Learning and Generalization in Overparameterized Normalizing Flows

Kulin Shah
Microsoft Research India

Amit Deshpande
Microsoft Research India

Navin Goyal
Microsoft Research India

Abstract

In supervised learning, it is known that overparameterized neural networks with one hidden layer provably and efficiently learn and generalize, when trained using stochastic gradient descent with a sufficiently small learning rate and suitable initialization. In contrast, the benefit of overparameterization in unsupervised learning is not well understood. Normalizing flows (NFs) constitute an important class of models in unsupervised learning for sampling and density estimation. In this paper, we theoretically and empirically analyze these models when the underlying neural network is a one-hidden-layer overparametrized network. Our main contributions are two-fold: (1) On the one hand, we provide theoretical and empirical evidence that for constrained NFs (this class of NFs underlies most NF constructions) with the one-hidden-layer network, overparametrization hurts training. (2) On the other hand, we prove that unconstrained NFs, a recently introduced model, can efficiently learn any reasonable data distribution under minimal assumptions when the underlying network is overparametrized and has one hidden-layer.

1 Introduction

Neural network models trained using gradient-based algorithms have been very effective in both supervised and unsupervised learning. This is surprising for two reasons: First, the optimization of training loss is typically non-smooth and non-convex and yet gradient-based methods often succeed in making the training loss very small. Second, even large neural networks whose number of parameters are more than the size of training data often generalize well on the unseen test data, instead of overfitting the seen training data. Recent work in supervised learning attempts to theoretically analyze these phenomena.

In supervised learning, the empirical risk minimization with quadratic or cross-entropy loss is a non-convex optimization problem even for one hidden layer fully connected network. In the last few years, it was realized that when the network is overparametrized, i.e. the hidden-layer size is large compared to the dataset size or some measure of complexity of the data, one can provably show efficient training and generalization for these networks. This hinges on the fact that overparametrization makes the optimization problem close to a convex one. See, e.g., Jacot et al. [2018], Du et al. [2018], Allen-Zhu et al. [2019], Zou et al. [2020], Arora et al. [2019].

The role of overparameterization and its effect on provable training and generalization guarantees for neural networks is far less understood in unsupervised learning. Generative modeling of a probability distribution when we are given samples drawn from that distribution is an important, classical problem in statistics and unsupervised learning. The goal of a generative model is to generate new samples from the distribution and give a probability density estimate at any queried point. Popular categories of generative models based on neural networks include Generative Adversarial Networks (GANs) Goodfellow et al. [2014], Variational AutoEncoders (VAEs) (e.g., Kingma and Welling [2014]), and Normalizing Flows (NFs) (e.g., Rezende and Mohamed [2015]). All categories of models, especially GANs, have shown an impressive capability to generate samples of photo-realistic images but GANs and VAEs cannot give probability density estimates for new data points. All categories present various challenges in training such as mode collapse, posterior collapse, training instability, etc., e.g., Bowman et al. [2016], Salimans et al. [2016], Arora et al. [2018], Lucic et al. [2018].

Unlike GANs and VAEs, NFs can do both sampling and density estimation, leading to a potentially wider range of applications; see, e.g., the surveys Kobyzev et al.

Proceedings of the 25$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[2020], Papamakarios et al. [2019]. Theoretical understanding of learning and generalization in generative models remains a natural and important open question even after some recent work (Buhai et al. [2020], Kong and Chaudhuri [2020], Koehler et al. [2020], Lee et al. [2021]). Appendix J contains further literature review. In this paper, we focus on the theoretical analysis of NFs. For *constrained* NFs which underlies a large class of NF constructions, we show that theoretical analysis in the overparametrized regime runs into difficulties. This is also seen in experiments where overparametrization hurts the performance of constrained NFs in many settings. In contrast, a recent class of NFs called *unconstrained* NFs, admits provable training and generalization guarantees in the overparametrized setting. Before stating our contributions in detail, we introduce NFs followed by a very brief discussion of overparametrized supervised learning to provide the necessary context.

Normalizing Flows. The general idea behind normalizing flows (NFs) is as follows: let $X \in \mathbb{R}^d$ be a random variable coming from the data distribution and $Z \in \mathbb{R}^d$ be a random variable associated with *base* distribution which can be the standard Gaussian or exponential distribution. Given i.i.d. samples of X, the goal is to learn a differentiable invertible map $f_X : \mathbb{R}^d \to \mathbb{R}^d$ that transports the distribution of X to the distribution of Z: in other words, the distribution of $f_X^{-1}(Z)$ and X are same. (We tacitly assume that the distribution of X is nice enough to allow for the existence of f_X.) We assume that function f_X is autoregressive, means f_X is of the form $f_X(x) = \big(f_{X,1}(x_1), f_{X,2}(x_{1:2}), \ldots, f_{X,d}(x_{1:d})\big)$ where $f_{X,i} : \mathbb{R}^i \to \mathbb{R}$ and $x_{1:i}$ is first i dimension of a data sample x from X (i.e., if $x = (x_1, x_2, \ldots, x_d)$, then $x_{1:i} = (x_1, \ldots, x_i)$). The nice thing about autoregressive functions is that their invertibility is easily ensured by making $f_{X,i}(x_{1:i})$ a strictly monotonically increasing function in x_i for any fixed value of $x_{1:(i-1)}$. We will call such an f *monotonic autoregressive function.* Such a function is also called a Knothe–Rosenblatt map and is known to exist and be unique under very general conditions sufficient for our purposes, in particular for any pair of probability measures on $\mathbb{R}^d$ with density; see Chapter 2 in Santambrogio [2015].

Learning of f_X is done by representing a monotonic autoregressive map f by neural networks, setting up an appropriate loss function, and doing gradient-based training with the aim of achieving $f = f_X$. A number of approaches have been suggested for carrying out this general plan. We distinguish between two classes of approches: (1) *Represent f directly using neural networks.* In this approach there are d neural networks $N_1, \ldots, N_d$ with $f_i(x_{1:i}) = N_i(x_{1:i})$. Since the functions represented by standard neural networks are not necessarily monotone, the design of the neural network is *constrained* to make it monotone. For example, if $\{a_r, w_r, b_r\}_{r=1}^m$ are the parameters of the neural networks, with $a_r, w_r, b_r \in \mathbb{R}$ for each r, and ρ is a monotonically increasing activation function, then the univariate one-hidden layer network of the form $\sum_{r=1}^m a_r \, \rho\,(w_r x + b_r)$ can be made monotonically increasing by ensuring positivity of a_r and w_r. This can be done in multiple ways: for example, instead of a_r, w_r, one can use a_r^2, w_r^2 in the above expression; see, e.g., [Huang et al., 2018, Cao et al., 2019a]. (2) *Represent the Jacobian matrix $\frac{\partial f(x)}{\partial x}$ using neural networks.* In this approach, we model diagonal entries of the Jacobian by neural networks $\frac{\partial f_i(x_{1:i})}{\partial x_i} = \phi(N_i\,(x_{1:i}))$ where $\phi : \mathbb{R} \to \mathbb{R}^+$ takes on only positive values. Positivity of $\frac{\partial f_i(x_{1:i})}{\partial x_i}$ implies monotonicity of $f_i\,(x_{1:i})$ with respect to x_i. Note that the parameters are *unconstrained* in this approach. This approach is used by Wehenkel and Louppe [2019].

We will refer to the models in the first class as *constrained normalizing flows* (**CNFs**) and those in the second class as *unconstrained normalizing flows* (**UNFs**).

Most existing analyses for overparametrized neural networks in the supervised setting consider a linear approximation of the neural network, termed *pseudo-network* in Allen-Zhu et al. [2019]. The convexity property of loss function for pseudo-network and closeness between neural network and pseudo network help in proving convergence and generalization of neural network.

1.1 Our Contributions

In this paper, we study both CNFs and UNFs theoretically when the underlying network has one hidden-layer and empirically validate our theoretical findings. We now describe our contributions.

Architectural variants. The practical CNF and UNF architectures can be quite detailed involving multiple layer neural networks and stacking of flows. It is difficult to get a theoretical handle on such models—presently there are no satisfactory results even for two-hidden layers networks in the supervised learning setting. In this paper, we identify very simple and natural NF models (gleaned from the existing architectures) reducing the architecture to the essentials and yet providing satisfactory results in experiments. These models are the starting point of our analyses. A natural approach to analyze NFs is to adapt the successful techniques from supervised learning to NFs. While there is a natural definition of pseudo-network in the case of CNFs, for UNFs this is not clear. We are able to define linear approximations of the neural network to analyze the training of both CNFs and UNFs. However, one immediately encounters some new roadblocks: the

loss surface of the pseudo-networks is non-convex in both CNFs and UNFs for the simple NF models mentioned above. Therefore, analyzing pseudo-networks still remains difficult. Barring a major breakthrough in non-convex optimization for deep learning, one way to proceed is to find architectural variants of simple NFs that may lead to pseudo-networks with convex optimization problems without adverse effect on their empirical performance. We follow this path and identify novel variations that make the optimization problem for associated pseudo-network convex. It is pertinent that our variations are arguably natural.

Architectural variants for CNFs. To resolve the non-convexity arising from using a_r^2, w_r^2 as parameters, we simply impose the constraints $a_r \geq \epsilon$ and $w_r \geq \epsilon$ for all $r \in [m]$ where $[m] = \{1, \ldots, m\}$. To solve this constrained optimization problem, we use projected SGD, which in this case incurs essentially no extra cost over SGD due to the simplicity of the constraints. In our experiments, this variation slightly *improves* the training of NFs compared to the reparametrization approach mentioned above and may be of a separate interest in practical settings.

Architectural variants for UNFs. Similarly, for UNFs we identify two problems in the model of Wehenkel and Louppe [2019] that make the theoretical analysis difficult. We resolve these as follows: (1) Change in numerical integration method. Instead of Clenshaw–Curtis quadrature method for numerical integration employed in Wehenkel and Louppe [2019], we use the simple rectangle quadrature. This change makes the model slightly slower (in our experiments, it typically uses twice as many samples and time to get similar performance). (2) Change in the base distribution. We use the exponential distribution as the base distribution instead of the standard Gaussian distribution. In experiments, this does not cause any changes in performance. Note that NFs require *only* efficient sampling and density estimation from the base distribution but the Gaussian is far from the only distribution to have those properties.

Our results about these variants point to a dichotomy between these two classes of NFs:

Overparametrization hurts CNFs. Our theoretical findings provide evidence that overparametrization makes training slower. To be more precise, we show that in a bounded number of training iterations or for bounded change in weights such that neural networks and pseudo networks are close, overparameterized CNFs can not learn the target function. We also point out the reasons that lead overparametrization to adversely affect the training of CNFs. Our experimental results also validate our theoretical results and confirm that overparameterization in CNF makes training slower. Note that in supervised learning, it is known that overparameterization makes training faster [Neyshabur et al., 2015, Allen-Zhu et al., 2019]. Therefore, the finding that overparametrization is significantly detrimental to CNFs is *novel* and we are not aware of *any other* settings where overparametrization has such a strong negative effect. Thus, for theoretical analysis of CNFs, one must work with moderate-sized networks. But this is likely to be difficult as analysis of such networks has remained open even for supervised learning leading us to a "barrier".

Analysis of overparametrized UNFs. We theoretically analyze UNFs and prove that overparameterized networks for UNFs indeed learn the data distribution. To our knowledge, this is the *first "end-to-end"* analysis of an NF model—and in fact for *any* neural generative model using gradient-based algorithms for a sufficiently large class of distributions (please see Appendix J for additional extensive related work). This proof, while following the high-level scheme of supervised learning proofs, requires several new ideas, conceptual as well as technical, due to different settings and will be discussed in the sequel.

To summarize, our contributions include:

- We identify difficulties in the theoretical analysis of existing NF models. We resolve these by proposing new versions of these models without loss of experimental efficacy.
- We identify a "barrier" to the training convergence and generalization analysis of CNFs: overparametrization is detrimental to CNFs.
- We provide efficient training convergence and generalization analysis for UNFs. To our knowledge, this is the *first* result on training and generalization of NFs.
- We experimentally validate our theoretical claims.

Paper outline. Sec. 2 contains preliminaries, Sec. 3 contains our results on CNFs and Sec. 4 contains results on UNFs. Sec. 5 briefly describes our empirical studies. We conclude in 6. Appendix A contains outline of the appendix.

2 Preliminaries

In this section, we will continue our description of the problem of learning probability distributions using NFs and introduce necessary notation.

2.1 Problem of learning distributions in Normalizing Flows

Recall that the goal of NFs is to learn a probability distribution given via i.i.d. samples from the distribution. Let X be the random variable corresponding to the data distribution we want to learn. We denote the probability density (we often just say density) of X at $u \in \mathbb{R}^d$ by $p_X(u)$. We will work with distributions whose densities have a finite support.[1] We will furthermore assume $p_X(u) = 0$ when $\|u\|_2 \geq 1$, without loss of generality. Let Z be a random variable with either standard Gaussian or the *standard exponential distribution*. There seems to be no well-accepted definition of multidimensional exponential distribution; for our purposes the following natural definition will serve well. The density of the standard exponential distribution at $z = (z_1, z_2, \ldots, z_d) \in \mathbb{R}^d$ is given by $e^{-\sum_{i=1}^d z_i}$ when all $z_i \geq 0$, and by 0, otherwise. We will refer to the distribution of Z as the *base* distribution.

Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be monotonic autoregressive as defined previously; thus, f is invertible. Let $p_{f,Z}(\cdot)$ be the density of the random variable $f^{-1}(Z)$. Let $z = f(x)$. Then the standard change of density formula using the invertibility of f gives

$$p_{f,Z}(x) = p_Z(f(x)) \left| \det \left(\frac{\partial f(x)}{\partial x} \right) \right|. \tag{2.1}$$

We would like to choose f so that $p_{f,Z} = p_X$. As mentioned before, such an f always exists and is unique and we will denote it by F^*. If we can find F^*, then we can generate samples of X using $F^{*-1}(Z)$ since generating the samples of Z is easy and so is the inversion of F^* using monotonic autoregressive property. Similarly, we can evaluate density $p_X(x)$ using standard change of variable with F^* because $p_{F^*,Z}(x) = p_X(x)$. To find F^* from the data, we set up the maximum log-likelihood objective:

$$\begin{aligned} &\max_f \tfrac{1}{n} \sum_{x \in \mathcal{X}} \log p_{f,Z}(x) \\ &= \max_f \tfrac{1}{n} \big[\sum_{x \in \mathcal{X}} \log p_Z(f(x)) + \sum_{x \in \mathcal{X}} \log \big(\det \big(\tfrac{\partial f(x)}{\partial x} \big) \big) \big], \end{aligned} \tag{2.2}$$

where training set $\mathcal{X} \subset \mathbb{R}^d$ contains n i.i.d. samples of X, and the maximum is over differentiable invertible functions. When Z is standard exponential and f is monotonic autoregressive, then (2.2) simplifies to

$$\begin{aligned} &\min_f L(f, \mathcal{X}) = \tfrac{1}{n} \sum_{x \in \mathcal{X}} L(f, x) \text{ and} \\ &L(f, x) = \sum_{i=1}^d \big(f_i(x_{1:i}) - \log \big(\tfrac{\partial f_i(x_{1:i})}{\partial x_i} \big) \big). \end{aligned} \tag{2.3}$$

We denote average loss by $L(f, \mathcal{X}) = \frac{1}{n} \sum_{x \in \mathcal{X}} L(f, x)$. Informally, we expect that as $n \to \infty$, the optimum f_n in the above optimization problem satisfies $p_{f_n,Z} \to p_X$. To make the above optimization problem tractable, instead of f we work with d neural networks $N_1, N_2, \ldots, N_d$ as previously touched upon in our brief description of CNFs and UNFs. All our networks will have one hidden layer with the following basic form:

$$N(x; \theta) = \sum_{r=1}^m \bar{a}_r \, \rho(\langle \bar{w}_r + w_r, x \rangle + (\bar{b}_r + b_r)).$$

Here m is the size of the hidden layer, ρ is a strictly increasing activation function, the weights $\bar{a}_r, \bar{w}_r, \bar{b}_r$ are the initial weights chosen at random according to some distribution specified later, and w_r, b_r are offsets from the initial weights. We only train w_r and b_r, and the outer weights remain frozen at their initial values. Let $\bar{\theta} = (\bar{w}_1, \ldots, \bar{w}_m; \bar{b}_1, \ldots, \bar{b}_m)$ denote the vector of initial parameters and similarly $\theta = (w_1, \ldots, w_m; b_1, \ldots, b_m)$ denote the matrix of offsets from the initial weights. Similarly, we denote offsets at time step t by $\theta^{(t)}$ and the corresponding network by $N^{(t)}(x)$ or $N(x; \theta^{(t)})$.

2.2 Supervised learning analysis

We now very briefly outline a proof technique for analyzing training and generalization for one-hidden layer neural networks for supervised learning (e.g. Allen-Zhu et al. [2019]). For simplicity, we restrict the discussion to the realizable setting. Data $x \in \mathbb{R}^d$ is generated by some distribution D and the labels $y = h(x)$ are generated by some unknown function $h : \mathbb{R}^d \to \mathbb{R}$. The function h is assumed to have small "complexity" C_h which (informally speaking) measures the required size of a one-hidden-layer neural network with smooth activations to approximate h. The loss function is the square loss on the training set $\mathcal{X}$, that is, $L_s(N^{(t)}, \mathcal{X}) = \frac{1}{n} \sum_{x \in \mathcal{X}} L_s(N^{(t)}, x)$ with $L_s(N^{(t)}, x) = (N(x; \theta^{(t)}) - y)^2$. The training is done using SGD to update the parameters θ of the neural network.

The problem of optimizing the square loss is non-convex even for one-hidden layer networks. One instead works with the *pseudo-network* $P(x; \theta)$ which is the linear

[1] This is often without any real loss of generality because, for most purposes, light-tailed distribution (e.g., the Gaussian distribution) can be assumed to have a finite support. (Exception to this are heavy-tailed distributions which are seldom encountered; we believe our work here could be extended to deal with such distributions too.)

approximation of $N(x;\theta)$:

$$P(x;\theta) = \sum_{r=1}^{m} \bar{a}_r(\rho(\langle \bar{w}_r, x\rangle + \bar{b}_r) + \rho'(\langle \bar{w}_r, x\rangle + \bar{b}_r)\left(\langle w_r, x\rangle + b_r\right)).$$

Similarly to $N^{(t)}$ and $N(x;\theta^{(t)})$, we can also define $P^{(t)}$ and $P(x;\theta^{(t)})$ with parameters $\theta^{(t)}$. When the network is overparameterized, i.e. the network size m is sufficiently large compared to C_h, and the learning rate is small ($\eta = O(1/m)$), SGD iterates when applied to $L_s(N^{(t)}, x^{(t)})$ and $L_s(P^{(t)}, x^{(t)})$ remain close throughout. Moreover, the problem of optimizing $L_s(P^{(t)}, \mathcal{X})$ is a convex problem in $\theta^{(t)}$ for all t and thus can be analyzed with the existing methods. An *approximation* theorem then states that there exist parameters θ^* with small norm such that the pseudo-network with parameters θ^* is close to the target function. This together with the analysis of SGD shows that the pseudo-network, and hence the neural network too, achieves small training loss. Then by a Rademacher complexity argument that the neural network after $T = O(C_h/\epsilon^2)$ time steps has population loss within ϵ of the optimal loss, thus obtaining a generalization result.

3 Constrained Normalizing Flow

In this section, we will first describe problems in analyzing current CNF architectures. Then, we will describe a new architectural variant which is easy to analyze and our theoretical result on CNF.

3.1 Problems in analyzing CNF architectures

In CNFs, monotonic autoregressive functions $f(x) = (f_1(x_{1:1}), f_2(x_{1:2}), \ldots, f_d(x_{1:d}))$ are represented by d neural networks via $f_i(x_{1:i}) = N_i(x_{1:i}) = N(x_{1:i};\theta_i)$ where $N(x_{1:i};\theta_i)$ is given by

$$N(x_{1:i};\theta_i) = \tau \sum_{r=1}^{m} \bar{a}_{i,r}\, \rho(\langle \bar{w}_{i,r} + w_{i,r}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right)),$$

where τ is a normalization constant chosen to compensate for the effect of overparameterization. We use θ_i to denote parameters of $N_i(x_{1:i})$ and θ to denote parameters of all neural networks. To make $f_i(x_{1:i})$ monotonically increasing in x_i for each fixed $x_{1:i-1}$, we ensure that $\bar{a}_{i,r,i} \geq 0$, $\bar{w}_{i,r,i} + w_{i,r,i} \geq 0$ for all r. One way to do this is by replacing $\bar{a}_{i,r}$ and $\bar{w}_{i,r,i} + w_{i,r,i}$ by their functions that take on only positive values. For example, the square function would give us the neural network

$$N_i(x_{1:i}) = \tau \sum_{r=1}^{m} \bar{a}_{i,r}^2\, \rho(\langle \zeta\left(\bar{w}_{i,r} + w_{i,r}\right), x_{1:i}\rangle + \bar{b}_{i,r} + b_{i,r}),$$

where $\zeta : \mathbb{R}^i \to \mathbb{R}^i$ is given by $\zeta(y_1, y_2, \ldots, y_i) = \left(y_1, \ldots, y_{i-1}, y_i^2\right)$. After reparameterization, parameters have no constraints, and so this network can be trained using SGD. But we need to specify the (monotone) activation ρ to complete our description of CNF.

Activation function. Unlike supervised learning, the choice of the activation function needs more care for CNFs as we will now see. Let $\sigma(x)$ denote the ReLU activation. If we choose $\rho = \sigma$, then in (2.3) we have

$$\frac{\partial f_i(x_{1:i})}{\partial x_i} = \tau \sum_{r=1}^{m} \bar{a}_{i,r}^2\, (\bar{w}_{i,r,i} + w_{i,r,i})^2 \mathbb{I}[\langle \zeta(\bar{w}_{i,r} + w_{i,r}), x_{1:i}\rangle + \bar{b}_{i,r} + b_{i,r} \geq 0].$$

The derivative $\frac{\partial f_i(x_{1:i})}{\partial x_i}$ and consequently $\log(\det(\frac{\partial f(x)}{\partial x}))$ are discontinuous functions of x and θ. Gradient-based optimization algorithms are not applicable to problems with discontinuous objectives, and indeed this is reflected in experimental failure of such models. By the same argument, any activation with a discontinuous derivative is not admissible. Convex activations with continuous derivative (e.g. ELU(x)) also cannot be used because then $N(x_{1:i};\theta_i)$ is also a convex function of x_i, which need not be the case for the optimal f. Hence in such cases, $N(x_{1:i};\theta_i)$ can not approximate f. To our knowledge, among the commonly used activations tanh (and the closely-related sigmoid) is the only one that does not suffer from either of these defects and also works well in practice Cao et al. [2019b].

Non-convexity of pseudo-network. Pseudo-network with activation tanh is given by

$$P(x_{1:i};\theta_i) = \tau \textstyle\sum_{r=1}^{m} \bar{a}_{i,r}^2 \big(\tanh(\langle \zeta\left(\bar{w}_{i,r}\right), x_{1:i}\rangle + \bar{b}_{i,r}) + \tanh'(\langle \zeta\left(\bar{w}_{i,r}\right), x_{1:i}\rangle + \bar{b}_{i,r})\, (\langle \zeta\left(\bar{w}_{i,r} + w_{i,r}\right) - \zeta\left(\bar{w}_{i,r}\right), x_{1:i}\rangle + b_{i,r})\big).$$

Note that $P(x_{1:i};\theta_i)$ is not linear in $w_{i,r}$. Hence, it is not obvious that the loss function for the pseudo-network will remain convex in parameters; indeed, non-convexity can be confirmed in experiments.

3.2 A variant of CNF architecture

To overcome the non-convexity issue, we propose another formulation of CNFs. Here we use standard form of the neural network, but ensure the constraints $\bar{a}_{i,r} > 0$ and $\bar{w}_{i,r,i} > 0$ by the choice of the initialization distribution and $\bar{w}_{i,r,i} + w_{i,r,i} \geq \epsilon$ by using *projected SGD* for optimization.

$$N(x_{1:i};\theta_i) = \tau \sum_{r=1}^{m} \bar{a}_{i,r} \tanh\left(\langle \bar{w}_{i,r} + w_{i,r}, x_{1:i}\rangle + (\bar{b}_{i,r} + b_{i,r})\right),$$

with constraints $\bar{w}_{i,r,i} + w_{i,r,i} \geq \epsilon$, for all r.

$\epsilon > 0$ is a small constant to ensure strict monotonicity of $N(x_{1:i};\theta_i)$. These constraints are very simple and

projected SGD incurs very little overhead. The pseudo-network in this formulation is given by

$$P(x_{1:i};\theta_i) = P_c(x_{1:i}) + P_\ell(x_{1:i};\theta_i)$$

with constraints $\bar{w}_{i,r,i} + w_{i,r,i} \geq \epsilon$ for all r, where

$$P_c(x_{1:i}) = \tau \sum_{r=1}^{m} \bar{a}_{i,r} \tanh(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}) \quad \text{and}$$
$$P_\ell(x_{1:i};\theta_i) = \tau \sum_{r=1}^{m} \bar{a}_{i,r} \tanh'(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}) \left(\langle w_{i,r}, x_{1:i}\rangle + b_{i,r}\right).$$

Pseudo-network $P(x_{1:i};\theta_i)$ is linear in θ_i, therefore the objective in (2.3) with f_i replaced by $P(x_{1:i};\theta_i)$ is convex in θ_i and hence, in θ. Note that $P_c(x_{1:i})$ does not change during training, therefore $P_\ell(x_{1:i};\theta_i)$ must approximate the target function with $P_c(x_{1:i})$ subtracted.

3.3 Theoretical analysis of CNF

Our results for CNFs are negative: we identify barriers in the analysis of highly over-parameterized CNFs and show that surmounting these barriers entails analyzing moderately overparameterized neural networks—a long-open problem even in supervised learning. Let F^* denote the target function and $C(F^*)$ denote some complexity measure of F^*. Initial weights $\bar{a}_{i,r}$ and $\bar{w}_{i,r,i}$ are sampled from *half-normal* distribution with parameters $\left(0,\epsilon_a^2\right)$ and $\left(0,\sigma_{wb}^2\right)$, respectively. The half-normal random variable Y with parameters $\left(\mu,\sigma^2\right)$ is given by simply $\left|Y'\right|$ where $Y' \sim \mathcal{N}\left(\mu,\sigma^2\right)$. Here $\mathcal{N}\left(\mu,\sigma^2\right)$ denote the Gaussian distribution with mean μ and variance σ^2. The bias term $\bar{b}_{i,r}$ is sampled from $\mathcal{N}\left(0,\sigma_{wb}^2\right)$. We divide our analysis into two cases based on the value of σ_{wb}: (1) σ_{wb} is between $\frac{1}{\sqrt{m}}$ and $\frac{\epsilon}{C(F^*)\sqrt{\log(md)}}$, (2) σ_{wb} is between $\frac{\epsilon}{C(F^*)\sqrt{\log(md)}}$ and 1. In case (1) we have:

Theorem 3.1. *For any $\epsilon > 0$, for any $i \in [d]$, any hidden layer size $m \geq \Omega\big(\text{poly}\left(C(F^*), \frac{1}{\epsilon}\right)\big)$, by choosing learning rate $\eta = O\big(\frac{\epsilon}{m\tau\epsilon_a^2 \log m}\big)$ and $T = O(\frac{C(F^*)}{\epsilon^2})$, with at least probability 0.9, there exist constants $\alpha_i \in \mathbb{R}^i$ and $\beta \in \mathbb{R}$ for which projected SGD after T iterations gives*

$$|N(x_{1:i};\theta_i^{(T)}) - (\langle \alpha_i, x_{1:i}\rangle + \beta)| \leq O(\epsilon), \qquad (3.1)$$

for all x with $\|x\|_2 \leq 1$.

Theorem 3.1 tells us that if we choose η and T as suggested in the theorem statement then the function learned by overparametrized neural networks is close to a linear function. Recall from Sec. 2.2 that choosing similar values of η and T in supervised learning enables the provable successful training of the neural network. The same issue in approximation arises for *all* activations with continuous derivative. More details about case (1) is given in Appendix H. The result in case (2) is given by the next theorem.

Theorem 3.2. *For any constant $c > 0$ and any $\eta > 0$, $T > 1$, if norm of change in parameters $\|\theta^{(T)}\|_{1,2} \leq O(\frac{1}{\epsilon_a \sigma_{wb} \tau m^c \log m})$, then for all $i \in [d]$ and for all x with $\|x\|_2 \leq 1$, we have*

$$|P_\ell(x_{1:i};\theta_i^{(T)})| \leq O\big(\frac{1}{\sigma_{wb} m^c \sqrt{\log(md)}}\big).$$

Most extant theoretical analyses require that the change in weights from initialization is small so that the pseudo-network remains close to the neural network. Small change implies $|P_\ell(x_{1:i};\theta_i^{(T)})| = O(\frac{1}{m^c})$ for some constant $c > 0$. Therefore, $P_\ell(x;\theta^{(T)})$ can not in general approximate the target function (with $P_c(x_{1:i})$ subtracted). And the same happens with $N(x_{1:i};\theta_i^{(T)})$ because it is close to $P(x_{1:i};\theta_i^{(T)})$. More details about case (2) is provided in Appendix H.

We also show the negative effect of overparameterization for CNF in experiments (Section 5).

4 Unconstrained Normalizing Flow

In this section, we first describe our UNF model that we analyze and then present our main theoretical result on training and generalization of the UNF model.

4.1 Our UNF model

Unlike the constrained case, where we model $f(x)$ using neural networks, here we model the Jacobian $\frac{\partial f(x)}{\partial x}$ using d neural networks by setting

$$\frac{\partial f_i(x_{1:i})}{\partial x_i} = \phi(N(x_{1:i};\theta_i)),$$

where ϕ is $\mathsf{ELU} + 1$ function given by

$$\phi(u) = e^u \mathbb{I}[u < 0] + (u+1)\,\mathbb{I}[u \geq 0] \quad \text{and}$$
$$N(x_{1:i};\theta_i) = \sum_{r=1}^{m} \bar{a}_{i,r}\, \rho(\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right))$$

with $\rho = \mathsf{ReLU}$. In the expression for $N(x_{1:i};\theta_i)$ instead of $x_{1:i}$, we use $\tilde{x}_{1:i} \in \mathbb{R}^{i+1}$ to aid in analysis; the extra coordinate is added to make $\|\tilde{x}_{1:i}\|_2 = 1$. No normalization factor is needed in the expression for $N(x_{1:i};\theta_i)$ because of the choice of initialization distribution specified later. We can reconstruct f by integration:

$$f_1(x_{1:1}) = \int_{-1}^{x_1} \frac{\partial f_1(t)}{\partial t} \mathrm{d}t \quad \text{and}$$
$$f_i(x_{1:i}) = \int_{-1}^{x_i} \frac{\partial f_i(x_1, x_2, \ldots, x_{i-1}, t)}{\partial t} \mathrm{d}t$$

for $i \in [d]$. The lower limit in our integral is -1 because $\|x\|_2 \leq 1$ by our assumption on the support of the data distribution. We also denote $\frac{\partial f_i(x_{1:i})}{\partial x_i}$ by $\nabla_i f_i(x_{1:i})$. The monotonicity of f is achieved by ensuring that $\nabla_i f_i(x_{1:i})$ is positive for all x. Although positivity was the only useful property of ϕ mentioned by Wehenkel and Louppe [2019], it turns out to have several other properties which we will exploit in our proof: it is 1-Lipschitz and increasing, its derivative is 1-Lipschitz, and its second derivative is non-negative (except at 0, where it's not defined).

Quadrature. To reconstruct f, from the Jacobian we need to evaluate the integrals. While this cannot be done exactly, good approximation can be obtained via numerical integration (also known as quadrature). We estimate $f_i(x_{1:i})$ via the general quadrature formula by

$$\tilde{f}_i(x_{1:i}) = \sum_{j=1}^{Q} q_j \nabla_i f_i\left(\tau_j(x_{1:i})\right).$$

Here, Q is the number of quadrature points and the $q_1, \ldots, q_Q$ are the corresponding coefficients. We use simple rectangle quadrature, which arises in Riemann integration, and uses only positive coefficients with $q_j = \Delta_{x_i} := \frac{x_i+1}{Q}$ and $\tau_j(x_{1:i}) = (x_1, \ldots, x_{i-1}, -1 + j\Delta_{x_i})$.

Wehenkel and Louppe [2019] uses Clenshaw–Curtis quadrature where the coefficients q_i can be negative. Compared to Clenshaw–Curtis quadrature, the rectangle quadrature requires more points for similar accuracy (about doubling the number of quadrature points in our experiments). This is a small price to pay because rectangle quadrature makes the problem of minimizing the loss of the pseudo-network (defined shortly) easier to analyze via the positivity of the quadrature coefficients.

Exponential base distribution. Taking the standard Gaussian as a base distribution as in Wehenkel and Louppe [2019] causes two difficulties: it is not clear that the loss function in the pseudo-network is convex (see Remark F.2). Moreover, it is not clear that throughout training the Lipschitz constant of the loss function will remain bounded by an absolute constant and hence independent of the parameters. (This issue also arises in supervised learning, e.g. Allen-Zhu et al. [2019], though the authors seem to have not realized the problem and do not address it.) Both of these difficulties with the Gaussian can be circumvented by using the exponential as the base distribution. This does not cause any negative effects in our experiments.

Learner network parameterization and training procedure. We initialize $\bar{a}_{i,r} \sim \mathcal{N}(0, \epsilon_a^2)$, $\bar{w}_r \sim \mathcal{N}(0, \frac{1}{m})$ and $\bar{b}_r \sim \mathcal{N}(0, \frac{1}{m})$, where $\epsilon_a = O(\frac{\epsilon}{\log m})$ is a small constant. Additionally, using the estimates $\tilde{f}_i(x_{1:i})$, we get approximate loss function

$$\tilde{L}(\nabla f, x) = \sum_{i=1}^{d} \tilde{f}_i(x_{1:i}) - \sum_{r=1}^{d} \log\left(\nabla_i f_i(x_{1:i})\right).$$

Define average approximated loss as $\tilde{L}(\nabla f, \mathcal{X}) = \frac{1}{n}\sum_{x\in\mathcal{X}} \tilde{L}(\nabla f, x)$ and expected approximated loss ass $\tilde{L}(\nabla f, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\tilde{L}(\nabla f, x)$. The parameters of neural networks are updated using SGD:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \tilde{L}(\nabla f, x^{(t)})$$

where $\nabla_i f_i = \phi(N(x_{1:i}; \theta_i^{(t)}))$, and $x^{(t)} \in \mathcal{X}$ is chosen uniformly at random from the training set at each step. We assume that our data is generated from a target function $F^* = \left(F_1^*(x_{1:1}), F_2^*(x_{1:2}), \ldots, F_d^*(x_{1:d})\right)$, where $F_i^* : \mathbb{R}^i \to \mathbb{R}$. Thus, $F^{*-1}(Z) = X$.

Target function class. We consider target functions whose derivative are given by

$$\frac{\partial F_i^*(x_{1:i})}{\partial x_i} = \phi\left(\sum_{r=1}^{p_i} \mu_{i,r}^* \psi_{i,r}(\langle u_{i,r}^*, \tilde{x}_{1:i}\rangle\right)$$

where $|\mu_{i,r}^*| \leq 1, \|u_{i,r}^*\|_2 \leq 1$ for all $i \in [d]$ and $\psi_{i,r} : \mathbb{R} \to \mathbb{R}$ are smooth functions with Taylor expansion and p_i are positive integers. Our target function class is rich: the argument of ϕ is two-layer neural network with smooth activations.

