1, \cdots, \mathbf{W}_d$ are learnable weight matrices in the model. In order to generate a white-box adversarial example on the basis of $\mathbf{x}$ with the obtained $f_d(\mathbf{x})$, we shall further compute the prediction loss $\mathcal{L}(\mathbf{x}, \mathbf{y})$ of $\mathbf{x}$ and then backpropagate the loss to compute the gradient $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y})$, the backward propagation for $\mathbf{x}$ is formulated as

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}) = \mathbf{M}_1 \cdots \mathbf{M}_{d-1} \mathbf{W_d} \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{y})}{\partial f_d(\mathbf{x})}, \tag{4}$$

where $\mathbf{M}_i = \mathbf{W}_i \partial \sigma(f_i(\mathbf{x})) / \partial f_i(\mathbf{x})$. LinBP (Guo et al., 2020) keeps the computation in the forward pass unchanged while removing the influence of the ReLU activation function (*i.e.*, $\partial \sigma(f_i(\mathbf{x})) / \partial f_i(\mathbf{x})$) in the backward pass. That being said, LinBP computes:

$$\tilde{\nabla}_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}) = \mathbf{W}_1 \cdots \mathbf{W}_d \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{y})}{\partial f_d(\mathbf{x})}. \tag{5}$$

Since the partial derivative of the activation function has been removed, it is considered linear in the backward pass and thus called LinBP. In practice, LinBP may only remove some of the nonlinear derivatives, *e.g.,* Guo et al. (2020) only modified those starting from the $(m+1)$-th layer and used the following formulation:

$$\tilde{\nabla}_{\mathbf{x}}\mathcal{L}(\mathbf{x},\mathbf{y}) = \mathbf{M}_1 \cdots \mathbf{M}_m \mathbf{W}_{m+1} \cdots \mathbf{W}_d \frac{\partial \mathcal{L}(\mathbf{x},\mathbf{y})}{\partial f_d(\mathbf{x})}, \tag{6}$$

The method was shown to be effective in improving the transferability of the generated adversarial examples on some source model (Guo et al., 2020).

Although LinBP was only proposed for obtaining adversarial examples, we may also adopt it to compute the gradient with respect to model weights for training. Note that the gradient of loss with respect to $\mathbf{W}_i$ in the standard BP is formulated as

$$\nabla_{\mathbf{W}_i}\mathcal{L}(\mathbf{x},\mathbf{y}) = \sigma(f_{i-1}(\mathbf{x}))\left(\frac{\partial \sigma(f_i(\mathbf{x}))}{\partial f_i(\mathbf{x})}\mathbf{M}_{i+1}\cdots\mathbf{M}_{d-1}\mathbf{W}_d\frac{\partial \mathcal{L}(\mathbf{x},\mathbf{y})}{\partial f_d(\mathbf{x})}\right)^T, \tag{7}$$

where we somewhat abuse the notations and define $\sigma(f_0(\mathbf{x})) = \mathbf{x}$ for simplify. Similar to Eq. (5), a linearized version of the gradient is

$$\tilde{\nabla}_{\mathbf{W}_i}\mathcal{L}(\mathbf{x},\mathbf{y}) = \sigma(f_{i-1}(\mathbf{x}))\left(\mathbf{W}_{i+1}\cdots\mathbf{W}_d\frac{\partial \mathcal{L}(\mathbf{x},\mathbf{y})}{\partial f_d(\mathbf{x})}\right)^T. \tag{8}$$

3 Theoretical Analyses

In this section, we provide convergence analyses of LinBP, and compare it to the standard BP. There have been attempts for studying the training dynamics of neural networks (Du et al., 2019a; Arora et al., 2019; Tian, 2017; Du et al., 2019b). We mainly follow Tian (2017)'s work and consider the teacher-student framework with ReLU activation function and squared l_2 loss. We shall start from the theoretical analyses for white-box adversarial attack and then consider model training.

3.1 Theoretical Analyses for Adversarial Attack

Compare to the two-layer teacher-student frameworks in Tian (2017)'s, we here consider a more general model, which can be formulated as

$$g(\mathbf{W},\mathbf{V},\mathbf{x}) = \mathbf{V}\sigma(\mathbf{W}\mathbf{x}), \tag{9}$$

where $\mathbf{x} \in \mathbb{R}^{d_1}$ is the input data vector, $\mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$ and $\mathbf{V} \in \mathbb{R}^{d_3 \times d_2}$ are weight matrices, $\sigma(\cdot)$ is the ReLU function, *i.e.*, $\sigma(\cdot) = \max(\cdot, 0)$. In the teacher-student frameworks, we assume the teacher networks possesses the optimal adversarial example $\mathbf{x}^\star$, and the student network learn the adversarial example $\mathbf{x}$ from the teacher network, in which the loss function is set as the squared l_2 loss between the output of student and teacher networks, *i.e.*,

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2}\|g(\mathbf{W},\mathbf{V},\mathbf{x}) - g(\mathbf{W},\mathbf{V},\mathbf{x}^\star)\|_2^2. \tag{10}$$

We further define $D(\mathbf{W},\mathbf{x}) := \mathrm{diag}(\mathbf{W}\mathbf{x} > 0)$. From Eq. (9) and Eq. (10), we can easily obtain the analytic expression of the gradient with respect to $\mathbf{x}$ for standard BP:

$$\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}) = \mathbf{W}^T D(\mathbf{W},\mathbf{x})\mathbf{V}^T\mathbf{V}(D(\mathbf{W},\mathbf{x})\mathbf{W}\mathbf{x} - D(\mathbf{W},\mathbf{x}^\star)\mathbf{W}\mathbf{x}^\star). \tag{11}$$

Compared with Eq. (11), the gradient obtained from LinBP removes the derivatives of ReLU function in the backward pass, *i.e.,*

$$\tilde{\nabla}_{\mathbf{x}}\mathcal{L}(\mathbf{x}) = \mathbf{W}^T\mathbf{V}^T\mathbf{V}(D(\mathbf{W},\mathbf{x})\mathbf{W}\mathbf{x} - D(\mathbf{W},\mathbf{x}^\star)\mathbf{W}\mathbf{x}^\star). \tag{12}$$

Inspired by Theorem 1 in Tian (2017)'s work, we introduce the following lemma to give the analytic expression of the gradients in expectations in adversarial settings.

Lemma 1 *Denote $G(\mathbf{e},\mathbf{x}) := \mathbf{W}^T D(\mathbf{W},\mathbf{e})\mathbf{V}^T\mathbf{V}D(\mathbf{W},\mathbf{x})\mathbf{W}\mathbf{x}$, where $\mathbf{e} \in \mathbb{R}^{d_1}$ is a unit vector, $\mathbf{x} \in \mathbb{R}^{d_1}$ is the input data vector, $\mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$ and $\mathbf{V} \in \mathbb{R}^{d_3 \times d_2}$ are weight matrices. If $\mathbf{W}$ and $\mathbf{V}$ follow independent standard Gaussian distribution, we have*

$$\mathbb{E}\left(G(\mathbf{e},\mathbf{x})\right) = \frac{d_2}{2\pi}[(\pi - \Theta)\mathbf{x} + \|\mathbf{x}\|\sin\Theta\mathbf{e}],$$

where $\Theta \in [0,\pi]$ is the angle between $\mathbf{e}$ and $\mathbf{x}$.

The proof can be found in the appendix. With Lemma 1, the expectation of Eq. (11) and Eq. (12) can be formulated as

$$\mathbb{E}[\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x})] = G(\mathbf{x}/\|\mathbf{x}\|, \mathbf{x}) - G(\mathbf{x}/\|\mathbf{x}\|, \mathbf{x}^\star) = \frac{d_2}{2}(\mathbf{x} - \mathbf{x}^\star) + \frac{d_2}{2\pi}\left(\Theta\mathbf{x}^\star - \frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}\|}\sin\Theta\mathbf{x}\right), \quad (13)$$

and

$$\mathbb{E}[\tilde{\nabla}_{\mathbf{x}}\mathcal{L}(\mathbf{x})] = G(\mathbf{x}/\|\mathbf{x}\|, \mathbf{x}) - G(\mathbf{x}^\star/\|\mathbf{x}^\star\|, \mathbf{x}^\star) = \frac{d_2}{2}(\mathbf{x} - \mathbf{x}^\star), \quad (14)$$

respectively, where $\Theta \in [0, \pi]$ is the angle between $\mathbf{x}$ and $\mathbf{x}^\star$. We follow prior works (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018) and focus on l_∞ attacks. Different iterative gradient-based l_∞ attack methods may have slightly different update rules, here, for ease of unified analyses, we simplify their update rules as

$$\mathbf{x}^{(t+1)} = \text{Clip}(\mathbf{x}^{(t)} - \eta\nabla_{\mathbf{x}^{(t)}}\mathcal{L}(\mathbf{x}^{(t)})), \quad (15)$$

where $\text{Clip}(\cdot) = \min(\mathbf{x} + \epsilon\mathbf{1}, \max(\mathbf{x} - \epsilon\mathbf{1}, \cdot))$ performs element-wise input clip to guarantee that the intermediate results always stay in the range fulfilling the constraint of $\|\mathbf{x}^{(t+1)} - \mathbf{x}\|_\infty \leq \epsilon$. We use the updates of standard BP and LinBP, *i.e.*, $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x})$ and $\tilde{\nabla}_{\mathbf{x}}\mathcal{L}(\mathbf{x})$, to obtain $\{\mathbf{x}^{(t)}\}$ and $\{\tilde{\mathbf{x}}^{(t)}\}$, respectively. Under all the settings above, we propose the following theorem to give convergence analysis on LinBP.

Theorem 1 *For the two-layer teacher-student network formulated as Eq. (9), the adversarial attack sets Eq. (10) and Eq. (15) as the loss function and the update rule, respectively. Assume that* $\mathbf{W}$ *and* $\mathbf{V}$ *follow independent standard Gaussian distribution,* $\mathbf{x}^\star \sim N(\mu_1, \sigma_1^2)$, $\mathbf{x}^{(0)} \sim N(0, \sigma_2^2)$, *and* η *is reasonably small* [1]. *Let* $\mathbf{x}^{(t)}$ *and* $\tilde{\mathbf{x}}^{(t)}$ *be the adversarial examples generated in the* t*-th iteration of attack using BP and LinBP, respectively, then we have*

$$\mathbb{E}\|\mathbf{x}^\star - \tilde{\mathbf{x}}^{(t)}\|_1 \leq \mathbb{E}\|\mathbf{x}^\star - \mathbf{x}^{(t)}\|_1.$$

The proof is shown in the appendix. Theorem 1 shows that LinBP can produce adversarial examples closer to the optimal adversarial examples $\mathbf{x}^\star$ compared to standard BP in the same settings for any finite number of iteration steps, which means that LinBP can craft more powerful and destructive adversarial examples in the white-box attack. It is also straightforward to further derive from our proof that for a benign example with low prediction loss, LinBP produces a more powerful adversarial example starting from the benign one. The conclusion should be surprising because the popular white-box attack methods like FGSM (I-FGSM) and PGD mostly use the gradient obtained by the standard BP and yet we find LinBP may lead to stronger attacks. From our proof, it can be easily derived that the norm of Eq. (14) is bigger than the norm of Eq. (13), which means LinBP provide larger update. Also, the direction of the update obtained from LinBP is closer to the residual $\mathbf{x} - \mathbf{x}^\star$ compared with the standard BP. These may cause LinBP to produce more destructive adversarial examples and powerful attack even in white-box settings. We have conducted extensive experiments on deeper networks using common attack methods to verify our conclusions, which are shown in Section 4.2.