Target function complexity. We need to quantify the complexity of the functions: more complex functions allow representing more distributions but are also harder to learn. We begin by defining the complexity of univariate smooth functions used in the definition of target functions. Let $\psi : \mathbb{R} \to \mathbb{R}$ have Taylor expansion $\psi(y) = \sum_{j=0}^{\infty} c_j y^j$, then, for $\epsilon > 0$, its complexity $C_0(\psi, \epsilon)$ is given by

$$C_0(\psi, \epsilon) = O\left(\left(\sum_{i=0}^{\infty} (i+1)^{1.75}|c_i|\right) \text{poly}\left(\frac{1}{\epsilon}\right)\right)$$

which is a weighted norm of the Taylor coefficients. For example, when $\psi(y)$ is one of $\text{poly}(y), \sin(y), e^y - 1, \tanh(y)$, it is known that $C_0(\psi, \epsilon) = O(\text{poly}(\frac{1}{\epsilon}))$ [Arora et al., 2019, Allen-Zhu et al., 2019]. Very roughly, $C_0(\psi, \epsilon)$ captures how many samples are needed to learn ψ up to error ϵ. For F^* in our target class, complexity $C(F^*, \epsilon)$ is defined to be $\text{poly}(d, \max_{i\in[d]} p_i, \max_{i\in[d], r\in[p_i]} C_0(\psi_{i,r}, \epsilon))$.

4.2 Theoretical analysis of UNF

We state the main theorem for UNFs informally. (For the complete version, see Theorem G.6 in the appendix.)

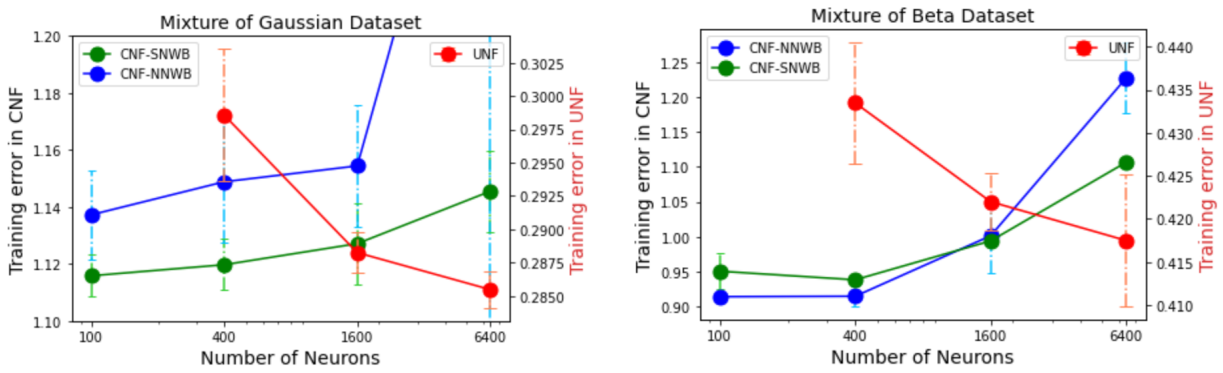

Figure 1: Effect of over-parameterization on training of CNF and UNF on mixture of Gaussian (left figure) and mixture of Beta (right figure) dataset

Theorem 4.1. *For any $\epsilon > 0$ and for any target function F^* with finite $\frac{\partial F_i^*(x_{1:i})}{\partial x_i}$ for all $i \in [d]$, hidden layer size $m \geq \frac{C(F^*,\epsilon)}{\epsilon^2}$, the number of samples $n \geq \frac{C(F^*,\epsilon)}{\epsilon^2}$, the number of quadrature points $Q \geq O(\frac{C(F^*,\epsilon)}{\epsilon})$ and total time steps $T \geq O(\frac{C(F^*,\epsilon)}{\epsilon^2})$ with probability at least 0.9, we have*

$$\mathbb{E}_{\text{sgd}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{x\sim\mathcal{D}}L(f^{(t)},x)\right] - \mathbb{E}_{x\sim\mathcal{D}}\left[L(F^*,x)\right] = O(\epsilon).$$

Recall that $\text{KL}(p_{F^*,Z}||p_{f^{(t)},Z}) = \mathbb{E}_X \log \frac{p_{F^*,Z}(X)}{p_{f^{(t)},Z}(X)}$, which gives $\mathbb{E}_{\text{sgd}}[\frac{1}{T}\sum_{t=0}^{T-1}\text{KL}(p_{F^*,Z}||p_{f^{(t)},Z})] = O(\epsilon)$. Using Pinsker's inequality, we can also bound the total variation distance between the learned and data distributions $p_{f_t,Z}$ and $p_{F^*,Z}$. The theorem can be interpreted as saying that the target density $p_{F^*,Z}$ of $X = F^{*-1}(Z)$ is close to the density given by the learned function, namely $p_{f^{(t)},Z}$ (which is the density of $(f^{(t)})^{-1}(Z)$). Note that Theorem 4.1 gives the learning guarantee for all probability distributions which has a two-layer low complexity neural network with smooth activation as the derivative of the target function F^*. An example of such functions is any positive low degree polynomial with small coefficients.

Proof Outline. The general outline of the proof follows that for supervised learning mentioned earlier, but details differ substantially and require new ideas. First, unlike prior work which only works with one neural network, NFs have d neural networks which are trained jointly. But we show that each neural network behaves essentially independently which allows us to analyze each neural network separately. Therefore, for each neural network $\nabla_i f_i(x_{1:i})$, we define its pseudo-network by

$$\nabla_i g_i(x_{1:i}) = \frac{\partial g_i(x_{1:i})}{\partial x_i} = \phi(P(x_{1:i};\theta_i)).$$

Note that our definition of pseudo-network is not a straightforward generalization from the supervised case: $\nabla_i g_i(x_{1:i})$ is not a linear approximation of $\nabla_i f_i(x_{1:i})$ because we are not taking linear approximation of final activation ϕ. For every $i \in [d]$, we show the existence of pseudo-networks close to the target function

$$\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}} \approx \phi\left(P(x_{1:i};\theta_i^*)\right)$$

for some parameters θ_i^* and for all x (Lemma E.8). However, for this we cannot directly use prior work: since our pseudo-network approximation is used in quadrature, it needs to be pointwise (close in L_∞) unlike only on average (close in L_1) as in the prior work. Next, we show that for each $i \in [d]$, the corresponding neural network and pseudo-network remain close during optimization and the same holds for the gradients of their respective loss functions (Section D on coupling). Specifically, for all $i \in [d]$, all $t \in [T]$ and all x, we show that

$$\nabla_i f_i^{(t)}(x_{1:i}) \approx \nabla_i g_i^{(t)}(x_{1:i}) \quad \text{(Lemma D.4)}$$
$$\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}(x_{1:i})\right) \approx \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}(x_{1:i})\right) \quad \text{(Lemma D.6)}$$
$$\tilde{L}\left(\nabla f^{(t)},x\right) \approx \tilde{L}\left(\nabla g^{(t)},x\right) \quad \text{(Lemma D.5)}$$
$$\nabla_\theta \tilde{L}\left(\nabla f^{(t)},x\right) \approx \nabla_\theta \tilde{L}\left(\nabla g^{(t)},x\right) \quad \text{(Lemma D.7)}.$$

Using coupling and independence of neural networks mentioned above, we show that SGD achieves near-minimum training loss (Theorem F.3), that is, for sufficient large T,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)},\mathcal{X})] \leq \tilde{L}(\nabla F^*,\mathcal{X}) + O(\epsilon).$$

Compared to the supervised setting the details in these sections are considerably more involved due to the presence of ∇f and $\tilde{f}$ and other features of the loss function. Finally, the full generalization result is proven

in Theorem G.6 showing that for sufficiently large T, population loss $L(f^{(T)}, \mathcal{D})$ is close to $L(F^*, \mathcal{D})$:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{\text{sgd}}\left[L(f^{(t)}, \mathcal{D})\right] \leq L(F^*, \mathcal{D}) + O(\epsilon).$$

This is proven by stringing together several approximate equalities. First, we show that the loss $\tilde{L}(\nabla F^*, x)$ (and $\tilde{L}(\nabla f^{(t)}, x)$) using the approximation via quadrature is close to the true loss $L(F^*, x)$ (respectively $L(f^{(t)}, x)$):

$$\tilde{L}(\nabla F^*, x) \approx L(F^*, x) \quad \text{and} \quad \tilde{L}(\nabla f^{(t)}, x) \approx L(f^{(t)}, x)$$

It is also shown that the empirical and population versions of approximate loss are close:

$$\tilde{L}(\nabla f^{(t)}, \mathcal{X}) \approx \tilde{L}(\nabla f^{(t)}, \mathcal{D}) \qquad \text{(Lemma G.3)}$$
$$\tilde{L}(\nabla F^*, \mathcal{X}) \approx \tilde{L}(\nabla F^*, \mathcal{D}) \qquad \text{(Lemma G.4).}$$

These results together with the optimization result mentioned earlier give Theorem G.6.

5 Experiments

In Sec. 3, we theoretically show that overparameterized neural networks in CNFs can not approximate the target function in the bounded time steps or in the bounded change in weights, and in Sec. 4, we show that highly overparameterized neural networks provably learn target distribution. We now give empirical evidence of these claims. In Fig. 1, we plot training error after a fixed number of training iterations for a different amount of over-parameterization for both CNF and UNF models on a mixture-of-Gaussian and a mixture-of-Beta distribution datasets. The left and right y-axes represent training error in CNF and UNF models, respectively. CNF-SNWB and CNF-NNWB denote CNF models with standard normal and normalized normal ($\mathcal{N}(0, \frac{1}{m})$) initialization of parameters, resp. We see that as we increase overparameterization in CNF models, training error becomes *larger* after a fixed number of training iterations, which means that larger CNF models need *larger* number of training iterations to learn the target function. But in UNFs, by increasing overparameterization, training error becomes *smaller*, which means that larger UNF models need *smaller* number of training iterations to learn the target function. Thus, our experimental results suggest that overparameterization in CNFs makes training *slower* and overparameterization in UNFs makes training *faster*. These experiments were done for a fixed learning rate. Similar patterns were observed for various different settings of learning rates except when training becomes unstable in CNFs. Since results in supervised learning also suggest that overparameterization makes training faster Neyshabur et al. [2015], our results on CNF are novel and surprising. Results on CNFs as well as results on UNFs on additional synthetic and real datasets, deeper models, various initializations, different learning rates and full experimental setup are given in Appendix I.

6 Conclusions and Limitaions

We gave the first end-to-end theoretical analysis of normalizing flows. We introduced the dichotomy between CNFs and UNFs: overparametrization seems to be hurting training of CNFs but for UNFs overparametrization does not hurt and we can analyze UNFs when the underlying network has one hidden-layer. We also proposed NF variants with desirable properties and these may find use in future work.

The main limitations of our work are the following which also suggests the main open problems: (1) A clear theoretical and empirical understanding of the role of overparameterization in CNFs remains an interesting open direction. As shown by our negative theoretical results, it seems necessary to analyze CNFs in the moderately overparametrized setting. However, this setting is not well-understood even in the supervised case. (2) For UNFs our analysis requires the overparametrized setting. (3) For the analysis we distill NF architectures to essentials—while this permits us to zero in on the main phenomena the more practical architectures are far more elaborate and performant and pose new theoretical challenges. (4) Our work assumes the autoregressive structure of the flow models. However, the role of overparameterized neural networks in other normalizing flow models such as coupling flows, residual flows, and other generative models such as VAEs is not well understood. (5) Our theoretical results have a one-hidden layer flow model but invertible flow models can be sequentially composed to construct an invertible map and in practice, flows models are sequentially composed to learn flexible target distributions. Extending our theoretical results for such models is an open problem.

References

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 8580–

8589, 2018. URL http://papers.nips.cc/paper/8076-neural-tangent-kernel-convergence-and-generalization-in-neural-networks.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *Proceedings of the 35th International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1810.02054.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6158–6169, 2019.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109:1–26, 03 2020. doi: 10.1007/s10994-019-05839-6.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 322–332. PMLR, 2019. URL http://proceedings.mlr.press/v97/arora19a.html.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/rezende15.html.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://www.aclweb.org/anthology/K16-1002.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242. 2016. URL http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf.

Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJehNfW0-.

Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Are gans created equal? a large-scale study. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 698–707, 2018.

I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *ArXiv*, abs/1912.02762, 2019.

Rares-Darius Buhai, Andrej Risteski, Yoni Halpern, and David Sontag. Empirical study of benefits of overparameterization in single-layer latent variable generative models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL https://proceedings.icml.cc/static/paper_files/icml/2020/5645-Paper.pdf.

Zhifeng Kong and Kamalika Chaudhuri. The expressive power of a class of normalizing flow models. volume 108 of *Proceedings of Machine Learning Research*, pages 3599–3609, Online, 26–28 Aug 2020. PMLR. URL http://proceedings.mlr.press/v108/kong20a.html.

Frederic Koehler, Viraj Mehta, and Andrej Risteski. Representational aspects of depth and conditioning in normalizing flows. *arXiv preprint arXiv:2010.01155*, 2020.

Holden Lee, Chirag Pabbaraju, Anish Sevekari, and Andrej Risteski. Universal approximation for log-concave distributions using well-conditioned normalizing flows, 2021.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians. Calculus of Variations, PDEs and Modeling.* Birkhäuser, 2015.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron C. Courville. Neural autoregressive flows. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2083–2092. PMLR, 2018. URL http://proceedings.mlr.press/v80/huang18d.html.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 511. AUAI Press, 2019a. URL http://auai.org/uai2019/proceedings/papers/511.pdf.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1543–1553, 2019. URL http://papers.nips.cc/paper/8433-unconstrained-monotonic-neural-networks.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *ICLR (Workshop)*, 2015.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 511. AUAI Press, 2019b. URL http://auai.org/uai2019/proceedings/papers/511.pdf.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems*, pages 6598–6608, 2019.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014.

Vaishnavh Nagarajan and J Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Rianne van den Berg, Leonard Hasenclever, Jakub Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In *proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2335–2344, 2017. ISBN 9781510860964.

Takeshi Teshima, I. Ishikawa, Koichi Tojo, Kenta Oono, M. Ikeda, and M. Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *ArXiv*, abs/2006.11469, 2020.

Qi Lei, Jason D. Lee, Alexandros G. Dimakis, and Constantinos Daskalakis. SGD learns one-layer networks in WGANs. In *In Proceedings of the 37th International COnference on Machine Learning*, 2020. URL https://proceedings.icml.cc/static/paper_files/icml/2020/4998-Paper.pdf.

Yogesh Balaji, Mohammadmahdi Sajedi, Neha Mukund Kalibhat, Mucong Ding, Dominik Stöger, Mahdi Soltanolkotabi, and Soheil Feizi. Understanding overparameterization in generative adversarial networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=C3qvk5IQIJY.

Yuanzhi Li and Zehao Dou. Making method of moments great again? – how can GANs learn the target distribution, 2020. URL https://arxiv.org/abs/2003.04033.

Thanh V. Nguyen, Raymond K. W. Wong, and Chinmay Hegde. On the dynamics of gradient descent for autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 2858–2867. PMLR, 2019a. URL http://proceedings.mlr.press/v89/nguyen19a.html.

Thanh V. Nguyen, Raymond K. W. Wong, and Chinmay Hegde. Benefits of jointly training autoencoders: An improved neural tangent kernel analysis. *CoRR*, abs/1911.11983, 2019b. URL http://arxiv.org/abs/1911.11983.

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. Overparameterized neural networks can implement associative memory, 2020. URL https://arxiv.org/abs/1909.12362.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

Justin Romberg. Maximum of a sequence of gaussian random variables. 2012. URL `http://cnx.org/contents/8bd316d8-6442-4f5a-a597-aef1d6202f87@1`.

Yuan-Chuan Li and Cheh-Chih Yeh. Some equivalent forms of bernoulli's inequality: A survey. *Applied Mathematics*, 4(07):1070, 2013.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

Supplementary Material: Learning and Generalization in Overparameterized Normalizing Flows

A Outline

In this section, we give outline of details and proofs of supplementary. We define common notations between Constrained Normalizing Flows results and Unconstrained Normalizing Flow results in Appendix B. Our results on CNFs from Section 3 from the main paper are discussed in detail in Theorem H.5 (Section H.1) and Theorem H.6 (Section H.2) and their proofs.

We give details about our result on UNFs (in Section 4) in Theorem G.6 and its proof (Section G). Our analysis begins with showing that if change in weights and biases from the initialization is small for a neural network, then training dynamics of the pseudo-network (linear approximation of neural network) is close to training dynamics of the neural network in Section D. In Section E, we show that with high probability there exist a pseudo-network which can approximate the derivative of target function. In Section F, we show that optimization problem for the pseudo-network is convex; therefore, combining results from Section E and Section D will give us the result that the loss of UNFs on the training data is close to the loss of target function. In section G, we prove generalization guarantees to test datasets and complete the proof of Theorem H.5.

We also provide experimental results to verify our theoretical claims on UNFs and CNFs in Section 5 and Section I. Discussion of related work is given in Section J.

B Notations

In this section, we define commonly used notations. We denote $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ as a concatenation of 2 vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. For any 2 vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, $\boldsymbol{\alpha} \odot \boldsymbol{\beta}$ denotes element wise multiplication of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ vector. We use $\|\boldsymbol{\alpha}\|_1$, $\|\boldsymbol{\alpha}\|_2$ and $\|\boldsymbol{\alpha}\|_\infty$ to denote L_1, L_2 and L_∞ norm of vector $\boldsymbol{\alpha}$. For any matrix $M \in \mathbb{R}^{m \times d}$, we denote matrix norm as

$$\|M\|_{p,q} = \left(\sum_{i \in [m]} \|m_i\|_p^q \right)^{1/q},$$

where $m_i \in \mathbb{R}^d$ denotes row vector of matrix M. We denote vector $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^m$. Big-$O$ and Big-Ω notation to hide only constants. We use log to denote natural logarithm. For any constant n, $[n]$ is denoted by set $\{1, 2, \ldots, n\}$. We use $\mathcal{N}(\mu, \sigma)$ to denote Gaussian distribution with mean μ and variance σ. We use $\mathbb{I}[E]$ to denote the indicator of the event E. We say a function $f : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz continuous if $\big|f(x) - f(y)\big| \leq L\|x - y\|_2$ for all $x, y \in \mathbb{R}^d$.

C Preliminaries

Recall that X is the random variable corresponding to the data distribution and Z is a random variable with standard Gaussian or multivariate exponential distribution. There seems to be no well-accepted definition of standard exponential distribution; for our purposes the following natural defintion will serve well. The density of the standard exponential distribution at $z = (z_1, z_2, \ldots, z_d) \in \mathbb{R}^d$ is given by $e^{-\sum_{i=1}^d z_i}$ when all $z_i \geq 0$, and by 0, otherwise. Let flow $f : \mathbb{R}^d \to \mathbb{R}^d$ be an monotonic autoregressive function. Then standard change of density formula using invertibility of f gives

$$p_{f,Z}(\mathbf{x}) = p_Z(\mathbf{z}) \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right).$$

To make $f(x) = (f_1(x_{1:1}), f_2(x_{1:2}), \ldots, f_d(x_{1:d}))$ an monotonic autoregressive function, we force function $f_i(x_{1:i})$ to be monotonic with respect to x_i for any fixed $x_{1:(i-1)}$ where x_i is i^{th} dimension of x. Recall that $x_{1:i}$ represents the vector including first i elements of vector x for any $i \in [1, d]$.

Unlike the constrained case where we model f using a neural network, in unconstrained case we model derivative of function using d neural networks. In normalizing flow, for all $i \in [1, d]$, we model $\frac{\partial f_i(x_{1:i})}{\partial x_i}$ using a neural network $N(x_{1:i}; \theta_i)$. To be specific,

$$\nabla_i f_i\left(x_{1:i}\right) = \frac{\partial f_i(x_{1:i})}{\partial x_i} = \phi\left(N(x_{1:i}; \theta_i)\right).$$

We denote ∇f as $\left(\nabla_1 f_1\left(x_{1:1}\right), \ldots, \nabla_r f_r\left(x_{1:r}\right), \ldots, \nabla_d f_d\left(x_{1:d}\right)\right)$. Here, ϕ is the $\mathsf{ELU}{+}1$ function given by $\phi(x) = e^x \mathbb{I}\left[x \leq 0\right] + (x+1)\,\mathbb{I}\left[x > 0\right]$ for all $x \in \mathbb{R}$. we use a one-hidden-layer neural network in $N(x_{1:i}; \theta_i)$, which is given by

$$N(x_{1:i}; \theta_i) = \sum_{r=1}^{m} \bar{a}_{i,r} \sigma\left(\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}\right)\right)$$

We construct $\tilde{x}_{1:i} \in \mathbb{R}^{i+1} = (x_1, x_2, \ldots, x_i, \sqrt{1 - \|x_{1:i}\|^2})$ such that $\|\tilde{x}_{1:i}\|_2 = 1$. We can reconstruct f by integration:

$$f_1(x_{1:1}) = \int_{-1}^{x_1} \tfrac{\partial f_1(t)}{\partial t} \mathrm{d}t \quad \text{and} \quad f_i(x_{1:i}) = \int_{-1}^{x_i} \tfrac{\partial f_i(x_1, x_2, \ldots, x_{i-1}, t)}{\partial t} \mathrm{d}t \text{ for } 1 < i \leq d.$$

The lower limit in our integral is -1 because $\|x\|_2 \leq 1$ by our assumption on the support of the data distribution. Note that to reconstruct f from the Jacobian, we need to evaluate the integrals. While this cannot be done exactly, good approximation can be obtained via numerical integration (also known as quadrature). We estimate $f_i(x_{1:i})$ via the general quadrature formula by

$$\tilde{f}_i(x_{1:i}) = \sum_{j=1}^{Q} q_j \nabla_i f_i\left(\tau_j\left(x_{1:i}\right)\right).$$

Here, Q is the number of quadrature points and the $q_1, \ldots, q_Q$ are the corresponding coefficients. We use simple rectangle quadrature, which arises in Riemann integration, and uses only positive coefficients with $q_j = \Delta_{x_i} := \frac{x_i + 1}{Q}$ and $\tau_j\left(x_{1:i}\right) = (x_1, \ldots, x_{i-1}, -1 + j\Delta_{x_i})$. The loss function for normalizing flows is given by

$$\tilde{L}\left(\nabla f, x\right) = -\log\left(p_Z\left(\tilde{f}(x)\right)\right) - \log\left(\Pi_{i=1}^{d} \nabla_i f_i\left(x_{1:i}\right)\right)$$

Using standard exponential distribution as a base distribution, we get

$$\tilde{L}\left(\nabla f, x\right) = \sum_{i=1}^{d} \tilde{f}_i(x_{1:i}) - \sum_{r=1}^{d} \log\left(\nabla_i f_i\left(x_{1:i}\right)\right) = \sum_{i=1}^{d} \tilde{L}_i\left(\nabla f, x\right) \tag{C.1}$$

where

$$\tilde{L}_i\left(\nabla f, x\right) = \tilde{f}_i(x_{1:i}) - \log\left(\nabla_i f_i\left(x_{1:i}\right)\right).$$

For our theoretical result, we consider target functions whose derivative are given by

$$\frac{\partial F_i^*\left(x_{1:i}\right)}{\partial x_i} = \phi(\sum_{r=1}^{p_i} \mu_{i,r}^* \psi_{i,r}(\langle u_{i,r}^*, \tilde{x}_{1:i} \rangle)),$$

where $|\mu_{i,r}^*| \leq 1, \|u_{i,r}^*\|_2 \leq 1$ for all $i \in [d]$ and $\psi_{i,r} : \mathbb{R} \to \mathbb{R}$ are smooth functions with Taylor expansion and p_i are positive integers. Our target function class is rich: the argument of ϕ is two-layer neural network with smooth activations.

We need to quantify the complexity of the functions: more complex functions allow representing more distributions but are also harder to learn. We begin by defining the complexity of univariate smooth functions used in the definition of target functions. Let $\psi : \mathbb{R} \to \mathbb{R}$ have Taylor expansion $\psi(y) = \sum_{j=0}^{\infty} c_j y^j$, then its complexity $C_0(\psi, \epsilon)$ for $\epsilon > 0$ is given by $O((\sum_{i=0}^{\infty}(i+1)^{1.75}|c_i|)\mathrm{poly}(\frac{1}{\epsilon}))$ which is a weighted norm of the Taylor coefficients. For example, when $\psi(y)$ is one of $\mathrm{poly}(y), \sin(y), e^y - 1, \tanh(y)$, it is known that $C_0(\psi, \epsilon) = O(\mathrm{poly}(\frac{1}{\epsilon}))$ Allen-Zhu et al. [2019]. Very roughly, $C_0(\psi, \epsilon)$ captures how many samples are needed to learn ψ up to error ϵ. For F^* in our target class, complexity $C(F^*, \epsilon)$ is defined to be $\mathrm{poly}(d, \max_{i \in [d]} p_i, \max_{i \in [d], r \in [p_i]} C_0(\psi_{i,r}, \epsilon))$.

For each neural network $\nabla_i f_i(x_{1:i})$, we define its pseudo-network by $\nabla_i g_i(x_{1:i}) = \frac{\partial g_i(x_{1:i})}{\partial x_i} = \phi(P(x_{1:i};\theta_i))$., where

$$P(x_{1:i};\theta_i) = \sum_{r=1}^{m} \bar{a}_{i,r}\sigma\left(\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right)$$

Note that our definition of pseudo-network is not the straightforward generalization from the supervised case: $\nabla_i g_i(x_{1:i})$ is not a linear approximation of $\nabla_i f_i(x_{1:i})$ because we are not taking linear approximation of final activation ϕ.

D Coupling

In this section, we will establish closeness between training dynamics of neural networks and pseudo network, which we will call as coupling. First, we will establish the coupling between $\nabla_i f_i(x_{1:i})$ and $\nabla_i g_i(x_{1:i})$ (Lemma D.4). Using coupling between $\nabla_i f_i(x_{1:i})$ and $\nabla_i g_i(x_{1:i})$, we prove coupling between $\tilde{L}_i\left(\nabla f^{(t)}, x\right)$ and $\tilde{L}_i\left(\nabla g^{(t)}, x\right)$ (Lemma D.5). We also prove coupling between gradient $\nabla_\theta \tilde{L}\left(\nabla f^{(t)}, x\right)$ and $\nabla_\theta \tilde{L}\left(\nabla g^{(t)}, x\right)$ in Lemma D.7, which will be used in proving global optimization of neural network in Section F.

We define λ_1 as

$$\lambda_1 = \sup_{t\in[T], i\in[d], r\in[m], w_{i,r}^{(t)}, b_{i,r}^{(t)}, |x|\le 1} \frac{\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)}{\phi\left(N(x_{1:i};\theta_i^{(t)})\right)}, \tag{D.1}$$

which will be used later in the proof of coupling between $\nabla_i f_i(x_{1:i})$ and $\nabla_i g_i(x_{1:i})$. The upper bound on λ_1 is useful to bound derivative of $\tilde{L}(\nabla f, x)$ w.r.t. $w_{i,r}$. We get the following upper bound on λ_1:

$$\begin{aligned}
\lambda_1 &= \sup_{t\in[T], i\in[d], r\in[m], w_{i,r}^{(t)}, b_{i,r}^{(t)}, |x|\le 1} \frac{\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)}{\phi\left(N(x_{1:i};\theta_i^{(t)})\right)} \\
&= \sup_{t\in[T], i\in[d], r\in[m], w_{i,r}^{(t)}, b_{i,r}^{(t)}, |x|\le 1} \frac{\exp\left(N(x_{1:i};\theta_i^{(t)})\right)\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})<0\right] + \mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})\ge 0\right]}{\exp\left(N(x_{1:i};\theta_i^{(t)})\right)\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})<0\right] + \left(N(x_{1:i};\theta_i^{(t)})+1\right)\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})\ge 0\right]} \\
&= \sup_{t\in[T], i\in[d], r\in[m], w_{i,r}^{(t)}, b_{i,r}^{(t)}, |x|\le 1} \mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})<0\right] + \frac{\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})\ge 0\right]}{N(x_{1:i};\theta_i^{(t)})+1} \\
&\le 1.
\end{aligned} \tag{D.2}$$

Define $\bar{\Lambda}$ as

$$\bar{\Lambda} := 6c_1\epsilon_a\sqrt{2\log m} \tag{D.3}$$

for any fixed constant $c_1 > 10$.

Recall that loss function in case of CNFs is given by

$$\begin{aligned}
\tilde{L}(\nabla f, x) &= \sum_{i=1}^{d} \tilde{f}_i(x_{1:i}) - \sum_{i=1}^{d} \log\left(\nabla_i f_i(x_{1:i})\right), \\
&= \sum_{i=1}^{d}\left(\sum_{j=1}^{Q} \Delta_x \nabla_i f_i^{(t)}\left(\tau_j(x_{1:i})\right)\right) - \sum_{i=1}^{d} \log\left(\nabla_i f_i(x_{1:i})\right), \\
&= \sum_{i=1}^{d}\left(\sum_{j=1}^{Q} \Delta_x \phi\left(N\left(\tau_j(x_{1:i}), \theta_i^{(t)}\right)\right)\right) - \sum_{i=1}^{d} \log\left(\phi\left(N\left(x_{1:i}, \theta_i^{(t)}\right)\right)\right),
\end{aligned}$$

Lemma D.1. *(Bound on change in weights) For every $i \in [d]$, for all $r \in [m]$, for any positive constant $c_1 \geq 10$ and for every $x_{1:i}$ with $\|x_{1:i}\|_2 \leq \frac{1}{2}$, with at least $1 - \frac{1}{c_1}$ probability over random initialization, bound on change in weights after t steps with learning rate η is given by*

$$\begin{aligned} \left\|w_{i,r}^{(t)}\right\|_2 &\leq \eta\bar{\Lambda}t, \\ \left|b_{i,r}^{(t)}\right| &\leq \eta\bar{\Lambda}t. \end{aligned}$$

Proof. By taking derivative of $\tilde{L}\left(\nabla f, x\right)$ w.r.t. $w_{i,r}$, we get

$$\begin{aligned} \left\|\frac{\partial\tilde{L}\left(\nabla f^{(t)}, x\right)}{\partial w_{i,r}}\right\|_2 \leq & \left\|\left(\sum_{j=1}^{Q}\Delta_x\phi'\left(N\left(\tau_j\left(x_{1:i}\right), \theta_i^{(t)}\right)\right)\bar{a}_{i,r}\sigma'\left(\langle\bar{w}_{i,r}+w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right)\right)\tilde{\tau}_j\left(x_{1:i}\right)\right)\right\|_2 \\ & + \left\|\frac{1}{\phi\left(N(x_{1:i};\theta_i^{(t)})\right)}\left(\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)\bar{a}_{i,r}\sigma'\left(\langle\bar{w}_{i,r}+w_{i,r}^{(t)}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right)\right)\tilde{x}_{1:i}\right)\right\|_2 \\ & \leq \sum_{j=1}^{Q}\left\|\Delta_x\phi'\left(N\left(\tau_j\left(x_{1:i}\right);\theta_i^{(t)}\right)\right)\bar{a}_{i,r}\sigma'\left(\langle\bar{w}_{i,r}+w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right)\right)\tilde{\tau}_j\left(x_{1:i}\right)\right\|_2 \\ & + \left|\frac{\phi'\left(N\left(x_{1:i};\theta_i^{(t)}\right)\right)}{\phi\left(N\left(x_{1:i};\theta_i^{(t)}\right)\right)}\right|\left\|\bar{a}_{i,r}\sigma'\left(\langle\bar{w}_{i,r}+w_{i,r}^{(t)}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right)\right)\tilde{x}_{1:i}\right\|_2. \end{aligned}$$

Using $\left|q_j\right| \leq \frac{2}{Q}$, $\|\tilde{\tau}_j\left(x_{1:i}\right)\| = 1$, $\|\tilde{x}_{1:i}\| = 1$ and $|\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)/\phi\left(N(x_{1:i};\theta_i^{(t)})\right)| \leq 1$ (by (D.2)), we get

$$\left\|\frac{\partial\tilde{L}\left(\nabla f^{(t)}, x\right)}{\partial w_{i,r}}\right\|_2 \leq 3\left|\bar{a}_{i,r}\right|.$$

Using Lemma K.4, with probability at least $1 - \frac{1}{c_1}$ we get

$$\left\|\frac{\partial\tilde{L}\left(\nabla f^{(t)}, x\right)}{\partial w_{i,r}}\right\|_2 \leq \bar{\Lambda} \tag{D.4}$$

where $\bar{\Lambda}$ is defined in (D.3). Using the same reasoning for $b_{i,r}$, with probability at least $1 - \frac{1}{c_1}$ we get

$$\begin{aligned} \left|\frac{\partial\tilde{L}\left(\nabla f^{(t)}, x\right)}{\partial b_{i,r}}\right| = & \left|\sum_{j=1}^{Q}\Delta_x\phi'\left(N\left(\tau_j\left(x_{1:i}\right);\theta_i^{(t)}\right)\right)\bar{a}_{i,r}\sigma'\left(\langle\bar{w}_{i,r}+w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right)\right)\right| \\ & + \left|\frac{1}{\phi\left(N(x_{1:i};\theta_i^{(t)})\right)}\left(\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)\bar{a}_{i,r}\sigma'\left(\langle\bar{w}_{i,r}+w_{i,r}^{(t)}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right)\right)\right)\right| \\ & \leq 3\left|\bar{a}_{i,r}\right|. \\ & \leq \bar{\Lambda}. \end{aligned} \tag{D.5}$$

Using (D.4), (D.5) and the fact that we are using SGD, we obtain

$$\begin{aligned} \left\|w_{i,r}^{(t)}\right\|_2 &\leq \eta\bar{\Lambda}t, \\ \left|b_{i,r}^{(t)}\right| &\leq \eta\bar{\Lambda}t. \end{aligned} \tag{D.6}$$

□

Lemma D.2. *(Bound on the number of changes in activation patterns) For every $i \in [d]$ and for all $r \in [m]$, suppose $\left\|w_{i,r}\right\|_2 \leq \Delta_i$ and $\left|b_{i,r}\right| \leq \Delta_i$. Then, for every $x_{1:i}$ such that $\|x_{1:i}\| \leq \frac{1}{2}$, with probability at least $1 - \exp\left(-\frac{32(c_4-1)^2 m^2 \Delta_i^2}{\pi}\right)$ over random initialization, the number of activation patterns that change is at most $c_4 \frac{4\Delta_i \sqrt{m}}{\sqrt{\pi}}$. In other words, for at most $c_4 \frac{4\Delta_i \sqrt{m}}{\sqrt{\pi}}$ fraction of $r \in [m]$, we have*

$$\mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}\right) \geq 0\right] \neq \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r} \geq 0\right]$$

for any positive constant $c_4 \geq 1$.

Proof. Define

$$\mathcal{H}_i := \{r \in [m] \mid \left|\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r}\right| \geq 4\Delta_i\}. \tag{D.7}$$

The set $\mathcal{H}_i$ contains indices of neurons for which indicator function doesn't change its value if change in weights is bounded by Δ_i. For every $x_{1:i}$ such that $\|x_{1:i}\|_2 \leq 1$ and for all $r \in [m]$, $\left|\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r}\right| \leq 2\Delta_i$. For all $r \in \mathcal{H}_i$, we have

$$\mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right) \geq 0\right] = \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r} \geq 0\right]. \tag{D.8}$$

Now, we need to bound the size of $\mathcal{H}_i$. We know that for all x with $\|x_{1:i}\|_2 \leq 1$, $\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r}$ is Gaussian with $\mathbb{E}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r}\right] = 0$ and $\mathrm{Var}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r}\right] = \frac{2}{m}$. Using Lemma K.5, we get

$$\Pr\left(\left|\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r}\right| \leq 4\Delta_i\right) \leq \frac{4\Delta_i \sqrt{m}}{\sqrt{\pi}}.$$

Using Fact K.7 (Hoeffding's inequality) for $\mathcal{H}_i$ (where $\overline{\mathcal{H}}_i = [m] \backslash \mathcal{H}_i$) for any positive constant $c_4 \geq 1$, we get

$$\begin{aligned}
\Pr\left(\left|\overline{\mathcal{H}}_i\right| \geq c_4 m \frac{4\Delta_i \sqrt{m}}{\sqrt{\pi}}\right) &\leq \exp\left(-2m\left((c_4 - 1)\left(\frac{4\Delta_i \sqrt{m}}{\sqrt{\pi}}\right)\right)^2\right), \\
&\leq \exp\left(-\frac{32(c_4-1)^2 m^2 \Delta_i^2}{\pi}\right),
\end{aligned}$$

which gives

$$\Pr\left(|\mathcal{H}_i| \geq m\left(1 - c_4 \frac{4\Delta_i \sqrt{m}}{\sqrt{\pi}}\right)\right) \geq 1 - \exp\left(-\frac{32(c_4-1)^2 m^2 \Delta_i^2}{\pi}\right).$$

□

Lemma D.3. *(Bound on the difference between $\nabla_i f_i^{(t)}(x_{1:i})$ and $\nabla_i g_i^{(t)}(x_{1:i})$) For every $i \in [d]$, for all x with $\|x\|_2 \leq \frac{1}{2}$ and for every time step $t \geq 1$, with probability at least $1 - \frac{1}{c_1}$ over random initialization, for any positive constants $c_1 > 10$, we have*

$$\left|\phi\left(N(x_{1:i}; \theta_i^{(t)})\right) - \phi\left(P(x_{1:i}; \theta_i^{(t)})\right)\right| \leq 24 c_1 \epsilon_a \Delta_i \left|\overline{\mathcal{H}}_i^{(t)}\right| \sqrt{2 \log m}.$$

Proof. Using 1-Lipschitz continuity of ϕ, we get

$$\left|\phi\left(N(x_{1:i}; \theta_i)\right) - \phi\left(P(x_{1:i}; \theta_i)\right)\right| \leq \left|N(x_{1:i}; \theta_i) - P(x_{1:i}; \theta_i)\right|.$$

We bound $\left|N(x_{1:i};\theta_i) - P(x_{1:i};\theta_i)\right|$:

$$\begin{aligned}
\left|N(x_{1:i};\theta_i) - P(x_{1:i};\theta_i)\right| \leq & \Bigg| \sum_{r\in[m]} \bar{a}_{i,r} \left(\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right)\right) \mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right) \geq 0\right] \\
& - \sum_{r\in[m]} \bar{a}_{i,r} \left(\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right)\right) \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right] \Bigg| \\
\leq & \Bigg| \sum_{r\in\overline{\mathcal{H}}_i} \bar{a}_{i,r} \left(\langle \bar{w}_{i,r} + w^{(t)}_{i,r}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b^{(t)}_{i,r}\right)\right) \left(\mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right) \geq 0\right]\right. \\
& \left. - \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right]\right) \Bigg| \\
\overset{\text{(i)}}{\leq} & \left|\overline{\mathcal{H}}^{(t)}_i\right| \left(2c_1\epsilon_a\sqrt{2\log m}\right)(4\Delta_i + 2\Delta_i)\,(2) \\
\leq & 24c_1\epsilon_a\Delta_i \left|\overline{\mathcal{H}}^{(t)}_i\right| \sqrt{2\log m}, \qquad \text{(D.9)}
\end{aligned}$$

where inequality (i) uses Lemma K.4 to upper bound $|\bar{a}_{i,r}|$ with probability at least $1 - \frac{1}{c_1}$. □

Lemma D.4. *(Final bound on the difference between $\nabla_i f_i^{(t)}(x_{1:i})$ and $\nabla_i g_i^{(t)}(x_{1:i})$) For every $i \in [d]$, for all x with $\|x\|_2 \leq 1$ and for every time step $t \geq 1$, with probability at least $1 - \frac{1}{c_1} - \exp\left(-\frac{32(c_4-1)^2\eta^2 m^2 \bar{\Lambda}^2 t^2}{\pi}\right)$ over the random initialization, and some positive constants $c_1 > 10$ and $c_4 \geq 1$, we have*

$$\left|\phi\left(N(x_{1:i};\theta_i^{(t)})\right) - \phi\left(P(x_{1:i};\theta_i^{(t)})\right)\right| \leq \left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| \leq \frac{192\eta^2 m^{1.5}\bar{\Lambda}^2 c_1 c_4 \epsilon_a t^2\sqrt{\log m}}{\sqrt{\pi}}. \qquad \text{(D.10)}$$

Proof. Using Lemma D.2 and Lemma D.3, we get

$$\begin{aligned}
\left|\phi\left(N(x_{1:i};\theta_i^{(t)})\right) - \phi\left(P(x_{1:i};\theta_i^{(t)})\right)\right| \leq & 24c_1\epsilon_a\Delta_i \left|\overline{\mathcal{H}}^{(t)}_i\right| \sqrt{2\log m} \\
\overset{\text{(i)}}{\leq} & 24c_1\epsilon_a\Delta_i \left(c_4 m \frac{4\Delta_i\sqrt{m}}{\sqrt{\pi}}\right)\sqrt{2\log m} \\
= & \frac{96\sqrt{2}c_1 c_4\epsilon_a\Delta_i^2 m^{1.5}\sqrt{\log m}}{\sqrt{\pi}} \\
= & \frac{192\eta^2 m^{1.5}\bar{\Lambda}^2 c_1 c_4\epsilon_a t^2\sqrt{\log m}}{\sqrt{\pi}}, \qquad \text{(D.11)}
\end{aligned}$$

where inequality (i) uses Lemma D.2 and the inequality follows with at least $1 - \frac{1}{c_1} - \exp\left(-\frac{32(c_4-1)^2\eta^2 m^2\bar{\Lambda}^2 t^2}{\pi}\right)$ probability. □

We denote the upper bound as $\Lambda^{(t)}_{np}$:

$$\Lambda^{(t)}_{np} := \frac{192\eta^2 m^{1.5}\bar{\Lambda}^2 c_1 c_4\epsilon_a t^2\sqrt{\log m}}{\sqrt{\pi}}.$$

Lemma D.5. *(Coupling of the loss functions) For every $i \in [d]$, for all x with $\|x\|_2 \leq 1$ and for every time step $t \geq 1$, with probability at least $1 - \frac{1}{c_1} - \exp\left(-\frac{32(c_4-1)^2\eta^2 m^2\bar{\Lambda}^2 t^2}{\pi}\right)$ over the random initialization, loss function of neural network and pseudo-network are close for some positive constant $c_1 > 10$ and $c_4 \geq 1$:*

$$\left|\tilde{L}_i\left(\nabla f^{(t)}, x\right) - \tilde{L}_i\left(\nabla g^{(t)}, x\right)\right| \leq 3\Lambda^{(t)}_{np}.$$

Using eq. (C.1)*, with probability at least* $1 - \frac{d}{c_1} - d\exp\left(-\frac{32(c_4-1)^2\eta^2 m^2 \bar{\Lambda}^2 t^2}{\pi}\right)$ *over the random initialization, we have*

$$\left|\tilde{L}\left(\nabla f^{(t)}, x\right) - \tilde{L}\left(\nabla g^{(t)}, x\right)\right| \leq 3d\Lambda_{np}^{(t)}.$$

Proof.

$$\begin{aligned}
\left|\tilde{L}_i\left(\nabla f^{(t)}, x\right) - \tilde{L}_i\left(\nabla g^{(t)}, x\right)\right| \leq & \left|\sum_{j=1}^{Q} \Delta_x\left(\nabla_i f_i\left(\tau_j\left(x_{1:i}\right)\right)\right) - \sum_{j=1}^{Q} \Delta_x\left(\nabla_i g_i\left(\tau_j\left(x_{1:i}\right)\right)\right)\right| \\
& + \left|\log\left(\nabla_i f_i\left(x_{1:i}\right)\right) - \log\left(\nabla_i g_i\left(x_{1:i}\right)\right)\right| \\
\overset{\text{(i)}}{\leq} & 2\left(\sup_{i\in[Q]}\left|\nabla_i f_i\left(x_{1:i}\right) - \nabla_i g_i\left(x_{1:i}\right)\right|\right) + \left|N(x_{1:i}; \theta_i^{(t)}) - P(x_{1:i}; \theta_i^{(t)})\right| \\
\overset{\text{(ii)}}{\leq} & 3\Lambda_{np}^{(t)},
\end{aligned}$$

where inequality (i) follows from 1-Lipschitz continuity of $\log\left(\phi(u)\right)$ with respect to u. Inequality (ii) uses Lemma D.3. Using the definition of $\tilde{L}$, with at least probability $1 - \frac{d}{c_1} - d\exp\left(-\frac{32(c_4-1)^2\eta^2 m^2 \bar{\Lambda}^2 t^2}{\pi}\right)$, we get

$$\begin{aligned}
\left|\tilde{L}\left(\nabla f^{(t)}, x\right) - \tilde{L}\left(\nabla g^{(t)}, x\right)\right| &\leq \sum_{i=1}^{d}\left|\tilde{L}_i\left(\nabla f^{(t)}, x\right) - \tilde{L}_i\left(\nabla g^{(t)}, x\right)\right| \\
&\leq 3d\Lambda_{np}^{(t)}
\end{aligned}$$

□

Lemma D.6. *(Coupling of the gradients of functions) For every* $i \in [d]$*, for all* x *with* $\|x\|_2 \leq 1$ *and for every time step* $t \geq 1$*, with probability at least* $1 - \frac{1}{c_1}$ *over random initialization, gradient of derivative of neural network function and derivative of pseudo-network function with respect to parameters are close for any positive constant* $c_1 > 10$

$$\left\|\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(x_{1:i}\right)\right) - \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)\right\|_{2,1} \leq 4c_1\epsilon_a\left(m\Lambda_{np}^{(t)} + 2\left|\overline{\mathcal{H}}_i^{(t)}\right|\right)\sqrt{2\log m}.$$

Proof. Recall that θ_i is given by

$$\theta_i = \begin{bmatrix} \theta_{i,1} \\ \vdots \\ \theta_{i,r} \\ \vdots \\ \theta_{i,m} \end{bmatrix}.$$

where $\theta_{i,r} = \left(w_{i,r}, b_{i,r}\right) \in \mathbb{R}^{i+2}$.

$$\begin{aligned}
\left\|\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(x_{1:i}\right)\right) - \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)\right\|_{2,1} \leq\ & \left\|\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right. \\
& \left. - \phi'\left(P(x_{1:i};\theta_i^{(t)})\right)\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
\leq\ & \left\|\phi'\left(N(x_{1:i};\theta_i^{(t)})\right)\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \phi'\left(P(x_{1:i};\theta_i^{(t)})\right)\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
& + \left\|\phi'\left(P(x_{1:i};\theta_i^{(t)})\right)\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \phi'\left(P(x_{1:i};\theta_i^{(t)})\right)\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
\leq\ & \left|\phi'\left(N(x_{1:i};\theta_i^{(t)})\right) - \phi'\left(P(x_{1:i};\theta_i^{(t)})\right)\right| \left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
& + \left|\phi'\left(P(x_{1:i};\theta_i^{(t)})\right)\right| \left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
\leq\ & \left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| \left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
& + \left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1},
\end{aligned}$$

where the last inequality follows from 1-Lipschitzness of ϕ' and $\phi'(x) \leq 1$ for all x. Now, we will bound $\left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1}$:

$$\begin{aligned}
\left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \leq\ & \left\|\left[(\mathbf{1}\bar{a}_{i,r}, \bar{a}_{i,r}) \odot (\tilde{x}_{1:i}, 1) \odot (\mathbf{1}\mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right) \geq 0\right]\right.\right. \\
& - \mathbf{1}\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right], \mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right) \geq 0\right] \\
& \left.\left. - \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right])\right]_m^{r=1}\right\|_{2,1} \\
\overset{\text{(i)}}{\leq}\ & \left(8c_1\epsilon_a\sqrt{2\log m}\right)\left|\overline{\mathcal{H}}_i^{(t)}\right| \\
\leq\ & 8c_1\epsilon_a\left|\overline{\mathcal{H}}_i^{(t)}\right|\sqrt{2\log m}, \qquad \text{(D.12)}
\end{aligned}$$

where inequality (i) follows from Lemma D.2 with atleast $1 - \frac{1}{c_1}$ probability. Now using Eq.(D.12), with atleast $1 - \frac{1}{c_1}$ probability, we get

$$\begin{aligned}
\left\|\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(x_{1:i}\right)\right) - \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)\right\|_{2,1} \leq & \left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| \left\|\left[(\mathbf{1}\bar{a}_{i,r}, \bar{a}_{i,r}) \odot (\tilde{x}_{1:i}, 1) \odot\right.\right. \\
& \left.\left.\left(\mathbf{1}\mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right) \geq 0\right], \mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{\tau}_j\left(x_{1:i}\right)\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right) \geq 0\right]\right)\right]_m^{r=1}\right\|_{2,1} \\
& + \left\|\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
\overset{\text{(D.12)}}{\leq}\ & 8c_1\epsilon_a m\Lambda_{np}^{(t)}\sqrt{2\log m} + 8c_1\epsilon_a\left|\overline{\mathcal{H}}_i^{(t)}\right|\sqrt{2\log m} \\
=\ & 8c_1\epsilon_a\left(m\Lambda_{np}^{(t)} + \left|\overline{\mathcal{H}}_i^{(t)}\right|\right)\sqrt{2\log m}.
\end{aligned}$$

□

Lemma D.7. *(Coupling of the gradient of loss) For every $i \in [d]$, for all x with $\|x\|_2 \leq 1$ and for every time step $t \geq 1$, with probability at least $1 - \frac{d}{c_1} - d\exp\left(-\frac{32(c_4-1)^2\eta^2m^2\bar{\Lambda}^2t^2}{\pi}\right)$ over random initialization, gradient of loss*

function with neural network and loss function with pseudo-network are close for some positive constant $c_1 > 10$ and $c_4 \geq 1$:

$$\left\|\nabla_\theta \tilde{L}\left(\nabla f^{(t)}, x\right) - \nabla_\theta \tilde{L}\left(\nabla g^{(t)}, x\right)\right\|_{2,1} \leq \frac{192 d\eta m^{1.5} \bar{\Lambda} c_1 c_4 \epsilon_a t\sqrt{\log m}}{\sqrt{\pi}} + 24 c_1 d\epsilon_a m \Lambda_{np}^{(t)} \sqrt{2\log m}.$$

Proof. We have

$$\begin{aligned}
\|\nabla_{\theta_i} \tilde{L}\left(\nabla f^{(t)}, x\right) - \nabla_{\theta_i} \tilde{L}\left(\nabla g^{(t)}, x\right)\|_{2,1} = & \left\| \sum_{j=1}^{Q} \Delta_x \nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right) - \frac{\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(x_{1:i}\right)\right)}{\nabla_i f_i^{(t)}\left(x_{1:i}\right)} \right. \\
& \left. - \sum_{j=1}^{Q} \Delta_x \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right) + \frac{\nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right)}{\nabla_i g_i^{(t)}\left(x_{1:i}\right)} \right\|_{2,1} \\
\leq & \underbrace{\left\| \sum_{j=1}^{Q} \Delta_x \nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right) - \sum_{j=1}^{Q} \Delta_x \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right) \right\|_{2,1}}_{\mathrm{I}} \\
& + \underbrace{\left\| \frac{\nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)}{\nabla_i g_i^{(t)}\left(x_{1:i}\right)} - \frac{\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(x_{1:i}\right)\right)}{\nabla_i f_i^{(t)}\left(x_{1:i}\right)} \right\|_{2,1}}_{\mathrm{II}}
\end{aligned}$$

We first bound I using Lemma D.6:

$$\begin{aligned}
\mathrm{I} &\leq \sum_{j=1}^{Q} \Delta_x \left\| \nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right) - \nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right) \right\|_1 \\
&\leq 16 c_1 \epsilon_a \left(m\Lambda_{np}^{(t)} + \left|\overline{\mathcal{H}}_i^{(t)}\right| \right) \sqrt{2\log m},
\end{aligned}$$

Now, we bound II:

$$
\begin{aligned}
\mathrm{II} =& \left\| \frac{\nabla_{\theta_i}\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)}{\nabla_i g_i^{(t)}\left(x_{1:i}\right)} - \frac{\nabla_{\theta_i}\left(\nabla_i f_i^{(t)}\left(x_{1:i}\right)\right)}{\nabla_i f_i^{(t)}\left(x_{1:i}\right)} \right\|_{2,1} \\
=& \Bigg\| \frac{\exp\left(P(x_{1:i};\theta_i^{(t)})\right)\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0\right]+\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})\geq 0\right]}{\exp\left(P(x_{1:i};\theta_i^{(t)})\right)\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0\right]+\left(P(x_{1:i};\theta_i^{(t)})+1\right)\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})\geq 0\right]}\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)}) \\
& - \frac{\exp\left(N(x_{1:i};\theta_i^{(t)})\right)\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})<0\right]+\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})\geq 0\right]}{\exp\left(N(x_{1:i};\theta_i^{(t)})\right)\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})<0\right]+\left(N(x_{1:i};\theta_i^{(t)})+1\right)\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})\geq 0\right]}\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\Bigg\|_{2,1} \\
=& \Bigg\| \left(\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0\right]+\frac{\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})\geq 0\right]}{\left(P(x_{1:i};\theta_i^{(t)})+1\right)}\right)\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)}) \\
& - \left(\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})<0\right]+\frac{\mathbb{I}\left[N(x_{1:i};\theta_i^{(t)})\geq 0\right]}{\left(N(x_{1:i};\theta_i^{(t)})+1\right)}\right)\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\Bigg\|_{2,1} \\
=& \underbrace{\left\|\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})-\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1}\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0,N(x_{1:i};\theta_i^{(t)})<0\right]}_{\mathrm{II}_1} \\
& + \underbrace{\left\|\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})-\frac{\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})}{N(x_{1:i};\theta_i^{(t)})+1}\right\|_{2,1}\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0,N(x_{1:i};\theta_i^{(t)})\geq 0\right]}_{\mathrm{II}_2} \\
& + \underbrace{\left\|\frac{\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})}{P(x_{1:i};\theta_i^{(t)})+1}-\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1}\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})\geq 0,N(x_{1:i};\theta_i^{(t)})<0\right]}_{\mathrm{II}_3} \\
& + \underbrace{\left\|\frac{\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})}{P(x_{1:i};\theta_i^{(t)})+1}-\frac{\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})}{N(x_{1:i};\theta_i^{(t)})+1}\right\|_{2,1}\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})\geq 0,N(x_{1:i};\theta_i^{(t)})\geq 0\right]}_{\mathrm{II}_4}.
\end{aligned}
$$

On simplifying II_2, we get

$$
\begin{aligned}
\mathrm{II}_2 \leq& \Bigg(\left|\frac{1}{N(x_{1:i};\theta_i^{(t)})+1}\right|\left\|\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})-\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1} \\
& + \left|\frac{N(x_{1:i};\theta_i^{(t)})}{1+N(x_{1:i};\theta_i^{(t)})}\right|\left\|\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1}\Bigg)\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0,N(x_{1:i};\theta_i^{(t)})\geq 0\right] \\
\overset{(D.10)}{\leq}& \left(\left\|\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})-\nabla_{\theta_i}N(x_{1:i};\theta_i^{(t)})\right\|_{2,1}+\Lambda_{np}^{(t)}\left\|\nabla_{\theta_i}P(x_{1:i};\theta_i^{(t)})\right\|_{2,1}\right)\mathbb{I}\left[P(x_{1:i};\theta_i^{(t)})<0,N(x;\theta^{(t)})\geq 0\right].
\end{aligned} \tag{D.13}
$$

Similarly, on simplifying II_3, we get

$$\begin{aligned}
\mathrm{II}_3 \leq & \left(\left| \frac{1}{P(x_{1:i};\theta_i^{(t)})+1} \right| \left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \right. \\
& \left. + \left| \frac{P(x_{1:i};\theta_i^{(t)})}{1+P(x_{1:i};\theta_i^{(t)})} \right| \left\| \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \right) \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) \geq 0, N(x_{1:i};\theta_i^{(t)}) < 0 \right] \qquad \text{(D.14)} \\
\leq & \left(\left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} + \Lambda_{np}^{(t)} \left\| \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \right) \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) \geq 0, N(x_{1:i};\theta_i^{(t)}) < 0 \right]. \qquad \text{(D.15)}
\end{aligned}$$

On simplifying II_4, we get

$$\begin{aligned}
\mathrm{II}_4 \leq & \left(\left\| \frac{\nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)})}{P(x_{1:i};\theta_i^{(t)})+1} - \frac{\nabla_{\theta_i} N(x;\theta^{(t)})}{P(x_{1:i};\theta_i^{(t)})+1} \right\|_{2,1} + \left\| \frac{\nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)})}{P(x_{1:i};\theta_i^{(t)})+1} \right. \right. \\
& \left. \left. - \frac{\nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)})}{N(x_{1:i};\theta_i^{(t)})+1} \right\|_{2,1} \right) \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) \geq 0, N(x_{1:i};\theta_i^{(t)}) \geq 0 \right] \\
\leq & \left(\frac{1}{P(x_{1:i};\theta_i^{(t)})+1} \left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \right. \\
& \left. + \frac{\left\| \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \Lambda_{np}^{(t)}}{\left(P(x_{1:i};\theta_i^{(t)})+1 \right)\left(N(x_{1:i};\theta_i^{(t)})+1 \right)} \right) \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) \geq 0, N(x_{1:i};\theta_i^{(t)}) \geq 0 \right] \\
\leq & \left(\left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} + \Lambda_{np}^{(t)} \left\| \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \right) \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) \geq 0, N(x_{1:i};\theta_i^{(t)}) \geq 0 \right]. \qquad \text{(D.16)}
\end{aligned}$$

Using (D.13), (D.14) and (D.16), we have

$$\begin{aligned}
\mathrm{II} \leq & \left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) - \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} + \Lambda_{np}^{(t)} \left\| \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) \geq 0 \right] \\
& + \Lambda_{np}^{(t)} \left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \mathbb{I}\left[P(x_{1:i};\theta_i^{(t)}) < 0, N(x_{1:i};\theta_i^{(t)}) \geq 0 \right].
\end{aligned}$$

Using (D.12), we get

$$\begin{aligned}
\mathrm{II} & \leq 8c_1\epsilon_a \left| \overline{\mathcal{H}}_i^{(t)} \right| \sqrt{2\log m} + \Lambda_{np}^{(t)} \left(\left\| \nabla_{\theta_i} P(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} + \left\| \nabla_{\theta_i} N(x_{1:i};\theta_i^{(t)}) \right\|_{2,1} \right) \\
& \leq 8c_1\epsilon_a \left| \overline{\mathcal{H}}_i^{(t)} \right| \sqrt{2\log m} + \Lambda_{np}^{(t)} \left(\left\| \left[(\mathbf{1}\bar{a}_{i,r}, \bar{a}_{i,r}) \odot (\tilde{x}_{1:i}, 1) \odot \left(\mathbf{1}\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r} \geq 0 \right], \right. \right. \right. \right. \\
& \quad \left. \left. \left. \left. \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r} \geq 0 \right] \right) \right]_m^{r=1} \right\|_{2,1} + \left\| \left[(\mathbf{1}\bar{a}_{i,r}, \bar{a}_{i,r}) \odot (\tilde{x}_{1:i}, 1) \odot \right. \right. \right. \\
& \quad \left. \left. \left. \left(\mathbf{1}\mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)} \right) \geq 0 \right], \mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)} \right) \geq 0 \right] \right) \right]_m^{r=1} \right\|_{2,1} \right) \\
& \leq 8c_1\epsilon_a \left| \overline{\mathcal{H}}_i^{(t)} \right| \sqrt{2\log m} + \Lambda_{np}^{(t)} \left(8c_1\epsilon_a m \sqrt{2\log m} \right) \\
& = 8c_1\epsilon_a \left(\left| \overline{\mathcal{H}}_i^{(t)} \right| + m\Lambda_{np}^{(t)} \right) \sqrt{2\log m}. \qquad \text{(D.17)}
\end{aligned}$$

Combining bounds on I and II, we get

$$\begin{aligned}\left\|\nabla_{\theta_i}\tilde{L}\left(\nabla f^{(t)},x\right)-\nabla_{\theta_i}\tilde{L}\left(\nabla g^{(t)},x\right)\right\|_{2,1} &\leq 16c_1\epsilon_a\left(m\Lambda_{np}^{(t)}+\left|\overline{\mathcal{H}}_i^{(t)}\right|\right)\sqrt{2\log m}\\ &+8c_1\epsilon_a\left(\left|\overline{\mathcal{H}}_i^{(t)}\right|+m\Lambda_{np}^{(t)}\right)\sqrt{2\log m}\\ &\leq 24c_1\epsilon_a\left(m\Lambda_{np}^{(t)}+\left|\overline{\mathcal{H}}_i^{(t)}\right|\right)\sqrt{2\log m}.\end{aligned}$$

Using Lemma D.1 and Lemma D.2, with at least $1-\frac{1}{c_1}-\exp\left(-\frac{32(c_4-1)^2\eta^2m^2\bar{\Lambda}^2t^2}{\pi}\right)$ probability, we get

$$\left\|\nabla_{\theta_i}\tilde{L}\left(\nabla f^{(t)},x\right)-\nabla_{\theta_i}\tilde{L}\left(\nabla g^{(t)},x\right)\right\|_{2,1}\leq\frac{192\eta m^{1.5}\bar{\Lambda}c_1c_4\epsilon_at\sqrt{\log m}}{\sqrt{\pi}}+24c_1\epsilon_am\Lambda_{np}^{(t)}\sqrt{2\log m}. \tag{D.18}$$

We can upper bound $\left\|\nabla_\theta\tilde{L}\left(\nabla f^{(t)},x\right)-\nabla_\theta\tilde{L}\left(\nabla g^{(t)},x\right)\right\|_{2,1}$ as

$$\begin{aligned}\left\|\nabla_\theta\tilde{L}\left(\nabla f^{(t)},x\right)-\nabla_\theta\tilde{L}\left(\nabla g^{(t)},x\right)\right\|_{2,1} &\leq\sum_{i=1}^{d}\left\|\nabla_{\theta_i}\tilde{L}\left(\nabla f^{(t)},x\right)-\nabla_{\theta_i}\tilde{L}\left(\nabla g^{(t)},x\right)\right\|_{2,1}\\ &\leq\frac{192d\eta m^{1.5}\bar{\Lambda}c_1c_4\epsilon_at\sqrt{\log m}}{\sqrt{\pi}}+24c_1d\epsilon_am\Lambda_{np}^{(t)}\sqrt{2\log m}\end{aligned}$$

where last inequality follows from $1-\frac{d}{c_1}-d\exp\left(-\frac{32(c_4-1)^2\eta^2m^2\bar{\Lambda}^2t^2}{\pi}\right)$. □

We define Γ as

$$\Gamma:=\frac{192d\eta m^{1.5}\bar{\Lambda}c_1c_4\epsilon_aT\sqrt{\log m}}{\sqrt{\pi}}+24c_1d\epsilon_am\Lambda_{np}^{(t)}\sqrt{2\log m}.$$

Note that Γ is an upper bound on $\left\|\nabla_\theta\tilde{L}\left(\nabla f^{(T)},x\right)-\nabla_\theta\tilde{L}\left(\nabla g^{(T)},x\right)\right\|_{2,1}$.

E Approximation

In this section, we will prove that each pseudo network can approximate any target function from target class with small offset θ^* from the weights of initialization. We first prove that expectation of multiplication of a fixed ω function and $\mathbb{I}\left[\langle w,x\rangle+b\geq 0\right]$ can approximate any smooth activation in target function (Lemma E.6). This is used to prove that $\nabla_i g_i^*\left(x_{1:i}\right)$ can approximate any target function in target class in L_∞ norm. Using Lipschitz continuity $\tilde{L}$ with respect to $\nabla_i g_i^*\left(x_{1:i}\right)$, we prove that $\tilde{L}(\nabla_i g_i^*,x)$ is close to $\tilde{L}(\nabla_i F^*,x)$, where F^* is any target function in the target class.

To prove results in this section, we require a number of new techniques on top of techniques from Allen-Zhu et al. [2019]. The target functions in Allen-Zhu et al. [2019] are more restricted because L_2-norm of weights in target function is equal to 1 (i.e., $\left|\mu_{i,r}^*\right|,\left\|u_{i,r}^*\right\|_2=1$). In our paper, we relax this condition and allow any weights with their norm bounded by 1 (i.e., $\left|\mu_{i,r}^*\right|,\left\|u_{i,r}^*\right\|_2\leq 1$). Our proof can easily be extended to weights bounded by any constant. Additionally, our proof requires to bound L_∞ approximation error between pseudo network $\nabla_i g_i^*\left(x_{1:i}\right)$ and target network, which is a stronger condition than L_1 approximation error given in Allen-Zhu et al. [2019], and requires a new proof technique.

Lemma E.1. *For any fixed constant $0<C\leq 1$ and even $i>0$, for any $x_1\in[0,C]$ and b, we have*

$$\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1+\beta\sqrt{C^2-x_1^2}}{C}\right)\mathbb{I}\left[\alpha\geq b\right]\right]=q_ix_1^i \text{ where}$$

$$q_i=\frac{(i-1)!!\exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}}\sum_{r=1,odd}^{(i-1)}\frac{(-1)^{\frac{i-r-1}{2}}}{r!!}\binom{i/2-1}{(r-1)/2}b^r.$$

Similarly, for any fixed constant $C > 0$ and odd $i > 0$, for any $x_1 \in [0, C]$ and b, we have

$$\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2 - x_1^2}}{C}\right)\mathbb{I}\left[\alpha \geq b\right]\right] = q_i x_1^i \text{ where}$$
$$q_i = \frac{(i-1)!!\exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}} \sum_{r=0,even}^{(i-1)} \frac{(-1)^{\frac{i-r-1}{2}}}{r!!}\binom{i/2-1}{(r-1)/2}b^r.$$

Proof. Using summation formula from Fact K.1, we have

$$h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2 - x_1^2}}{C}\right) = \sum_{k=0}^{i}\binom{i}{k}\left(\frac{\alpha x_1}{C}\right)^{i-k} h_k\left(\beta\sqrt{1-\frac{x_1^2}{C^2}}\right).$$

Expanding $h_k\left(\beta\sqrt{1-\frac{x_1^2}{C^2}}\right)$ using multiplication formula of Hermite polynomial from Fact K.1, we get

$$h_k\left(\beta\sqrt{1-\frac{x_1^2}{C^2}}\right) = \sum_{j=0}^{\lfloor\frac{k}{2}\rfloor}\left(1-\frac{x_1^2}{C^2}\right)^{\frac{k-2j}{2}}\left(-\frac{x_1^2}{C^2}\right)^j\binom{k}{2j}\frac{(2j)!}{j!}2^{-j}h_{k-2j}(\beta). \tag{E.1}$$

Using Fact K.2, for even k, we have

$$\mathbb{E}_{\beta\sim\mathcal{N}(0,1)}\left[h_k\left(\beta\sqrt{1-\frac{x_1^2}{C^2}}\right)\right] = \left(-\frac{x_1^2}{C^2}\right)^{k/2}\frac{k!}{(k/2)!}2^{-k/2}, \tag{E.2}$$

and for odd k,

$$\mathbb{E}_{\beta\sim\mathcal{N}(0,1)}\left[h_k\left(\beta\sqrt{1-\frac{x_1^2}{C^2}}\right)\right] = 0. \tag{E.3}$$

Using Eq. (E.1), Eq. (E.2) and Eq.(E.3), we get

$$\begin{aligned}\mathbb{E}_{\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2 - x_1^2}}{C}\right)\right] &= \sum_{k=0,\text{even}}^{i}\binom{i}{k}\left(\frac{\alpha x_1}{C}\right)^{i-k}\left(-\frac{x_1^2}{C^2}\right)^{k/2}\frac{k!}{(k/2)!}(-2)^{-k/2}\\ &= \frac{x_1^i}{C^i}\sum_{k=0,\text{even}}^{i}\binom{i}{k}\alpha^{i-k}\frac{k!}{(k/2)!}(-2)^{-k/2}.\end{aligned}$$

Using $\mathbb{I}\left[\frac{\alpha}{C} \geq b\right]$ in the expectation, we have

$$\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2 - x_1^2}}{C}\right)\mathbb{I}\left[\alpha \geq b\right]\right] = \frac{x_1^i}{C^i}\sum_{k=0,\text{even}}^{i}\binom{i}{k}\mathbb{E}_{\alpha\sim\mathcal{N}(0,1)}\left[\alpha^{i-k}\mathbb{I}\left[\alpha \geq b\right]\right]\frac{k!}{(k/2)!}(-2)^{-k/2}. \tag{E.4}$$

Define $B_{i,b}$ as

$$B_{i,b} := \mathbb{E}_{\alpha\sim\mathcal{N}(0,1)}\left[\alpha^i\mathbb{I}\left[\alpha \geq b\right]\right].$$

Now, we divide our proof in two parts. In (a), we complete the proof for even $i > 0$ and in (b), we do it for odd i.

(a) Using Lemma E.2, for even $i \geq 0$, we have

$$B_{i,b} = (i-1)!!\Phi(0,1;b) + \phi(0,1;b) \sum_{j=1,j\text{odd}}^{i-1} \frac{(i-1)!!}{j!!} b^j$$

Using Eq. (E.4), we have

$$\begin{aligned}
&\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2-x_1^2}}{C}\right)\mathbb{I}[\alpha \geq b]\right] \\
=&\frac{x_1^i}{C^i}\left(\sum_{k=0,\text{even}}^{i}\binom{i}{k} B_{i-k,b}\frac{k!}{(k/2)!}(-2)^{-k/2}\right) \\
=&\frac{x_1^i}{C^i}\left(\sum_{k=0,\text{even}}^{i}\binom{i}{k}(i-k-1)!!\Phi(0,1;b)\frac{k!}{(k/2)!}(-2)^{-k/2}\right) \\
&+\frac{x_1^i}{C^i}\phi(0,1;b)\left(\sum_{k=0,\text{even}}\binom{i}{k}\left(\sum_{j=1,\text{odd}}^{i-k-1}\frac{(i-k-1)!!}{j!!}b^j\right)\frac{k!}{(k/2)!}(-2)^{-k/2}\right).
\end{aligned}$$

Using

$$\begin{aligned}
\sum_{k=0,\text{even}}^{i}\binom{i}{k}(i-k-1)!!\frac{k!}{(k/2)!}(-2)^{-k/2} &= \sum_{k=0,\text{even}}^{i}\frac{i!\,(i-k-1)!!k!\,(-2)^{-k/2}}{(i-k)!k!(k/2)!} \\
&= \sum_{k=0,\text{even}}^{i}\frac{i!(-1)^{k/2}}{(i-k)!!(k/2)!2^{k/2}} \\
&= (i-1)!!\sum_{k=0,\text{even}}^{i}\frac{i!!(-1)^{k/2}}{(i-k)!!(k/2)!2^{k/2}} \\
&= (i-1)!!\sum_{k=0,\text{even}}^{i}\binom{i/2}{k/2}(-1)^{k/2} \\
&= 0,
\end{aligned}$$

we get

$$\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2-x_1^2}}{C}\right)\mathbb{I}[\alpha \geq b]\right] = \frac{x_1^i}{C^i}(i-1)!!\phi(0,1;b)\sum_{r=1,\text{odd}}^{i-1} c_r b^r \tag{E.5}$$

where c_r is given by

$$\begin{aligned}
c_r &:= \frac{1}{(i-1)!!} \sum_{k=0,\text{even}}^{i-r-1} \binom{i}{k} \frac{(i-k-1)!!k! \left(-2\right)^{-k/2}}{r!!(k/2)!} \\
&= \frac{1}{(i-1)!!} \sum_{k=0,\text{even}}^{i-r-1} \frac{i! \left(i-k-1\right)!!k! \left(-2\right)^{-k/2}}{(i-k)!k!r!!(k/2)!} \\
&= \sum_{k=0,\text{even}}^{i-r-1} \frac{i!! \ \left(-2\right)^{-k/2}}{(i-k)!! \ r!!(k/2)!} \\
&= \sum_{k=0,\text{even}}^{i-r-1} \binom{i/2}{k/2} \frac{\left(-1\right)^{k/2}}{r!!} \\
&= \sum_{j=0,\text{even}}^{(i-r-1)/2} \binom{i/2}{j} \frac{\left(-1\right)^{j}}{r!!} \\
&= \sum_{j=0,\text{even}}^{(i-r-1)/2} \binom{i/2}{j} \frac{\left(-1\right)^{j}}{r!!} \\
&= \sum_{j=0,\text{even}}^{(i-r-1)/2} \binom{j-i/2-1}{j} \frac{1}{r!!} \\
&= \frac{1}{r!!} \binom{-i/2+\left(i-r-1\right)/2}{\left(i-r-1\right)/2} \\
&= \frac{\left(-1\right)^{(i-r-1)/2}}{r!!} \binom{i/2-1}{\left(i-r-1\right)/2} \\
&= \frac{\left(-1\right)^{(i-r-1)/2}}{r!!} \binom{i/2-1}{\left(r-1\right)/2}.
\end{aligned}$$

Using value of c_r in Eq.(E.5), we get the required result.