3.2 Theoretical Analyses for Model Training

For model training, we adopt the same idea in adversarial attack to analyze the performance of LinBP. Yet we mainly consider the shallow one-layer teacher-student framework since the student network may not converge to the teacher network in two-layer or deeper models. We will show in Section 4.2 that many of our results hold in more complex network architectures. The one-layer network can be formulated as

$$h(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^T\mathbf{w}), \quad (16)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $\mathbf{w} \in \mathbb{R}^d$ is the weight vector, and $\sigma(\cdot)$ is the ReLU function. Given a set of training samples, we obtain $h(\mathbf{X}, \mathbf{w}) = \sigma(\mathbf{X}\mathbf{w})$, where $\mathbf{X} = [\mathbf{x}_1^T; ...; \mathbf{x}_N^T]$ is the input data matrix, $\mathbf{x}_k \in \mathbb{R}^d$ is the k-th training samples, for $k = 1, ..., N$. We assume the teacher network have the optimal weight, and the loss function can be formulated as

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\|h(\mathbf{X}, \mathbf{w}) - h(\mathbf{X}, \mathbf{w}^\star)\|_2^2, \quad (17)$$

[1] See Eq. (21) for more details of the constraint.

where $\mathbf{w}$ and $\mathbf{w}^\star$ are the weight vector for the student network and teacher network, respectively. Therefore, we can derive the partial gradient with respect to $\mathbf{w}$ for standard BP:

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \mathbf{X}^T\mathbf{D}(\mathbf{X},\mathbf{w})(\mathbf{D}(\mathbf{X},\mathbf{w})\mathbf{X}\mathbf{w} - \mathbf{D}(\mathbf{X},\mathbf{w}^\star)\mathbf{X}\mathbf{w}^\star). \tag{18}$$

Also, the partial gradient to $\mathbf{w}$ for LinBP is formulated as

$$\tilde{\nabla}_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \mathbf{X}^T(\mathbf{D}(\mathbf{X},\mathbf{w})\mathbf{X}\mathbf{w} - \mathbf{D}(\mathbf{X},\mathbf{w}^\star)\mathbf{X}\mathbf{w}^\star). \tag{19}$$

While training the simple one-layer network with SGD, the update rule can be formulated as

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla_{\mathbf{w}^{(\mathbf{t})}}\mathcal{L}(\mathbf{w}^{(\mathbf{t})}), \tag{20}$$

where we use $\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w})$ and $\tilde{\nabla}_{\mathbf{w}}\mathcal{L}(\mathbf{w})$ shown in Eq. (18) and Eq. (19) to obtain $\{\mathbf{w}^{(t)}\}$ and $\{\tilde{\mathbf{w}}^{(t)}\}$ for standard BP and LinBP, respectively. Similar to Theorem 1, we have the following theorem to analyze the training effect of LinBP.

Theorem 2 *For the one-layer teacher-student network formulated as Eq. (16), the training task sets Eq. (17) and Eq. (20) as the loss function and the update rule, respectively. Assume that* $\mathbf{X}$ *is generated from standard Gaussian distribution,* $\mathbf{w}^\star \sim N(\mu_1, \sigma_1^2)$, $\mathbf{w}^{(0)} \sim N(0, \sigma_2^2)$, *and* η *is reasonably small*[2]. *Let* $\mathbf{w}^{(t)}$ *and* $\tilde{\mathbf{w}}^{(t)}$ *be the weight vectors obtained in the* t*-th iteration of training using standard BP and LinBP respectively. Then we have*

$$\mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t)}\|_1 \leq \mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t)}\|_1.$$

The proof is also deferred in the appendix. Theorem 2 shows that LinBP may let the weight vector get closer toward the optimal target weight vector $\mathbf{w}^\star$ relative to standard BP on the same learning rate and training iterations, which means LinBP can also lead to faster convergence in model training when the assumptions are satisfied.

Discussions about η and t. In Theorem 1 and Theorem 2, we assume that the update step size in adversarial attacks and the learning rate in model training are reasonably small. Here we would like to talk more about such assumptions. Precisely, we mean that η should be small enough to satisfy the following constraints: for Theorem 1, it is required that

$$|\sum_{j=0}^{m-1}\frac{\eta d_2}{2\pi}(1-\frac{\eta d_2}{2})^{m-1-j}\mathbf{p}_{ji}| < |(1-\frac{\eta d_2}{2})^m(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)})|, \tag{21}$$

for $m = 1, \ldots, t$ and $i = 1, \ldots, d_1$ indicating the i-th entry of a d_1-dimensional vector, where $\mathbf{p}_j = \theta_j\mathbf{x}^\star - \frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}^{(j)}\|}\sin\theta_j\mathbf{x}^{(j)}$. And for Theorem 2,

$$|\sum_{j=0}^{m-1}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{m-1-j}\mathbf{q}_{ji}| < |(1-\frac{\eta N}{2})^m(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})|, \tag{22}$$

for $m = 1, \ldots, t$ and $i = 1, \ldots, d$, where $\mathbf{q}_j = \theta_j\mathbf{w}^\star - \frac{\|\mathbf{w}^\star\|}{\|\mathbf{w}^{(j)}\|}\sin\theta_j\mathbf{w}^{(j)}$. It is nontrivial to obtain analytic solutions for Eq. (21) and Eq. (22). However, we can know that they are both more likely to hold when t is small. Since the maximum number of learning iterations for generating adversarial examples is often set to be small (at most several hundred) in methods like I-FGSM and PGD, the assumption is easier to be fulfilled than in the setting of model training (which can take tens of thousands of iterations), indicating that the superiority of LinBP can be more obvious in performing adversarial attacks. We will give empirical discussions in Section 4.

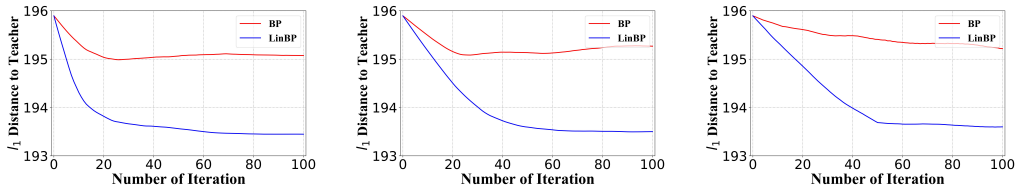
4 EXPERIMENTS

In this section, we provide experimental results on synthetic data (see Section 4.1) and real data (see Section 4.2) to confirm our theorem and compare LinBP against BP in more practical settings for white-box adversarial attack/defense and model training, respectively. The practical experiments show that our theoretical results are hold in variety of different model architectures. All the experiments were performed on NVIDIA GeForce RTX 1080 Ti and the code was implemented on PyTorch (Paszke et al., 2019). Our code is included in our supplementary materials.

[2]See Eq. (22) for a precious formulation of the constraint.

4.1 SIMULATION EXPERIMENTS

Adversarial attack. We constructed a two-layer neural network and performed adversarial attack following the teacher-student framework described in Section 3.1. To be more specific, the victim model is formulated in Eq. (9) and the loss function is given in Eq. (10). Eq. (15) is the update rule to generate adversarial examples. Following the assumption in Theorem 1, the weight matrices $\mathbf{V}$ and $\mathbf{W}$ were generated from independent standard Gaussian distributions. Similarly, the optimal adversarial example $\mathbf{x}^\star$ and the initial adversarial example $\mathbf{x}^{(0)}$ were obtained via sampling from Gaussian distributions, *i.e.,* $\mathbf{x}^\star \sim N(\mu_1, \sigma_1^2)$ and $\mathbf{x}^{(0)} \sim N(0, \sigma_2^2)$, where μ_1, σ_1, and σ_2 can in fact be arbitrary constants. Here we set $\mu_1 = 1.0$, $\sigma_1 = 2.0$, and $\sigma_2 = 1.0$. In our experiments, we set $d_1 = 100$, $d_2 = 20$, $d_3 = 10$, $\eta = 0.001$, and $\epsilon = 0.25$. And the maximum number of iteration step was set as 100. We randomly sampled 10 sets of weight matrices, optimal adversarial examples, and initial adversarial examples for different methods. The l_1 distance between the obtained student adversarial examples and the optimal adversarial examples from the teacher is shown in Figure 1(a) . The results show that LinBP can make the adversarial examples converge faster to the optimal ones, indicating that LinBP helps to craft more powerful examples in the same hyperparameter settings, which clearly confirms our Theorem 1. As we mentioned in Section 3, the norm of the update obtained from LinBP is larger than that obtained from the standard BP, which may cause faster convergence for LinBP. For deeper discussions, we have conducted two more experiments to analyze the effect of LinBP. We used l_2 norm and l_∞ norm to normalize the update obtained from LinBP and standard BP. η was set as 0.05 and 0.005, respectively. The results are shown in Figure 1(b) and Figure 1(c) , where we can find our conclusion holds with gradient normalization.



(a) Original updates (b) l_2 normalized updates (c) l_∞ normalized updates

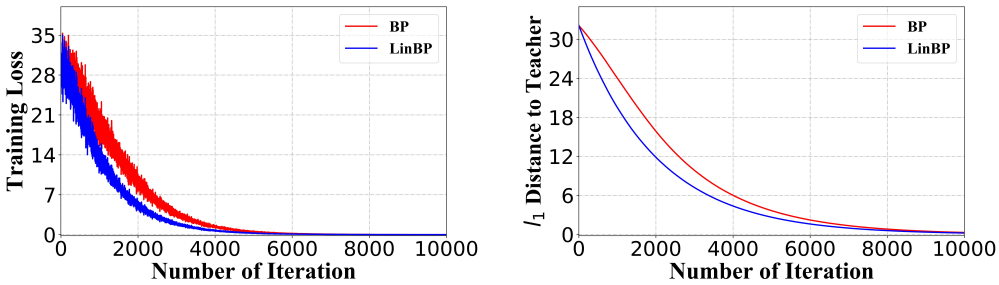
Figure 1: LinBP leads to more powerful white-box adversarial examples which are closer to the optimal ones.

Model training. As described in Section 3.2, we constructed the one-layer neural network formulaed as Eq. (16) and trained it following the student-teacher framework. Note that Eq. (17) is the loss function. Similar to the adversarial attack experiments, the input data matrix $\mathbf{X}$ was generated from the standard Gaussian distribution and the optimal weight vector $\mathbf{w}^\star$ in the teacher network and the initial weight vector $\mathbf{w}^{(0)}$ in the student network were obtained via sampling from a Gaussian distributions with $\mu_1 = 1.0$, $\mu_2 = 0.0$, $\sigma_1 = 3.0$, and $\sigma_2 = 1.0$. We used SGD for optimization and set $N = 100$, $d = 10$, and $\eta = 0.001$ in our experiments. The maximal number of optimization iteration was as 10000 to guarantee training convergence. We used the l_1 distance between the weight vector in the teacher model and that in the student model to evaluate their difference. Figure 2 illustrates the experimental results over 10 runs and compares the two methods. From the results we can observe that LinBP leads to more accurate approximation (*i.e.*, smaller l_1 distance) to the teacher as well as lower training loss in comparison to BP, which clearly confirms our Theorem 2, indicating that LinBP can lead to faster convergence in the same settings.

4.2 MORE PRACTICAL EXPERIMENTS

We also conducted experiments on more practical settings for adversarial attack and model training on MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky et al., 2009), using DNNs with a variety of different architectures, including a simple MLP, LeNet-5 (LeCun et al., 1998), ResNet-50 (He et al., 2016), DenseNet-161 (Huang et al., 2017), and MobileNetV2 (Sandler et al., 2018). License of the datasets and models can be found in the official paper or GitHub.

Adversarial attack on DNNs. We performed white-box adversarial attacks on CIFAR-10 using several different DNNs, including VGG-16, ResNet-50, DenseNet-161, and MobileNetV2. We used the test set of CIFAR-10 to implement I-FGSM (Kurakin et al., 2017) on pre-trained models collected



(a) Faster decrease of training loss (b) Lower approximation error to the teacher

Figure 2: LinBP leads to lower training loss and better approximation to the teacher model, especially at the early stage of training.

on GitHub to generate adversarial examples for this experiment. The attack success rate of these adversarial examples was adopted to evaluate the performance of the attack, and only the correctly classified benign CIFAR-10 images were chosen for generating adversarial examples. The update step size of the iterative attack η was fixed to be $2/255$. We tested with different settings of the maximum number of learning step K and the perturbation budget ϵ. I-FGSM was performed following these settings and we summarize the attack performance using LinBP and BP in Table 1. We considered $K \in \{5, 10\}$ and $\epsilon \in \{0.05, 0.1\}$. It is easy to see in Table 1 that LinBP gains consistently higher attack success rates with I-FGSM, showing that it can help generate more powerful adversarial examples.

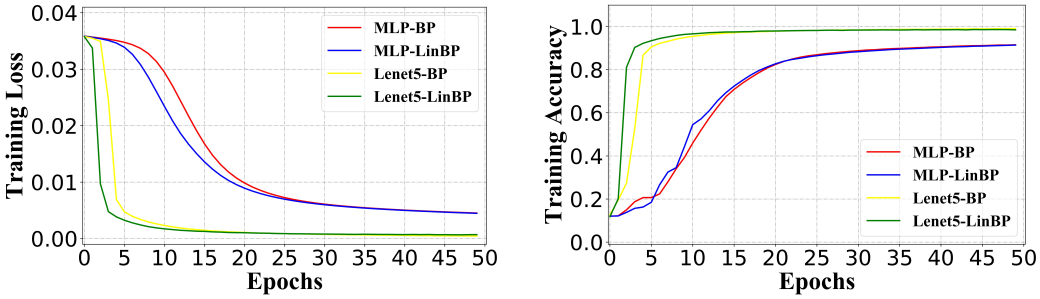
Table 1: Success rate of white-box adversarial attacks using I-FGSM with LinBP and BP in different settings. Higher success rate (*i.e.*, lower prediction accuracy on the adversarial examples) indicates more powerful attack.

Method	K	ϵ	VGG-16	ResNet-50	DenseNet-161	MobileNetV2
BP	5	0.05	71.37%	66.97%	58.25%	91.88%
	5	0.1	71.37%	66.97%	58.25%	91.88%
	10	0.05	90.99%	86.25%	82.94%	98.56%
	10	0.1	93.25%	90.76%	87.53%	98.87%
LinBP	5	0.05	**97.80%**	**90.23%**	**82.56%**	**99.85%**
	5	0.1	**97.80%**	**90.23%**	**82.56%**	**99.85%**
	10	0.05	**99.90%**	**98.67%**	**96.31%**	**100.0%**
	10	0.1	**100.00%**	**99.72%**	**99.81%**	**100.0%**

As we have analyzed, the superiority of LinBP can be more significant with smaller t. In general, for performing adversarial attacks, the maximum number of optimization steps is rather limited and normally at most hundreds. Under such circumstance, the constraint of η in Eq. (21) is easier to be fulfilled and thus LinBP can be consistently better than BP in most test cases. We have also tested with $K = 500$ and using both LinBP and BP gained 100% attack success rate with $K = 500$.

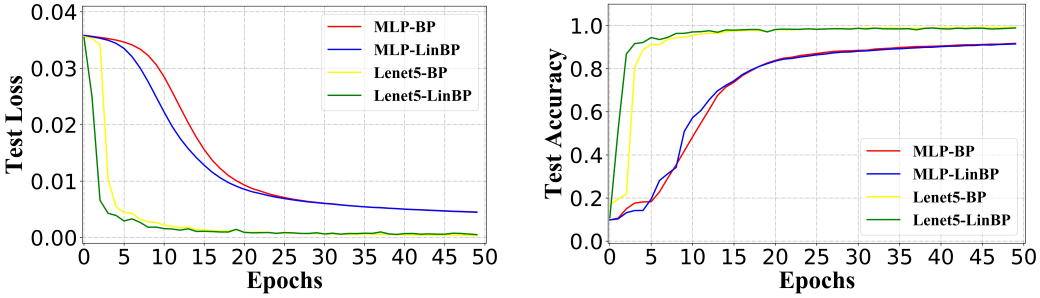
Model training. We first trained and evaluated the MLP and LeNet-5 on MNIST. The MLP has four parameterized and learnable layers, and the numbers of its hidden layer units are 400, 200, and 100. We used SGD for optimization and the learning rate was 0.001 and 0.005 for the MLP and LeNet-5, respectively. The training batch size was set to 64 and the training process lasted for at most 50 epochs. The training loss and training accuracy are illustrated in Figure 3, and note that we fix the random seed to eliminate unexpected randomness in training. We can easily observe from the training curves in Figure 3 that the obtained MLP and LeNet-5 models show lower prediction loss and higher accuracy when incorporating LinBP, especially in the early dozens of epochs, which suggests that the incorporation of LinBP can be beneficial to the convergence of SGD. The same observation can be made on the test set of MNIST and the results are shown in Figure 4.

We further report our experimental results on CIFAR-10. ResNet-50 (He et al., 2016), DenseNet-161 (Huang et al., 2017), and MobileNetV2 (Sandler et al., 2018) were trained and evaluated on the dataset. The architecture of these networks and their detailed settings can be found in their papers.



(a) Decrease faster in training loss (b) Increase faster in training accuracy

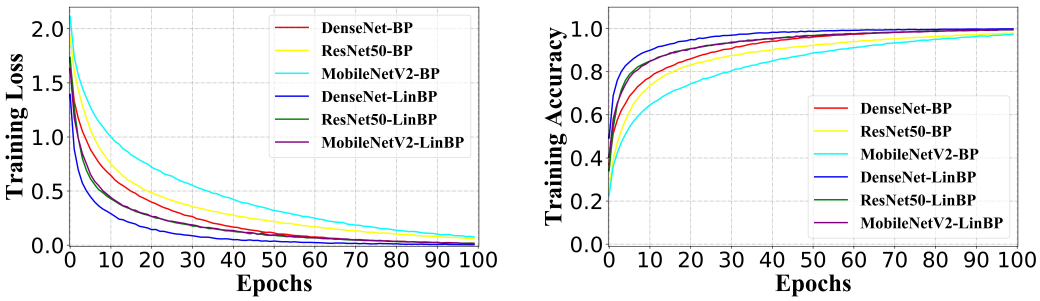
Figure 3: Compare the training loss and training accuracy of the MLP and LeNet-5 on MNIST using LinBP and BP.



(a) Decrease faster in test loss (b) Increase faster in test accuracy

Figure 4: Compare the test loss and test accuracy of the MLP and LeNet-5 on MNIST using LinBP and BP.

The optimizer was SGD and the learning rate was set to 0.002. We set the batch size as 128 and trained for 100 epochs. We evaluated the prediction loss and accuracy of trained models using LinBP and BP. Similar to the experiment on MNIST, we fixed the random seed. The training results in Figure 5 and test results in Figure 6 demonstrate that equipped with LinBP, the models achieved lower prediction loss and higher prediction accuracy in the same training settings, especially when the training just got started. Nevertheless, as training progressed, the superiority of LinBP became less obvious and finally the performance of LinBP became just comparable to that of the standard BP, which is consistent with our theoretical discussions in Section 3.2.

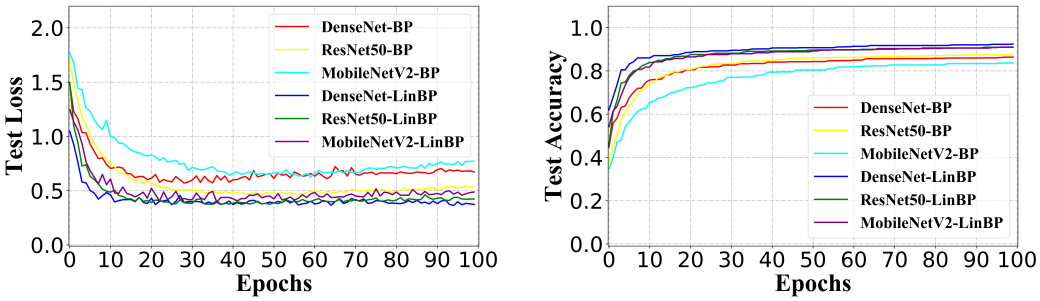


(a) Decrease faster in training loss (b) Increase faster in training accuracy

Figure 5: Compare the training loss and training accuracy of ResNet-50, DenseNet-161, and MobileNetV2 on CIFAR-10 using LinBP and BP.

We also noticed that in certain scenarios, optimization with LinBP may fail to converge on very large DNN models (and large learning rates, *e.g.*, VGG-Net (Simonyan & Zisserman, 2015)), and it is ascribed to unsatisfactory approximation to the gradient and needs to be further studied in future work. Decreasing the learning rate appropriately may relieve the problem to some extent.

Adversarial training for DNNs. Since the discovery of adversarial examples, the vulnerability of DNNs has been intensively discussed. Tremendous effort has been devoted to improve the



(a) Decrease faster in test loss

(b) Increase faster in test accuracy

Figure 6: Compare the test loss and test accuracy of ResNet-50, DenseNet-161, and MobileNetV2 on CIFAR-10 using LinBP and BP.

robustness of DNNs. Thus far, a variety of methods have been proposed, in which adversarial training (Goodfellow et al., 2015; Madry et al., 2018) has become an indispensable procedure in many application scenarios. In this context, we would like to study how LinBP can be adopted to further enhance the robustness of DNNs. We used PGD to generate adversarial examples to construct our dataset on which we train our model using SGD. There exist multiple strategies of adopting LinBP in adversarial training, *i.e.*, 1) computing gradients for updating model parameters using LinBP just like in the model training experiment and 2) generated adversarial examples using LinBP as in the adversarial attack experiment. We observed that the first strategy is beneficial, and further applying the second strategy in combination slightly increases the performance in latter epochs (See Figure 6). Specifically, we applied the classification accuracy on adversarial examples using the standard BP to evaluate the model robustness. The experiment was performed on CIFAR-10 using MobileNetV2, where the attack step size was $2/255$, $\epsilon = 8/255$, $K = 5$ and the training learning rate was 0.01.

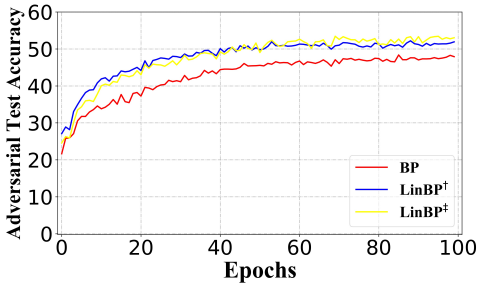


Figure 7: Compare the robustness of MobileNetV2 models shielded with different adversarial training methods. We use "LinBP†" and "LinBP‡" to indicate models trained using the first strategy and the combination of the two strategies as described in the above paragraph, respectively.