(b) By Lemma E.2 for odd $i > 0$, we get

$$B_{i,b} = \phi\left(0,1;b\right) \sum_{j=0,\text{even}}^{i-1} \frac{(i-1)!!}{j!!} b^j.$$

Using Eq.(E.4), we get

$$\begin{aligned}
&\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)} \left[h_i \left(\frac{\alpha x_1 + \beta\sqrt{C^2 - x_1^2}}{C} \right) \mathbb{I}\left[\alpha \geq b\right] \right] \\
&= \frac{x_1^i}{C^i} \sum_{k=0,\text{even}}^{i} \binom{i}{k} B_{i-k,b} \frac{k!}{(k/2)!} \left(-2\right)^{-k/2} \\
&= \frac{x_1^i}{C^i} \phi\left(0,1;b\right) \sum_{k=0,\text{even}}^{i} \binom{i}{k} \left(\sum_{j=0,\text{even}}^{i-k-1} \frac{(i-k-1)!!}{j!!} b^j \right) \frac{k!}{(k/2)!} \left(-2\right)^{-k/2} \\
&= \frac{x_1^i}{C^i} \phi\left(0,1;b\right) (i-1)!! \sum_{r=0,r\text{even}}^{i-1} c_r b^r
\end{aligned} \tag{E.6}$$

where c_r is given by

$$c_r = \frac{1}{(i=1)!!} \sum_{k=0,\text{even}}^{i-r-1} \frac{(i-k-1)!!}{r!!} \frac{k!}{(k/2)!} \left(-2\right)^{-k/2}.$$

By a similar calculation given in part (a), we get

$$c_r = \frac{\left(-1\right)^{(i-r-1)/2}}{r!!} \binom{i/2-1}{\left(r-1\right)/2}.$$

Using value of c_r in Eq.(E.6), we get the required result.

□

Lemma E.2. *Define $B_{i,b}$ as*

$$B_{i,b} := \mathbb{E}_{\alpha\sim\mathcal{N}(0,1)}\left[\alpha^i \mathbb{I}\left[\alpha \geq b\right]\right].$$

and define $\Phi\left(0,1;b\right)$ and $\phi\left(0,1;b\right)$ as

$$\Phi\left(0,1;b\right) = \Pr_{\alpha\sim\mathcal{N}(0,1)}\left[\alpha \geq b\right]$$
$$\phi\left(0,1;b\right) = \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-b^2}{2}\right)$$

For any b, we have

$$\text{for even } i \geq 0: \qquad B_{i,b} = (i-1)!!\Phi\left(0,1;b\right) + \phi\left(0,1;b\right)\sum_{j=1,odd}^{i-1}\frac{(i-1)!!}{j!!}b^j \tag{E.7}$$

$$\text{for odd } i > 0: \qquad B_{i,b} = \phi\left(0,1;b\right)\sum_{j=1,even}^{i-1}\frac{(i-1)!!}{j!!}b^j \tag{E.8}$$

Proof. The lemma follows from Lemma A.7 of Allen-Zhu et al. [2019]. □

We will use two different view of the randomness. Define w_0 as $w_0 = (\alpha_1, \beta_1)$ and $x = \left(x_1, \sqrt{C^2 - x_1^2}\right)$ where α_1 and β_1 are standard normal random variables and C is any positive constant. In alternative view of randomness, we write w_0 as

$$w_0 = \frac{\langle w_0, x\rangle}{\|x\|^2}x + \frac{\langle w_0, x^\perp\rangle}{\|x^\perp\|^2}x^\perp$$

where $x^\perp = \left(\sqrt{C^2 - x_1^2}, -x_1\right)$. Define $\alpha' = \langle w_0, x\rangle$ and $\beta' = \langle w_0, x^\perp\rangle$ where α' and β' are normal random variables with 0 mean and C^2 variance. Using definitions of α' and β', we get

$$w_0 = \frac{\alpha'}{C^2}x + \frac{\beta'}{C^2}x^\perp = \frac{\alpha}{C}x + \frac{\beta}{C}x^\perp$$

where α and β are standard normal random variable.

Lemma E.3. *For every integer $i \geq 1$, there exists a constant q_i' with $\left|q_i'\right| \geq \frac{(i-1)!!}{200i^2C^i}$ such that*

$$\text{for even } i: \qquad x_1^i = \frac{1}{q_i'}\mathbb{E}_{w_0\sim\mathcal{N}(0,\mathbf{I}),b_0\sim\mathcal{N}(0,1)}\left[h_i\left(\alpha_1\right)\mathbb{I}\left[0 \leq -b_0 \leq 1/(2i)\right]\mathbb{I}\left[\frac{\langle w_0, x\rangle}{C} + b_0 \geq 0\right]\right]$$
$$\text{for odd } i: \qquad x_1^i = \frac{1}{q_i'}\mathbb{E}_{w_0\sim\mathcal{N}(0,\mathbf{I}),b_0\sim\mathcal{N}(0,1)}\left[h_i\left(\alpha_1\right)\mathbb{I}\left[\left|b_0\right| \leq 1/(2i)\right]\mathbb{I}\left[\frac{\langle w_0, x\rangle}{C} + b_0 \geq 0\right]\right]$$

Proof. First, we will prove for even i. By Lemma E.1, we get

$$\begin{aligned}
&\mathbb{E}_{w_0\sim\mathcal{N}(0,\mathbf{1}),b_0\sim\mathcal{N}(0,1)}\left[h_i\left(\alpha_1\right)\mathbb{I}\left[0 \leq -b_0 \leq 1/(2i)\right]\mathbb{I}\left[\frac{\langle w_0, x\rangle}{C} + b_0 \geq 0\right]\right] \\
&= \mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1 + \beta\sqrt{C^2 - x_1^2}}{C}\right)\mathbb{I}\left[\alpha \geq -b_0\right]\right]\mathbb{I}\left[0 \leq -b_0 \leq 1/(2i)\right]\right] \\
&= \mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[q_i\mathbb{I}\left[0 \leq -b_0 \leq 1/(2i)\right]\right]x_1^i
\end{aligned} \tag{E.9}$$

where

$$q_i = \frac{(i-1)!! \exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}} \sum_{r=0,\text{even}}^{(i-1)} \frac{(-1)^{\frac{i-r-1}{2}}}{r!!} \binom{i/2-1}{(r-1)/2} (-b_0)^r .$$

Now, we try to bound the coefficient $\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[q_i \mathbb{I}\left[0 \le -b_0 \le 1/(2i)\right]\right]$. Define c_r as

$$c_r := \frac{(-1)^{\frac{i-r-1}{2}}}{r!!} \binom{i/2-1}{(r-1)/2}.$$

For $0 \le -b_0 \le 1/(2i)$ and for all odd r with $1 < r \le i-1$,

$$\left|c_r (-b_0)^r\right| = \left|\frac{(-1)^{\frac{i-r-1}{2}}}{r!!} \binom{i/2-1}{(r-1)/2} (-b_0)^r\right| \le \left|\frac{(-1)^{\frac{i-r+1}{2}}}{(r-2)!!} \binom{i/2-1}{(r-3)/2} (-b_0)^r\right| \le \frac{1}{4}\left|c_{r-2} (-b_0)^{r-2}\right| .$$

Using above relation, we get

$$\begin{aligned}
\left|\sum_{r=1,\text{odd}}^{i-1} c_r (-b_0)^r\right| &\ge \left||c_1 b_0| - \left|\sum_{r=3,\text{odd}} c_r (-b_0)^r\right|\right| \\
&\ge \left||c_1 b_0| - \left|\sum_{r=1}^{\infty} \frac{1}{4^r} |c_1 (b_0)|\right|\right| \\
&\ge \left||c_1 b_0| - \frac{1}{3}\|c_1 b_0|\right| \\
&\ge \frac{2}{3}|c_1 b_0| ,
\end{aligned}$$

and

$$\text{sign}\left(\sum_{r=1,\text{odd}}^{i-1} c_r (-b_0)^r\right) = \text{sign}\left(c_1 (-b_0)\right) = \text{sign}\left(c_1\right).$$

Using Eq.(E.9), we get

$$\begin{aligned}
&\left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[q_i \mathbb{I}\left[0 \le -b_0 \le 1/(2i)\right]\right]\right| \\
&= \left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\frac{(i-1)!! \exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}} \sum_{r=0,\text{even}}^{(i-1)} c_r (-b_0)^r \mathbb{I}\left[0 \le -b_0 \le 1/(2i)\right]\right]\right| \\
&= \left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\frac{(i-1)!! \exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}} \text{sign}\left(\sum_{r=0,\text{even}}^{(i-1)} c_r (-b_0)^r\right)\left|\sum_{r=0,\text{even}}^{(i-1)} c_r (-b_0)^r\right| \mathbb{I}\left[0 \le -b_0 \le 1/(2i)\right]\right]\right| \\
&\ge \left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\frac{(i-1)!! \exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}} \text{sign}\left(c_1\right) \frac{2}{3}|c_1\ b_0| \mathbb{I}\left[0 \le -b_0 \le 1/(2i)\right]\right]\right| \\
&\ge \frac{(i-1)!!}{100 i^2 C^i}.
\end{aligned}$$

This completes the proof for even i. Similarly for odd i, using Lemma E.1, we get

$$\begin{aligned}
&\mathbb{E}_{w_0\sim\mathcal{N}(0,\mathbf{1}),b_0\sim\mathcal{N}(0,1)}\left[h_i\left(\alpha_1\right)\mathbb{I}\left[|b_0|\le 1/(2i)\right]\mathbb{I}\left[\frac{\langle w_0,x\rangle}{C}+b_0\ge 0\right]\right]\\
&=\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\mathbb{E}_{\alpha,\beta\sim\mathcal{N}(0,1)}\left[h_i\left(\frac{\alpha x_1+\beta\sqrt{C^2-x_1^2}}{C}\right)\mathbb{I}\left[\alpha\ge -b_0\right]\right]\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]\\
&=\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[q_i\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]x_1^i
\end{aligned}\tag{E.10}$$

where

$$q_i=\frac{(i-1)!!\exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}}\sum_{r=0,\text{even}}^{(i-1)}\frac{(-1)^{\frac{i-r-1}{2}}}{r!!}\binom{i/2-1}{(r-1)/2}b^r.$$

Now, we will try to bound $\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[q_i\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]$. Define c_r as

$$c_r:=\frac{(-1)^{\frac{i-r-1}{2}}}{r!!}\binom{i/2-1}{(r-1)/2}.$$

For $|b_0|\le 1/(2i)$ and for all even r with $1<r\le i-1$, we get

$$\left|c_r\left(-b_0\right)^r\right|=\left|\frac{(-1)^{\frac{i-r-1}{2}}}{r!!}\binom{i/2-1}{(r-1)/2}\left(-b_0\right)^r\right|\le\left|\frac{(-1)^{\frac{i-r+1}{2}}}{(r-2)!!}\binom{i/2-1}{(r-3)/2}\left(-b_0\right)^r\right|\le\frac{1}{4}\left|c_{r-2}\left(-b_0\right)^{r-2}\right|.$$

Using above relation, we get

$$\left|\sum_{r=1,\text{odd}}^{i-1}c_r\left(-b_0\right)^r\right|\ge\left||c_0|-\left|\sum_{r=2,\text{even}}c_r\left(-b_0\right)^r\right|\right|\ge\left||c_0|-\left|\sum_{r=1}^{\infty}\frac{1}{4^r}|c_0|\right|\right|\ge\left||c_0|-\frac{1}{3}\|c_0|\|\right|=\frac{2}{3}|c_0|=\frac{2}{3}\left|\binom{i/2-1}{-1/2}\right|>\frac{1}{2i},$$

and

$$\text{sign}\left(\sum_{r=1,\text{odd}}^{i-1}c_r\left(-b_0\right)^r\right)=\text{sign}\left(c_0\right).$$

Using the formula of q_i in Eq. (E.10), we have

$$\begin{aligned}
&\left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[q_i\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]\right|\\
&=\left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\frac{(i-1)!!\exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}}\sum_{r=0,\text{even}}^{(i-1)}c_rb^r\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]\right|\\
&=\left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\frac{(i-1)!!\exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}}\text{sign}\left(\sum_{r=0,\text{even}}^{(i-1)}c_rb^r\right)\left|\sum_{r=0,\text{even}}^{(i-1)}c_rb^r\right|\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]\right|\\
&\ge\left|\mathbb{E}_{b_0\sim\mathcal{N}(0,1)}\left[\frac{(i-1)!!\exp\left(-\frac{b^2}{2}\right)}{C^i\sqrt{2\pi}}\text{sign}\left(c_0\right)\frac{1}{2i}\mathbb{I}\left[|b_0|\le 1/(2i)\right]\right]\right|\\
&\ge\frac{(i-1)!!}{100i^2C^i}
\end{aligned}$$

This completes the proof for odd i. □

Lemma E.4. *For any constant $C \leq 1$ and for any arbitary function $\psi : [-C, C] \mapsto \mathbb{R}$, we have*

$$\psi(x_1) = c_0 + \sum_{i=1}^{\infty} c_i' \mathbb{E}_{w_0 \sim \mathcal{N}(0,\mathbf{1}), b_0 \sim \mathcal{N}(0,1)} \left[h_i(\alpha_1) \mathbb{I}\left[G_i(b_0)\right] \mathbb{I}\left[\frac{\langle w_0, x\rangle}{C} + b_0 \geq 0\right]\right]$$

where $w_0 = (\alpha_1, \beta_1)$, $c_i = i^{th}$ coefficient of taylor series of ψ function,

$$\left|c_i'\right| \leq \frac{200 i^2 |c_i|}{(i-1)!!} \qquad \text{and} \qquad G_i(b_0) = \begin{cases} |b_0| \leq 1/(2i) & \text{if } i \text{ is odd} \\ 0 < -b_0 \leq 1/(2i) & \text{if } i \text{ is even} \end{cases}$$

Proof. Using Taylor expansion of function $\psi(x_1)$, we get

$$\begin{aligned} \psi(x_1) &= c_0 + \sum_{i=1,\text{odd}}^{\infty} c_i x_1^i + \sum_{i=2,\text{even}}^{\infty} c_i x_1^i \\ &= c_0 + \sum_{i=1}^{\infty} c_i' \mathbb{E}_{\alpha,\beta,b_0 \sim \mathcal{N}(0,1)} \left[h_i(\alpha_1) \mathbb{I}\left[G_i(b_0)\right] \mathbb{I}\left[\frac{\langle x, w_0\rangle}{C} + b_0 \geq 0\right]\right] \end{aligned}$$

where above relation follows from Lemma E.3 and c_i' is given by

$$c_i' = \frac{c_i}{q_i'}, \qquad \left|c_i'\right| \leq \frac{200 i^2 |c_i| C^i}{(i-1)!!} \quad \text{and} \quad G_i(b_0) = \begin{cases} |b_0| \leq 1/(2i) & \text{if } i \text{ is odd} \\ 0 < -b_0 \leq 1/(2i) & \text{if } i \text{ is even} \end{cases}$$

□

Lemma E.5. *For any $\epsilon \in (0,1)$ and any positive integer i, setting $B_i \stackrel{\text{def}}{=} 100 i^{1/2} + 10\sqrt{\log \frac{1}{\epsilon}}$, we have*

1. $\sum_{i=1}^{\infty} \mathbb{E}_{z \sim \mathcal{N}(0,1)} \left[\left|h_i(z)\right| \mathbb{I}\left[|z| \geq B_i\right]\right] \leq \epsilon/8$
2. $\sum_{i=1}^{\infty} \mathbb{E}_{z \sim \mathcal{N}(0,1)} \left[\left|h_i(B_i)\right| \mathbb{I}\left[|z| \geq B_i\right]\right] \leq \epsilon/8$
3. $\sum_{i=1}^{\infty} \mathbb{E}_{z \sim \mathcal{N}(0,1)} \left[\left|h_i(z)\right| \mathbb{I}\left[|z| \leq B_i\right]\right] \leq \frac{1}{2}\mathfrak{C}_\epsilon(\psi)$

The Lemma is same as Claim C.2 of Allen-Zhu et al. [2019].

Lemma E.6. *For any positive integer d, for any $\epsilon \in (0,1)$, for every function ψ, every $\epsilon \in (0,1)$, every $u^*, x \in \mathbb{R}^d$ with $\|u^*\|_2 \leq 1$ and $\|x\|_2 = 1$, there exist a function $\omega : \mathbb{R}^3 \to [-\mathfrak{C}_\epsilon(\psi), \mathfrak{C}_\epsilon(\psi)]$ such that*

$$\left| \mathbb{E}_{w \sim \mathcal{N}(0,\mathbf{I}), b_0 \sim \mathcal{N}(0,1)} \left[\omega\left(\langle w, u^*\rangle, b_0, \|u^*\|\right) \mathbb{I}\left[\langle w, x\rangle + b_0 \geq 0\right]\right] - \psi\left(\langle u^*, x\rangle\right) \right| \leq \epsilon. \tag{E.11}$$

Proof. Define $\hat{h}_i(\alpha_1) \stackrel{\text{def}}{=} h_i(\alpha_1) \mathbb{I}\left[|\alpha_1| \leq B_i\right] + h_i\left(\text{sign}(\alpha_1) B_i\right) \mathbb{I}\left[|\alpha_1| > B_i\right]$. From Lemma E.4, we get

$$\begin{aligned} \psi(x_1) &= c_0 + \sum_{i=1}^{\infty} c_i' \mathbb{E}_{\alpha,\beta,b_0 \sim \mathcal{N}(0,1)} \left[h_i(\alpha_1) \mathbb{I}\left[G_i(b_0)\right] \mathbb{I}\left[\frac{\langle x, w_0\rangle}{C} + b_0 \geq 0\right]\right] \\ &= c_0 + R'(x_1) + \sum_{i=1}^{\infty} c_i' \mathbb{E}_{\alpha,\beta,b_0 \sim \mathcal{N}(0,1)} \left[\hat{h}_i(\alpha_1) \mathbb{I}\left[G_i(b_0)\right] \mathbb{I}\left[\frac{\langle x, w_0\rangle}{C} + b_0 \geq 0\right]\right]. \end{aligned}$$

where

$$R'(x_1) = \sum_{i=1}^{\infty} c_i' \mathbb{E}_{\alpha,\beta,b_0 \sim \mathcal{N}(0,1)} \left[\left(h_i(\alpha_1) \mathbb{I}\left[|\alpha_1| > B_i\right] - h_i\left(\text{sign}(\alpha_1) B_i\right) \mathbb{I}\left[|\alpha_1| > B_i\right]\right) \mathbb{I}\left[G_i(b_0)\right] \mathbb{I}\left[\frac{\langle x, w_0\rangle}{C} + b_0 \geq 0\right]\right].$$

Using Lemma E.5, we have $\left|R'(x_1)\right| \leq \epsilon/4$. Define $\omega(\alpha_1, b_0, C)$ as

$$\omega(\alpha_1, b_0, C) = 2c_0 + \sum_{i=1}^{\infty} c_i' \hat{h}_i(\alpha_1) \mathbb{I}\left[G_i\left(b_0\right)\right].$$

Using definition of $\omega(\alpha_1, b_0, C)$, we get

$$\left| \mathbb{E}_{\alpha_1,\beta_1,b_0\sim\mathcal{N}(0,1)} \left[\omega\left(\alpha_1, b_0, C\right) \mathbb{I}\left[\frac{\alpha_1 x_1 + \beta_1 \sqrt{C^2 - x_1^2}}{C} + b_0 \geq 0 \right] \right] \right| \leq \epsilon/4.$$

Using Lemma E.5, we have

$$\left|\omega\left(\alpha_1, b_0, C\right)\right| \leq 2c_0 + \frac{\epsilon}{8} + \frac{1}{2}\mathfrak{C}_\epsilon\left(\psi\right) \leq \mathfrak{C}_\epsilon\left(\psi\right)$$

This proves that for every function ψ, every $\epsilon \in (0,1)$, every constant $C \in \mathbb{R}$ and for every $x_1 \in [-C, C]$, there exist a function $\omega : \mathbb{R}^3 \to [-\mathfrak{C}_\epsilon\left(\psi\right), \mathfrak{C}_\epsilon\left(\psi\right)]$ such that we have

$$\left| \mathbb{E}_{\alpha_1,\beta_1,b_0\sim\mathcal{N}(0,1)} \left[\omega\left(\alpha_1, b_0, C\right) \mathbb{I}\left[\frac{\alpha_1 x_1 + \beta_1 \sqrt{C^2 - x_1^2}}{C} + b_0 \geq 0 \right] \right] - \psi(x_1) \right| \leq \epsilon. \tag{E.12}$$

We denote $u_i^{*\perp}$ for $2 \leq i \leq d$ as $d-1$ orthogonal vectors of u^* with $\|u_i^{*\perp}\| = \|u^*\|$. Now, using projection of w on u^*, we get

$$w = \frac{\langle w, u^* \rangle}{\|u^*\|^2} u^* + \sum_{i=2}^{d} \frac{\langle w, u_i^{*\perp} \rangle}{\|u_i^{*\perp}\|^2} u_i^{*\perp} = \frac{\alpha_1'}{\|u^*\|^2} u^* + \sum_{i=2}^{d} \frac{\alpha_i'}{\|u^*\|^2} u_i^{*\perp} \tag{E.13}$$

where α_i' for any i such that $1 \leq i \leq d$ is a normal random variable with 0 mean and $\|u^*\|^2$ variance. Define x_1' as $x_1' = \langle u^*, x \rangle$. Similarly, define $x_i' = \langle u_i^{*\perp}, x \rangle$ for $2 \leq i \leq d$. Now, dot product $\langle w, x \rangle$ can be written as

$$\begin{aligned}
\langle w, x \rangle &= \frac{1}{\|u^*\|^2} \langle \alpha_1' u^* + \sum_{i=2}^{d} \alpha_i' u_i^{*\perp}, x \rangle \\
&= \frac{1}{\|u^*\|^2} \left(\alpha_1' x_1' + \sum_{i=2}^{d} \alpha_i' x_i' \right) \\
&= \frac{1}{\|u^*\|^2} \left(\alpha_1' x_1' + \beta_1' \sqrt{\|u^*\|^2 - x_1'^2} \right) \\
&= \frac{1}{\|u^*\|} \left(\alpha_1 x_1' + \beta_1 \sqrt{\|u^*\|^2 - x_1'^2} \right)
\end{aligned} \tag{E.14}$$

where last inequality follows from $\left(\sum_{i=1}^{d} x_i'^2\right) = \|u^*\|^2$. Here α_1 and β_1 are standard normal random variables. Setting $C = \|u^*\|$ and using Eq.(E.12), Eq. (E.13) and Eq.(E.14), we get

$$\left| \mathbb{E}_{w\sim\mathcal{N}(0,\mathbf{I}),b_0\sim\mathcal{N}(0,1)} \left[\omega\left(\langle w, u^* \rangle, b_0, \|u^*\|\right) \mathbb{I}\left[\langle w, x \rangle + b_0 \geq 0\right] \right] - \psi\left(\langle u^*, x \rangle\right) \right| \leq \epsilon$$

□

Lemma E.7. *For all $i \in [d]$, for any $\epsilon \in (0,1)$, for any derivative of target function $\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}}$ and for any x with $\|x\| \leq 1$, there exist a set of parameters θ_i^* such that we have*

$$\left| \mathbb{E}_{\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)} \left[P_\ell(x_{1:i}; \theta_i^*) \right] - \phi^{-1}\left(\frac{\partial F_i^*\left(x_{1:i}\right)}{\partial x_{1:i}} \right) \right| \leq p_i \epsilon.$$

Moreover, L_∞ norm of θ_i^* is given by

$$\|\theta_i^*\|_{2,\infty} \leq \frac{\sqrt{\pi}\left(\sum_{r=1}^{p_i} U_{\omega_{i,r}}\right)}{m\epsilon_a\sqrt{2}}.$$

Proof. We denote pseudo network with parameters θ_i^* as:

$$P_\ell(x_{1:i};\theta_i^*) = \sum_{r=1}^{m} \bar{a}_{i,r}\left(\langle w_{i,r}^*, \tilde{x}_{1:i}\rangle + b_{i,r}^*\right)\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right].$$

Similarly, $\nabla_i g_i^*(x_{1:i})$ is given by $\phi\left(P_\ell(x_{1:i};\theta_i^*)\right)$. We will use function $\omega_{i,j}$ to approximate a neuron of target function $\psi_{i,j}$ for all $i \in [d], j \in [p_i]$. Setting $w_{i,r}^*$ and $b_{i,r}^*$ as

$$w_{i,r}^* = \frac{\sqrt{\pi}\text{sign}\left(\bar{a}_{i,r}\right)}{m\epsilon_a\sqrt{2}} \sum_{j=1}^{p_i} \mu_{i,j}^* \omega_{i,j}\left(\sqrt{m}\langle \bar{w}_{i,r}, u_{i,j}^*\rangle, \sqrt{m}\bar{b}_{i,r}, \|u_{i,j}^*\|\right) v_{i,j}^*,$$
$$b_{i,r}^* = 0,$$

we get

$$\begin{aligned}
&\left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P_\ell(x_{1:i};\theta_i^*)\right] - \phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial \tilde{x}_{1:i}}\right)\right| \\
=&\left|m\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[\bar{a}_{i,r}\left(\langle w_{i,r}^*, \tilde{x}_{1:i}\rangle + b_{i,r}^*\right)\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right]\right] - \phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial \tilde{x}_{1:i}}\right)\right| \\
=&\left|\frac{\sqrt{\pi}}{\epsilon_a\sqrt{2}}\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[\bar{a}_{i,r}\text{sign}\left(\bar{a}_{i,r}\right)\sum_{j=1}^{p_i}\mu_{i,j}^*\omega_{i,j}\left(\sqrt{m}\langle \bar{w}_{i,r}, u_{i,j}^*\rangle, \sqrt{m}\bar{b}_{i,r}, \|u_{i,j}^*\|\right)\langle v_{i,j}^*, \tilde{x}_{1:i}\rangle\right.\right. \\
&\left.\left.\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right]\right] - \phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial \tilde{x}_{1:i}}\right)\right| \\
=&\left|\mathbb{E}_{\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[\sum_{j=1}^{p_i}\mu_{j,r}^*\omega_{i,j}\left(\sqrt{m}\langle \bar{w}_{i,r}, u_{j,r}^*\rangle, \sqrt{m}\bar{b}_{i,r}, \|u_{j,r}^*\|\right)\langle v_{j,r}^*, \tilde{x}_{1:i}\rangle\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right]\right]\right. \\
&\left. - \sum_{j=1}^{p_i}\mu_{i,j}^*\psi_{i,j}\left(\langle u_{i,j}^*, \tilde{x}_{1:i}\rangle\right)\left(\langle v_{i,j}^*, \tilde{x}_{1:i}\rangle\right)\right| \\
\leq& p_i\epsilon
\end{aligned}$$

Bounding $\|w_{i,r}^*\|$, we get

$$\begin{aligned}
\|w_{i,r}^*\|_2 &= \left\|\frac{\sqrt{\pi}\text{sign}\left(\bar{a}_{i,r}\right)}{m\epsilon_a\sqrt{2}}\sum_{j=1}^{p_i}\mu_{i,j}^*\omega_{i,j}\left(\sqrt{m}\langle \bar{w}_{i,r}, u_{i,j}^*\rangle, \sqrt{m}\bar{b}_{i,r}, \|u_{i,j}^*\|\right)v_{i,j}^*\right\|_2 \\
&\leq \frac{\sqrt{\pi}\left(\sum_{r=1}^{p_i}U_{\omega_{i,r}}\right)}{m\epsilon_a\sqrt{2}}.
\end{aligned}$$

□

Define upper bound on $\|w_{i,r}^*\|_2$ as

$$U_{w_i^*} = \frac{\sqrt{\pi}\left(\sum_{r=1}^{p_i}U_{\omega_{i,r}}\right)}{m\epsilon_a\sqrt{2}}$$

Lemma E.8. *For any $i \in [d]$, for any $\epsilon \in (0,1)$, for any derivative of target function $\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}}$, for any $m \geq \Omega\left(\frac{d^{10}\left(\sum_{i=1}^{d}\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^{12}}{\epsilon_a^2\epsilon^8}\right)$ and for any x with $\|x\| \leq \frac{1}{2}$, there exist a set of parameters θ_i^* such that, with atleast $1 - \frac{1}{c_1} - \frac{1}{c_2} - \frac{1}{c_3} - \exp\left(-\frac{\epsilon^2}{2mC_i^2}\right) - \exp\left(-\frac{32(c_4-1)^2m^2U_{w_i^*}^2}{\pi}\right)$ probability, we have*

$$\left|\phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}}\right) - P(x_{1:i};\theta_i^*)\right| \leq (p_i+1)\epsilon + \frac{192c_1c_4\epsilon_a m^{1.5}U_{w_i^*}^2\sqrt{2\log m}}{\sqrt{\pi}}.$$

Proof. We divide $\left|\phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}}\right) - P(x_{1:i};\theta_i^*)\right|$ into five parts as

$$\begin{aligned}
\left|\phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}}\right) - P(x_{1:i};\theta_i^*)\right| \leq & \underbrace{\left|\phi^{-1}\left(\frac{\partial F_i^*(x_{1:i})}{\partial x_{1:i}}\right) - \mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P_\ell(x_{1:i};\theta_i^*)\right]\right|}_{\text{I}} \\
& + \underbrace{\left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P_\ell(x_{1:i};\theta_i^*)\right] - \mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P(x_{1:i};\theta_i^*)\right]\right|}_{\text{II}} \\
& + \underbrace{\left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P(x_{1:i};\theta_i^*)\right] - \mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[N(x_{1:i};\theta_i^*)\right]\right|}_{\text{III}} \\
& + \underbrace{\left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[N(x_{1:i};\theta_i^*)\right] - N(x_{1:i};\theta_i^*)\right|}_{\text{IV}} \\
& + \underbrace{\left|N(x_{1:i};\theta_i^*) - P(x_{1:i};\theta_i^*)\right|}_{\text{V}}. \qquad \text{(E.15)}
\end{aligned}$$

We know that the first part $\text{I} \leq p_i\epsilon$ from Lemma E.7. Since $\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P_c(x_{1:i};\theta_i^*)\right] = 0$, the second term $\text{II} = 0$. Using Lemma D.2 and Lemma D.3 for bounding the third term III, we get

$$\begin{aligned}
\text{III} &= \left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P(x_{1:i};\theta_i^*)\right] - \mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[N(x_{1:i};\theta_i^*)\right]\right| \\
&= \left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[P(x_{1:i};\theta_i^*) - N(x_{1:i};\theta_i^*)\right]\right| \\
&\leq \mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[\left|P(x_{1:i};\theta_i^*) - N(x_{1:i};\theta_i^*)\right|\right] \\
&\leq \mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[24c_1\epsilon_a U_{w_i^*}\left|\overline{\mathcal{H}}_i\right|\sqrt{2\log m}\right] \\
&\leq 24c_1\epsilon_a U_{w_i^*}\left(c_4 m\frac{4U_{w_i^*}\sqrt{m}}{\sqrt{\pi}}\right)\sqrt{2\log m} \\
&= \frac{96c_1c_4\epsilon_a m^{1.5}U_{w_i^*}^2\sqrt{2\log m}}{\sqrt{\pi}}. \qquad \text{(E.16)}
\end{aligned}$$

We will use technique from Yehudai and Shamir [2019] to bound the fourth term IV. Define a function $\mathbf{N}_i$ as

$$\mathbf{N}_i = \mathbf{N}_i\left(\left(\bar{a}_{i,1},\bar{w}_{i,1},\bar{b}_{i,1}\right),\ldots,\left(\bar{a}_{i,m},\bar{w}_{i,m},\bar{b}_{i,m}\right)\right) = \sup_x\left|\mathbb{E}_{\bar{a}_{i,r}\sim\mathcal{N}\left(0,\epsilon_a^2\right),\bar{w}_{i,r},\bar{b}_{i,r}\sim\mathcal{N}\left(0,\frac{1}{m}\right)}\left[N(x_{1:i};\theta_i^*)\right] - N(x_{1:i};\theta_i^*)\right|.$$

We will now bound the expectation of $\mathbf{N}_i$ using McDiarmid's inequality (Fact K.13). For every $1 \leq r \leq m$, we get

$$\begin{aligned}
&\Bigg|\mathbf{N}_i\left(\left(\bar{a}_{i,1},\bar{w}_{i,1},\bar{b}_{i,1}\right)\ldots\left(\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}\right)\ldots\left(\bar{a}_{i,m},\bar{w}_{i,m},\bar{b}_{i,m}\right)\right) \\
&\quad - \mathbf{N}_i\left(\left(\bar{a}_{i,1},\bar{w}_{i,1},\bar{b}_{i,1}\right)\ldots\left(\bar{a}'_{i,r},\bar{w}'_{i,r},\bar{b}'_{i,r}\right)\ldots\left(\bar{a}_{i,m},\bar{w}_{i,m},\bar{b}_{i,m}\right)\right)\Bigg| \\
&= \sup_x\left|\bar{a}_{i,r}\sigma\left(\langle\bar{w}_{i,r}+w^*_{i,r},\tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r}+b^*_{i,r}\right)\right) - \bar{a}'_{i,r}\sigma\left(\langle\bar{w}'_{i,r}+w^*_{i,r},\tilde{x}_{1:i}\rangle + \left(\bar{b}'_{i,r}+b^*_{i,r}\right)\right)\right| \\
&= \left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\frac{2\left(c_2+c_3\right)\sqrt{2\log m}}{m} + 2U_{w_i^*}\right),
\end{aligned}$$

where last inequality follows with atleast $1-\frac{1}{c_1}-\frac{1}{c_2}-\frac{1}{c_3}$ probability by applying Lemma K.4 on $\left[\bar{a}_{i,r}\right]_{r=1}^{m}$, $\left[\bar{w}_{i,r}\right]_{r=1}^{m}$ and $\left[\bar{b}_{i,r}\right]_{r=1}^{m}$. Define $\mathbf{C}_i$ as

$$\mathbf{C}_i = \left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\frac{2\left(c_2+c_3\right)\sqrt{2\log m}}{m} + 2U_{w_i^*}\right)$$

Using Lemma 26.2 from Shalev-Shwartz and Ben-David [2014], we get

$$\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\mathbf{N}_i\right] \leq \frac{2}{m}\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\sup_x\left|\sum_{r=1}^{m}\xi_r\bar{a}_{i,r}\sigma\left(\langle\bar{w}_{i,r}+w_{i,r}^*,\right\rangle + \left(\bar{b}_{i,r}+b_{i,r}^*\right)\right)\right|\right]$$

where $\xi_1,\xi_2,\ldots,\xi_m$ are independent Rademacher random variables. Using Lipschitz continuity of ReLU activation, we get

$$\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\mathbf{N}_i\right] \leq \frac{2}{m}\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\sup_x\left|\sum_{r=1}^{m}\xi_r\bar{a}_{i,r}\left(\langle\bar{w}_{i,r}+w_{i,r}^*,\tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r}+b_{i,r}^*\right)\right)\right|\right]$$

Using Lemma 26.10 from Shalev-Shwartz and Ben-David [2014], we get

$$\begin{aligned}
\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\mathbf{N}_i\right] &\leq \mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\frac{\max_{r\in[m]}\left\|a_{i,r}\left(\bar{w}_{i,r}+w_{i,r}^*\right)\right\|_2}{\sqrt{m}}\right] \\
&\quad + 2\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\frac{\max_{r\in[m]}\left\|a_{i,r}\left(\bar{b}_{i,r}+b_{i,r}^*\right)\right\|_2}{\sqrt{m}}\right] \\
&\leq \left(2\frac{\left(2c_1\epsilon_a\sqrt{2\log m}\right)}{\sqrt{m}}\left(\frac{2c_2\sqrt{2\log m}}{\sqrt{m}} + U_{w_{i,r}^*}\right)\right) + 2\frac{\left(2c_1\epsilon_a\sqrt{2\log m}\right)}{\sqrt{m}}\frac{2c_3\sqrt{2\log m}}{\sqrt{m}}.
\end{aligned}$$

For $m \geq \Omega\left(\frac{d^{10}\left(\sum_{i=1}^{d}\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^{12}}{\epsilon_a^2\epsilon^8}\right)$, we have $U_{w_{i,r}^*} \leq \frac{\left(c_2+c_3\right)\sqrt{2\log m}}{\sqrt{m}}$ and therefore, we get

$$\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[\mathbf{N}_i\right] \leq \frac{24c_1\left(c_2+c_3\right)\epsilon_a\log m}{\sqrt{m}}.$$

Using McDiarmid's inequality (Fact K.13), we get

$$\Pr\left(\mathbf{N}_i - \frac{24c_1\left(c_2+c_3\right)\epsilon_a\log m}{\sqrt{m}} \geq \frac{\epsilon}{2}\right) \leq \Pr\left(\mathbf{N}_i - \mathbb{E}\left[\mathbf{N}_i\right] \geq \frac{\epsilon}{2}\right) \leq \exp\left(-\frac{\epsilon^2}{2mC_i^2}\right)$$

For $m \geq \Omega\left(\frac{d^{10}\left(\sum_{i=1}^{d}\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^{12}}{\epsilon_a^2\epsilon^8}\right)$, with at least $1-\frac{1}{c_1}-\frac{1}{c_2}-\frac{1}{c_3}-\exp\left(-\frac{\epsilon^2}{2mC_i^2}\right)$ probability, for all x with $\|x\|_2 \leq 1$, we have

$$\left|\mathbb{E}_{\bar{a}_{i,r},\bar{w}_{i,r},\bar{b}_{i,r}}\left[N(x_{1:i};\theta_i^*)\right] - N(x_{1:i};\theta_i^*)\right| \leq \epsilon \tag{E.17}$$

To bound V, by Eq. (D.9), we know

$$\begin{aligned}
\mathrm{V} &= \left|N(x_{1:i};\theta_i^*) - P(x_{1:i};\theta_i^*)\right| \\
&\overset{\text{(i)}}{\leq} 24c_1\epsilon_a U_{w_{i,r}^*}\left|\overline{\mathcal{H}}_i\right|\sqrt{2\log m} \\
&\overset{\text{(ii)}}{\leq} 24c_1\epsilon_a U_{w_{i,r}^*}\left(c_4 m\frac{4U_{w_{i,r}^*}\sqrt{m}}{\sqrt{\pi}}\right)\sqrt{2\log m} \\
&= \frac{96c_1c_4\epsilon_a m^{1.5}U_{w_{i,r}^*}^2\sqrt{2\log m}}{\sqrt{\pi}},
\end{aligned} \tag{E.18}$$

where inequality (i) follows from Eq. (D.9) with atleast $1-\frac{1}{c_1}$ probability and inequality (ii) follows from Lemma D.2 with atleast $1-\frac{1}{c_1}-\exp\left(-\frac{32(c_4-1)^2m^2U^2_{w^*_{i,r}}}{\pi}\right)$. Using Lemma E.7, Eq.(E.15), Eq.(E.16), Eq.(E.17) and Eq.(E.18), with atleast $1-\frac{1}{c_1}-\frac{1}{c_2}-\frac{1}{c_3}-\exp\left(-\frac{\epsilon^2}{2mC_i^2}\right)-\exp\left(-\frac{32(c_4-1)^2m^2U^2_{w^*_{i,r}}}{\pi}\right)$ probability, we get

$$\begin{aligned}\left|\phi^{-1}\left(\frac{\partial F_i^*\left(x_{1:i}\right)}{\partial x_{1:i}}\right)-P(x_{1:i};\theta_i^*)\right| &\le p_i\epsilon+\frac{96c_1c_4\epsilon_a m^{1.5}U^2_{w^*_{i,r}}\sqrt{2\log m}}{\sqrt{\pi}}+\epsilon+\frac{96c_1c_4\epsilon_a m^{1.5}U^2_{w^*_{i,r}}\sqrt{2\log m}}{\sqrt{\pi}}\\ &=\left(p_i+1\right)\epsilon+\frac{192c_1c_4\epsilon_a m^{1.5}U^2_{w^*_{i,r}}\sqrt{2\log m}}{\sqrt{\pi}}.\end{aligned}$$

□

Lemma E.9. *For any $\epsilon\in(0,1)$, for any target function F^*, for any $m\ge\Omega\left(\frac{d^{10}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^{12}}{\epsilon_a^2\epsilon^8}\right)$ and for any x with $\|x\|_2\le 1$, there exist a set of parameters $\theta^*=\left(\theta_1^*,\theta_2^*,\ldots,\theta_d^*\right)$ such that, with atleast $1-\frac{d}{c_1}-\frac{d}{c_2}-\frac{d}{c_3}-d\exp\left(-\frac{\epsilon^2}{2mC_i^2}\right)-d\exp\left(-\frac{32(c_4-1)^2m^2U^2_{w_i^*}}{\pi}\right)$ probability, we get*

$$\left|\tilde{L}\left(\nabla g^*,x\right)-\tilde{L}\left(\nabla F^*,x\right)\right|\le 3\left(\sum_{i=1}^d p_i+d\right)\epsilon+\frac{576c_1c_4\epsilon_a m^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^d U^2_{w_i^*}\right).$$

Proof. Using definition of $\tilde{L}$, we get

$$\begin{aligned}\left|\tilde{L}\left(\nabla g^*,x\right)-\tilde{L}\left(\nabla F^*,x\right)\right| &\le \left|\sum_{i=1}^d\sum_{j=1}^Q\Delta_x\left(\nabla_i g_i^*\left(\tau_j\left(x_{1:i}\right)\right)\right)-\sum_{i=1}^d\sum_{j=1}^Q\Delta_x\left(\nabla_i F_i^*\left(\tau_j\left(x_{1:i}\right)\right)\right)\right|\\ &\quad+\left|\sum_{i=1}^d\log\left(\nabla_i g_i^*\left(x_{1:i}\right)\right)-\sum_{i=1}^d\log\left(\nabla_i F_i^*\left(x_{1:i}\right)\right)\right|\\ &\le \sum_{i=1}^d\sum_{j=1}^Q\Delta_x\left|\phi\left(P\left(\tau_j\left(x_{1:i}\right),\theta_i^*\right)\right)-\left(\nabla_i F_i^*\left(\tau_j\left(x_{1:i}\right)\right)\right)\right|\\ &\quad+\sum_{i=1}^d\left|\log\left(\nabla_i g_i^*\left(x_{1:i}\right)\right)-\log\left(\nabla_i F_i^*\left(x_{1:i}\right)\right)\right|\\ &\overset{\text{(i)}}{\le} \sum_{i=1}^d\sum_{j=1}^Q\Delta_x\left|P\left(\tau_j\left(x_{1:i}\right),\theta_i^*\right)-\phi^{-1}\left(\nabla_i F_i^*\left(\tau_j\left(x_{1:i}\right)\right)\right)\right|\\ &\quad+\sum_{i=1}^d\left|P(x_{1:i};\theta_i^*)-\phi^{-1}\left(\nabla_i F_i^*\left(x_{1:i}\right)\right)\right|\\ &\le 2\left(\sum_{i=1}^d p_i+d\right)\epsilon+\frac{384c_1c_4\epsilon_a m^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^d U^2_{w_i^*}\right)\\ &\quad+\left(\sum_{i=1}^d p_i+d\right)\epsilon+\frac{192c_1c_4\epsilon_a m^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^d U^2_{w_i^*}\right)\\ &\le 3\left(\sum_{i=1}^d p_i+d\right)\epsilon+\frac{576c_1c_4\epsilon_a m^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^d U^2_{w_i^*}\right),\end{aligned}$$

where inequality (i) follows from 1-Lipschitz continuity of $\phi(\cdot)$ and $\log\left(\phi\left(\cdot\right)\right)$. The upper bound on $\|\theta^*\|_{2,\infty}$ is given by

$$\|\theta^*\|_{2,\infty}\le\sum_{i=1}^d\|\theta_i^*\|_{2,\infty}\le\sum_{i=1}^d\frac{\sqrt{\pi}\left(\sum_{r=1}^{p_i}U_{\omega_{i,r}}\right)}{m\epsilon_a\sqrt{2}}=\frac{\sqrt{\pi}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{\omega_{i,r}}\right)}{m\epsilon_a\sqrt{2}}.$$

□

We define upper bound on $\|\theta^*\|_{2,\infty}$ as U_{θ^*}:

$$U_{\theta^*} = \frac{\sqrt{\pi}\left(\sum_{i=1}^{d}\sum_{r=1}^{p_i} U_{\omega_{i,r}}\right)}{m\epsilon_a\sqrt{2}}.$$

F Optimization

This section shows that SGD on the loss of the neural network can be closely approximated by the SGD on the loss of the pseudo-network (Theorem F.3). Since the loss function of the pseudo-network is convex in its parameters (Lemma F.1), we get global optimization of the pseudo network, and hence, global optimization of the neural network. Moreover, there exist a pseudo-network which can approximation the target function and achieve training loss close to the trainign loss of the target function (Section E). Therefore, SGD on the loss of the neural network can achieve training loss comparable to training loss of the target function (Theorem F.3).

First, we will start with proving convexity of the loss function of the pseudo-network.

Lemma F.1. *(Convexity of the loss function of the pseudo-network) The loss function of the pseudo-network is convex with respect to the parameters of the neural network, and therefore, loss $\tilde{L}$ satisfies first order condition of convexity for all $t \in [T]$ and for all x with $\|x\|_2 \leq 1$:*

$$\tilde{L}(\nabla g^{(t)}, \mathcal{X}) - \tilde{L}(\nabla g^*, \mathcal{X}) \leq \langle \nabla_\theta \tilde{L}(\nabla g^{(t)}, \mathcal{X}), \theta^{(t)} - \theta^* \rangle.$$

Proof. We decompose the loss function of the pseudo-network for each dimension into two parts:

$$\tilde{L}\left(\nabla g^{(t)}, x\right) = \sum_{i=1}^{d}\left(\sum_{j=1}^{Q} \Delta_x \nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right) - \log\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)\right) = \sum_{i=1}^{d}\left(\tilde{L}_{i,1}(\nabla g^{(t)}, x) + \tilde{L}_{i,2}(\nabla g^{(t)}, x)\right),$$

where

$$\tilde{L}_{i,1}(\nabla g^{(t)}, x) = \sum_{j=1}^{Q} \Delta_x \nabla_i f_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right) \qquad \text{and} \qquad \tilde{L}_{i,2}(\nabla g^{(t)}, x) \qquad = -\log\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right).$$

We prove convexity of both $\tilde{L}_{i,1}(\nabla g^{(t)}, x)$ and $\tilde{L}_{i,2}(\nabla g^{(t)}, x)$. We can write $\tilde{L}_{i,1}\left(\nabla g^{(t)}, x\right)$ as

$$\tilde{L}_{i,1}\left(\nabla g^{(t)}, x\right) = \sum_{j=1}^{Q} \Delta_x \phi\left(P(x_{1:i}; \theta_i^{(t)})\right).$$

Note that $\phi\left(P(x_{1:i}; \theta_i^{(t)})\right)$ is convex in $P(x_{1:i}; \theta_i^{(t)})$ and $P(x_{1:i}; \theta_i^{(t)})$ is linear in $\theta_i^{(t)}$. As composition of any convex and linear function is convex, $\phi\left(P(x_{1:i}; \theta_i^{(t)})\right)$ is convex. The first part of loss function $\tilde{L}_{i,1}\left(\nabla g^{(t)}, x\right)$ is convex in $\theta_i^{(t)}$ because sum of convex functions is also convex. By writing $\tilde{L}_{i,2}\left(\nabla g^{(t)}, x\right)$ in parts, we get

$$\begin{aligned}
\tilde{L}_{i,2}\left(\nabla g^{(t)}, x\right) &= -\log\left(\phi\left(P(x_{1:i}; \theta_i^{(t)})\right)\right) \\
&= -\log\left(\exp\left(P(x_{1:i}; \theta_i^{(t)})\right)\mathbb{I}\left[P(x_{1:i}; \theta_i^{(t)}) \leq 0\right] + \left(P(x_{1:i}; \theta_i^{(t)}) + 1\right)\mathbb{I}\left[P(x_{1:i}; \theta_i^{(t)}) \geq 0\right]\right) \\
&= -P(x_{1:i}; \theta_i^{(t)})\mathbb{I}\left[P(x_{1:i}; \theta_i^{(t)}) \leq 0\right] - \log\left(P(x_{1:i}; \theta_i^{(t)}) + 1\right)\mathbb{I}\left[P(x_{1:i}; \theta_i^{(t)}) \geq 0\right].
\end{aligned}$$

Using last equality in the above equation, we can see that $\tilde{L}_{i,2}$ is convex in $P(x_{1:i}; \theta_i^{(t)})$ and we know that $P(x_{1:i}; \theta_i^{(t)})$ is linear in $\theta_i^{(t)}$. Therefore, $\tilde{L}_{i,2}$ is convex in $\theta_i^{(t)}$ because composition of any convex and linear function is a convex function. As $\tilde{L}_{i,1}$ and $\tilde{L}_{i,2}$ are convex, $\tilde{L}$ is also convex in $\theta_i^{(t)}$ because sum of convex functions is a convex function. □

Remark F.2. *When we use the standard Gaussian for the base distribution, then the loss function will be:*

$$\tilde{L}\left(\nabla g^{(t)}, x\right) = \sum_{i=1}^{d}\left(\left(\sum_{j=1}^{Q}\Delta_x \nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right)^2 - \log\left(\nabla_i g_i^{(t)}\left(x_{1:i}\right)\right)\right)$$
$$= \sum_{i=1}^{d}\left(\tilde{L}_{i,1}(\nabla g^{(t)}, x) + \tilde{L}_{i,2}(\nabla g^{(t)}, x)\right).$$

Note that the second term in the decomposition $\tilde{L}_{i,2}$ is convex with same argument given in Lemma F.1 and the first term $\tilde{L}_{i,1}$ is given by

$$\tilde{L}_{i,1}\left(\nabla g^{(t)}, x\right) = \left(\sum_{j=1}^{Q}\Delta_x \nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)\right)^2.$$

Using the same argument given in Lemma F.1, we get that $\sum_{j=1}^{Q}\Delta_x \nabla_i g_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right)$ is convex in $\theta_i^{(t)}$ but each summand in $\tilde{L}_{i,1}$ is square of convex function, which may not be convex in $\theta_i^{(t)}$. Therefore, $\tilde{L}_{i,1}$ can be non-convex in $\theta_i^{(t)}$.

Recall that average loss of function $f^{(t)}$ on training set $\mathcal{X}$ is defined as $\tilde{L}\left(\nabla f^{(t)}, \mathcal{X}\right)$:

$$\tilde{L}\left(\nabla f^{(t)}, \mathcal{X}\right) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\tilde{L}\left(\nabla f^{(t)}, x\right).$$

Similarly, average loss for $g^{(t)}$ and average loss for F^* is denoted by $\tilde{L}\left(\nabla g^{(t)}, \mathcal{X}\right)$ and $\tilde{L}\left(\nabla F^*, \mathcal{X}\right)$, respectively.

Theorem F.3. *(SGD achieves near-optimal loss) For every $\epsilon \in (0,1)$, for every $m > \text{poly}\left(U_{\theta^*}, d, \frac{1}{\epsilon}\right)$, learning rate $\eta = \tilde{O}\left(\frac{1}{m\epsilon}\right)$ and number of steps $T = O\left(\frac{U_{\theta^*}^2 \log m}{\epsilon^2}\right)$ such that, with at least 0.94 probability, we get*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)}, \mathcal{X})] - \tilde{L}(\nabla F^*, \mathcal{X}) \le O(\epsilon).$$

Proof. Recall that ∇g^* is a pseudo network which approximates the target function ∇F^*. From Lemma F.1, we know that $\tilde{L}(\nabla g^{(t)}, \mathcal{X})$ is convex in parameters θ, which gives

$$\begin{aligned}\tilde{L}(\nabla g^{(t)}, \mathcal{X}) - \tilde{L}(\nabla g^*, \mathcal{X}) &\le \langle \nabla_\theta \tilde{L}(\nabla g^{(t)}, \mathcal{X}), \theta^{(t)} - \theta^*\rangle \\ &\le \|\nabla_\theta \tilde{L}(\nabla g^{(t)}, \mathcal{X}) - \nabla_\theta \tilde{L}(\nabla f^{(t)}, \mathcal{X})\|_{2,1}\|\theta^{(t)} - \theta^*\|_{2,\infty} \\ &\quad + \langle \nabla_\theta \tilde{L}(\nabla f^{(t)}, \mathcal{X}), \theta^{(t)} - \theta^*\rangle. \end{aligned} \tag{F.1}$$

Recall that SGD update at time t is given by

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \tilde{L}(\nabla f^{(t)}, x^{(t)}).$$

Using SGD update at time t, We have

$$\begin{aligned}\|\theta^{(t+1)} - \theta^*\|_{2,2}^2 &= \|\theta^{(t)} - \eta \nabla_\theta \tilde{L}(\nabla f^{(t)}, x^{(t)}) - \theta^*\|_{2,2}^2 \\ &= \|\theta^{(t)} - \theta^*\|_{2,2}^2 + \eta^2\|\nabla_\theta \tilde{L}(\nabla f^{(t)}, x^{(t)})\|_{2,2}^2 - 2\eta\langle \theta^{(t)} - \theta^*, \nabla_\theta \tilde{L}(\nabla f^{(t)}, x^{(t)})\rangle.\end{aligned}$$

By taking expectation wrt x_t, we get

$$\mathbb{E}_{x^{(t)}}\left[\|\theta^{(t+1)} - \theta^*\|_{2,2}^2\right] = \|\theta^{(t)} - \theta^*\|_{2,2}^2 + \eta^2 \mathbb{E}_{x^{(t)}}\left[\|\nabla_\theta \tilde{L}(\nabla f^{(t)}, x^{(t)})\|_{2,2}^2\right] - 2\eta\langle \nabla_\theta \tilde{L}(\nabla f^{(t)}, \mathcal{X}), \theta^{(t)} - \theta^*\rangle. \tag{F.2}$$

Putting value of $\langle \nabla_\theta \tilde{L}(\nabla f^{(t)}, \mathcal{X}), \theta^{(t)} - \theta^*\rangle$ from Eq.(F.2) to (F.1), we get

$$\begin{aligned}\tilde{L}(\nabla g^{(t)}, \mathcal{X}) - \tilde{L}(\nabla g^*, \mathcal{X}) &\le \left\|\nabla_\theta \tilde{L}(\nabla g^{(t)}, \mathcal{X}) - \nabla_\theta \tilde{L}(\nabla f^{(t)}, x)\right\|_{2,1}\|\theta^{(t)} - \theta^*\|_{2,\infty} \\ &\quad + \frac{\|\theta^{(t)} - \theta^*\|_{2,2}^2 - \mathbb{E}_{x^{(t)}}\|\theta^{(t+1)} - \theta^*\|_{2,2}^2}{2\eta} \\ &\quad + \frac{\eta}{2}\mathbb{E}_{x^{(t)}}\|\nabla_\theta \tilde{L}(f'^{(t)}, x^{(t)})\|_{2,2}^2.\end{aligned}$$

By (D.3), (D.4) and (D.5), with atleast $1 - \frac{1}{c_1}$ probability, we have

$$\left\|\nabla_\theta \tilde{L}(\nabla f^{(t)}, x^{(t)})\right\|_{2,2}^2 \leq 2m\bar{\Lambda}^2.$$

Averaging from $t = 0$ to $T-1$, we get

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}\left[\tilde{L}(\nabla g^{(t)}, \mathcal{X})\right] - \tilde{L}(\nabla g^*, \mathcal{X}) \leq \frac{1}{T}\sum_{t=0}^{T-1}\left[\left\|\nabla_\theta \tilde{L}(\nabla g^{(t)}, \mathcal{X}) - \nabla_\theta \tilde{L}(\nabla f^{(t)}, x)\right\|_{2,1}\|\theta^{(t)} - \theta^*\|_{2,\infty}\right] \\ + \frac{\|\theta^{(0)} - \theta^*\|_{2,2}^2}{2\eta T} \\ + \frac{\eta}{2}\frac{1}{T}\sum_{t=0}^{T-1}\left[\mathbb{E}_{x^{(t)}}\|\nabla_\theta \tilde{L}(f'^{(t)}, x^{(t)})\|_{2,2}^2\right].\end{aligned}$$

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla g^{(t)}, \mathcal{X})] - \tilde{L}(\nabla g^*, \mathcal{X}) &\leq \Gamma\left(\sup_{t\in[T]}\|\theta^{(t)}\|_{2,\infty} + \|\theta^*\|_{2,\infty}\right) + \frac{\|\theta^{(0)} - \theta^*\|_{2,2}^2}{2\eta T} + \eta m\bar{\Lambda}^2 \\ &= \Gamma\left(\sup_{t\in[T]}\|\theta^{(t)}\|_{2,\infty} + \|\theta^*\|_{2,\infty}\right) + \frac{\|\theta^*\|_{2,2}^2}{2\eta T} + \eta m\bar{\Lambda}^2, \qquad \text{(F.3)}\end{aligned}$$

where last inequality follows with atleast $1 - \frac{d}{c_1} - d\exp\left(-\frac{32(c_4-1)^2\eta^2 m^2\bar{\Lambda}^2 t^2}{\pi}\right)$. Recall that Γ was defined in (D.18). The last equality also uses the fact that initial change in weights $\theta^{(0)}$ is equal to $(0, 0, \ldots, 0)$. Using Lemmas D.5 and E.9 respectively, with probability at least $1 - \frac{d}{c_1} - \frac{d}{c_2} - \frac{d}{c_3} - \sum_{t=1}^{T} d\exp\left(-\frac{32(c_4-1)^2\eta^2 m^2\bar{\Lambda}^2 t^2}{\pi}\right) - d\exp\left(-\frac{\epsilon^2}{2mC_i^2}\right) - d\exp\left(-\frac{32(c_4-1)^2 m^2 U_{w_i^*}^2}{\pi}\right)$ we have

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)}, \mathcal{X})] - \tilde{L}(\nabla g^*, \mathcal{X}) &\leq \Gamma\left(\sup_{t\in[T]}\|\theta^{(t)}\|_{2,\infty} + \|\theta^*\|_{2,\infty}\right) + \frac{\|\theta^*\|_{2,2}^2}{2\eta T} + \eta m\bar{\Lambda}^2 + 3\Lambda_{np}^{(t)}, \\ \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)}, \mathcal{X})] - \tilde{L}(\nabla F^*, \mathcal{X}) &\leq \Gamma\left(\sup_{t\in[T]}\|\theta^{(t)}\|_{2,\infty} + \|\theta^*\|_{2,\infty}\right) + \frac{\|\theta^*\|_{2,2}^2}{2\eta T} + \eta m\bar{\Lambda}^2 \\ &\quad + 3\Lambda_{np}^{(t)} + 3\left(\sum_{i=1}^{d} p_i + d\right)\epsilon + \frac{576c_1c_4\epsilon_a m^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^{d} U_{w_i^*}^2\right).\end{aligned}$$

We now choose values of η and T:

$$\begin{aligned}\eta &= \frac{\epsilon}{m\bar{\Lambda}^2} \\ &= \frac{\epsilon}{m\left(6c_1\epsilon_a\sqrt{2\log m}\right)^2} \\ &= \frac{\epsilon}{72c_1^2 m\epsilon_a^2\log m}, \\ T &:= \frac{\|\theta^*\|_{2,2}^2}{2\eta\epsilon} \\ &\leq mU_{\theta^*}^2\frac{72c_1^2 m\epsilon_a^2\log m}{2\epsilon^2} \\ &= \frac{72c_1^2 m^2 U_{\theta^*}^2\epsilon_a^2\log m}{2\epsilon^2},\end{aligned} \qquad \text{(F.4)}$$

where we use chosen value of η to get upper bound on T. Using above inequalities, we get the following equalities:

$$\begin{aligned}\frac{\|\theta^*\|_{2,2}^2}{2\eta T} &= \frac{\|\theta^*\|_{2,2}^2}{2\eta}\frac{2\eta\epsilon}{\|\theta^*\|_{2,2}^2} = \epsilon, \\ \eta m\bar{\Lambda}^2 &= \frac{\epsilon}{m\bar{\Lambda}^2}m\bar{\Lambda}^2 = \epsilon.\end{aligned}$$

Using Lemma E.9, we get

$$\|\theta^*\|_{2,\infty} \leq U_{\theta^*},$$
$$\|\theta^*\|_{2,2} \leq \sqrt{m}\|\theta^*\|_{2,\infty} = \sqrt{m}U_{\theta^*}.$$

To get value of m, we will first upper bound $\sup_{t\in[T]} \|\theta^{(t)}\|_\infty$, $\sup_{t\in[T]} \|\theta^{(t)}\|_\infty + \|\theta^*\|_\infty$ and Γ:

$$\begin{aligned}
\sup_{t\in[T]} \|\theta^{(t)}\|_\infty &= \sup_{t\in[T]} \eta\bar{\Lambda}t = \eta\bar{\Lambda}T = \frac{\|\theta^*\|_2^2\bar{\Lambda}}{2\epsilon} \leq mU_{\theta^*}^2 \frac{\left(6c_1\epsilon_a\sqrt{2\log m}\right)}{2\epsilon} \\
&= \frac{\left(3c_1 mU_{\theta^*}^2\epsilon_a\sqrt{2\log m}\right)}{\epsilon} \\
\sup_{t\in[T]} \|\theta^{(t)}\|_\infty + \|\theta^*\|_\infty &\leq \frac{\left(3c_1 mU_{\theta^*}^2\epsilon_a\sqrt{2\log m}\right)}{\epsilon} + U_{\theta^*}^2 \leq \frac{\left((1+3c_1)\, mU_{\theta^*}^2\epsilon_a\sqrt{2\log m}\right)}{\epsilon} \\
\Gamma &= \frac{192d\eta m^{1.5}\bar{\Lambda}c_1c_4\epsilon_a t\sqrt{\log m}}{\sqrt{\pi}} + 24c_1 d\epsilon_a m\Lambda_{np}^{(t)}\sqrt{2\log m} \\
&\leq \frac{192d\eta m^{1.5}\bar{\Lambda}c_1c_4\epsilon_a t\sqrt{\log m}}{\sqrt{\pi}} + 24c_1 d\epsilon_a m\sqrt{2\log m}\left(\frac{192\eta^2 m^{1.5}\bar{\Lambda}^2 c_1c_4\epsilon_a t^2\sqrt{\log m}}{\sqrt{\pi}}\right) \\
&\leq \frac{192d\eta m^{1.5}\bar{\Lambda}c_1c_4\epsilon_a t\sqrt{\log m}}{\sqrt{\pi}} + \frac{4608\sqrt{2}c_1^2c_4d\epsilon_a^2\eta^2t^2m^{2.5}\log m\bar{\Lambda}^2}{\sqrt{\pi}} \\
&\leq \frac{192dm^{1.5}c_1c_4\epsilon_a\sqrt{\log m}}{\sqrt{\pi}}\left(\frac{mU_{\theta^*}^2}{2\epsilon}\right)\left(6c_1\epsilon_a\sqrt{2\log m}\right) \\
&\quad + \frac{4608\sqrt{2}c_1^2c_4d\epsilon_a^2m^{2.5}\log m}{\sqrt{\pi}}\left(\frac{mU_{\theta^*}^2}{2\epsilon}\right)^2\left(6c_1\epsilon_a\sqrt{2\log m}\right)^2 \\
&\leq \frac{576\sqrt{2}dm^{2.5}c_1^2c_4\epsilon_a^2U_{\theta^*}^2\log m}{\epsilon\sqrt{\pi}} + \frac{82944\sqrt{2}c_1^4c_4d\epsilon_a^4m^{4.5}U_{\theta^*}^4(\log m)^2}{\sqrt{\pi}\epsilon^2} \\
&\leq \frac{165888\sqrt{2}c_1^4c_4d\epsilon_a^4m^{4.5}U_{\theta^*}^4(\log m)^2}{\sqrt{\pi}\epsilon^2}.
\end{aligned}$$

Multiplication of Γ and $\left(\sup_{t\in[T]} \|\theta^{(t)}\|_\infty + \|\theta^*\|_\infty\right)$ will be

$$\begin{aligned}
\Gamma\left(\sup_{t\in[T]} \|\theta^{(t)}\|_\infty + \|\theta^*\|_\infty\right) &\leq \frac{165888\sqrt{2}c_1^4c_4d\epsilon_a^4m^{4.5}U_{\theta^*}^4(\log m)^2}{\sqrt{\pi}\epsilon^2}\left(\frac{\left((1+3c_1)\, mU_{\theta^*}^2\epsilon_a\sqrt{2\log m}\right)}{\epsilon}\right) \\
&= \frac{331776c_1^4(1+3c_1)\, c_4d\epsilon_a^5m^{5.5}U_{\theta^*}^6(\log m)^{2.5}}{\sqrt{\pi}\epsilon^3} \\
&= \frac{331776c_1^4(1+3c_1)\, c_4d\epsilon_a^5m^{5.5}(\log m)^{2.5}}{\sqrt{\pi}\epsilon^3}\left(\frac{\sqrt{\pi}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)}{m\epsilon_a\sqrt{2}}\right)^6 \\
&= \frac{41472\pi^{2.5}c_1^4(1+3c_1)\, c_4d(\log m)^{2.5}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^6}{\sqrt{m}\epsilon^3\epsilon_a}.
\end{aligned}$$

Taking m as

$$m \geq \Omega\left(\frac{c_1^8c_4^2d^2(1+3c_1)^2\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^{12}}{\epsilon_a^2\epsilon^8}\right), \tag{F.5}$$

we get

$$\Gamma\left(\sup_{t\in[T]}\|\theta^{(t)}\|_\infty+\|\theta^*\|_\infty\right)\leq\epsilon.$$

Using (D.10), we get

$$\begin{aligned}\Lambda_{np}^{(t)}&=\left(\frac{192\eta^2m^{1.5}\bar{\Lambda}^2c_1c_4\epsilon_at^2\sqrt{\log m}}{\sqrt{\pi}}\right)\\&\leq\left(\frac{192m^{1.5}c_1c_4\epsilon_a\sqrt{\log m}}{\sqrt{\pi}}\right)\left(\frac{mU_{\theta^*}^2}{2\epsilon}\right)^2\left(6c_1\epsilon_a\sqrt{2\log m}\right)^2\\&=\frac{3456m^{3.5}c_1^3c_4\epsilon_a^3U_{\theta^*}^4\left(\log m\right)^{1.5}}{\epsilon^2\sqrt{\pi}}.\end{aligned}\tag{F.6}$$

Using given choice of m from (F.5), we get

$$\begin{aligned}\Lambda_{np}^{(t)}&\leq\frac{3456m^{3.5}c_1^3c_4\epsilon_a^3\left(\log m\right)^{1.5}}{\epsilon^2\sqrt{\pi}}\left(\frac{\sqrt{\pi}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)}{m\epsilon_a\sqrt{2}}\right)^4\\&=\frac{864\pi^{1.5}c_1^3c_4\left(\log m\right)^{1.5}}{\epsilon_a\epsilon^2}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^4\left(\frac{\epsilon_a^2\epsilon^8}{c_1^8c_4^2d^2\left(1+3c_1\right)^2\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^{12}}\right)^{0.5}\\&=O\left(\frac{\epsilon^2\left(\log m\right)^{1.5}}{c_1\left(1+3c_1\right)\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^2}\right)\\&\leq O\left(\epsilon\right).\end{aligned}$$

Similarly, using given choice of m from (F.5), we get

$$\begin{aligned}\frac{576c_1c_4\epsilon_am^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^dU_{w_i^*}^2\right)&=\frac{576c_1c_4\epsilon_am^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^d\left(\frac{\sqrt{\pi}\left(\sum_{r=1}^{p_i}U_{h_{i,r}}\right)}{m\epsilon_a\sqrt{2}}\right)^2\right)\\&\leq\frac{288\sqrt{\pi}c_1c_4\sqrt{2\log m}}{m^{0.5}\epsilon_a}\left(\sum_{i=1}^d\sum_{r=1}^{p_i}U_{h_{i,r}}\right)^2\\&\leq O\left(\epsilon\right).\end{aligned}$$

Using Eq.(F.4) and Eq.(F.5), with at least $1-\frac{d}{c_1}-\frac{d}{c_2}-\frac{d}{c_3}-\sum_{t=1}^T d\exp\left(-\frac{32(c_4-1)^2\eta^2m^2\bar{\Lambda}^2t^2}{\pi}\right)-d\exp\left(-\frac{\epsilon^2}{2mC_i^2}\right)-d\exp\left(-\frac{32(c_4-1)^2m^2U_{w_i^*}^2}{\pi}\right)$ probability, we get

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)},\mathcal{X})]-\tilde{L}(\nabla F^*,\mathcal{X})\leq\ &\Gamma\left(\sup_{t\in[T]}\|\theta^{(t)}\|_{2,\infty}+\|\theta^*\|_{2,\infty}\right)+\frac{\|\theta^*\|_{2,2}^2}{2\eta T}+\eta m\bar{\Lambda}^2\\&+3\Lambda_{np}^{(t)}+3\left(\sum_{i=1}^dp_i+d\right)\epsilon+\frac{576c_1c_4\epsilon_am^{1.5}\sqrt{2\log m}}{\sqrt{\pi}}\left(\sum_{i=1}^dU_{w_i^*}^2\right)\\\leq&O(\epsilon)+3\left(\sum_{i=1}^dp_i+d\right)\epsilon.\end{aligned}$$

Taking $c_1 = 100d, c_2 = 100d, c_3 = 100d, c_4 = d+1, \epsilon_a = \frac{\epsilon}{6000 \log m} \leq \epsilon$ and rescaling ϵ as $\epsilon / \left(\sum_{i=1}^{d} p_i + d\right)$, with at least $0.97 - \sum_{t=1}^{T} d \exp\left(-\frac{32 d^2 \eta^2 m^2 \bar{\Lambda}^2 t^2}{\pi}\right) - d \exp\left(-\frac{\epsilon^2}{2mC_i^2}\right) - d \exp\left(-\frac{32 d^2 m^2 U_{w_i^*}^2}{\pi}\right)$ probability, we get

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)}, \mathcal{X})] - \tilde{L}(\nabla F^*, \mathcal{X}) \leq \ O\left(\epsilon\right).$$

To find the lower bound on probability, we use $\sum_{t=1}^{T} \frac{1}{t^2} \leq \sum_{t=1}^{\infty} \frac{1}{t^2} \leq 2$:

$$\sum_{t=1}^{T} d \exp\left(-\frac{32 d^2 \eta^2 m^2 \bar{\Lambda}^2 t^2}{\pi}\right) \overset{\text{(i)}}{\leq} \sum_{t=1}^{T} \frac{d\pi}{32 d^2 \eta^2 m^2 \bar{\Lambda}^2 t^2} = \frac{\pi \bar{\Lambda}^4}{16 d \epsilon^2 \bar{\Lambda}^2} \leq \frac{\pi \bar{\Lambda}^2}{16 d \epsilon^2} \leq \frac{\pi}{3200} \leq 0.01.$$

where inequality (i) follows from $\exp(-x) \leq \frac{1}{x}$ for all $x \geq 0$. To find lower bound on $d \exp\left(-\frac{\epsilon^2}{2mC_i^2}\right)$, we use same inequality:

$$d \exp\left(-\frac{\epsilon^2}{2mC_i^2}\right) \leq \frac{2dmC_i^2}{\epsilon^2} = \frac{2dm}{\epsilon^2} \left(2c_1 \epsilon_a \sqrt{2 \log m}\right) \left(\frac{2\left(c_2 + c_3\right)\sqrt{2 \log m}}{m} + 2\frac{\sqrt{\pi}\left(\sum_{r=1}^{p_i} U_{h_{i,r}}\right)}{m \epsilon_a \sqrt{2}}\right)^2 \leq 0.01$$

where last inequality follows from given choice (Eq. (F.5)) of sufficiently high m. Now, we will lower bound $d \exp\left(-\frac{32 d^2 m^2 U_{w_i^*}^2}{\pi}\right)$ quantity:

$$d \exp\left(-\frac{32 d^2 m^2 U_{w_i^*}^2}{\pi}\right) \leq \frac{\pi d}{32 d^2 m^2 U_{w_i^*}^2} = \frac{\pi d}{32 d^2 m^2} \frac{2 m^2 \epsilon_a^2}{\pi \left(\sum_{r=1}^{p_i} U_{h_{i,r}}\right)^2} = \frac{\epsilon_a^2}{16 d \left(\sum_{r=1}^{p_i} U_{h_{i,r}}\right)^2} \leq 0.01$$

where last inequality follows from the value of ϵ_a. Finally, we can say that, with at least 0.94 probability, we get

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\text{sgd}}[\tilde{L}(\nabla f^{(t)}, \mathcal{X})] - \tilde{L}(\nabla F^*, \mathcal{X}) \leq O(\epsilon).$$

□

G Generalization

In this section, we prove generalization guarantees to complement our optimization result, and complete the proof of our main theorem (Theorem G.6) about efficiently learning distributions using univariate normalizing flows. Recall that $\tilde{L}(\nabla f^{(t)}, \mathcal{X})$ denotes an empirical average of $\tilde{L}(\nabla f^{(t)}, \mathcal{X})$ over training data and $\tilde{L}(\nabla f^{(t)}, \mathcal{D})$ denotes expectation with respect to underlying data distribution. The proof in this section can be broadly divided two parts. First, we prove that empirical average $\tilde{L}(\nabla f^{(t)}, \mathcal{X})$ and $\tilde{L}(\nabla F^*, \mathcal{X})$ are close to expectation $\tilde{L}(\nabla f^{(t)}, \mathcal{D})$ and $\tilde{L}(\nabla F^*, \mathcal{D})$, respectively (Lemma G.3 and Lemma G.4). Second, we prove that $\tilde{L}(\nabla f^{(t)}, \mathcal{D})$ and $\tilde{L}(\nabla F^*, \mathcal{D})$ are close to $L(f^{(t)}, \mathcal{D})$ and $L(F^*, \mathcal{D})$, respectively (Theorem G.6).