5 Conclusions

We have studied the optimization convergence of the standard BP and compare to the very recent method called LinBP which skips ReLUs during the backward pass. Theoretical analyses have been carefully performed in two popular application scenarios, *i.e.*, white-box adversarial attack and model training. In addition to the benefit on black-box transferability which has already been shown in Guo et al. (2020)'s work, we have proven in this paper that LinBP also leads to generate more destructive adversarial examples and faster convergence in the same hyperparameter settings. Experimental results on simulated data confirms our theoretical results. Extensive experiments on MNIST and CIFAR-10 further validate that the theoretical results hold in practical settings on a variety of DNN architectures. Our theoretical and empirical discussions in the paper perform thorough analysis on LinBP, and may become an inspiration for more researches on convergence analysis in neural-network-involved learning tasks.

ETHICS STATEMENT

Our theoretical and empirical discussions in the paper are related to machine learning robustness. Make more destructive adversarial samples will fool the trained model and hamper the model robustness. That may cause potential privacy and security concerns. We have read the codes of ethics, and we guarantee that our work conforms to them.

REPRODUCIBILITY STATEMENT

For our theoretical results, we state the full set of assumptions and the complete proof can be found in the appendix. In our supplemental materials, we submit the entire codes for all the experiments in the paper. We use the public datasets MNIST and CIFAR-10 which can be directly downloaded in Pytorch.

REFERENCES

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *ICML*, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE symposium on security and privacy (SP)*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *ICML*, 2019a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*, 2019b.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. In *NeurIPS*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

Yann LeCun. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pp. 21–28, 1988.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM on Asia conference on computer and communications security*, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *ICML*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

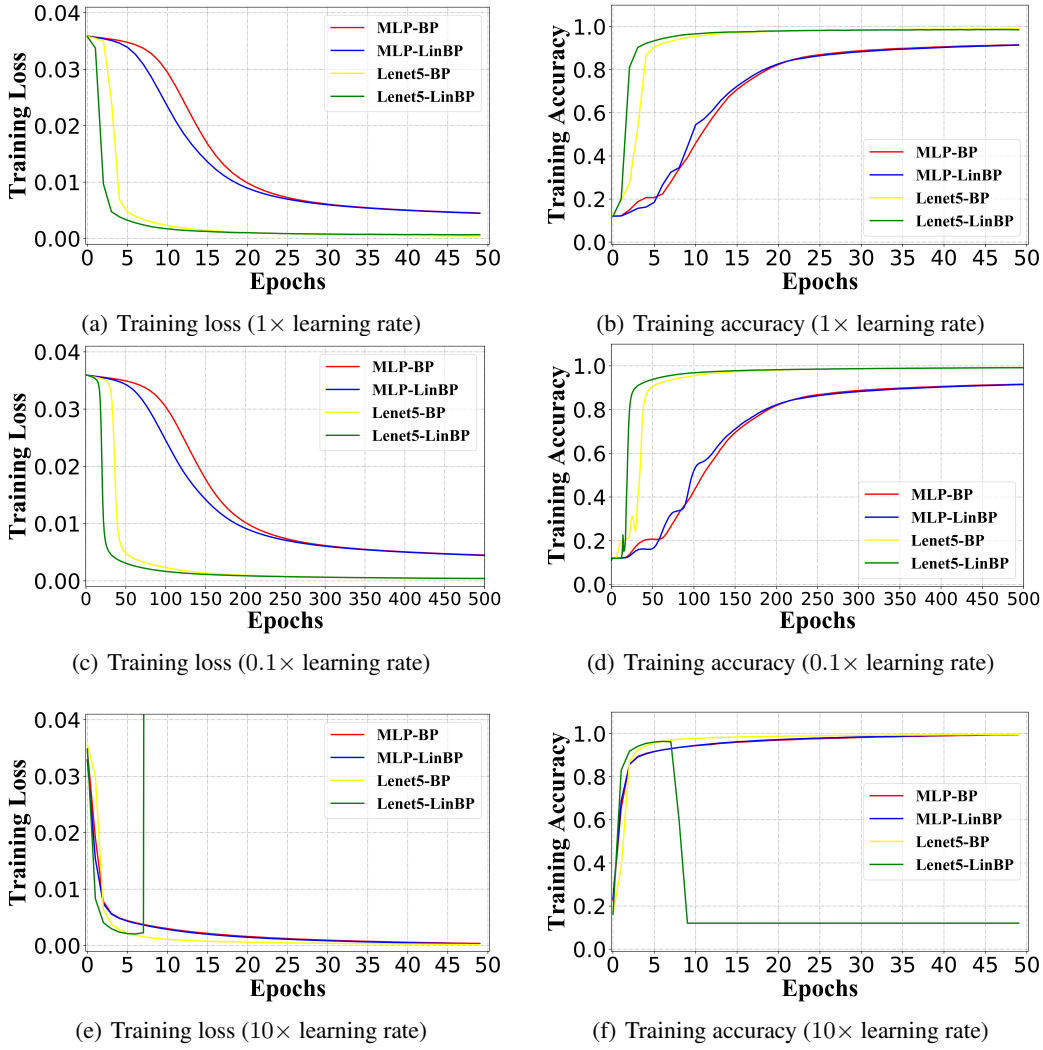
A Appendix

A.1 The Influence of η

As discussed in the main paper, the superiority of LinBP would be more obvious with reasonably small η. We tried using LinBP with varying η, and observed that slightly decreasing the learning rate for model training stabilized model training using LinBP and guaranteed the superiority of LinBP. Here, for adequate comparison, we report results using two more η values than in the main paper in Figure 8. The basic learning rate η is set as 0.001 and 0.005 for MLP and LeNet-5 in training MNIST model just equals to the experiments in the main paper.. We further scaled the basic learning rate by a factor of 10 and 0.1 in the experiment and report results in Figure 8.

It can be seen that, when scaled the learning rate by $10\times$, the performance of LinBP and BP are similar on the MLP but on the LeNet-5 the training failed to converge with LinBP while it still converged with BP [3]. The results conform that a reasonably small η help stabilize training using LinBP and guarantees its superiority.

[3]The same observation was also made when training VGG-16 on CIFAR-10 using LinBP and a learning rate of 0.005.



(a) Training loss (1× learning rate)

(b) Training accuracy (1× learning rate)

(c) Training loss (0.1× learning rate)

(d) Training accuracy (0.1× learning rate)

(e) Training loss (10× learning rate)

(f) Training accuracy (10× learning rate)

Figure 8: The influence of setting different learning rate to LinBP and BP for training the MLP and LeNet-5 on MNIST.

A.2 PROOFS

In this subsection, we will give theoretical proofs on Lemma 1, Theorem 1 and Theorem 2 in the main paper. Note that our proofs are mainly based on the following theorem in Tian (2017)'s work.

Theorem 1 in Tian (2017) *Denote* $F(\mathbf{e}, \mathbf{w}) := \mathbf{X}^T D(\mathbf{X}, \mathbf{e}) D(\mathbf{X}, \mathbf{w}) \mathbf{X}\mathbf{w}$*, where* $\mathbf{e}$ *is a unit vector and* $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times d}$ *is the input data matrix. If* $\mathbf{X}$ *is standard Gaussian distributed, we have*

$$\mathbb{E}\left(F(\mathbf{e}, \mathbf{w})\right) = \frac{N}{2\pi}[(\pi - \theta)\mathbf{w} + \|\mathbf{w}\| \sin\theta \mathbf{e}],$$

where $\theta \in [0, \pi]$ *is the angle between* $\mathbf{e}$ *and* $\mathbf{w}$.

A.2.1 PROOF OF LEMMA 1

We first recall the contents of Lemma 1.

Lemma 1 *Denote* $G(\mathbf{e}, \mathbf{x}) := \mathbf{W}^T D(\mathbf{W}, \mathbf{e}) \mathbf{V}^T \mathbf{V} D(\mathbf{W}, \mathbf{x}) \mathbf{W}\mathbf{x}$*, where* $\mathbf{e} \in \mathbb{R}^{d_1}$ *is a unit vector,* $\mathbf{x} \in \mathbb{R}^{d_1}$ *is the input data vector,* $\mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$ *and* $\mathbf{V} \in \mathbb{R}^{d_3 \times d_2}$ *are weight matrices. If* $\mathbf{W}$ *and* $\mathbf{V}$

follow independent standard Gaussian distribution, we have

$$\mathbb{E}\left(G(\mathbf{e},\mathbf{x})\right) = \frac{d_2}{2\pi}[(\pi-\Theta)\mathbf{x} + \|\mathbf{x}\|\sin\Theta\mathbf{e}],$$

where $\Theta \in [0,\pi]$ *is the angle between* $\mathbf{e}$ *and* $\mathbf{x}$.

Proof. Assume $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_{d_2}]^T$ and $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_{d_2}]$, where $\mathbf{w}_i \in \mathbb{R}^{d_1}$ and $\mathbf{v}_i \in \mathbb{R}^{d_3}$, for $i = 1, \cdots, d_2$. For $G(\mathbf{e},\mathbf{x})$, we have

$$G(\mathbf{e},\mathbf{x}) = \sum_{i:\mathbf{w}_i^T\mathbf{x}\geq 0, \mathbf{w}_i^T\mathbf{e}\geq 0} \sum_{j:\mathbf{w}_j^T\mathbf{x}\geq 0, \mathbf{w}_j^T\mathbf{e}\geq 0} \sum_{d=1}^{d_3} v_{id}v_{dj}\mathbf{w}_i\mathbf{w}_j^T\mathbf{x}. \tag{23}$$

We consider the expectation of $G(\mathbf{e},\mathbf{x})$, there is

$$\mathbb{E}(G(\mathbf{e},\mathbf{x})) = \sum_{i=1}^{d_2}\sum_{j=1}^{d_2}\mathbb{E}(\sum_{d=1}^{d_3} v_{id}v_{dj})\mathbb{E}(\mathbf{w}_i\mathbf{w}_j^T \cdot \mathbb{I}_{(\mathbf{w}_i^T\mathbf{x}\geq 0, \mathbf{w}_i^T\mathbf{e}\geq 0, \mathbf{w}_j^T\mathbf{x}\geq 0, \mathbf{w}_j^T\mathbf{e}\geq 0)}(\mathbf{x},\mathbf{e})\mathbf{x}), \tag{24}$$

where $\mathbb{I}_A(x)$ is the indicator function, *i.e.*, $\mathbb{I}_A(x)$ equals 1 if $x \in A$ and equals 0 if $x \notin A$. As the assumption that $\mathbf{W}$ and $\mathbf{V}$ follow independent standard Gaussian distribution, we find $\mathbb{E}(\sum_{d=1}^{d_3} v_{id}v_{dj}) = 0$ when $i \neq j$ and $\mathbb{E}(\sum_{d=1}^{d_3} v_{id}v_{dj}) = 1$ when $i = j$. Therefore, Eq. (24) can be simplified as

$$\mathbb{E}(G(\mathbf{e},\mathbf{x})) = \sum_{i=1}^{d_2}\mathbb{E}(\mathbf{w}_i\mathbf{w}_i^T \cdot \mathbb{I}_{(\mathbf{w}_i^T\mathbf{x}\geq 0, \mathbf{w}_i^T\mathbf{e}\geq 0)}(\mathbf{x},\mathbf{e})\mathbf{x}). \tag{25}$$