Recall that the approximate loss function $\tilde{L}$ is given by

$$\tilde{L}\left(\nabla f^{(t)}, x\right) = \sum_{i=1}^{d} \left(\sum_{j=1}^{Q} \Delta_x \phi\left(N\left(\tau_j\left(x_{1:i}\right); \theta_i^{(t)}\right)\right) - \log\left(\phi\left(N\left(x_{1:i}; \theta_i^{(t)}\right)\right)\right)\right),$$

where

$$N(x_{1:i}, \theta_i^{(t)}) = \sum_{r=1}^{m} \bar{a}_{i,r} \sigma\left(\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right)\right).$$

Similarly, we define $\tilde{L}\left(\nabla F^*, x\right)$ for the target function F^*.

Lemma G.1. *(Empirical Rademacher complexity for two-layer neural network) For every constant $B > 0$, for any number of training samples $n \geq 1$, for any time $t \geq 1$, with probability at least $1 - \frac{1}{c_1}$ over random initialization, the empirical Rademacher complexity is bounded by*

$$\frac{1}{n}\mathbb{E}_{\xi\in\{\pm 1\}^n}\left[\sup_{\max_{r\in[m]}\left\|w_{i,r}^{(t)}\right\|,\left|b_{i,r}^{(t)}\right|\leq B}\sum_{j=1}^{n}\xi_j N\left(\left(x_{1:i}\right)_j,\theta_i^{(t)}\right)\right] \leq \frac{8c_1\epsilon_a Bm\sqrt{2\log m}}{\sqrt{n}},$$

where $\left(x_{1:i}\right)_j$ denotes first i dimension of j^{th} training example.

Proof. Using part (a) of Lemma K.16, we get that $\{x \mapsto \langle w_{i,r}^{(t)}, \tilde{x}_{1:i}\rangle + b_{i,r}^{(t)} \mid \left\|w_{i,r}^{(t)}\right\|_2 \leq B, \left|b_{i,r}^{(t)}\right| \leq B\}$ has Rademacher complexity $\frac{2B}{\sqrt{n}}$. Using part (b) of Lemma K.16, we get that $\{x \mapsto \langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right) \mid \left\|w_{i,r}^{(t)}\right\|_2 \leq B, \left|b_{i,r}^{(t)}\right| \leq B, \bar{w}_{i,r} \sim \mathcal{N}\left(0, \frac{1}{m}\mathbf{1}\right), \bar{b}_{i,r} \sim \mathcal{N}\left(0, \frac{1}{m}\right)\}$ has Rademacher complexity $\frac{2B}{\sqrt{n}}$. Using part (c) of Lemma K.16, we get that class of functions in $\mathcal{F} = \{x \mapsto N(x_{1:i}; \theta_i^{(t)}) \mid \max_{r\in[m]}\left\|w_{i,r}^{(t)}\right\|_2 \leq B, \max_{r\in[m]}\left|b_{i,r}^{(t)}\right| \leq B\}$ has Rademacher complexity

$$\hat{\mathcal{R}}\left(\mathcal{X};\mathcal{F}\right) \leq 2\|\mathbf{a}\|_1\frac{2B}{\sqrt{n}} \overset{\text{(i)}}{\leq} \frac{8c_1\epsilon_a Bm\sqrt{2\log m}}{\sqrt{n}},$$

where inequality (i) follows from Lemma K.4 with at least $1 - \frac{1}{c_1}$ probability over random initialization.

□

We denote $M_{\nabla F^*}$ and $m_{\nabla F^*}$ as maximum and minimum value of ∇F^*:

$$M_{\nabla F^*} = \max_{i\in[d], x\in\mathbb{R}^d} \nabla_i F_i^*\left(x_{1:i}\right)\left(x_{1:i}\right),$$
$$m_{\nabla F^*} = \min_{i\in[d], x\in\mathbb{R}^d} \nabla_i F_i^*\left(x_{1:i}\right)\left(x_{1:i}\right).$$

We find upper bound on maximum and lower bound on minimum value of the loss $\tilde{L}$ for the target function F^* in terms of $M_{\nabla F^*}$ and $m_{\nabla F^*}$:

$$\sup_x \tilde{L}\left(\nabla F^*, x\right) = \max_x \sum_{i=1}^{d}\left(\sum_{j=1}^{Q}\Delta_x\left(\nabla_i F_i^*\left(x_{1:i}\right)\left(\tau_j\left(x_{1:i}\right)\right)\right) - \log\nabla_i F_i^*\left(x_{1:i}\right)\right) \leq 2dM_{\nabla F^*} - d\log\left(m_{\nabla F^*}\right),$$
$$\inf_x \tilde{L}\left(\nabla F^*, x\right) = \min_x \sum_{i=1}^{d}\left(\sum_{j=1}^{Q}\Delta_x\left(\nabla_i F_i^*\left(x_{1:i}\right)\left(\tau_j\left(x_{1:i}\right)\right)\right) - \log\nabla_i F_i^*\left(x_{1:i}\right)\right) \geq 2dm_{\nabla F^*} - d\log\left(M_{\nabla F^*}\right),$$

and define them respectively as $M_{\tilde{L}}$ and $m_{\tilde{L}}$:

$$\begin{aligned} M_{\tilde{L}} &= 2dM_{\nabla F^*} - d\log\left(m_{\nabla F^*}\right), \\ m_{\tilde{L}} &= 2dm_{\nabla F^*} - d\log\left(M_{\nabla F^*}\right). \end{aligned} \tag{G.1}$$

Lemma G.2. *(Small value of neural network at initialization) For any dimension $i \in [d]$, for any constant $c_1 > 10, c_2 > 10$ and $c_3 > 10$, with probability at least $0.99 - \frac{1}{c_1} - \frac{1}{c_2} - \frac{1}{c_3}$, we have*

$$\left|\sum_{r=1}^{m}\bar{a}_{i,r}\sigma\left(\langle\bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right)\right| \leq 16\sqrt{(d+1)\log(d+1)}c_1c_2\epsilon_a\left(\log m\right) + 16c_1c_3\epsilon_a\left(\log m\right).$$

Proof. Suppose, for any given x, there are m' indicators with value 1. Without loss of generality, we can assume

that indicators from $r = 1$ to $r = m'$ is 1. Then,

$$\left|\sum_{r=1}^{m} \bar{a}_{i,r}\left(\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right)\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right]\right| = \left|\sum_{r=1}^{m'} \bar{a}_{i,r}\left(\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right)\right|$$
$$= \left|\langle x, \sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r}\rangle + \sum_{r=1}^{m'} \bar{a}_{i,r}\bar{b}_{i,r}\right|$$

Now, applying Hoeffding's inequality (Fact K.8) on any dimension $j \in [d+1]$ for the sum in first part of the above equation, with atleast $1 - \frac{1}{c_1} - \frac{1}{c_2}$ probability, we get

$$\Pr\left(\left|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r,j}\right| \geq t\right) \leq \exp\left(-\frac{2t^2 m}{m'\left(2c_1\epsilon_a\sqrt{2\log m}\right)^2\left(2c_2\sqrt{2\log m}\right)^2}\right)$$
$$\leq \exp\left(-\frac{t^2}{32c_1^2c_2^2\epsilon_a^2\left(\log m\right)^2}\right). \tag{G.2}$$

Using union bound, we get

$$\Pr\left(\bigcup_{j\in[d+1]}\left(\left|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r,j}\right| \geq t\right)\right) \leq (d+1)\exp\left(-\frac{t^2}{32c_1^2c_2^2\epsilon_a^2\left(\log m\right)^2}\right)$$

Using definition of $L_\infty-$norm, we have

$$\Pr\left(\left\|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r}\right\|_\infty \geq t\right) \leq (d+1)\exp\left(-\frac{t^2}{32c_1^2c_2^2\epsilon_a^2\left(\log m\right)^2}\right)$$

Plugging $t = 16\sqrt{\log(d+1)}c_1c_2\epsilon_a(\log m)$ in above equation, with probability at least $1 - \exp(-8) - \frac{1}{c_1} - \frac{1}{c_2}$, we have

$$\left\|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r}\right\|_\infty \leq 16\sqrt{\log(d+1)}c_1c_2\epsilon_a(\log m),$$

and using relation between L_2 and L_∞ norm, we have

$$\left\|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r}\right\|_2 \leq \sqrt{d+1}\left\|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r}\right\|_\infty \leq 16\sqrt{(d+1)\log(d+1)}c_1c_2\epsilon_a(\log m). \tag{G.3}$$

Similarly, using Hoeffding's inequality (Fact K.8), with at least $1 - \frac{1}{c_1} - \frac{1}{c_3}$ probability, we get

$$\Pr\left(\left|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{b}_{i,r}\right| \geq t\right) \leq \exp\left(-\frac{2t^2 m}{m'\left(2c_1\epsilon_a\sqrt{2\log m}\right)^2\left(2c_3\sqrt{2\log m}\right)^2}\right)$$
$$\leq \exp\left(-\frac{t^2}{32c_1^2c_3^2\epsilon_a^2\left(\log m\right)^2}\right).$$

Plugging $t = 16c_1c_3\epsilon_a(\log m)$, with at least $1 - \exp(-8) - \frac{1}{c_1} - \frac{1}{c_3}$ probability, we get

$$\left|\sum_{r=1}^{m'} \bar{a}_{i,r}\bar{b}_{i,r}\right| \leq 16c_1c_3\epsilon_a(\log m). \tag{G.4}$$

Using Eq.(G.3) and Eq.(G.4), with probability at least $0.99 - \frac{1}{c_1} - \frac{1}{c_2} - \frac{1}{c_3}$, we have

$$\begin{aligned}\left|\sum_{r=1}^{m} \bar{a}_{i,r}\left(\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right)\mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r} \geq 0\right]\right| &= \left|\langle x, \sum_{r=1}^{m'} \bar{a}_{i,r}\bar{w}_{i,r}\rangle + \sum_{r=1}^{m'} \bar{a}_{i,r}\bar{b}_{i,r}\right| \\ &\leq 16\sqrt{(d+1)\log(d+1)}c_1c_2\epsilon_a(\log m) + 16c_1c_3\epsilon_a(\log m).\end{aligned} \tag{G.5}$$

This completes the proof. □

Lemma G.3. *For any constant T, for any dimension $i \in [d]$, any time $1 \leq t \leq T$, any $\epsilon \in (0,1)$, suppose that the number of samples n satisfies*

$$n \geq O\left(\frac{\left(M_{\tilde{L}} - m_{\tilde{L}}\right)^2 (Q+1)^2 d^2 \log(d)\, \epsilon_a^4 U_{\theta^*}^4 m^4 (\log m)^2}{\epsilon^4}\right). \tag{G.6}$$

Then, with at least 0.98 probability over random initialization, the population loss of any functions of the set $\{x \mapsto N(x_{1:i}; \theta_i^{(t)}) \mid \left\|w_{i,r}^{(t)}\right\|_2 \leq \eta\bar{\Lambda}T, \left|b_{i,r}^{(t)}\right| \leq \eta\bar{\Lambda}T \ \ \forall r \in [m]\}$ is close to the empirical loss, i.e.

$$\left|\mathbb{E}_{x\in\mathcal{D}}\left[\tilde{L}\left(\nabla f^{(t)}, x\right)\right] - \tilde{L}\left(\nabla f^{(t)}, \mathcal{X}\right)\right| \leq \epsilon.$$

Proof. We know that the loss for i^{th} dimension $\tilde{L}_i\left(\nabla f^{(t)}, x\right)$ depends on neural network $N(x_{1:i}; \theta_i^{(t)})$ through $\left(N\left(\tau_1(x_{1:i}); \theta_i^{(t)}\right), N\left(\tau_2(x_{1:i}); \theta_i^{(t)}\right), \ldots, N\left(\tau_Q(x_{1:i}); \theta_i^{(t)}\right), N\left(x_{1:i}; \theta_i^{(t)}\right)\right)$ vector. Using Fact K.17, with at least $1-\delta$ probability, we get

$$\sup_{N\in\mathcal{F}}\left|\mathbb{E}_{x\sim\mathcal{D}}\left[\tilde{L}_i\left(\nabla f^{(t)}, x\right)\right] - \frac{1}{n}\sum_{i=1}^{n}\tilde{L}_i\left(\nabla f^{(t)}, x\right)\right| \leq 2\sqrt{2}L_s(Q+1)\hat{\mathcal{R}}(\mathcal{X};\mathcal{F}) + b_i\sqrt{\frac{\log\frac{1}{\delta}}{2n}} \tag{G.7}$$

where $\mathcal{F} = \{x \mapsto N(x_{1:i}; \theta_i^{(t)}) \mid \left\|w_{i,r}^{(t)}\right\|_2 \leq \eta\bar{\Lambda}T, \left|b_{i,r}^{(t)}\right| \leq \eta\bar{\Lambda}T \ \ \forall r \in [m]\}$. In the above equation, constant b_i denotes upper bound on the loss $\tilde{L}_i$ and $L_{s,i}$ denote standard Lipschitz constant of $\tilde{L}_i$ with respect to $\left(N\left(\tau_1(x_{1:i}); \theta_i^{(t)}\right), N\left(\tau_2(x_{1:i}); \theta_i^{(t)}\right), \ldots, N\left(\tau_Q(x_{1:i}); \theta_i^{(t)}\right), N\left(x_{1:i}; \theta_i^{(t)}\right)\right)$. We denote $L_{c,i,j}$ as j^{th} coordinate-wise Lipschitz continuity of loss $\tilde{L}_i$ function as following:

$$\begin{aligned}
L_{c,i,j} &\leq \sup_{N\in\mathcal{F}, \|x\|_2\leq 1}\left|\Delta_x\phi'\left(N\left(\tau_j(x_{1:i}), \theta_i^{(t)}\right)\right)\right| \\
&\leq \sup_{N\in\mathcal{F}, \|x\|_2\leq 1}\frac{2}{Q}\left|\phi'\left(N\left(\tau_j(x_{1:i}), \theta_i^{(t)}\right)\right)\right| \\
&\leq \frac{2}{Q} \quad \forall j \in [Q], \\
L_{c,i,Q+1} &\leq \sup_{N\in\mathcal{F}, \|x\|_2\leq 1}\frac{\phi'\left(N(x_{1:i}; \theta_i^{(t)})\right)}{\phi\left(N(x_{1:i}; \theta_i^{(t)})\right)} \\
&= \sup_{N\in\mathcal{F}, \|x\|_2\leq 1}\frac{\exp\left(N(x_{1:i}; \theta_i^{(t)})\right)\mathbb{I}\left[N(x_{1:i}; \theta_i^{(t)}) \leq 0\right] + \mathbb{I}\left[N(x_{1:i}; \theta_i^{(t)}) \geq 0\right]}{\exp\left(N(x_{1:i}; \theta_i^{(t)})\right)\mathbb{I}\left[N(x_{1:i}; \theta_i^{(t)}) \leq 0\right] + \left(N(x_{1:i}; \theta_i^{(t)}) + 1\right)\mathbb{I}\left[N(x_{1:i}; \theta_i^{(t)}) \geq 0\right]} \\
&= \sup_{N\in\mathcal{F}, \|x\|\leq 1}\mathbb{I}\left[N(x_{1:i}; \theta_i^{(t)}) \leq 0\right] + \frac{1}{N(x_{1:i}; \theta_i^{(t)}) + 1}\mathbb{I}\left[N(x_{1:i}; \theta_i^{(t)}) \geq 0\right] \\
&\leq 1
\end{aligned}$$

Using Lemma K.6, standard Lipschitz constant of $\tilde{L}_i$ is given by

$$L_{s,i} \leq \sqrt{\sum_{j=1}^{Q+1} L_{c,i,j}^2} \leq \sqrt{\frac{4}{Q}+1} \leq 2 \tag{G.8}$$

To get constant b_i (i.e., upper bound on $\tilde{L}_i$), we use Lipschitz property of $\tilde{L}_i$. We construct $\tilde{f}_i$ such that $\left(\nabla_i \tilde{f}_i\left(\tau_1\left(x_{1:i}\right)\right), \nabla_i \tilde{f}_i\left(\tau_2\left(x_{1:i}\right)\right), \ldots, \nabla_i \tilde{f}_i\left(\tau_Q\left(x_{1:i}\right)\right), \nabla_i \tilde{f}_i\left(x_{1:i}\right)\right) = (1,1,\ldots,1,1)$.

$$\begin{aligned}
\left|\tilde{L}_i\left(\nabla f^{(t)}, x\right) - \tilde{L}_i\left(\nabla \tilde{f}^{(t)}, x\right)\right| =& \left|\sum_{j=1}^{Q} \Delta_x \nabla_i f_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right) - \sum_{j=1}^{Q} \Delta_x \nabla_i \tilde{f}_i\left(\tau_j\left(x_{1:i}\right)\right)\right| \\
&+ \left|\log\left(\phi\left(N(x_{1:i};\theta_i^{(t)})\right)\right) - \log\left(\phi\left(N\left(x_{1:i},\tilde{\theta}_i\right)\right)\right)\right| \\
&\leq \sum_{j=1}^{Q} \Delta_x \left|\phi\left(N\left(\tau_j\left(x_{1:i}\right),\theta_i^{(t)}\right)\right) - \phi\left(N\left(\tau_j\left(x_{1:i}\right),\tilde{\theta}_i\right)\right)\right| \\
&+ \left|\log\left(\phi\left(N(x_{1:i};\theta_i^{(t)})\right)\right)\right| \\
&\leq \sum_{j=1}^{Q} \Delta_x \left|N\left(\tau_j\left(x_{1:i}\right),\theta_i^{(t)}\right)\right| + \left|N(x_{1:i};\theta_i^{(t)})\right|
\end{aligned} \tag{G.9}$$

Note that $\tilde{L}\left(\nabla f^{(t)}, x\right)$ depends upon $\left(N\left(\tau_1\left(x_{1:i}\right);\theta_i^{(t)}\right), N\left(\tau_2\left(x_{1:i}\right);\theta_i^{(t)}\right), \ldots, N\left(\tau_Q\left(x_{1:i}\right);\theta_i^{(t)}\right), N\left(x_{1:i};\theta_i^{(t)}\right)\right)$ vector and similarly, $\tilde{L}\left(\nabla \tilde{f}^{(t)}, x\right)$ depends upon $(0,0,0,\ldots,0,0)$. Finding upper bound $N(x_{1:i};\theta_i^{(t)})$ for all $x \in \mathbb{R}^d$ with $\|x\|_2 \leq 1$, we get

$$\begin{aligned}
\sup_{N\in\mathcal{F}, \|x\|\leq 1} N(x_{1:i};\theta_i^{(t)}) \leq& \sup_{\left\|w_{i,r}^{(t)}\right\|_2 \leq \eta\bar{\Lambda}T, \left|b_{i,r}^{(t)}\right| \leq \eta\bar{\Lambda}T, \|x\|_2 \leq 1} P(x_{1:i};\theta_i^{(t)}) + \Lambda_{np}^{(T)} \\
\leq& \sup_{\left\|w_{i,r}^{(t)}\right\|_2 \leq \eta\bar{\Lambda}T, \left|b_{i,r}^{(t)}\right| \leq \eta\bar{\Lambda}T, \|x\|_2 \leq 1} \sum_{r=1}^{m} \bar{a}_{i,r}\sigma\left(\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right) \\
&+ \sum_{r=1}^{m} \bar{a}_{i,r}\left(\langle w_{i,r}^{(t)}, \tilde{x}_{1:i}\rangle + b_{i,r}^{(t)}\right)\sigma\left(\langle \bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right) + \Lambda_{np}^{(T)} \\
\overset{\text{(i)}}{\leq}& 16\sqrt{(d+1)\log(d+1)}c_1c_2\epsilon_a(\log m) + 16c_1c_3\epsilon_a(\log m) \\
&+ m\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(2\eta\bar{\Lambda}T\right) + \Lambda_{np}^{(T)} \\
\overset{\text{(ii)}}{\leq}& 16\sqrt{(d+1)\log(d+1)}c_1c_2\epsilon_a(\log m) + 16c_1c_3\epsilon_a(\log m) \\
&+ m\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(12c_1\epsilon_a\sqrt{2\log m}\right)\left(\frac{mU_{\theta^*}^2}{2\epsilon}\right) + \Lambda_{np}^{(T)} \\
=& 16\sqrt{(d+1)\log(d+1)}c_1c_2\epsilon_a(\log m) + 16c_1c_3\epsilon_a(\log m) \\
&+ m\left(24c_1^2\epsilon_a^2\log m\left(\frac{mU_{\theta^*}^2}{\epsilon}\right)\right) + \Lambda_{np}^{(T)} \\
\leq& O\left(\frac{m^2\epsilon_a^2U_{\theta^*}^2\log m}{\epsilon}\right)
\end{aligned}$$

where inequality (i) follows from Lemma G.2, Lemma K.4 and Eq.(D.6). The inequality (ii) uses our choices of η

and T from Eq.(F.4). We define K as upper bound on $\sup_{N\in\mathcal{F},\,\|x\|_2\le 1} N(x_{1:i};\theta_i^{(t)})$:

$$K := O\left(\frac{m^2\epsilon_a^2 U_{\theta^*}^2 \log m}{\epsilon}\right). \tag{G.10}$$

Using value of K and Eq.(G.9), we get upper bound b_i on $\tilde{L}_i$:

$$b_i = 2K + K + \tilde{L}_i\left(\nabla\tilde{f}^{(t)}, x\right) = 3K + 2.$$

Using value of b_i in Eq.(G.7) and Lemma G.1, with at least $0.99 - \delta - \frac{1}{c_1} - \frac{1}{c_2} - \frac{1}{c_3}$ probability, we get

$$\begin{aligned}&\sup_{N\in\mathcal{F}}\left|\mathbb{E}_{x\in\mathcal{D}}\left[\tilde{L}_i\left(\nabla f^{(t)}, x\right)\right] - \frac{1}{n}\sum_{i=1}^{n}\tilde{L}_i\left(\nabla f^{(t)}, x_i\right)\right| \\ &\qquad\le 4\sqrt{2}\,(Q+1)\,\frac{8c_1\epsilon_a\eta\bar{\Lambda}Tm\sqrt{2\log m}}{\sqrt{n}} + (3K+2)\sqrt{\frac{\log\frac{1}{\delta}}{2n}}.\end{aligned}$$

By summing over all dimension $i\in[d]$, with atleast $0.99 - d\delta - \frac{d}{c_1} - \frac{d}{c_2} - \frac{d}{c_3}$ probability, we get

$$\begin{aligned}\sup_{N\in\mathcal{F}}\left|\mathbb{E}_{x\in\mathcal{D}}\left[\tilde{L}\left(\nabla f^{(t)}, x\right)\right] - \frac{1}{n}\sum_{i=1}^{n}\tilde{L}\left(\nabla f^{(t)}, x_i\right)\right| &\le \sum_{i=1}^{d}\sup_{N\in\mathcal{F}}\left|\mathbb{E}_{x\in\mathcal{D}}\left[\tilde{L}_i\left(\nabla f^{(t)}, x\right)\right] - \frac{1}{n}\sum_{i=1}^{n}\tilde{L}_i\left(\nabla f^{(t)}, x_i\right)\right|, \\ &\le 4\sqrt{2}d\,(Q+1)\,\frac{8c_1\epsilon_a\eta\bar{\Lambda}Tm\sqrt{2\log m}}{\sqrt{n}} + (3K+2)\,d\sqrt{\frac{\log\frac{1}{\delta}}{2n}}.\end{aligned}$$

Using $\delta = \frac{0.001}{d}$ and our choice of n given in (G.6), with probability at least 0.989, we have

$$\sup_{N\in\mathcal{F}}\left|\mathbb{E}_{x\in\mathcal{D}}\left[\tilde{L}_i\left(\nabla f^{(t)}, x\right)\right] - \frac{1}{n}\sum_{i=1}^{n}\tilde{L}_i\left(\nabla f^{(t)}, x_i\right)\right| \le \epsilon.$$

□

Lemma G.4. *(Concentration on approximated loss of target function) Suppose n is sufficiently high such that it satisfies*

$$n \ge O\left(\frac{\left(M_{\tilde{L}} - m_{\tilde{L}}\right)^2 (Q+1)^2 d^2 \log(d)\,\epsilon_a^4 U_{\theta^*}^4 m^4 (\log m)^2}{\epsilon^4}\right).$$

If n satisfies above condition, then with at least 0.9999 probability, population loss of target function $F^{*\prime}$ is close to empirical loss i.e.

$$\left|\mathbb{E}_{x\sim\mathcal{D}}\left[\tilde{L}\left(\nabla F^*, x\right)\right] - \tilde{L}\left(\nabla F^*, \mathcal{X}\right)\right| \le \epsilon.$$

Proof. Using Hoeffding's inequality (Fact K.8), we have

$$\Pr\left(\left|\mathbb{E}_{x\sim\mathcal{D}}\left[\tilde{L}\left(\nabla F^*, \mathcal{X}\right)\right] - \tilde{L}\left(\nabla F^*, \mathcal{X}\right)\right| \ge \epsilon\right) \le \exp\left(-\frac{2n\epsilon^2}{\left(M_{\tilde{L}} - m_{\tilde{L}}\right)^2}\right)$$

Taking n as

$$n \ge O\left(\frac{\left(M_{\tilde{L}} - m_{\tilde{L}}\right)^2 (Q+1)^2 d^2 \log(d)\,\epsilon_a^4 U_{\theta^*}^4 m^4 (\log m)^2}{\epsilon^4}\right),$$

with at least probability 0.9999, we get

$$\left|\mathbb{E}_{x\sim\mathcal{D}}\left[\tilde{L}\left(\nabla F^*, x\right)\right] - \tilde{L}\left(\nabla F^*, \mathcal{X}\right)\right| \leq \epsilon \tag{G.11}$$

□

Corollary G.5. *Under same setting as Theorem F.3 and*

$$n \geq O\left(\frac{\left(M_{\tilde{L}} - m_{\tilde{L}}\right)^2 (Q+1)^2 d^2 \log(d)\, \epsilon_a^4 U_{\theta^*}^4 m^4 (\log m)^2}{\epsilon^4}\right)$$

then with at least 0.94 *probability, we get*

$$\mathbb{E}_{\text{sgd}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{x\sim\mathcal{D}}\left[\tilde{L}(f'^{(t)}, x)\right]\right] - \mathbb{E}_{x\sim\mathcal{D}}\left[\tilde{L}(\nabla F^*, x)\right] \leq O(\epsilon).$$

Proof. The corollary follows from Theorem F.3, Lemma G.3 and Lemma G.4. □

Before stating our main theorem, we recall and define necessary terms used in stating the theorem. Recall that

$$\begin{aligned}
M_{\nabla F} &= \max_{i\in[d], x\in\mathbb{R}^d} \nabla_i F_i\left(x_{1:i}\right) = \max_{i\in[d], x\in\mathbb{R}^d} \frac{\partial F_i\left(x_{1:i}\right)}{\partial x_i},\\
m_{\nabla F} &= \min_{i\in[d], x\in\mathbb{R}^d} \nabla_i F_i\left(x_{1:i}\right) = \min_{i\in[d], x\in\mathbb{R}^d} \frac{\partial F_i\left(x_{1:i}\right)}{\partial x_i},\\
M_{\nabla^2 F} &= \max_{i\in[d], x\in\mathbb{R}^d} \nabla_i^2 F_i\left(x_{1:i}\right) = \max_{i\in[d], x\in\mathbb{R}^d} \frac{\partial^2 F_i\left(x_{1:i}\right)}{\partial x_i^2},\\
m_{\nabla^2 F} &= \min_{i\in[d], x\in\mathbb{R}^d} \nabla_i^2 F_i\left(x_{1:i}\right) = \min_{i\in[d], x\in\mathbb{R}^d} \frac{\partial^2 F_i\left(x_{1:i}\right)}{\partial x_i^2},\\
M_{\tilde{L}} &= \sup_x \tilde{L}\left(\nabla F^*, x\right) = 2M_{\nabla F} - \log\left(m_{\nabla F}\right),\\
m_{\tilde{L}} &= \inf_x \tilde{L}\left(\nabla F^*, x\right) = 2m_{\nabla F} - \log\left(M_{\nabla F}\right).
\end{aligned}$$

Recall that for any function $\psi : \mathbb{R} \to \mathbb{R}$ with Taylor expansion $\psi(y) = \sum_{j=0}^{\infty} c_j y^j$, then its complexity $C_0(\psi, \epsilon)$ for any $\epsilon > 0$ is given by

$$C_0(\psi, \epsilon) = O((\sum_{i=0}^{\infty}(i+1)^{1.75}|c_i|)\text{poly}(\tfrac{1}{\epsilon})),$$

which is a weighted norm of the Taylor coefficients. Recall that we define upper bound on complexity of learning any ψ function as

$$U_\psi = \max_{i\in[d], j\in[p_i]} C_0\left(\psi_{i,j}, \epsilon\right).$$

Now, we will state our main theorem.

Theorem G.6. *(loss function is close to optimal) For every* $\epsilon \in (0,1)$*, for every* $m > \text{poly}\left(U_\psi, d, \left(\max_{i\in[d]} p_i\right), \frac{1}{\epsilon}\right), \eta = \tilde{O}\left(\frac{1}{m\epsilon}\right)$ *and* $T = O\left(\frac{d^2\left(\max_{i\in[d]} p_i\right)^2 U_\psi^2 \log m}{\epsilon^2}\right)$*, for any target function* $F^{*\prime}$ *with finite second order derivative and number of quadrature points* $Q \geq \frac{2dM_{\nabla^2 F^*} + 2dK_2}{\epsilon}$ *and number of training points* $n \geq O\left(\frac{\left(M_{\tilde{L}} - m_{\tilde{L}}\right)^2 (Q+1)^2 d^6 \log(d)\left(\max_{i\in[d]} p_i\right)^4 U_\psi^4 m^4 (\log m)^2}{\epsilon^4}\right)$*, with at least* 0.94 *probability, we have*

$$\mathbb{E}_{\text{sgd}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{x\sim\mathcal{D}}\left[L(f^{(t)}, x)\right]\right] - \mathbb{E}_{x\sim\mathcal{D}}\left[L(F^*, x)\right] \leq O(\epsilon),$$

where K_2 is given by

$$K_2 = O\left(\frac{m^2 U_\psi^2 d^6 \left(\max_{i\in[d]} p_i\right)}{\epsilon}\right).$$

Proof. First, we will try to bound for all $x \in \mathbb{R}^d$ with $\|x\|_2 \leq \frac{1}{2}$:

$$\begin{aligned}\left|\tilde{L}(\nabla F^*, x) - L(F^*, x)\right| \leq& \left|\sum_{i=1}^{d}\sum_{j=1}^{Q} \Delta_x \nabla_i F_i^*\left(x_{1:i}\right)\left(\tau_j\left(x_{1:i}\right)\right) - F_i^*\left(x_{1:i}\right)\right| \\ \leq& \sum_{i=1}^{d}\left|\sum_{j=1}^{Q} \Delta_x \nabla_i F_i^*\left(x_{1:i}\right)\left(\tau_j\left(x_{1:i}\right)\right) - F_i^*\left(x_{1:i}\right)\right| \\ \leq& \frac{2dM_{\nabla^2 F}}{Q}.\end{aligned}$$

Similarly, bounding error for $f'^{(t)}$ for all $x \in \mathbb{R}^d$ with $\|x\|_2 \leq \frac{1}{2}$, we will get

$$\begin{aligned}\left|\tilde{L}(\nabla f^{(t)}, x) - L(f^{(t)}, x)\right| \leq& \left|\sum_{i=1}^{d}\left(\sum_{j=1}^{Q} \Delta_x \nabla_i f_i^{(t)}\left(\tau_j\left(x_{1:i}\right)\right) - f_i^{(t)}(x_{1:i})\right)\right| \\ \leq& \frac{2d\left(\sup_{x,i\in[d],t\in[T]} \nabla_i^2 f_i^{(t)}(x_{1:i})\right)}{Q}.\end{aligned}$$

To get $\sup_{x,i\in[d],t\in[T]} \nabla_i^2 f_i^{(t)}(x_{1:i})$, we will use Eq.(G.10).

$$
\begin{aligned}
\sup_{x,i\in[d],t\in[T]} \nabla_i^2 f_i^{(t)}(x_{1:i}) &= \sup_{x,i\in[d],t\in[T]} \left| \frac{\partial^2 f_i^{(t)}(x_{1:i})}{\partial x_i^2} \right| \\
&= \sup_{x,i\in[d],t\in[T]} \left| \frac{\partial}{\partial x_i} \left(\phi\left(N(x_{1:i};\theta_i^{(t)}) \right) \right) \right| \\
&\leq \sup_{x,i\in[d],t\in[T]} \left| \frac{\partial}{\partial x_i} N(x_{1:i};\theta_i^{(t)}) \right| \\
&\leq \sup_{x,i\in[d],t\in[T]} \left| \sum_{r=1}^{m} \bar{a}_{i,r} \sigma'\left(\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)} \right) \right) \left(\left(\bar{w}_{i,r,i} + w_{i,r,i}^{(t)} \right) \right. \right. \\
&\quad \left. \left. + \left(\bar{w}_{i,r,i+1} + w_{i,r,i+1}^{(t)} \right) \frac{x_i}{\sqrt{1-\|x_{1:i}\|^2}} \right) \right| \\
&= \sup_{x,i\in[d],t\in[T]} \sum_{r\in\mathcal{H}_i} \bar{a}_{i,r} \mathbb{I}\left[\langle \bar{w}_{i,r}, \tilde{x}_{1:i} \rangle + \bar{b}_{i,r} \geq 0 \right] \left(\left(\bar{w}_{i,r,i} + w_{i,r,i}^{(t)} \right) \right. \\
&\quad \left. + \left(\bar{w}_{i,r,i+1} + w_{i,r,i+1}^{(t)} \right) \frac{x_i}{\sqrt{1-\|x_{1:i}\|^2}} \right) + \sum_{r\in\overline{\mathcal{H}}_i^{(t)}} \bar{a}_{i,r} \mathbb{I}\left[\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, \tilde{x}_{1:i} \rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)} \right) \geq 0 \right] \\
&\quad \left(\left(\bar{w}_{i,r,i} + w_{i,r,i}^{(t)} \right) + \left(\bar{w}_{i,r,i+1} + w_{i,r,i+1}^{(t)} \right) \frac{x_i}{\sqrt{1-\|x_{1:i}\|^2}} \right) \\
&\overset{\text{(i)}}{\leq} 32 c_1 c_2 \epsilon_a (\log m) + 2m \left(2c_1\epsilon_a \sqrt{2\log m} \right) (\eta \bar{\Lambda} T) \\
&\quad + \left(c_4 m \frac{4\eta\bar{\Lambda}T\sqrt{m}}{\sqrt{\pi}} \right) \left(2c_1\epsilon_a\sqrt{2\log m} \right) \left(\frac{2c_2\sqrt{2\log m}}{\sqrt{m}} \right) \\
&\quad + \left(c_4 m \frac{4\sqrt{m}}{\sqrt{\pi}} \right) \left(2c_1\epsilon_a\sqrt{2\log m} \right) (\eta\bar{\Lambda}T)^2 \\
&\leq 32 c_1 c_2 \epsilon_a (\log m) + 2m \left(2c_1\epsilon_a \sqrt{2\log m} \right) \left(\frac{\left(3c_1 m U_{\theta^*}^2 \epsilon_a \sqrt{2\log m} \right)}{\epsilon} \right) \\
&\quad + \left(c_4 m \frac{4\sqrt{m}}{\sqrt{\pi}} \right) \left(2c_1\epsilon_a\sqrt{2\log m} \right) \left(\frac{2c_2\sqrt{2\log m}}{\sqrt{m}} \right) \left(\frac{\left(3c_1 m U_{\theta^*}^2 \epsilon_a \sqrt{2\log m} \right)}{\epsilon} \right) \\
&\quad + \left(c_4 m \frac{4\sqrt{m}}{\sqrt{\pi}} \right) \left(2c_1\epsilon_a\sqrt{2\log m} \right) \left(\frac{\left(3c_1 m U_{\theta^*}^2 \epsilon_a \sqrt{2\log m} \right)}{\epsilon} \right)^2 \\
&\overset{\text{(ii)}}{\leq} O\left(d^2\epsilon \right) + O\left(d^2 m^2 U_{\theta^*}^2 \epsilon \right) + O\left(m^2 U_{\theta^*}^2 d^4 \epsilon \right) + O\left(m^{3.5} U_{\theta^*}^4 \epsilon^2 \right) \\
&\leq O\left(m^2 U_{\theta^*}^2 d^4 \epsilon \right),
\end{aligned}
$$

where inequality (i) follows by plugging $t = 16 c_1 c_2 \epsilon_a \log m$ in Eq.(G.2), with . Define K_2 as upper bound on $\nabla_i^2 f_i^{(t)}(x_{1:i})$,

$$
K_2 = O\left(m^2 U_{\theta^*}^2 d^4 \epsilon \right) = O\left(\frac{m^2 U_\psi^2 d^6 \left(\max_{i\in[d]} p_i \right)}{\epsilon} \right).
$$

Taking Q as

$$
Q \geq \frac{2dM_{\nabla^2 F^*} + 2dK_2}{\epsilon} \tag{G.12}
$$

Using given value of Q, we get that

$$\left|\tilde{L}(\nabla F^*, x) - L(F^*, x)\right| \leq \epsilon, \tag{G.13}$$

$$\left|\tilde{L}(\nabla f^{(t)}, x) - L(f^{(t)}, x)\right| \leq \epsilon. \tag{G.14}$$

Using these relations, we get

$$\mathbb{E}_{\text{sgd}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{x\sim\mathcal{D}}\left[L(f^{(t)}, x)\right]\right] - \mathbb{E}_{x\sim\mathcal{D}}\left[L(F^*, x)\right] \leq O(\epsilon).$$

By the definition of KL divergence, we get

$$\mathbb{E}_{\text{sgd}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\text{KL}\left(p_{F^*,Z}||p_{f^{(t)},Z}\right)\right] \leq O(\epsilon).$$

□

H Problem in Training of Constrained Normalizing Flow

In this section, we provide details of why different initializations cause problems (described in section 3) in the training of Constrained Normalizing Flows. Recall that the loss function of normalizing flow with Gaussian distribution as base distribution is given by

$$L_G(f, x) = \frac{f(x)^T f(x)}{2} - \log\left(\left|\det\left(\frac{\partial f(x)}{\partial x}\right)\right|\right),$$

where function $f(x) : \mathbb{R}^d \to \mathbb{R}^d$ is parameterized using d neural networks $N_1, N_2, \ldots, N_d$. The i^{th} dimension of the function $f_i(x_{1:i}) = N(x_{1:i}; \theta_i)$. The neural network in CNF is defined as

$$N(x_{1:i}; \theta_i) = \tau\sum_{r=1}^{m}\bar{a}_{i,r}\tanh\left(\langle\bar{w}_{i,r} + w_{i,r}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}\right)\right),$$

with constraints $\bar{w}_{i,r,i} + w_{i,r,i} \geq \epsilon$, for all $r \in [m]$ and $i \in [d]$.

Here, $\epsilon > 0$ is a small constant and τ is a normalization constant which only depends on m. We use θ_i to denote parameters of $N(x_{1:i}; \theta_i)$ and θ to denote parameters of all neural networks. Initial weights $\bar{a}_{i,r}$ and $\bar{w}_{i,r,i}$ are sampled from *half-normal* distribution with parameters $\left(0, \epsilon_a^2\right)$ and $\left(0, \sigma_{wb}^2\right)$, resp. The half-normal random variable Y with parameters $\left(\mu, \sigma^2\right)$ is given by simply $|X|$ where $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$. Here $\mathcal{N}\left(\mu, \sigma^2\right)$ denote the Gaussian distribution with mean μ and variance σ^2. Other weights ($\bar{b}_{i,r}, \bar{w}_{i,r,j}$ for $j \neq i$) are sampled from $\mathcal{N}\left(0, \sigma_{wb}^2\right)$. We optimize the objective using *projected SGD*. Note that in this case, the constraints are very simple and projected SGD incurs very little overhead.

The pseudo network function is given by $g(x) = \left(g_1(x_{1:1}), g_2(x_{1:2}), \ldots, g_d(x_{1:d})\right)$, where $g_i(x_{1:i}) = P(x_{1:i}; \theta_i)$ is given by

$$P(x_{1:i}; \theta_i) = \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left(\tanh\left(\langle\bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right) + \tanh'\left(\langle\bar{w}_{i,r}, \tilde{x}_{1:i}\rangle + \bar{b}_{i,r}\right)\left(\langle w_{i,r}, x_{1:i}\rangle + b_{i,r}\right)\right)$$

with constraints $\bar{w}_{i,r,i} + w_{i,r,i} \geq \epsilon$ for all r. We decompose pseudo network in two parts:

$$P(x_{1:i}; \theta_i) = P_c(x_{1:i}) + P_\ell(x_{1:i}; \theta_i),$$

where $P_c(x_{1:i})$ and $P_\ell(x_{1:i}; \theta_i)$ is given by

$$P_c(x_{1:i}) = \tau\sum_{r=1}^{m}\bar{a}_{i,r}\tanh(\langle\bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}$$

$$P_\ell(x_{1:i}; \theta_i) = \tau\sum_{r=1}^{m}\bar{a}_{i,r}\tanh'(\langle\bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r})\left(\langle w_{i,r}, x_{1:i}\rangle + b_{i,r}\right).$$

The loss function for pseudo network is given b

$$L_G(g,x) = \frac{g(x)^T g(x)}{2} - \log\left(\left|\det\left(\frac{\partial g(x)}{\partial x}\right)\right|\right) = \sum_{i=1}^{d} g_i(x_{1:i}) - \sum_{i=1}^{d} \log\left(\frac{\partial g_i(x_{1:i})}{\partial x_i}\right)$$

where $g(x) = \left(g_i(x_{1:1}), g_i(x_{1:1}), \ldots, g_i(x_{1:n})\right)$. The pseudo network $P(x_{1:i};\theta_i)$, which approximates the neural network $N(x_{1:i};\theta_i)$, will be

$$P(x_{1:i};\theta_i) = \tau \sum_{r=1}^{m} \bar{a}_{i,r}\left(\tanh(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}) + \tanh'(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r})\left(\langle w_r, x_{1:i}\rangle + b_r\right)\right),$$

with constraints $\bar{w}_{i,r,i} + w_{i,r,i} \geq \epsilon$, for all $r \in [m]$. We decompose $P(x_{1:i};\theta_i)$ into two parts: $P(x_{1:i};\theta_i) = P_c(x_{1:i}) + P_\ell(x_{1:i};\theta_i)$, where

$$P_c(x_{1:i}) = \tau \sum_{r=1}^{m} \bar{a}_{i,r}\tanh(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}) \quad \text{and} \quad P_\ell(x_{1:i};\theta_i) = \tau \sum_{r=1}^{m} \bar{a}_{i,r}\tanh'(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r})\left(\langle w_r, x_{1:i}\rangle + b_r\right).$$

Note that $P_c(x_{1:i})$ only depends upon initialization and does not depend on parameters θ_i.

Let F^* denote the target function and $C(F^*)$ denote some complexity measure of F^*. We devide our analysis into two cases based on variance of $\bar{w}_{i,r}$ and $\bar{b}_{i,r}$. (1) In the first case, standard deviation σ_{wb} satisfies $\frac{\epsilon^2}{C(F^*)\sqrt{\log(md)}} \leq \sigma_{wb} \leq 1$. (2) In the second case, standard deviation σ_{wb} satisfies $\frac{1}{\sqrt{m}} \leq \sigma_{wb} \leq \frac{\epsilon^2}{C(F^*)\sqrt{\log(md)}}$. We call the first case *larger variance initalization* case and the second one *smaller variance intialization* case. Analysis for larger variance case is given in Section H.2 and analysis for smaller variance case is given in Section H.1.

H.1 Problem in optimization for smaller variance initialization case

In this section, we will provide details about the problem in smaller variance initialization case for Constrained Normalizing Flows (CNFs). We prove in Theorem H.5 that if we choose small learning rate η and number of time steps T according to the theorem statement, then function learned by sufficiently overparameterized CNFs is close to a linear function. To prove the theorem, we start by bounding maximum possible change in weights $\left\|w_{i,r}^{(t)}\right\|$ and biases $\left|b_r^{(t)}\right|$ during $t = T$ iterations in Lemma H.1. Using bound on change in weights, we establish closeness between function value given by neural networks and function value given by pseudo networks (Lemma H.3). We, then, prove that for any $t \in [T]$, pseudo network at time t is close to a linear function (Lemma H.4). Using closeness between neural network and pseudo network and linearity of pseudo network, we get that neural networks are close to a linear function for given small learning rate η and number of time steps T. Note that choosing similar values of η and T in supervised learning enables the provable *successful* training of neural network. The same issue in approximation arises for *all* activations with continuous derivative.

Recall that neural network $N(x_{1:i};\theta_i^{(t)})$ is given by

$$N(x_{1:i};\theta_i^{(t)}) = \sum_{r=1}^{m} \bar{a}_{i,r}\tanh\left(\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right)\right),$$

and derivative $\frac{\partial N(x_{1:i};\theta_i^{(t)})}{\partial x_i}$ is given by

$$\frac{\partial N(x_{1:i};\theta_i^{(t)})}{\partial x_i} = \sum_{r=1}^{m} \bar{a}_{i,r}\tanh'\left(\langle \bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right)\right)\left(\bar{w}_{i,r,i} + w_{i,r,i}^{(t)}\right).$$

We denote $\frac{\partial N(x_{1:i};\theta_i^{(t)})}{\partial x_i}$ as $N'(x_{1:i};\theta_i^{(t)})$.

Lemma H.1. *(Bound on change in weights and biases) For every x with $\|x\| \leq 1$, every $i \in [d]$ and time step $t \geq 1$, upper bound on weights $w_{i,r}^{(t)}$ and biases $b_{i,r}^{(t)}$ is given by following with at least $1 - \frac{d}{c_1} - \frac{d}{c_2}$ probability for*

any constant $c_1 > 10$, $c_2 > 10$.

$$\left\|w_{i,r}^{(t)}\right\|_2 \leq \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1 + 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t - 1\right)$$

$$\left|b_{i,r}^{(t)}\right| \leq 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t + \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1 + 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t - 1\right)$$

Proof. We first find upper bound on the derivative of loss function and $w_{i,r}$ and $b_{i,r}$. We denote $\boldsymbol{\alpha}_i = (0, 0, \ldots, 0, 1) \in \mathbb{R}^i$. By taking derivative of $L_G(f^{(t)}, x)$ with respect to $w_{i,r}$, we get

$$\begin{aligned}\frac{\partial L_G(f^{(t)}, x)}{\partial w_{i,r}} =& \tau N(x_{1:i}; \theta_i^{(t)})(\bar{a}_{i,r}x_{1:i}\tanh'(\langle\bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right))) \\ &- \frac{\tau}{N'(x_{1:i}; \theta_i^{(t)})}\Bigg(\boldsymbol{\alpha}_i\bar{a}_{i,r}\Big(\tanh'(\langle\bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right)) \\ &+ \left(\bar{w}_{i,r,i} + w_{i,r,i}^{(t)}\right)x_{1:i}\tanh''(\langle\bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right))\Big)\Bigg).\end{aligned}$$

We assume that $L_G(f^{(t)}, x)$ is $\tilde{L}_1$-lipschitz continuous wrt N and $\tilde{L}_2$-lipschitz continuous wrt N'. Assuming $|\tanh'(.)| \leq 1$ and $\|x\|_2 \leq 1$, we have

$$\left\|\frac{\partial L_G(f^{(t)}, x)}{\partial w_{i,r}}\right\|_2 \leq \tau\tilde{L}_1\bar{a}_{i,r} + \tau\tilde{L}_2\bar{a}_{i,r}\left(1 + \left|\bar{w}_{i,r,i} + w_{i,r,i}^{(t)}\right||\tanh''(\langle\bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right))|\right).$$

Assuming $|\tanh''(.)| \leq 1$, we get

$$\left\|\frac{\partial L_G(f^{(t)}, x)}{\partial w_{i,r}}\right\|_2 \leq \tau\tilde{L}_1\bar{a}_{i,r} + \tau\tilde{L}_2\bar{a}_{i,r}\left(1 + \left|w_{i,r,i}^{(t)}\right| + \left|\bar{w}_{i,r,i}\right|\right).$$

Using Lemma K.4 for $\bar{a}_{i,r}$ and $\bar{w}_{i,r,i}$, with probability at least $1 - \frac{1}{c_1} - \frac{1}{c_2}$, we have

$$\left\|\frac{\partial L_G(f^{(t)}, x)}{\partial w_{i,r}}\right\|_2 \leq \left(2c_1\epsilon_a\tau\sqrt{2\log m}\right)\left(\tilde{L}_1 + \tilde{L}_2\left(1 + \left|w_{i,r,i}^{(t)}\right| + 2c_2\sigma_{wb}\sqrt{2\log(md)}\right)\right). \tag{H.1}$$

For projected gradient descent, we get

$$\begin{aligned}\left\|w_{i,r}^{(t)}\right\|_2 &\leq \eta\sum_{j=0}^{t-1}\left\|\frac{\partial L_G(f^{(j)}, x^{(j)})}{\partial w_{i,r}}\right\|_2 \\ &\leq \eta\sum_{j=0}^{t-1}\left(\left(2c_1\epsilon_a\tau\sqrt{2\log m}\right)\left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\sigma_{wb}\tilde{L}_2\sqrt{2\log(md)}\right) + \left(2c_1\epsilon_a\tau\sqrt{2\log m}\right)\tilde{L}_2|w_{i,r,i}^{(j)}|\right) \\ &\leq \left(2\eta c_1\epsilon_a\tau\sqrt{2\log m}\right)\left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t + \left(2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)\left(\sum_{j=0}^{t-1}\left\|w_{i,r}^{(j)}\right\|_2\right).\end{aligned}$$

By defining α and β as

$$\begin{aligned}\alpha &= \left(2\eta c_1\tau\epsilon_a\sqrt{2\log m}\right)\left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right) \\ \beta &= \left(2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right),\end{aligned}$$

we get

$$\left\|w_{i,r}^{(t)}\right\|_2 \leq \alpha t + \beta\left(\sum_{j=0}^{t-1}\left\|w_{i,r}^{(j)}\right\|_2\right), \tag{H.2}$$

where
$$\begin{aligned}\sum_{j=0}^{t-1}\left\|w_{i,r}^{(j)}\right\|_2 &\leq \alpha(t-1) + (1+\beta)\left(\sum_{j=0}^{t-2}\left\|w_{i,r}^{(j)}\right\|_2\right)\\ &\leq \alpha\left((t-1)+(1+\beta)(t-2)\right) + (1+\beta)^2\left(\sum_{j=0}^{t-3}\left\|w_{i,r}^{(j)}\right\|_2\right)\\ &\leq \alpha\left((t-1)+(1+\beta)(t-2)+(1+\beta)^2(t-3)\right) + (1+\beta)^3\left(\sum_{j=0}^{t-4}\left\|w_{i,r}^{(j)}\right\|_2\right).\end{aligned} \tag{H.3}$$

In general, for any $t' \in \{0, 1, \ldots, t-1\}$, we can write

$$\sum_{j=0}^{t-1}\left\|w_{i,r}^{(j)}\right\|_2 \leq \alpha\left(\sum_{j=1}^{t-t'-1}(1+\beta)^{j-1}(t-j)\right) + (1+\beta)^{(t-t'-1)}\left(\sum_{j=0}^{t'}\left\|w_{i,r}^{(j)}\right\|_2\right).$$

By taking $t' = 0$, we get

$$\sum_{j=0}^{t-1}\left\|w_{i,r}^{(j)}\right\|_2 \leq \alpha\left(\sum_{j=1}^{t-1}(1+\beta)^{j-1}(t-j)\right).$$

Note that $\sum_{j=1}^{t-1}(1+\beta)^{(j-1)}(t-j)$ is sum of an arithmetic-geometric progression (AGP). Using Fact K.14, we can simplify the above sum as

$$\begin{aligned}\sum_{j=0}^{t-1}\left\|w_{i,r}^{(j)}\right\|_2 &\leq \alpha\left(\sum_{j=1}^{t-1}(1+\beta)^{j-1}(t-j)\right)\\ &= \alpha\left(\frac{(t-1)-(1+\beta)^{t-1}}{-\beta} - \frac{(1+\beta)\left(1-(1+\beta)^{t-2}\right)}{\beta^2}\right)\\ &= \alpha\left(\frac{\beta(1+\beta)^{t-1} - \beta(t-1) - (1+\beta) + (1+\beta)^{t-1}}{\beta^2}\right)\\ &= \alpha\left(\frac{(1+\beta)^t - (1+\beta t)}{\beta^2}\right)\end{aligned} \tag{H.4}$$

Using Eq.(H.4) to bound $\left\|w_{i,r}^{(t)}\right\|_2$ in Eq. (H.2), we get

$$\begin{aligned}\left\|w_{i,r}^{(t)}\right\|_2 &\leq \alpha\left(t + \beta\left(\frac{(1+\beta)^t - (1+\beta t)}{\beta^2}\right)\right)\\ &= \alpha\left(\frac{(1+\beta)^t - 1}{\beta}\right)\\ &= \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1 + 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t - 1\right).\end{aligned}$$

This completes the proof of upper bounding $\left\|w_{i,r}^{(t)}\right\|_2$. We use a similar procedure for $\left|b_{i,r}^{(t)}\right|$. By taking derivative

$\frac{\partial L_G(f^{(t)},x)}{\partial b_{i,r}}$, we get

$$\begin{aligned}\frac{\partial L_G(f^{(t)},x)}{\partial b_{i,r}} =& N(x_{1:i};\theta_i^{(t)})\tau\left(\bar{a}_{i,r}\tanh'(\langle\bar{w}_{i,r}+w_{i,r}^{(t)},x_{1:i}\rangle+\left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right))\right)\\ &-\frac{\tau}{N'(x_{1:i};\theta_i^{(t)})}\left(\bar{a}_{i,r}(\bar{w}_{i,r,i}+w_{i,r,i}^{(t)})\tanh''(\langle\bar{w}_{i,r}+w_{i,r}^{(t)},x_{1:i}\rangle+\left(\bar{b}_{i,r}+b_{i,r}^{(t)}\right))\right).\end{aligned}$$

We assume that $L_G(f^{(t)},x)$ is $\tilde{L}_1$-lipschitz wrt N and $\tilde{L}_2$-lipschitz wrt N'. Additionaly, using $|\tanh'(\cdot)|\le 1$ and $|\tanh''(\cdot)|\le 1$, we get

$$\left|\frac{\partial L_G(f^{(t)},x)}{\partial b_{i,r}}\right|\le\tilde{L}_1\bar{a}_{i,r}\tau+\tilde{L}_2\bar{a}_{i,r}\tau\left(\bar{w}_{i,r,i}+|w_{i,r,i}^{(t)}|\right).$$

Using Lemma K.4 for $\bar{a}_{i,r}$ and $\bar{w}_{i,r}$, with probability at least $1-\frac{1}{c_1}-\frac{1}{c_2}$, we get

$$\left|\frac{\partial L_G(f^{(t)},x)}{\partial b_{i,r}}\right|\le\left(2c_1\epsilon_a\sqrt{2\log m}\right)\tau\left(\tilde{L}_1+\tilde{L}_2|w_{i,r,i}^{(t)}|+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right). \tag{H.5}$$

For projected gradient descent, summing from time step $j=0$ to $j=t-1$, we get

$$\begin{aligned}|b_{i,r}^{(t)}|&\le\eta\sum_{j=0}^{t-1}\left|\frac{\partial L_G(f^{(j)},x^{(j)})}{\partial b_r}\right|\\ &=2\eta c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t+2\eta c_1\epsilon_a\tau\tilde{L}_2\sqrt{2\log m}\left(\sum_{j=0}^{t-1}|w_{i,r,i}^{(j)}|\right)\\ &\le 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t+2\eta c_1\epsilon_a\tau\tilde{L}_2\sqrt{2\log m}\left(\sum_{j=0}^{t-1}\left\|w_{i,r}^{(t)}\right\|_2\right).\end{aligned}$$

Using Eq.(H.4), we get

$$\begin{aligned}|b_{i,r}^{(t)}|&\le 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t\\ &+2\eta c_1\epsilon_a\tilde{L}_2\sqrt{2\log m}\left(\frac{\tilde{L}_1+\tilde{L}_2+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{2\eta c_1\epsilon_a\sqrt{2\log m}\tilde{L}_2^2}\right)\left(\left(1+2\eta c_1\tau\epsilon_a\sqrt{2\log m}\tilde{L}_2\right)^t-1\right)\\ &=2\eta c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t\\ &+\left(\frac{\tilde{L}_1+\tilde{L}_2+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1+2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t-1\right).\end{aligned}$$

This completes the proof. □

Define $\Lambda_w^{(t)}$ and $\Lambda_b^{(t)}$ as upper bound on $\left\|w_{i,r}^{(t)}\right\|_2$ and $\left|b_{i,r}^{(t)}\right|$:

$$\begin{aligned}\Lambda_w^{(t)}&=\left(\frac{\tilde{L}_1+\tilde{L}_2+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1+2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t-1\right),\\ \Lambda_b^{(t)}&=2\eta c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t\\ &\quad+\left(\frac{\tilde{L}_1+\tilde{L}_2+2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1+2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t-1\right).\end{aligned}$$

Lemma H.2. *For any $\epsilon > 0$, target function F^* with some complexity measure $C(F^*)$, any σ_{wb} which satisfy $\frac{1}{\sqrt{m}} \leq \sigma_{wb} \leq \frac{\epsilon}{C(F^*)\sqrt{\log(md)}}$, any hidden layer size $m \geq \Omega\left(poly\left(C(F^*), d, \frac{1}{\epsilon}\right)\right)$, any learning rate $\eta \leq \frac{c_9\epsilon}{m\tau\epsilon_a^2 \log m}$ and $T \leq \frac{c_{10}C(F^*)}{\epsilon^2}$, with at least $1 - \frac{d}{c_1} - \frac{d}{c_2}$ probability, we get*

$$\Lambda_w^{(t)} \leq \left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\left(\frac{4\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon\epsilon_a\sqrt{\log m}}\right)$$
$$\Lambda_b^{(t)} \leq \left(3\tilde{L}_1 + 2\tilde{L}_2 + 6c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\left(\frac{2\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon_a\epsilon\sqrt{\log m}}\right)$$

Proof. To simplify expression of $\Lambda_w^{(t)}$, we will use Fact K.12. First, we will check the condition for Fact K.12:

$$\begin{aligned}
2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2(t-1) &\leq 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2(T-1) \\
&\leq 2\left(\frac{c_9\epsilon}{m\tau\epsilon_a^2\log m}\right)c_1\epsilon_a\tau\sqrt{2\log m}\left(\frac{c_{10}C(F^*)}{\epsilon^2} - 1\right) \\
&= \frac{2\sqrt{2}c_1c_9c_{10}C(F^*)}{\epsilon_a\epsilon m\sqrt{\log m}}.
\end{aligned}$$

Choosing sufficiently high m such that $m \geq \Omega\left(\text{poly}\left(C(F^*), d, \frac{1}{\epsilon}\right)\right)$, we get

$$2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2(t-1) \leq 0.5.$$

By choosing sufficiently high m, the condition of Fact K.12 satisfies. Now, simplifying expression of $\Lambda_w^{(t)}$ using Fact K.12, we get

$$\begin{aligned}
\Lambda_w^{(t)} &= \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1 + 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t - 1\right) \\
&\leq \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(4\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2 t\right) \\
&\leq \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(4\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2 T\right) \\
&\leq \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(4c_1\epsilon_a\tau\tilde{L}_2\sqrt{2\log m}\left(\frac{c_9\epsilon}{m\tau\epsilon_a^2\log m}\right)\left(\frac{c_{10}C(F^*)}{\epsilon^2}\right)\right) \\
&= \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(4c_1\epsilon_a\tau\tilde{L}_2\sqrt{2\log m}\left(\frac{c_9\epsilon}{m\tau\epsilon_a^2\log m}\right)\left(\frac{c_{10}C(F^*)}{\epsilon^2}\right)\right) \\
&= \left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\left(\frac{4\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon\epsilon_a\sqrt{\log m}}\right).
\end{aligned}$$

Simplifying expression of $\Lambda_b^{(t)}$ in simillar manner as $\Lambda_w^{(t)}$, we get

$$\begin{aligned}
\Lambda_b^{(t)} &= 2\eta c_1 \epsilon_a \tau \sqrt{2\log m}\left(\tilde{L}_1 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)t \\
&+ \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(\left(1 + 2\eta c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\right)^t - 1\right) \\
&\leq 2c_1\epsilon_a\tau\sqrt{2\log m}\left(\tilde{L}_1 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\left(\frac{c_9\epsilon}{m\tau\epsilon_a^2\log m}\right)\left(\frac{c_{10}C(F^*)}{\epsilon^2}\right) \\
&\quad + \left(\frac{\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}}{\tilde{L}_2}\right)\left(4c_1\epsilon_a\tau\sqrt{2\log m}\tilde{L}_2\left(\frac{c_9\epsilon}{m\tau\epsilon_a^2\log m}\right)\left(\frac{c_{10}C(F^*)}{\epsilon^2}\right)\right) \\
&= \left(\tilde{L}_1 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\left(\frac{2\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon_a\epsilon\sqrt{\log m}}\right) \\
&\quad + \left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\left(\frac{4\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon_a\epsilon\sqrt{\log m}}\right) \\
&= \left(\frac{2\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon_a\epsilon\sqrt{\log m}}\right)\left(3\tilde{L}_1 + 2\tilde{L}_2 + 6c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right).
\end{aligned}$$

□

Lemma H.3. *(Coupling between neural network and pseudo network) For every x with $\|x\|_2 \leq 1$, every $i \in [d]$ and every time step $t \leq T$, with probability at least $1 - \frac{d}{c_1} - \frac{d}{c_2}$ over random initialization, we have*

$$\left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| \leq 2c_1\epsilon_a\tau m\sqrt{2\log m}\left(\left(\Lambda_w^{(t)}\right)^2 + {\Lambda_b^{(t)}}^2\right)$$

Proof. Bounding difference between $N(x_{1:i};\theta_i^{(t)})$ and $P(x_{1:i};\theta_i^{(t)})$, we get

$$\begin{aligned}
\left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| &= \Bigg|\tau\sum_{r=1}^m \bar{a}_{i,r}\tanh(\langle\bar{w}_{i,r} + w_{i,r}^{(t)}, x_{1:i}\rangle + \left(\bar{b}_{i,r} + b_{i,r}^{(t)}\right)) \\
&\quad - \tau\sum_{r=1}^m \bar{a}_{i,r}\left(\tanh(\langle\bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}) + \bar{a}_{i,r}\tanh'(\langle\bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r})(\langle w_{i,r}^{(t)}, x_{1:i}\rangle + b_{i,r}^{(t)})\right)\Bigg| \\
&= \left|\frac{\tau}{2}\sum_{r=1}^m \bar{a}_{i,r}\tanh''(\xi_r)\left(\langle w_{i,r}^{(t)}, x_{1:i}\rangle + b_{i,r}^{(t)}\right)^2\right|
\end{aligned}$$

for some $\xi_r \in \mathbb{R}$. Using $|\tanh''(\xi_r)| \leq 1$ and $(\langle w_{i,r}^{(t)}, x_{1:i}\rangle + b_{i,r}^{(t)})^2 \leq 2\left(\langle w_{i,r}^{(t)}, x_{1:i}\rangle^2 + \left(b_{i,r}^{(t)}\right)^2\right)$, we get

$$\left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| \leq \tau\sum_{r=1}^m \bar{a}_{i,r}\left(\langle w_{i,r}^{(t)}, x_{1:i}\rangle^2 + \left(b_{i,r}^{(t)}\right)^2\right).$$

Using Lemma K.4 and using $\|x\|_2 \leq 1$, with at least $1 - \frac{1}{c_1}$ probability, we have

$$\begin{aligned}
\left|N(x_{1:i};\theta_i^{(t)}) - P(x_{1:i};\theta_i^{(t)})\right| &\leq 2c_1\epsilon_a\tau\sqrt{2\log m}\sum_{r=1}^m\left(\left\|w_{i,r}^{(t)}\right\|_2^2 + \left(b_r^{(t)}\right)^2\right) \\
&\leq 2c_1\epsilon_a\tau\sqrt{2\log m}\left(\|W_i^{(t)}\|_{2,2}^2 + \|B_i^{(t)}\|_2^2\right) \\
&\leq 2c_1\epsilon_a\tau m\sqrt{2\log m}\left(\left(\Lambda_w^{(t)}\right)^2 + \left(\Lambda_b^{(t)}\right)^2\right).
\end{aligned}$$

Using union bound for all $i \in [d]$, we complete the proof. □

Lemma H.4. *For any $\epsilon \in (0, \frac{1}{d^3})$, every $i \in [d]$ and every time step $t \le T$, any target function F^* with some complexity measure of target function $C(F^*)$, any variance σ_{wb} with $\frac{1}{\sqrt{m}} \le \sigma_{wb} \le \frac{\epsilon^2}{C(F^*)\sqrt{\log(md)}}$, any hidden layer size $m \ge \Omega\left(poly\left(C(F^*), d, \frac{1}{\epsilon}\right)\right)$, choosing learning rate $\eta = O\left(\frac{\epsilon}{m\tau\epsilon_a^2 \log m}\right)$ and $T = O\left(\frac{C(F^*)}{\epsilon^2}\right)$, with probability at least $1 - \frac{d}{c_1} - \frac{d}{c_2} - \frac{d}{c_3}$ over random initialization, we get*

$$\left| P_\ell(x_{1:i}; \theta_i^{(t)}) - \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(\langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right) \right| \le O(\epsilon).$$

Proof. Recalling the definition of $P_\ell(x_{1:i}; \theta_i^{(t)})$:

$$P_\ell(x_{1:i}; \theta_i^{(t)}) = \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(\tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) \left(\langle w_r^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right) \right).$$

Subtracting the linear function from $P_\ell(x_{1:i}; \theta_i^{(t)})$ will give us the following:

$$\begin{aligned} \left| P_\ell(x_{1:i}; \theta_i^{(t)}) - \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(\langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right) \right| &\le \left| \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(\left(\tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right) \left(\langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right) \right) \right| \\ &\le \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right| \left| \langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right|. \quad \text{(H.6)} \end{aligned}$$

First, we will try to find upper bound on $\left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right|$:

$$\left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right| = \left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - \tanh'(0) \right| \overset{\text{(i)}}{\le} \left| \langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r} \right|,$$

where inequality (i) follows from 1-Lipschitz continuity of $\tanh'(\cdot)$ function. Using Lemma K.4 on $\bar{w}_{i,r}$ and $\bar{b}_{i,r}$, with probability at least $1 - \frac{1}{c_2} - \frac{1}{c_3}$, we get that

$$\left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right| \le 2c_2\sigma_{wb}\sqrt{2\log(md)} + 2c_3\sigma_{wb}\sqrt{2\log m}.$$