We then introduce a coordinate system, where $\mathbf{e} = [1, 0, \cdots, 0]^T$, $\mathbf{x} = \|\mathbf{x}\|[\cos\Theta, \sin\Theta, 0, \cdots, 0]^T$ are hold. Thus, $\mathbf{w_i} = [r\cos\phi_i, r\sin\phi_i, w_{i,3}, \cdots, w_{i,d_1}]^T$. We then can rewrite Eq. (25) as

$$\begin{aligned}\mathbb{E}(G(\mathbf{e},\mathbf{x})) &= \mathbb{E}\sum_{i:\mathbf{w}_i^T\mathbf{x}\geq 0, \mathbf{w}_i^T\mathbf{e}\geq 0} \mathbf{w}_i\mathbf{w}_i^T\mathbf{x} \\ &= \mathbb{E}\sum_{i:\phi_i\in[-\pi/2+\Theta,\pi/2]} \mathbf{w}_i\mathbf{w}_i^T\mathbf{x}\end{aligned} \tag{26}$$

Then we compute the following equation,

$$\begin{aligned}R(\phi_0) = \mathbb{E}\left[\frac{1}{N}\sum_{i:\phi_i\in[0,\phi_0]} \mathbf{w_i}\mathbf{w_i}^T\right] &= \mathbb{E}\left[\mathbf{w}\mathbf{w}^T|\phi\in[0,\phi_0]\right]\mathbb{P}\left[\phi\in[0,\phi_0]\right] \\ &= \int_{-\infty}^{\infty}\cdots\int_0^{\phi_0}\int_0^{\infty}\mathbf{w}\mathbf{w}^T p(r)p(\phi)\prod_{i=3}^{d} p(w_i)rdrd\phi dw_3\ldots dw_{d_1},\end{aligned} \tag{27}$$

where $p(r) = e^{-\frac{r^2}{2}}$ and $p(\phi) = \frac{1}{2\pi}$. Since $\mathbf{W}$ follows gaussian distribution, the off-diagonal and diagonal elements except the first 2×2 block are equal to 0 and $\frac{\phi_0}{2\pi}$ respectively. The first 2×2 block can be formulated as

$$\begin{aligned}R(\phi_0)_{[1:2,1:2]} &= \int_0^{\phi_0}\int_0^{\infty}\begin{bmatrix}r\cos\phi\\ r\sin\phi\end{bmatrix}\begin{bmatrix}r\cos\phi & r\sin\phi\end{bmatrix}p(r)p(\phi)rdrd\phi \\ &= \int_0^{\infty}\frac{r^3e^{-\frac{r^2}{2}}}{2\pi}dr\int_0^{\phi_0}\begin{bmatrix}\cos^2\phi & \cos\phi\sin\phi\\ \cos\phi\sin\phi & \sin^2\phi\end{bmatrix}d\phi. \\ &= \frac{1}{4\pi}\begin{bmatrix}2+\sin 2\phi_0 & 1-\cos 2\phi_0\\ 1-\cos 2\phi_0 & 2-\sin 2\phi_0\end{bmatrix}\end{aligned} \tag{28}$$

Further we have

$$R(\phi_0) = \frac{1}{2\pi}\mathbf{I}_d + \frac{1}{4\pi}\begin{bmatrix}\sin 2\phi_0 & 1-\cos 2\phi_0 & \mathbf{0}\\ 1-\cos 2\phi_0 & -\sin 2\phi_0 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix} \tag{29}$$

Then Eq. (26) can be formulated as

$$\begin{aligned}\mathbb{E}(G(\mathbf{e},\mathbf{x})) &= d_2\left(R(\frac{\pi}{2}) - R(-\frac{\pi}{2}+\Theta)\right)x \\ &= d_2\frac{\pi-\Theta}{2\pi}\mathbf{I}_d + \frac{d_2}{4\pi}(\begin{bmatrix}0 & 2\\ 2 & 0\end{bmatrix} - \begin{bmatrix}\sin(2\Theta-\pi) & 1-\cos(2\Theta-\pi)\\ 1-\cos(2\Theta-\pi) & -\sin(2\Theta-\pi)\end{bmatrix})\|x\|\begin{bmatrix}\cos\Theta\\ \sin\Theta\end{bmatrix} \\ &= d_2\frac{\pi-\Theta}{2\pi}x + \frac{d_2\|x\|}{4\pi}\begin{bmatrix}2\sin\Theta\\ 0\end{bmatrix} \\ &= \frac{d_2}{2\pi}[(\pi-\Theta)\mathbf{x} + \|\mathbf{x}\|\sin\Theta\mathbf{e}]\end{aligned} \tag{30}$$

A.2.2 LEMMA FOR FURTHER PROOF.

Before we delve deep into Theorem 1 and Theorem 2, we first propose a lemma for convenience. The lemma is described as follows,

Lemma 2 *Let α_i define as*

$$\alpha_i = (u\cdot\mathbf{x}_i^\star - v\cdot\mathbf{x}_i)sgn(\mathbf{x}_i^\star - \mathbf{x}_i) = \begin{cases}u\cdot\mathbf{x}_i^\star - v\cdot\mathbf{x}_i, & if \quad \mathbf{x}_i^\star > \mathbf{x}_i,\\ v\cdot\mathbf{x}_i - u\cdot\mathbf{x}_i^\star, & if \quad \mathbf{x}_i^\star < \mathbf{x}_i,\end{cases}$$

where u and v are constants, $\mathbf{x}_i^\star \sim N(\mu_1,\sigma_1^2)$ and $\mathbf{x}_i \sim N(\mu_2,\sigma_2^2)$. The expectation of α_i can be formulated as

$$\mathbb{E}(\alpha_i) = 2\gamma(u\cdot\sigma_1^2 + v\cdot\sigma_2^2) + (u\cdot\mu_1 - v\cdot\mu_2)(2\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star) - 1),$$

where $\gamma = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} > 0$. Further we have $\mathbb{E}(\alpha_i) > 0$ when $u > 0$, $v > 0$ and $\mu_2 = 0$.

Proof. From the definition of α_i, the expectation of α_i can be formulated as

$$\mathbb{E}(\alpha_i) = \mathbb{E}(u\cdot\mathbf{x}_i^\star - v\cdot\mathbf{x}_i|\mathbf{x}_i^\star > \mathbf{x}_i)\mathbf{P}(\mathbf{x}_i^\star > \mathbf{x}_i) + \mathbb{E}(v\cdot\mathbf{x}_i - u\cdot\mathbf{x}_i^\star|\mathbf{x}_i^\star < \mathbf{x}_i)\mathbf{P}(\mathbf{x}_i^\star < \mathbf{x}_i). \tag{31}$$

Further, we have $\mathbf{x}_i^\star - \mathbf{x}_i \sim N(\mu_1-\mu_2, \sigma_1^2+\sigma_2^2)$. Therefore, we can conclude that $\mathbf{P}(\mathbf{x}_i^\star > \mathbf{x}_i) = 1 - F(0)$ and $\mathbf{P}(\mathbf{x}_i^\star < \mathbf{x}_i) = F(0)$, where $F(\cdot)$ denotes the cumulative distribution function for $N(\mu_1-\mu_2, \sigma_1^2+\sigma_2^2)$.

We then solve the conditional cumulative distribution function for $\mathbf{x}_i^\star$ when $\mathbf{x}_i^\star > \mathbf{x}_i$. We define $f_1(\cdot)$, $f_2(\cdot)$ denote the probability density function of $\mathbf{x}_i^\star$ and $\mathbf{x}_i$ and $F_1(\cdot)$, $F_2(\cdot)$ denote the cumulative distribution function of $\mathbf{x}_i^\star$ and $\mathbf{x}_i$. There we have

$$\begin{aligned}G_1(x) &= \mathbf{P}(\mathbf{x}_i^\star \le x|\mathbf{x}_i^\star > \mathbf{x}_i) \\ &= \frac{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star \le x)}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)}\int_{-\infty}^{x}\mathbf{P}(y < \mathbf{x}_i^\star \le x)f_2(y)dy \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)}\int_{-\infty}^{x}(F_1(x) - F_1(y))f_2(y)dy.\end{aligned} \tag{32}$$

Therefore, the probability density function can be formulated as

$$\begin{aligned}g_1(x) &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)}\left[(F_1(x)\int_{-\infty}^{x}f_2(y)dy)' - (\int_{-\infty}^{x}(F_1(y)f_2(y)dy)'\right] \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)}\left[(f_1(x)\int_{-\infty}^{x}f_2(y)dy) + F_1(x)f_2(x) - F_1(x)f_2(x)\right] \\ &= \frac{f_1(x)F_2(x)}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)}.\end{aligned} \tag{33}$$

Similarly, we first solve the conditional cumulative distribution function for $\mathbf{x}_i$ when $\mathbf{x}_i^\star > \mathbf{x}_i$. There we have,

$$\begin{aligned} G_2(x) &= 1 - \mathbf{P}(\mathbf{x}_i > x | \mathbf{x}_i^\star > \mathbf{x}_i) \\ &= 1 - \frac{\mathbf{P}(x < \mathbf{x}_i < \mathbf{x}_i^\star)}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \\ &= 1 - \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_x^\infty \mathbf{P}(x < \mathbf{x}_i < y) f_1(y) dy \\ &= 1 - \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_x^\infty (F_2(y) - F_2(x)) f_1(y) dy. \end{aligned} \tag{34}$$

The probability density function can be formulated as

$$\begin{aligned} g_2(x) &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \left[(F_2(x) \int_x^\infty f_1(y) dy)' - (\int_x^\infty (F_2(y) f_1(y) dy)' \right] \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \left[(f_2(x) \int_x^\infty f_1(y) dy) - F_2(x) f_1(x) + F_2(x) f_1(x) \right] \\ &= \frac{f_2(x)(1 - F_1(x))}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)}. \end{aligned} \tag{35}$$

We assume $\phi(\cdot)$ and $\Phi(\cdot)$ represent the probability density function and cumulative distribution function for standard gaussian distribution. There we have

$$\begin{aligned} \mathbb{E}(\mathbf{x}_i^\star | \mathbf{x}_i^\star > \mathbf{x}_i) &= \int_{-\infty}^\infty x g_1(x) dx \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_{-\infty}^\infty \frac{x}{\sigma_1} \phi(\frac{x - \mu_1}{\sigma_1}) \Phi(\frac{x - \mu_2}{\sigma_2}) dx \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_{-\infty}^\infty \frac{x}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2} \Phi(\frac{x - \mu_2}{\sigma_2}) dx \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_{-\infty}^\infty \left[\frac{x - \mu_1}{\sqrt{2\pi}\sigma_1} + \frac{\mu_1}{\sqrt{2\pi}\sigma_1} \right] e^{-(x-\mu_1)^2/2\sigma_1^2} \Phi(\frac{x - \mu_2}{\sigma_2}) dx \end{aligned} \tag{36}$$

The integral can be devided into two part. The first part can be formulated as

$$\begin{aligned} \int_{-\infty}^\infty \frac{x - \mu_1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2} \Phi(\frac{x - \mu_2}{\sigma_2}) dx &= \frac{\sigma_1}{\sqrt{2\pi}} \int_{-\infty}^\infty \Phi(\frac{x - \mu_2}{\sigma_2}) \frac{d(-e^{-(x-\mu_1)^2/2\sigma_1^2})}{dx} dx \\ &= \frac{\sigma_1}{\sqrt{2\pi}} \int_{-\infty}^\infty e^{-(x-\mu_1)^2/2\sigma_1^2} \frac{d\Phi(\frac{x-\mu_2}{\sigma_2})}{dx} dx \\ &= \frac{\sigma_1}{2\sigma_2\pi} \int_{-\infty}^\infty e^{-(x-\mu_1)^2/2\sigma_1^2 - (x-\mu_2)^2/2\sigma_2^2} dx \\ &= \frac{\sigma_1}{2\sigma_2\pi} e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} \frac{\sigma_1\sigma_2\sqrt{2\pi}}{\sqrt{\sigma_1^2+\sigma_2^2}} \\ &= \frac{\sigma_1^2}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} \end{aligned} \tag{37}$$

Step 3 equals Step 4 because there is $\int_{-\infty}^\infty e^{-(ax^2+bx+c)} dx = e^{(b^2-4ac)/4a} \sqrt{\frac{\pi}{a}}$. The second part can be formulated as

$$\int_{-\infty}^\infty \frac{\mu_1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2} \Phi(\frac{x - \mu_2}{\sigma_2}) dx = \mu_1 \int_{-\infty}^\infty f_1(x) F_2(x) dx \tag{38}$$

From Eq. (33) we have

$$\int_{-\infty}^\infty g_1(x) dx = \frac{\int_{-\infty}^\infty f_1(x) F_2(x) dx}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} = 1 \tag{39}$$

Therefore we have

$$\int_{-\infty}^{\infty} \frac{\mu_1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2} \Phi(\frac{x-\mu_2}{\sigma_2}) dx = \mu_1 \int_{-\infty}^{\infty} f_1(x) F_2(x) dx = \mu_1 \mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star) \tag{40}$$

As the result in Eq. (37) and Eq. (40), Eq. (36) can be formulated as

$$\begin{aligned} \mathbb{E}(\mathbf{x}_i^\star | \mathbf{x}_i^\star > \mathbf{x}_i) &= \frac{\frac{\sigma_1^2}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)}}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} + \mu_1 \\ &= \frac{\gamma \sigma_1^2}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} + \mu_1, \end{aligned} \tag{41}$$

where $\gamma = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} > 0$.

Similarly, we have

$$\begin{aligned} \mathbb{E}(\mathbf{x}_i | \mathbf{x}_i^\star > \mathbf{x}_i) &= \int_{-\infty}^{\infty} x g_2(x) dx \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_{-\infty}^{\infty} \frac{x}{\sigma_2} \phi(\frac{x-\mu_2}{\sigma_2})(1 - \Phi(\frac{x-\mu_1}{\sigma_1})) dx \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_{-\infty}^{\infty} \frac{x}{\sqrt{2\pi}\sigma_2} e^{-(x-\mu_2)^2/2\sigma_2^2}(1 - \Phi(\frac{x-\mu_1}{\sigma_1})) dx \\ &= \frac{1}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} \int_{-\infty}^{\infty} \left[\frac{x-\mu_2}{\sqrt{2\pi}\sigma_2} + \frac{\mu_2}{\sqrt{2\pi}\sigma_2} \right] e^{-(x-\mu_2)^2/2\sigma_2^2}(1 - \Phi(\frac{x-\mu_1}{\sigma_1})) dx \end{aligned} \tag{42}$$

The integral can also be devided into two part. The first part can be formulated as

$$\begin{aligned} &\int_{-\infty}^{\infty} \frac{x-\mu_2}{\sqrt{2\pi}\sigma_2} e^{-(x-\mu_2)^2/2\sigma_2^2}(1 - \Phi(\frac{x-\mu_1}{\sigma_1})) dx \\ &= \frac{\sigma_2}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [1 - \Phi(\frac{x-\mu_1}{\sigma_1})] \frac{d(-e^{-(x-\mu_2)^2/2\sigma_2^2})}{dx} dx \\ &= \frac{\sigma_2}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-(x-\mu_2)^2/2\sigma_2^2} \frac{d(1 - \Phi(\frac{x-\mu_1}{\sigma_1}))}{dx} dx \\ &= \frac{-\sigma_2}{2\sigma_1\pi} \int_{-\infty}^{\infty} e^{-(x-\mu_1)^2/2\sigma_1^2 - (x-\mu_2)^2/2\sigma_2^2} dx \\ &= \frac{-\sigma_2}{2\sigma_1\pi} e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} \frac{\sigma_1\sigma_2\sqrt{2\pi}}{\sqrt{\sigma_1^2+\sigma_2^2}} \\ &= \frac{-\sigma_2^2}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} \end{aligned} \tag{43}$$

The second part can be formulated as

$$\int_{-\infty}^{\infty} \frac{\mu_2}{\sqrt{2\pi}\sigma_2} e^{-(x-\mu_2)^2/2\sigma_2^2}(1 - \Phi(\frac{x-\mu_1}{\sigma_1})) dx = \mu_2 \int_{-\infty}^{\infty} f_2(x)(1 - F_1(x)) dx \tag{44}$$

From Eq. (35) we have

$$\int_{-\infty}^{\infty} g_2(x) dx = \frac{\int_{-\infty}^{\infty} f_2(x)(1 - F_1(x)) dx}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} = 1 \tag{45}$$

Therefore we have

$$\begin{aligned} \int_{-\infty}^{\infty} \frac{\mu_2}{\sqrt{2\pi}\sigma_2} e^{-(x-\mu_2)^2/2\sigma_2^2}(1 - \Phi(\frac{x-\mu_1}{\sigma_1})) dx &= \mu_2 \int_{-\infty}^{\infty} f_2(x)(1 - F_1(x)) dx \\ &= \mu_2 \mathbf{P}(w_i < w_i^\star) \end{aligned} \tag{46}$$

Therefore, Eq. (42) can be formulated as

$$\begin{aligned}\mathbb{E}(\mathbf{x}_i|\mathbf{x}_i^\star > \mathbf{x}_i) &= \frac{\frac{-\sigma_2^2}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)}}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} + \mu_2 \\ &= \frac{-\gamma\sigma_2^2}{\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star)} + \mu_2.\end{aligned} \tag{47}$$

Due to symmetry, we also have

$$\mathbb{E}(\mathbf{x}_i|\mathbf{x}_i^\star < \mathbf{x}_i) = \frac{\gamma\sigma_2^2}{\mathbf{P}(\mathbf{x}_i > \mathbf{x}_i^\star)} + \mu_2,$$

and

$$\mathbb{E}(\mathbf{x}_i^\star|\mathbf{x}_i^\star < \mathbf{x}_i) = \frac{-\gamma\sigma_1^2}{\mathbf{P}(\mathbf{x}_i > \mathbf{x}_i^\star)} + \mu_1.$$

Therefore, Eq. (31) can be formulated as

$$\begin{aligned}\mathbb{E}(\alpha_i) =&\gamma(u\cdot\sigma_1^2 + v\cdot\sigma_2^2) + (u\cdot\mu_1 - v\cdot\mu_2)\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star) \\ &+ \gamma(v\cdot\sigma_2^2 + u\cdot\sigma_1^2) + (v\cdot\mu_2 - u\cdot\mu_1)\mathbf{P}(\mathbf{x}_i > \mathbf{x}_i^\star) \\ =&2\gamma(u\cdot\sigma_1^2 + v\cdot\sigma_2^2) + (u\cdot\mu_1 - v\cdot\mu_2)(2\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star) - 1),\end{aligned} \tag{48}$$

where $\gamma = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}e^{-(\mu_1-\mu_2)^2/2(\sigma_1^2+\sigma_2^2)} > 0$. When $\mu_2 = 0$, we have

$$\mathbb{E}(\alpha_i) = 2\gamma(u\cdot\sigma_1^2 + v\cdot\sigma_2^2) + u\mu_1(2\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star) - 1). \tag{49}$$

If $u > 0$ and $v > 0$, we have $\gamma(u\cdot\sigma_1^2 + v\cdot\sigma_2^2) > 0$, and

$$u\mu_1(2\mathbf{P}(\mathbf{x}_i < \mathbf{x}_i^\star) - 1) = u\mu_1(1 - 2F(0)), \tag{50}$$

where F is the cumulative distribution function for $\mathbf{x}_i^\star - \mathbf{x}_i$, and follows the gaussian distribution $N(\mu_1, \sigma_1^2 + \sigma_2^2)$. If $\mu_1 > 0$, we have $F(0) < 0.5$, and Eq. (50) is greater than 0. If $\mu_1 < 0$, we have $F(0) > 0.5$, and Eq. (50) is also greater than 0. Therefore, we have $u\mu_1(1 - 2F(0)) \geq 0$. In sum, we have $\mathbb{E}(\alpha_i) > 0$ when $u > 0$, $v > 0$ and $\mu_2 = 0$.

A.2.3 Proof of Theorem 1

We first recall the update rules for BP and LinBP, which are formulated as

$$\mathbf{x}^{(t+1)} = \text{Clip}(\mathbf{x}^{(t)} - \eta\nabla_{\mathbf{x}^{(t)}}\mathcal{L}(\mathbf{x}^{(t)})), \tag{51}$$

and

$$\tilde{\mathbf{x}}^{(t+1)} = \text{Clip}(\tilde{\mathbf{x}}^{(t)} - \eta\tilde{\nabla}_{\tilde{\mathbf{x}}}^{(t)}\mathcal{L}(\tilde{\mathbf{x}}^{(t)})), \tag{52}$$

respectively, where $\text{Clip}(\cdot) = \min(\mathbf{x} + \epsilon\mathbf{1}, \max(\mathbf{x} - \epsilon\mathbf{1}, \cdot))$ and the expectation of the gradient obtained by BP and LinBP can be computed from Lemma 1, *i.e.*,

$$\mathbb{E}[\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x})] = G(\mathbf{x}/\|\mathbf{x}\|, \mathbf{x}) - G(\mathbf{x}/\|\mathbf{x}\|, \mathbf{x}^\star) = \frac{N}{2}(\mathbf{x} - \mathbf{x}^\star) + \frac{N}{2\pi}\left(\Theta\mathbf{x}^\star - \frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}\|}\sin\Theta\mathbf{x}\right), \tag{53}$$

and

$$\mathbb{E}[\tilde{\nabla}_{\mathbf{x}}\mathcal{L}(\mathbf{x})] = G(\mathbf{x}/\|\mathbf{x}\|, \mathbf{x}) - G(\mathbf{x}^\star/\|\mathbf{x}^\star\|, \mathbf{x}^\star) = \frac{N}{2}(\mathbf{x} - \mathbf{x}^\star), \tag{54}$$

, respectively. Note $\Theta \in [0, \pi]$ is the angle between $\mathbf{x}$ and $\mathbf{x}^\star$. Then we begin our proof.

Theorem 1 *For the two-layer teacher-student network formulated as Eq. (9), the adversarial attack sets Eq. (10) and Eq. (15) as the loss function and the update rule, respectively. Assume that* $\mathbf{W}$ *and* $\mathbf{V}$ *follow independent standard Gaussian distribution,* $\mathbf{x}^\star \sim N(\mu_1, \sigma_1^2)$, $\mathbf{x}^{(0)} \sim N(0, \sigma_2^2)$, *and* η *is reasonably small* [4]. *Let* $\mathbf{x}^{(t)}$ *and* $\tilde{\mathbf{x}}^{(t)}$ *be the adversarial examples generated in the* t*-th iteration of attack using BP and LinBP, respectively, then we have*

$$\mathbb{E}\|\mathbf{x}^\star - \tilde{\mathbf{x}}^{(t)}\|_1 \leq \mathbb{E}\|\mathbf{x}^\star - \mathbf{x}^{(t)}\|_1.$$

[4] See Eq. (67) for more details of the constraint.

Proof We first analyse the property of the Eq. (51). Assume $\mathbf{x}^{(v)}$ is the first $\{\mathbf{x}^{(t)}\}$ to satisfy $|\mathbf{x}_i^{(v)} - \mathbf{x}_i^{(0)}| = \epsilon$, which also means $|\mathbf{x}_i^{(v-1)} - \mathbf{x}_i^{(0)}| < \epsilon$. If $\mathbf{x}_i^{(v)} = \mathbf{x}_i^{(0)} + \epsilon$, from Eq. (51), we have

$$\begin{aligned} x_i^{(v)} &= \text{Clip}(\mathbf{x}_i^{(v-1)} - \frac{\eta d_2}{2}(\mathbf{x}_i^{(v-1)} - \mathbf{x}_i^{\star})) \\ &= \text{Clip}((1 - \frac{\eta d_2}{2})\mathbf{x}_i^{(v-1)} + \frac{\eta d_2}{2}\mathbf{x}_i^{\star}) \\ &= \mathbf{x}_i^{(0)} + \epsilon. \end{aligned} \tag{55}$$

As $|\mathbf{x}_i^{(v-1)} - \mathbf{x}_i^{(0)}| < \epsilon$, we have $\mathbf{x}^{\star} > \mathbf{x}^{(0)} + \epsilon$. Therefore, for the $v+1$-th step, we have

$$\begin{aligned} x_i^{(v+1)} &= \text{Clip}(\mathbf{x}_i^{(v)} - \frac{\eta d_2}{2}(\mathbf{x}_i^{(v)} - \mathbf{x}_i^{\star})) \\ &= \text{Clip}((1 - \frac{\eta d_2}{2})\mathbf{x}_i^{(v)} + \frac{\eta d_2}{2}\mathbf{x}_i^{\star}) \\ &= \text{Clip}((1 - \frac{\eta d_2}{2})(\mathbf{x}_i^{(0)} + \epsilon) + \frac{\eta d_2}{2}\mathbf{x}_i^{\star}) \\ &= \mathbf{x}_i^{(0)} + \epsilon, \end{aligned} \tag{56}$$

note that the final step is established because $(1 - \frac{\eta d_2}{2})(\mathbf{x}_i^{(0)} + \epsilon) + \frac{\eta d_2}{2}\mathbf{x}_i^{\star} > \mathbf{x}_i^{(0)} + \epsilon$. Further we have $\mathbf{x}_i^{(t)} = \mathbf{x}_i^{(0)} + \epsilon$ for $\forall t > v$. If $\mathbf{x}_i^{(v)} = \mathbf{x}_i^{(0)} - \epsilon$, from Eq. (51), we have

$$\begin{aligned} x_i^{(v)} &= \text{Clip}((1 - \frac{\eta d_2}{2})\mathbf{x}_i^{(v-1)} + \frac{\eta d_2}{2}\mathbf{x}_i^{\star}) \\ &= \mathbf{x}_i^{(0)} - \epsilon. \end{aligned} \tag{57}$$

As $|\mathbf{x}_i^{(v-1)} - \mathbf{x}_i^{(0)}| < \epsilon$, we have $\mathbf{x}^{\star} < \mathbf{x}^{(0)} - \epsilon$. Therefore, for the $v+1$-th step, we have

$$\begin{aligned} x_i^{(v+1)} &= \text{Clip}((1 - \frac{\eta d_2}{2})\mathbf{x}_i^{(v)} + \frac{\eta d_2}{2}\mathbf{x}_i^{\star}) \\ &= \text{Clip}((1 - \frac{\eta d_2}{2})(\mathbf{x}_i^{(0)} - \epsilon) + \frac{\eta d_2}{2}\mathbf{x}_i^{\star}) \\ &= \mathbf{x}_i^{(0)} - \epsilon. \end{aligned} \tag{58}$$

Therefore we have for $\forall t > v$, $\mathbf{x}_i^{(t)} = \mathbf{x}_i^{(0)} - \epsilon$. In sum, for $\forall t > v$, we have $|\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(0)}| = \epsilon$. Similar conclusion can be for made for Eq. (52), therefore we can find that if a $\mathbf{x}^{(t)}$ (or $\tilde{\mathbf{x}}^{(t)}$) achieve the bound of Clip, the following steps will keep the bounded value. We then let $\mathbf{p}_j = \theta_j \mathbf{x}^{\star} - \frac{\|\mathbf{x}^{\star}\|}{\|\mathbf{x}^{(j)}\|} \sin\theta_j \mathbf{x}^{(j)}$, the l_1 distance bewteen $\mathbf{x}^{(t)}$ ($\tilde{\mathbf{x}}^{(t)}$) and $\mathbf{x}^{\star}$ can be formulated as

$$\begin{aligned} \mathbb{E}\|\mathbf{x}^{\star} - \mathbf{x}^{(t+1)}\|_1 &= \mathbb{E}\|\mathbf{x}^{\star} - \text{Clip}(\mathbf{x}^{(t)} - \eta \nabla_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)}))\|_1 \\ &= \mathbb{E}\| H\left((1 - \frac{\eta d_2}{2})(\mathbf{x}^{\star} - \mathbf{x}^{(t)}) + \frac{\eta d_2}{2\pi}\mathbf{p}_t\right) \|_1 \\ &= \mathbb{E}\| H\left((1 - \frac{\eta d_2}{2})^{t+1}(\mathbf{x}^{\star} - \mathbf{x}^{(0)}) + \sum_{j=0}^{t} \frac{\eta d_2}{2\pi}(1 - \frac{\eta d_2}{2})^{t-j}\mathbf{p}_j\right) \|_1, \end{aligned} \tag{59}$$

and

$$\begin{aligned} \mathbb{E}\|\mathbf{x}^{\star} - \tilde{\mathbf{x}}^{(t+1)}\|_1 &= \mathbb{E}\|\mathbf{x}^{\star} - \text{Clip}(\tilde{\mathbf{x}}^{(t)} - \eta \tilde{\nabla}_{\tilde{\mathbf{x}}}^{(t)} \mathcal{L}(\tilde{\mathbf{x}}^{(t)}))\|_1 \\ &= \mathbb{E}\| H\left((1 - \frac{\eta d_2}{2})^{t+1}(\mathbf{x}^{\star} - \tilde{\mathbf{x}}^{(0)})\right) \|_1, \end{aligned} \tag{60}$$

where $H(\cdot) = \min(\mathbf{x}^{\star} - \mathbf{x}^{(0)} + \epsilon\mathbf{1}, \max(\mathbf{x}^{\star} - \mathbf{x}^{(0)} - \epsilon\mathbf{1}, \cdot))$. From Eq. (59) and Eq. (60), further we have

$$\begin{aligned} \mathbb{E}\|\mathbf{x}^{\star} - \mathbf{x}^{(t+1)}\|_1 &= \mathbb{E}\| H\left((1 - \frac{\eta d_2}{2})^{t+1}(\mathbf{x}^{\star} - \mathbf{x}^{(0)}) + \sum_{j=0}^{t} \frac{\eta d_2}{2\pi}(1 - \frac{\eta d_2}{2})^{t-j}\mathbf{p}_j\right) \|_1 \\ &= \sum_{i=0}^{d} \mathbb{E}| H\left((1 - \frac{\eta d_2}{2})^{t+1}(\mathbf{x}_i^{\star} - \mathbf{x}_i^{(0)}) + \sum_{j=0}^{t} \frac{\eta d_2}{2\pi}(1 - \frac{\eta d_2}{2})^{t-j}\mathbf{p}_{ji}\right)_i |, \end{aligned} \tag{61}$$

and

$$\begin{aligned}\mathbb{E}\|\mathbf{x}^\star - \tilde{\mathbf{x}}^{(t+1)}\|_1 &= \mathbb{E}\|H\left((1-\frac{\eta d_2}{2})^{t+1}(\mathbf{x}^\star - \tilde{\mathbf{x}}^{(0)})\right)\|_1 \\ &= \sum_{i=0}^{d}\mathbb{E}|H\left((1-\frac{\eta d_2}{2})^{t+1}(\mathbf{x}_i^\star - \tilde{\mathbf{x}}_i^{(0)})\right)_i|.\end{aligned} \tag{62}$$

Note that $\mathbf{x}^{(0)} = \tilde{\mathbf{x}}^{(0)}$ in the theorem. If $|\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}| < \epsilon$, then for $\forall t$, $|\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}| < \epsilon$ and $|\mathbf{x}_i^\star - \tilde{\mathbf{x}}_i^{(0)}| < \epsilon$, which the Clip function can be removed in this case. Eq. (61) and Eq. (62) can be formulated as

$$\begin{aligned}\mathbb{E}\|\mathbf{x}^\star - \mathbf{x}^{(t+1)}\|_1 &= \sum_{i=0}^{d}\mathbb{E}|(1-\frac{\eta d_2}{2})^{t+1}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}) + \sum_{j=0}^{t}\frac{\eta d_2}{2\pi}(1-\frac{\eta d_2}{2})^{t-j}\mathbf{p}_{ji}| \\ &= \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta d_2}{2})^{t+1}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}) + \sum_{j=0}^{t}\frac{\eta d_2}{2\pi}(1-\frac{\eta d_2}{2})^{t-j}\mathbf{p}_{ji}\right)\operatorname{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}),\end{aligned} \tag{63}$$

and

$$\mathbb{E}\|\mathbf{x}^\star - \tilde{\mathbf{x}}^{(t+1)}\|_1 = \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta d_2}{2})^{t+1}(\mathbf{x}_i^\star - \tilde{\mathbf{x}}_i^{(0)})\right)\operatorname{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}). \tag{64}$$

If $|\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}| \geq \epsilon$, the sign of $H(\cdot)_i$ is determined by the sign of $\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}$. Therefore Eq. (61) and Eq. (62) can be formulated as

$$\begin{aligned}\mathbb{E}\|\mathbf{x}^\star - \mathbf{x}^{(t+1)}\|_1 &= \sum_{i=0}^{d}\mathbb{E}|(1-\frac{\eta N}{2})^{t+1}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}) + \sum_{j=0}^{t}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{t-j}\mathbf{p}_{ji}| \\ &= \sum_{i=0}^{d}\mathbb{E}H\left((1-\frac{\eta N}{2})^{t+1}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}) + \sum_{j=0}^{t}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{t-j}\mathbf{p}_{ji}\right)_i\operatorname{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}),\end{aligned} \tag{65}$$

and

$$\mathbb{E}\|\mathbf{x}^\star - \tilde{\mathbf{x}}^{(t+1)}\|_1 = \sum_{i=0}^{d}\mathbb{E}H\left((1-\frac{\eta N}{2})^{t+1}(\mathbf{x}_i^\star - \tilde{\mathbf{x}}_i^{(0)})\right)_i\operatorname{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}). \tag{66}$$

Recall the assumption that η is sufficiently small, to be exact, it should satisfy the following constraints,

$$|\sum_{j=0}^{m-1}\frac{\eta d_2}{2\pi}(1-\frac{\eta d_2}{2})^{m-1-j}\mathbf{p}_{ji}| < |(1-\frac{\eta d_2}{2})^{m}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)})|, \tag{67}$$

for $m = 1, \ldots, t$ and $i = 1, \ldots, d$. Under the constraints, we find the sign of $\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}$ determine the sign of $\left((1-\frac{\eta N}{2})^{t-j}\mathbf{p}_{ji}\right)_i$. Therefore, when we compare Eq. (63) and Eq. (64), Eq. (65) and Eq. (66), we actually compare the sign of $\mathbb{E}(\mathbf{p}_{ji}\operatorname{sgn}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})$. We let $\zeta^{(j)} = (1 - \frac{\eta d_2}{2} + \frac{\eta N\|\mathbf{x}^\star\|}{2\pi\|\mathbf{x}^{(j)}\|}\sin\theta_j)$ and $\beta^{(j)} = \frac{\eta d_2}{2}(1-\frac{\theta}{\pi})$. As $\theta_j \in [0, \pi]$ and $1 - \frac{\eta d_2}{2} > 0$, we have $\zeta^{(j)} > 0$ and $\beta^{(j)} > 0$ for $\forall j$. There we have

$$\begin{aligned}\mathbb{E}\mathbf{x}^{(t+1)} &= \mathbb{E}\left(\zeta^{(t)}\mathbf{x}^{(t)} + \beta^{(t)}\mathbf{x}^\star\right) \\ &= \left(\prod_{i=0}^{t}\zeta^{(i)}\right)\mathbf{x}^{(0)} + \left(\sum_{i=0}^{t}\prod_{j=i+1}^{t}\zeta^{(j)}\beta^{(i)}\right)\mathbf{x}^\star.\end{aligned} \tag{68}$$

We then solve the following equation,

$$\begin{aligned}\mathbb{E}\left(\mathbf{p}_{si}\text{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)})\right) =& \mathbb{E}\left((\theta_s\mathbf{x}_i^\star - \frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}^{(s)}\|}\sin\theta_s\mathbf{x}_i^{(s)})\text{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)})\right)\\ =& \mathbb{E}\left(\theta_s - \frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}^{(s)}\|}\sin\theta_s\sum_{i=0}^{s-1}\prod_{j=i+1}^{s-1}\zeta^{(j)}\beta^{(i)}\right)\mathbf{x}^\star\text{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}) -\\ & \mathbb{E}\left(\frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}^{(s)}\|}\sin\theta_s\prod_{i=0}^{t-1}\zeta^{(i)}\right)\mathbf{x}_i^{(0)}\text{sgn}(\mathbf{x}_i^\star - \mathbf{x}_i^{(0)}).\end{aligned} \tag{69}$$

We have $\theta_s - \frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}^{(s)}\|}\sin\theta_s\sum_{i=0}^{s-1}\prod_{j=i+1}^{s-1}\zeta^{(j)}\beta^{(i)} > 0$ and $\frac{\|\mathbf{x}^\star\|}{\|\mathbf{x}^{(s)}\|}\sin\theta_s\prod_{i=0}^{t-1}\zeta^{(i)} > 0$. Recall that $\mathbf{x}^\star \sim N(\mu_1, \sigma_1^2)$ and $\mathbf{x}^{(0)} \sim N(0, \sigma_2^2)$, using the Lemma 2, we have

$$\mathbb{E}(\mathbf{p}_{ji}\text{sgn}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)}) > 0. \tag{70}$$

With Eq. (70), we find Eq. (63) and Eq. (65) are greater than Eq. (64) and Eq. (66), respectively. That is to say, we have

$$\mathbb{E}\|\mathbf{x}^\star - \tilde{\mathbf{x}}^{(t+1)}\|_1 \leq \mathbb{E}\|\mathbf{x}^\star - \mathbf{x}^{(t+1)}\|_1\|_1.$$

A.2.4 Proof of Theorem 2

Theorem 2 *For the one-layer teacher-student network formulated as Eq. (16), the training task sets Eq. (17) and Eq. (20) as the loss function and the update rule, respectively. Assume that* $\mathbf{X}$ *is generated from standard Gaussian distribution,* $\mathbf{w}^\star \sim N(\mu_1, \sigma_1^2)$, $\mathbf{w}^{(0)} \sim N(0, \sigma_2^2)$, *and* η *is reasonably small*[5]. *Let* $\mathbf{w}^{(t)}$ *and* $\tilde{\mathbf{w}}^{(t)}$ *be the weight vectors obtained in the* t*-th iteration of training using standard BP and LinBP respectively. Then we have*

$$\mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t)}\|_1 \leq \mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t)}\|_1.$$

Proof. We first recall the partial gradient to $\mathbf{W}$ for BP and LinBP, which are formulated as

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \mathbf{X}^T\mathbf{D}(\mathbf{X}, \mathbf{w})(\mathbf{D}(\mathbf{X}, \mathbf{w})\mathbf{X}\mathbf{w} - \mathbf{D}(\mathbf{X}, \mathbf{w}^\star)\mathbf{X}\mathbf{w}^\star), \tag{71}$$

and

$$\tilde{\nabla}_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \mathbf{X}^T(\mathbf{D}(\mathbf{X}, \mathbf{w})\mathbf{X}\mathbf{w} - \mathbf{D}(\mathbf{X}, \mathbf{w}^\star)\mathbf{X}\mathbf{w}^\star), \tag{72}$$

respectively. From Theorem 1 in Tian (2017), we can calculate the expectation of Eq. (71) and Eq. (72) as

$$\mathbb{E}[\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w})] = \frac{N}{2}(\mathbf{w} - \mathbf{w}^\star) + \frac{N}{2\pi}\left(\theta\mathbf{w}^\star - \frac{\|\mathbf{w}^\star\|}{\|\mathbf{w}\|}\sin\theta\mathbf{w}\right), \tag{73}$$

where $\theta \in [0, \pi]$ is the angle between $\mathbf{w}$ and $\mathbf{w}^\star$, and

$$\mathbb{E}[\tilde{\nabla}_{\mathbf{w}}\mathcal{L}(\mathbf{w})] = \frac{N}{2}(\mathbf{w} - \mathbf{w}^\star). \tag{74}$$

The expectation of l_1 distance between $\mathbf{w}^{(t+1)}$ ($\tilde{\mathbf{w}}^{(t+1)}$) and $\mathbf{w}^\star$ can be formulated as

$$\begin{aligned}\mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t+1)}\|_1 &= \mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t)} + \eta\nabla_{\mathbf{w}^{(t)}}\mathcal{L}(\mathbf{w}^{(t)})\|_1\\ &= \mathbb{E}\|(1 - \frac{\eta N}{2})(\mathbf{w}^\star - \mathbf{w}^{(t)}) + \frac{\eta N}{2\pi}\left(\theta\mathbf{w}^\star - \frac{\|\mathbf{w}^\star\|}{\|\mathbf{w}^{(t)}\|}\sin\theta\mathbf{w}^{(t)}\right)\|_1,\end{aligned} \tag{75}$$

and

$$\begin{aligned}\mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t+1)}\|_1 &= \mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t)} + \eta\tilde{\nabla}_{\mathbf{w}^{(t)}}\mathcal{L}(\mathbf{w}^{(t)})\|_1\\ &= \mathbb{E}\|(1 - \frac{\eta N}{2})(\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t)})\|_1.\end{aligned} \tag{76}$$

[5]See Eq. (79) for a precious formulation of the constraint.

We assume $\mathbf{q}_j = \theta_j \mathbf{w}^\star - \frac{\|\mathbf{w}^\star\|}{\|\mathbf{w}^{(j)}\|} \sin\theta_j \mathbf{w}^{(j)}$. Therefore, Eq. (75) can be formulated as

$$\begin{aligned}
\mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t+1)}\|_1 &= \mathbb{E}\|(1-\frac{\eta N}{2})(\mathbf{w}^\star - \mathbf{w}^{(t)}) + \frac{\eta N}{2\pi}\mathbf{q}_t\|_1 \\
&= \mathbb{E}\|(1-\frac{\eta N}{2})^2(\mathbf{w}^\star - \mathbf{w}^{(t-1)}) + (1-\frac{\eta N}{2})\frac{\eta N}{2\pi}\mathbf{q}_{t-1} + \frac{\eta N}{2\pi}\mathbf{q}_t\|_1 \\
&= \mathbb{E}\|(1-\frac{\eta N}{2})^{t+1}(\mathbf{w}^\star - \mathbf{w}^{(0)}) + \sum_{j=0}^{t}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{t-j}\mathbf{q}_j\|_1.
\end{aligned} \tag{77}$$

And Eq. (76) can be formulated as

$$\begin{aligned}
\mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t+1)}\|_1 &= \mathbb{E}\|(1-\frac{\eta N}{2})(\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t)})\|_1 \\
&= \mathbb{E}\|(1-\frac{\eta N}{2})^{t+1}(\mathbf{w}^\star - \tilde{\mathbf{w}}^{(0)})\|_1.
\end{aligned} \tag{78}$$

Note that $\tilde{\mathbf{w}}^{(0)} = \mathbf{w}^{(0)}$. As mentioned in Theorem 1, we assume η is is sufficiently small, to be exact, it should satisfy the following constraints,

$$|\sum_{j=0}^{m}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{m-j}\mathbf{q}_{ji}| < |(1-\frac{\eta N}{2})^{m+1}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})|, \tag{79}$$

for $m = 1, \ldots, t$ and $i = 1, \ldots, d$. Under the constraints, $\mathbf{w}_i^\star - \mathbf{w}_i^{(0)}$ determine the sign of $\mathbf{w}_i^\star - \mathbf{w}_i^{(t+1)}$ in Eq. (77), then Eq. (77) and Eq. (78) can be calculated as

$$\begin{aligned}
&\mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t+1)}\|_1 \\
&= \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta N}{2})^{t+1}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)}) + \sum_{j=0}^{t}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{t-j}\mathbf{q}_{ji}\right)\mathrm{sgn}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)}) \\
&= \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta N}{2})^{t+1}|\mathbf{w}_i^\star - \mathbf{w}_i^{(0)}| + \sum_{j=0}^{t}\frac{\eta N}{2\pi}(1-\frac{\eta N}{2})^{t-j}\mathbf{q}_{ji}\mathrm{sgn}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})\right),
\end{aligned} \tag{80}$$

and

$$\begin{aligned}
\mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t+1)}\|_1 &= \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta N}{2})^{t+1}(\mathbf{w}_i^\star - \tilde{\mathbf{w}}_i^{(0)})\mathrm{sgn}(\mathbf{w}_i^\star - \tilde{\mathbf{w}}_i^{(0)})\right) \\
&= \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta N}{2})^{t+1}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})\mathrm{sgn}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})\right) \\
&= \sum_{i=0}^{d}\mathbb{E}\left((1-\frac{\eta N}{2})^{t+1}|\mathbf{w}_i^\star - \mathbf{w}_i^{(0)}|\right).
\end{aligned} \tag{81}$$

Similar to Theorem 2, using Lemma 2, we have

$$\mathbb{E}\left(\mathbf{q}_{ji}\mathrm{sgn}(\mathbf{w}_i^\star - \mathbf{w}_i^{(0)})\right) > 0. \tag{82}$$

With Eq. (82), we find Eq. (80) is greater than Eq. (81), i.e.,

$$\mathbb{E}\|\mathbf{w}^\star - \tilde{\mathbf{w}}^{(t+1)}\|_1 \leq \mathbb{E}\|\mathbf{w}^\star - \mathbf{w}^{(t+1)}\|_1.$$