Using above inequality in Eq. (H.6), with probability at least $1 - \frac{1}{c_1} - \frac{1}{c_3}$ over random initialization, we get

$$\begin{aligned} \left| P_\ell(x_{1:i}; \theta_i^{(t)}) - \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(\langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right) \right| &\le \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right| \left| \langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right| \\ &\le \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(2c_2\sigma_{wb}\sqrt{2\log(md)} + 2c_3\sigma_{wb}\sqrt{2\log m} \right) \left| \langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right|. \end{aligned}$$

Using Lemma K.4 and Lemma H.1, with probability at least $1 - \frac{1}{c_1} - \frac{1}{c_2} - \frac{1}{c_3}$, we get

$$\begin{aligned} &\left| P_\ell(x_{1:i}; \theta_i^{(t)}) - \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left(\langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right) \right| \\ &\le \tau \sum_{r=1}^{m} \bar{a}_{i,r} \left| \tanh'(\langle \bar{w}_{i,r}, x_{1:i} \rangle + \bar{b}_{i,r}) - 1 \right| \left| \langle w_{i,r}^{(t)}, x_{1:i} \rangle + b_{i,r}^{(t)} \right| \\ &\le m\tau \left(2c_1\epsilon_a\sqrt{2\log m} \right) \left(2c_2\sigma_{wb}\sqrt{2\log(md)} + 2c_3\sigma_{wb}\sqrt{2\log m} \right) \left(\Lambda_w^{(t)} + \Lambda_b^{(t)} \right) \\ &\le 8c_1 \left(c_2\sqrt{\log(md)} + c_3\sqrt{\log m} \right) \epsilon_a\sigma_{wb} m\tau\sqrt{\log m} \left(\Lambda_w^{(t)} + \Lambda_b^{(t)} \right). \end{aligned}$$

Using bound on $\Lambda_w^{(t)}$ and $\Lambda_b^{(t)}$ from Lemma H.2, we get

$$\begin{aligned}
&\left|P_\ell(x_{1:i};\theta_i^{(t)}) - \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left(\langle w_{i,r}^{(t)}, x_{1:i}\rangle + b_{i,r}^{(t)}\right)\right| \\
&\le 8c_1\left(c_2\sqrt{\log(md)} + c_3\sqrt{\log m}\right)\epsilon_a\sigma_{wb}m\tau\sqrt{\log m}\Bigg(\left(\frac{2\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon_a\epsilon\sqrt{\log m}}\right)\left(5\tilde{L}_1 + 4\tilde{L}_2\right. \\
&\quad \left.+ 10c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)\Bigg) \\
&\le \frac{16\sqrt{2}c_1^2c_9c_{10}\left(c_2\sqrt{\log(md)} + c_3\sqrt{\log m}\right)\sigma_{wb}\tau C(F^*)}{\epsilon}\left(5\tilde{L}_1 + 4\tilde{L}_2 + 10c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right).
\end{aligned}$$

Using $\sigma_{wb} \le \frac{\epsilon^2}{C(F^*)\sqrt{\log(md)}}$ and re-scaling ϵ by $\frac{\epsilon}{d^3}$, we get

$$\left|P_\ell(x_{1:i};\theta_i^{(t)}) - \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left(\langle w_{i,r}^{(t)}, x_{1:i}\rangle + b_{i,r}^{(t)}\right)\right| \le O(\tau\epsilon).$$

Using $\tau \le 1$ for $\sigma_{wb} \ge \frac{1}{\sqrt{m}}$, we get

$$\left|P_\ell(x_{1:i};\theta_i^{(t)}) - \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left(\langle w_{i,r}^{(t)}, x_{1:i}\rangle + b_{i,r}^{(t)}\right)\right| \le O(\epsilon).$$

□

Theorem H.5. *For any $\epsilon \in (0, \frac{1}{d^3})$, any $i \in [d]$, any target function F^* with some complexity measure of target function $C(F^*)$, any σ_{wb} which satisfy $\frac{1}{\sqrt{m}} \le \sigma_{wb} \le \frac{\epsilon}{C(F^*)\sqrt{\log(md)}}$, any hidden layer size $m \ge \Omega\left(poly\left(C(F^*), d, \frac{1}{\epsilon}\right)\right)$, choosing normalization constant τ such that $\left|P_c(x)\right| \le O(\epsilon)$, learning rate $\eta = O\left(\frac{\epsilon}{m\tau\epsilon_a^2\log m}\right)$ and $T = O\left(\frac{C(F^*)}{\epsilon^2}\right)$, with probability at least 0.9 over random initialization, Projected SGD on neural network after T iterations*

$$\left|N(x_{1:i};\theta_i^{(T)}) - \left(\langle\alpha_i, x_{1:i}\rangle + \beta_i\right)\right| \le O(\epsilon), \tag{H.7}$$

where α_i and β_i are given by

$$\alpha_i = \tau\sum_{r=1}^{m}\bar{a}_{i,r}w_{i,r}^{(T)} \quad \textit{and} \quad \beta_i = \tau\sum_{r=1}^{m}\bar{a}_{i,r}b_{i,r}^{(T)}.$$

Proof. By decomposing the difference between $N(x_{1:i};\theta_i^{(T)})$ and $\left(\langle\alpha_i, x_{1:i}\rangle + \beta_i\right)$ into two parts, we get

$$\left|N(x_{1:i};\theta_i^{(T)}) - \left(\langle\alpha, x_{1:i}\rangle + \beta\right)\right| \le \underbrace{\left|N(x_{1:i};\theta_i^{(T)}) - P(x_{1:i};\theta_i^{(T)})\right|}_{\text{I}} + \underbrace{\left|P(x_{1:i};\theta_i^{(T)}) - \left(\langle\alpha_i, x_{1:i}\rangle + \beta_i\right)\right|}_{\text{II}}. \tag{H.8}$$

Using Lemma H.3, we can bound I:

$$\begin{aligned}
\text{I} &\le 2c_1\epsilon_a\tau m\sqrt{2\log m}\left(\left(\Lambda_w^{(t)}\right)^2 + \left(\Lambda_b^{(t)}\right)^2\right) \\
&\le 2c_1\epsilon_a\tau m\sqrt{2\log m}\left(\left(\Lambda_w^{(t)}\right)^2 + \left(\Lambda_b^{(t)}\right)^2\right) \\
&\le 2c_1\epsilon_a\tau m\sqrt{2\log m}\left(\frac{2\sqrt{2}c_1c_9c_{10}C(F^*)}{m\epsilon\epsilon_a\sqrt{\log m}}\right)^2\left(4\left(\tilde{L}_1 + \tilde{L}_2 + 2c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)^2\right. \\
&\quad \left.+ \left(3\tilde{L}_1 + 2\tilde{L}_2 + 6c_2\tilde{L}_2\sigma_{wb}\sqrt{2\log(md)}\right)^2\right).
\end{aligned}$$

Choosing sufficienty high m such that $m \geq \Omega\left(\text{poly}\left(C(F^*), \frac{1}{\epsilon}\right)\right)$, we get

$$\text{I} \leq O\left(\epsilon\right)$$

To bound II, we use Lemma H.4.

$$\text{II} \leq \left|P_c(x_{1:i})\right| + \left|P_\ell(x_{1:i}; \theta_i^{(t)}) - \left(\langle \alpha_i, x_{1:i}\rangle + \beta_i\right)\right| \leq O(\epsilon)$$

Using Eq. (H.8), we get

$$\left|N(x_{1:i}; \theta_i^{(t)}) - \left(\langle \alpha_i, x_{1:i}\rangle + \beta_i\right)\right| \leq O(\epsilon)$$

□

H.2 Problem in optimization for larger variance initialization case

In this section, we will provide details about the problem for larger variance initialization case. Recall that $P_c\left(x_{1:i}\right)$ only depends upon initialization and does not depend on θ_i. Hence, it can not approximate the target function after the training, therefore $P_\ell(x_{1:i}; \theta_i)$ needs to approximate target function with $P_c\left(x_{1:i}\right)$ subtracted but in this case, we prove in Theorem H.6 that if norm of change in weights $\|\theta^{(T)}\|_{2,1}$ is small then $\left|P_\ell(x_{1:i}; \theta_i)\right|$ is very small for sufficiently large m; therefore, it can not approximate every target function. We also provide reasons and details in Lemma H.7 about the requirement of small norm of change in weights $\|\theta^{(T)}\|_{2,1}$. In short, small norm of change in weights is required to maintain coupling between neural networks and pseudo networks. For large variance initialization case, we have

Theorem H.6. *(small value of $P_\ell(x_{1:i}; \theta_i^{(T)})$) For any standard deviation $\frac{\epsilon^2}{C(F^*)\sqrt{\log m}} \leq \sigma_{wb} \leq 1$, for any $i \in [d]$, any constant $c_8 > 0$ and any $\eta > 0$, $T > 1$, if upper bound on norm of change of parameters is given by*

$$\|\theta_i^{(T)}\|_{2,1} \leq O\left(\frac{1}{d^2 \epsilon_a \sigma_{wb} \tau m^{c_8} \log m}\right),$$

then for all $x \in \mathbb{R}^d$ with $\|x\|_2 \leq 1$, with probabillity at least 0.99, we have

$$\left|P_\ell(x_{1:i}; \theta_i^{(T)})\right| \leq \frac{1}{3\sqrt{2} c_2 \sigma_{wb} m^{c_8} \sqrt{\log m}} = O\left(\frac{1}{d \sigma_{wb} m^{c_8} \sqrt{\log\left(md\right)}}\right).$$

Given upper bound on $\|\theta^{(T)}\|_{2,1}$ is necessary to ensure closeness between neural network and pseudo network (More details given in Lemma H.7).

Proof. Using the definition of $P_\ell(x_{1:i}; \theta_i^{(t)})$, we get

$$\begin{aligned}
\left|P_\ell(x_{1:i}; \theta_i^{(t)})\right| &= \left|\tau \sum_{r=1}^{m} \bar{a}_{i,r} \tanh'\left(\langle \bar{w}_{i,r}, x_{1:i}\rangle + \bar{b}_{i,r}\right)\left(\langle w_{i,r}^{(t)}, x\rangle + b_{i,r}^{(t)}\right)\right| \\
&\overset{\text{(i)}}{\leq} \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right) \sum_{r=1}^{m}\left(\left\|w_{i,r}^{(t)}\right\|_2 + \left|b_{i,r}^{(t)}\right|\right) \\
&\leq \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1} + \left\|B_i^{(t)}\right\|_1\right),
\end{aligned}$$

where inequality (i) follows from Lemma K.4 with at least $1 - \frac{1}{c_1}$ probability. Using upper bound on $\|\theta_i^{(T)}\|_{2,1}$ from the theorem statement, with at least $1 - \frac{1}{c_1}$ probability, we get

$$\begin{aligned}
\left|P_\ell(x_{1:i}; \theta_i^{(t)})\right| &\leq \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\|W_i^{(T)}\|_{2,1} + \|B_i^{(T)}\|_1\right) \\
&\leq \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(2\|\theta_i^{(T)}\|_{2,1}\right) \\
&\leq \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\frac{1}{12 c_1 c_2 \epsilon_a \sigma_{wb} \tau m^{c_8}\sqrt{\log m \log\left(md\right)}}\right) \\
&\leq \frac{1}{3\sqrt{2} c_2 \sigma_{wb} m^{c_8}\sqrt{\log\left(md\right)}}.
\end{aligned}$$

This completes the proof. □

Recall that we denote derivative $\frac{\partial f_i(x_{1:i})}{\partial x_i}$ as $\nabla_i f_i(x_{1:i})$. Similarly, we use $\nabla_i g_i(x_{1:i})$ to derivative of $g_i(x_{1:i})$.

Lemma H.7. *(Requirement of having small $L_{2,1}$-norm of change in weights $\|\theta_i^{(T)}\|_{2,1}$) For any constant $c_8 > 0$, for all $i \in [d]$, if following bound either on $\|\theta_i^{(T)}\|_{2,1}$ holds,*

$$\|\theta_i^{(T)}\|_{2,1} = \omega\left(\frac{1}{d^2\epsilon_a\sigma_{wb}\tau m^{c_8}\sqrt{\log(m)\log(md)}}\right)$$

then, with at least 0.98 probability, coupling between $\nabla_i f_i^{(t)}(x_{1:i})$ and $\nabla_i g_i^{(t)}(x_{1:i})$ can be lost. More precisely,

$$\left|\nabla_i f_i^{(t)}(x_{1:i}) - \nabla_i g_i^{(t)}(x_{1:i})\right| \le \omega\left(\frac{1}{m^{c_8}}\right)$$

Proof. First, we will find upper bound on difference between $\nabla_i f_i^{(t)}(x_{1:i})$ and $\nabla_i g_i^{(t)}(x_{1:i})$:

$$\begin{aligned}
\left|\nabla_i f_i^{(t)}(x_{1:i}) - \nabla_i g_i^{(t)}(x_{1:i})\right| &= \Bigg|\tau\sum_{r=1}^{m}\bar{a}_{i,r}\big((w_{i,r,i}^{(t)}+\bar{w}_{i,r,i})\Big(\tanh'(\langle\bar{w}_{i,r}+w_{i,r}^{(t)},x_{1:i}\rangle+\Big(\bar{b}_{i,r}+b_{i,r}^{(t)}\Big)\big)\\
&\quad-\tanh'(\langle\bar{w}_{i,r},x_{1:i}\rangle+\bar{b}_{i,r})\Big)-\tanh''(\langle\bar{w}_{i,r},x_{1:i}\rangle+\bar{b}_{i,r})\left(\bar{w}_{i,r,i}(\langle w_{i,r}^{(t)},x_{1:i}\rangle+b_{i,r}^{(t)})\right)\Bigg)\Bigg|\\
&\le \tau\sum_{r=1}^{m}2\bar{a}_{i,r}\left|w_{i,r,i}^{(t)}+\bar{w}_{i,r,i}\right|\left|\langle w_{i,r}^{(t)},x_{1:i}\rangle+b_{i,r}^{(t)}\right|+\tau\sum_{r=1}^{m}\bar{a}_{i,r}\bar{w}_{i,r,i}\left|\langle w_{i,r}^{(t)},x_{1:i}\rangle+b_{i,r}^{(t)}\right|\\
&= \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left|\langle w_{i,r}^{(t)},x_{1:i}\rangle+b_{i,r}^{(t)}\right|\left(2\left|w_{i,r,i}^{(t)}+\bar{w}_{i,r,i}\right|+\bar{w}_{i,r,i}\right)\\
&= \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left|\langle w_{i,r}^{(t)},x_{1:i}\rangle+b_{i,r}^{(t)}\right|\left(2|w_{i,r,i}^{(t)}|+3\bar{w}_{i,r,i}\right)\\
&\le \tau\sum_{r=1}^{m}\bar{a}_{i,r}\left(\left\|w_{i,r}^{(t)}\right\|_2+\left|b_{i,r}^{(t)}\right|\right)\left(2\left\|W_i^{(t)}\right\|_{\infty,\infty}+6c_2\sigma_{wb}\sqrt{2\log(md)}\right)\\
&\overset{\text{(i)}}{\le} \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1}+\left\|B_i^{(t)}\right\|_1\right)\left(2\left\|W_i^{(t)}\right\|_{\infty,\infty}+6c_2\sigma_{wb}\sqrt{2\log(md)}\right),
\end{aligned}$$

where inequality (i) follows from Lemma K.4 with probability at least $1-\frac{1}{c_1}-\frac{1}{c_2}$. Using bounds on norm $\|\theta_i^{(T)}\|_{2,1}$, we get

$$\begin{aligned}
\left|\nabla_i f_i^{(t)}(x_{1:i}) - \nabla_i g_i^{(t)}(x_{1:i})\right| &\le \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1}+\left\|B_i^{(t)}\right\|_1\right)\left(2\left\|W_i^{(t)}\right\|_{\infty,\infty}+6c_2\sigma_{wb}\sqrt{2\log(md)}\right)\\
&\le \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1}+\left\|B_i^{(t)}\right\|_1\right)2\left\|W_i^{(t)}\right\|_{\infty,\infty}\\
&\quad+\tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1}+\left\|B_i^{(t)}\right\|_1\right)6c_2\sigma_{wb}\sqrt{2\log(md)}\\
&\le \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1}+\left\|B_i^{(t)}\right\|_1\right)2\left\|W_i^{(t)}\right\|_{\infty,\infty}\\
&\quad+\tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(2\left\|\theta_i^{(T)}\right\|_{2,1}\right)6c_2\sigma_{wb}\sqrt{2\log(md)}\\
&\le \tau\left(2c_1\epsilon_a\sqrt{2\log m}\right)\left(\left\|W_i^{(t)}\right\|_{2,1}+\left\|B_i^{(t)}\right\|_1\right)2\left\|W_i^{(t)}\right\|_{\infty,\infty}+\omega\left(\frac{1}{m^{c_8}}\right)\\
&\le \omega\left(\frac{1}{m^{c_8}}\right).
\end{aligned}$$

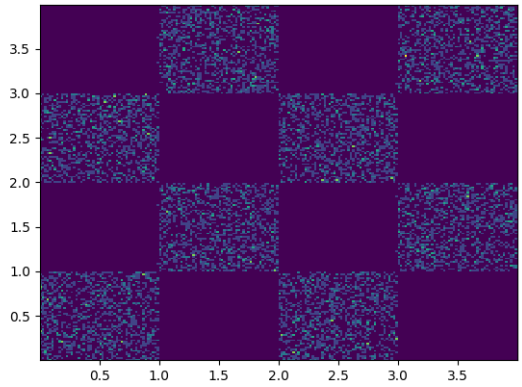

Figure 2: Grid dataset

Using union bound on all $i \in [d]$, with probability atleast $1 - \frac{d}{c_1} - \frac{d}{c_2}$, for all $i \in [d]$, we get

$$\left|\nabla_i f_i^{(t)}\left(x_{1:i}\right) - \nabla_i g_i^{(t)}\left(x_{1:i}\right)\right| \leq \omega\left(\frac{1}{m^{c_8}}\right).$$

Taking $c_1 = 100d$ and $c_2 = 100d$, with atleast 0.98 probability, for all $i \in [d]$, we get

$$\left|\nabla_i f_i^{(t)}\left(x_{1:i}\right) - \nabla_i g_i^{(t)}\left(x_{1:i}\right)\right| \leq \omega\left(\frac{1}{m^{c_8}}\right).$$

□

I Additional experiments

In this section, we show experimental results for both CNF and UNF on different datasets. First, we describe experimental setup in Subsection I.1. Then, we discuss our main observations for constrained normalizing flow and unconstrained normalizing flow in Subsection I.2 and I.3. In Subsection I.5, we plot training curves for both CNF and UNF for different learning rates and datasets. Codes for the experiments are available at `https://github.com/kulinshah98/overparam-NFs`.

I.1 Experimental Setup

Datasets. We use five synthetic datasets for our experiments. All datasets contain 10,000 data points. The details about the datasets are given below:

- *Mixture of Gaussian Dataset:* Data in this dataset lies in 1D and is generated from mixture of 2 Gaussians with means at 2.5 and -2.5. The standard deviation of both Gaussians is 1.
- *Mixture of Beta Dataset:* Data in this dataset lies in 1D and is generated from mixture of 3 Beta distribution. The parameters of Beta distributions are given by (5, 30), (30, 5) and (30, 30).
- *Grid Dataset:* Data in this dataset lies in 2D. Figure of the data is given in 2. Brightness at any point in this 2D plot represents the unnormalized probability density of that point.
- *5D Mixture of Gaussian dataset:* Data in this dataset lies in 5D and is generated from mixture of 10 Gaussians.

Architecture. We use similar architecture as described in Section 3 and Section 4 for both constrained and unconstrained normalizing flows. In all our experiments, we fix the weights of the output layer and train the weights and biases of the hidden layer. In UNFs, we use one-hidden layer network for all datasets while in CNFs, we use one-hidden layer network for 1D datasets and use two-hidden layer network for Grid dataset and three-hidden layer network for 5D Mixture of Gaussian dataset. We initialize weights of neural network as described in Section 3 and Section 4. We choose ϵ_a (standard deviation of top layer of neural networks in both UNF and CNF) from $\{0.15, 0.2, 0.25\}$ using the training error after a fixed number of iteration as a metric to evaluate.

Training Procedure. We use same training procedure for both constrained and unconstrained normalizing flows as described in Section 3 and Section 4. We use same base distribution as used in theoretical results for both CNF and UNF (i.e., standard Gaussian for CNFs and standard exponential for UNFs). Although, we believe that our experimental result can hold for all common distributions as a base distribution. In all our experiments, we use mini-batch SGD with batch size 32 for the training.

All our results are averaged over 5 different iterations. We used NVIDIA Tesla P100 GPU for approx 1000 hours to generate our final experimental results. Our experimental results validate the dichotomy between constrained and unconstrained normalizing flows which was established in Section 3 and Section 4.

I.2 Results for constrained normalizing flow

In Section 3, we suggested that high overparameterization may adversely affect training for constrained normalizing flows . In this section, we give empirical evidence for our claims. We use Gaussian distribution as a base distribution in all our experiments of constrained normalizing flow. We experiment with two different initialization for weights and biases of the hidden layer. 1) Gaussian distribution with zero-mean and $1/m$ variance ($\sigma_{wb}^2 = 1/m$) where m is number of neurons in hidden layer. We call CNF with this initialization as CNF-NNWB (CNF with Normalized Normal initialization for Weights and Biases) and 2) Standard Gaussian distribution ($\sigma_{wb}^2 = 1$). We denote CNF with this initialization as CNF-SNWB (CNF with Standard Normal initialization for Weights and Biases). We observe training error and L_2 distance of parameters from initialization after a fixed number of iterations for both CNF-NNWB and CNF-SNWB. We made following two observations:

Effect of overparameterization on training speed of CNF. In Figure 3 and Figure 4, we plot width of neural networks versus training error after a fixed number of iterations and for a fixed learning rate. We see that training error for CNF models increases as we increase overparameterization of neural networks, which means that to reach a fixed training error, larger models take *more* number of training updates. This shows that for any fixed learning rate, as we increase overparameterization in CNF, training speed *decreases*. This phenomenon is consistent across different datasets, different learning rates and different initializations. This result is *novel* and *surprising* because in supervised learning, overparameterization helps in *faster* convergence for a fixed learning rate Neyshabur et al. [2015] and we are not aware of *any other* settings where overparametrization has such strong negative effect.

Effect of overparameterization on L_2 distance of parameters from initialization. Figure 5 has plots of width of neural networks versus L_2 distance of parameters from the initialization after a fixed number of training iterations. From the figure, we see that as we increase overparameterization in CNF models, L_2 distance from the initialization also increases. From our previous observation, we know that after a fixed number of training iterations, training error increases as overparameterization increases. Combining experiment on L_2 distance with our previous observation, we get that to achieve same training error, more overparameterized model have larger L_2 distance compared to their smaller counterparts. This result is *surprising* because in supervised learning, it is known that more overparameterized model have smaller distance of parameters from the initialization Nagarajan and Kolter [2019].

I.3 Results for unconstrained normalizing flow

In Section 4, we prove that overparameterized neural network can efficiently learn the data distribution. In this section, we will provide empirical evidence that overparameterization helps in training of UNF. Similar to CNF, we study training error and L_2 distance of parameters from initialization after a fixed number of training

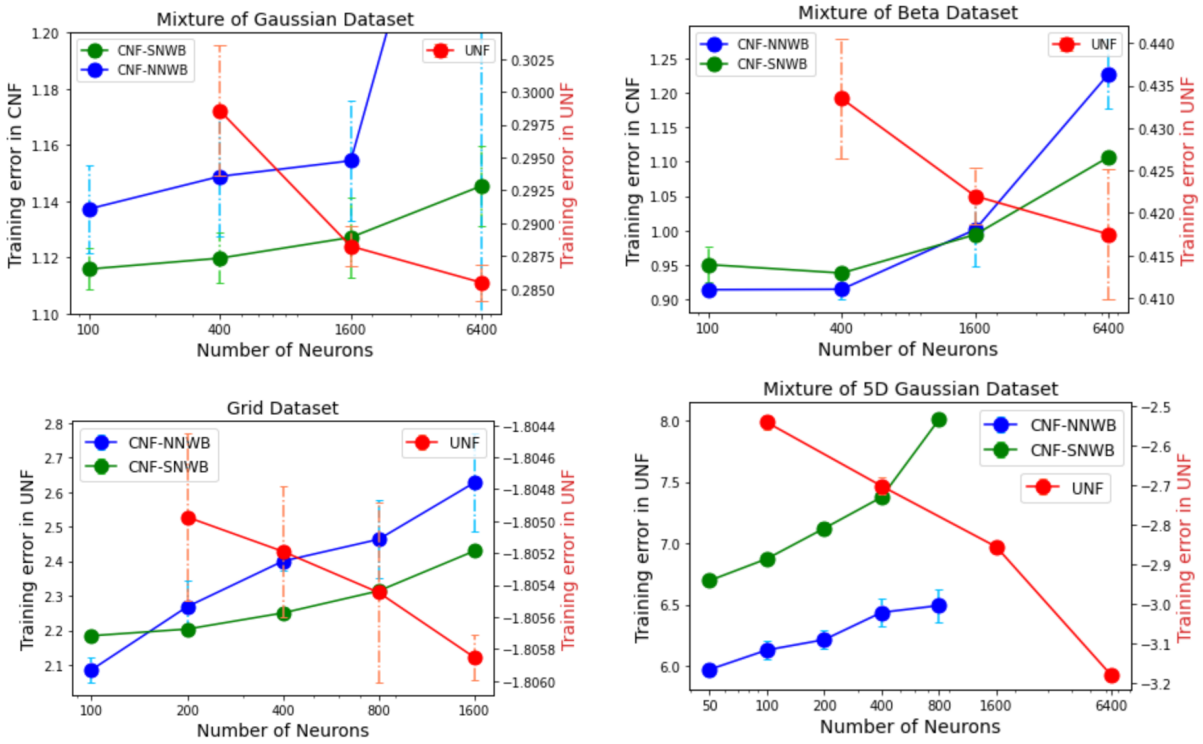

Figure 3: Comparison between CNF and UNF of training error after a fixed number of training iterations

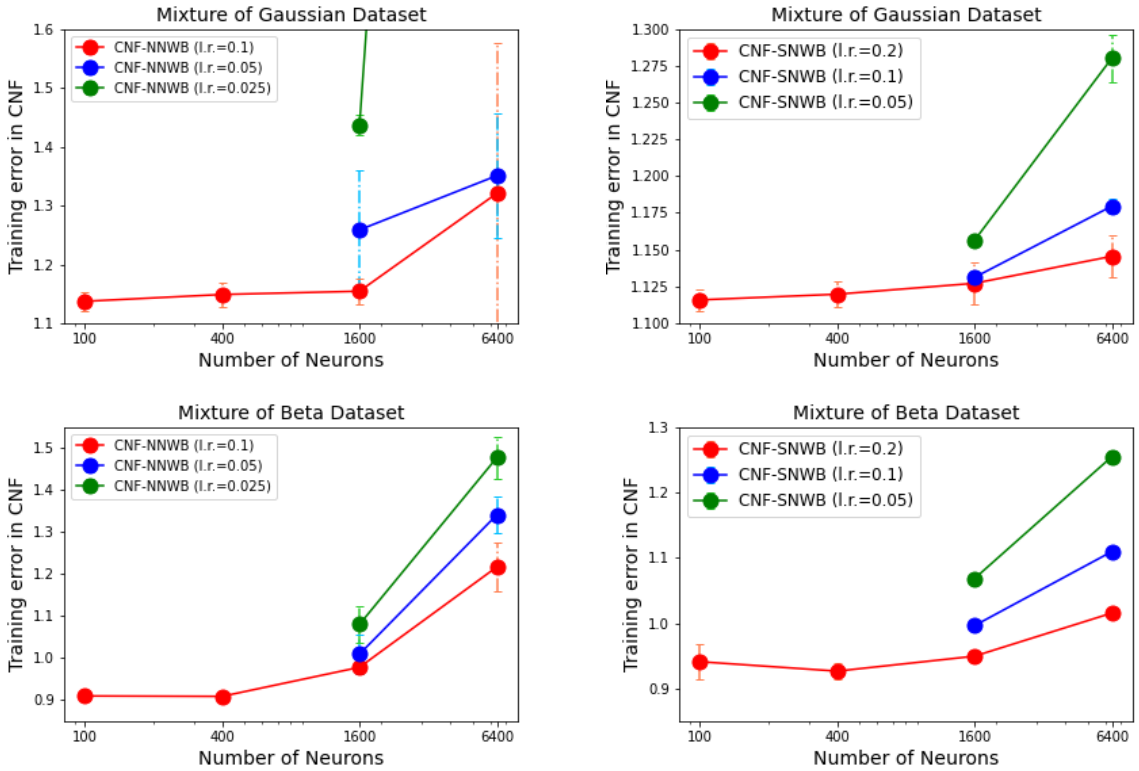

Figure 4: Training error of CNF-NNWB and CNF-SNWB after a fixed number of training iterations for different learning rates

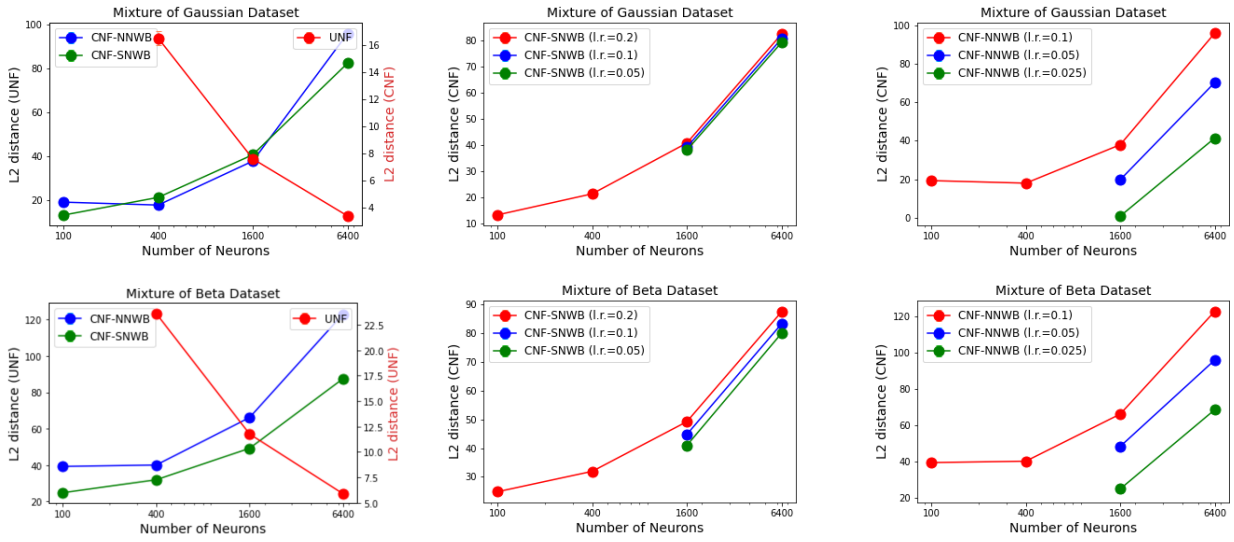

Figure 5: Comparison of L_2 distance from initialization between UNF and CNF models

iterations. We made following two observations:

Effect of overparameterization on training speed of UNF. In Figure 3, we see that training error after a fixed number of iterations decreases with increasing width of neural networks in UNF, which means that to reach a fixed training error, larger models need *smaller* number of training updates. This implies that for any fixed learning rate, increasing overparameterization in UNF *increases* training speed. This trend is consistent with supervised learning, where it is known that overparameterization helps in *faster* convergence for a fixed learning rate Neyshabur et al. [2015].

Effect of overparameterization on L_2 distance of parameters from initialization. Figure 5 shows that as we increase overparameterization in UNF models, L_2 distance of parameters from the initialization decreases. Our previous observation was that after a fixed number of training iterations, training error decreases or remains almost same as overparameterization increases. Combining our observation on L_2 distance with our previous observation, we get that to achieve a fixed training error, more overparameterized model require smaller L_2 distance compared to their less overparameterized counterparts. This result is *consistent* with supervised learning, where it is known that more overparameterized model have smaller distance of parameters from the initialization Nagarajan and Kolter [2019].

I.4 Results on Miniboone dataset

To show experimental results on a real-world dataset, we use miniboone dataset [Dua and Graff, 2017]. The dataset contains examples of electron neutrino and muon neutrino. This dataset contains around 30K examples and lies in 43 dimensions. To test our phenomenon, we modify the official implementation of block neural autoregressive flow (BNAF) [Cao et al., 2019b] for CNF and Unconstrained Monotonic Neural Network Flow [Wehenkel and Louppe, 2019] for UNF. We use 3 hidden layers for CNF and 3 hidden layers for both embedding network and derivative network. We use one flow model for both of them and use a mini-batch SGD optimizer with a learning rate of 0.001. The figure to illustrate the change in training error by changing the width of the network for each dimension is plotted in 6. From the figure, we see that the training error for CNFs increases with an increase in width of the network whereas the training error for UNFs decreases with an increase in width of the network. This observation supports our theoretical results.

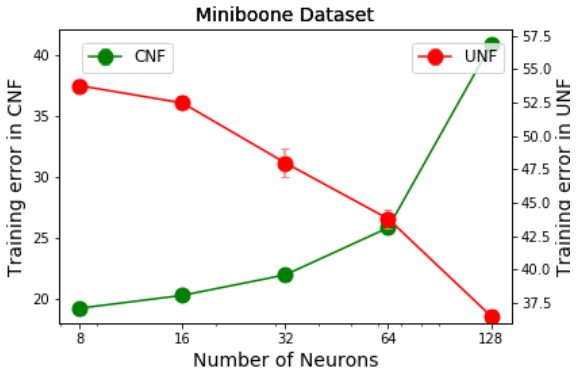

Figure 6: Training error of CNF and UNF after a fixed number of training epochs.

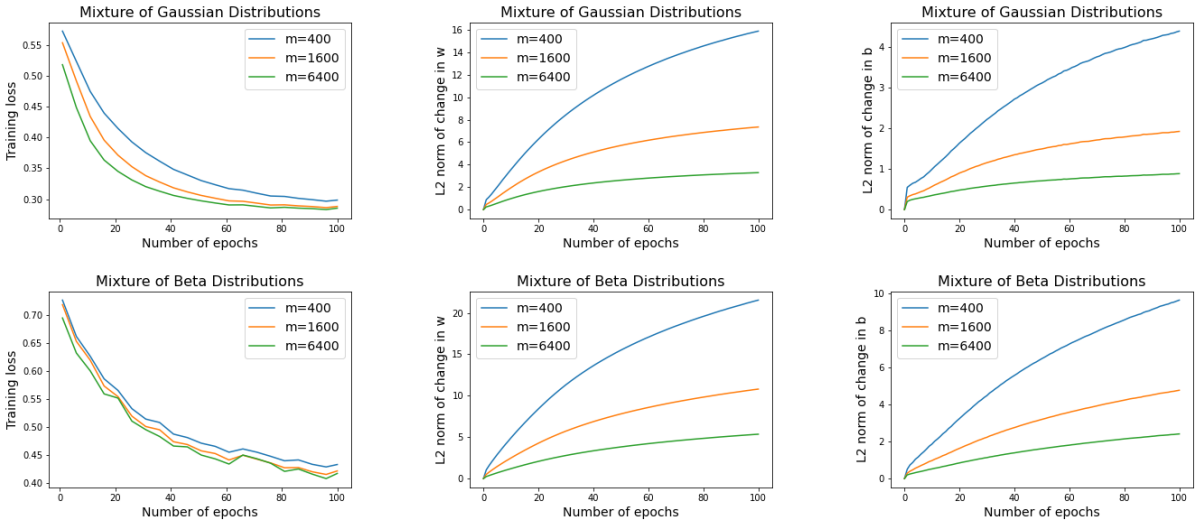

Figure 7: Effect of over-parameterization on training of unconstrained normalizing flow on mixture of Gaussian and mixture of beta distributions

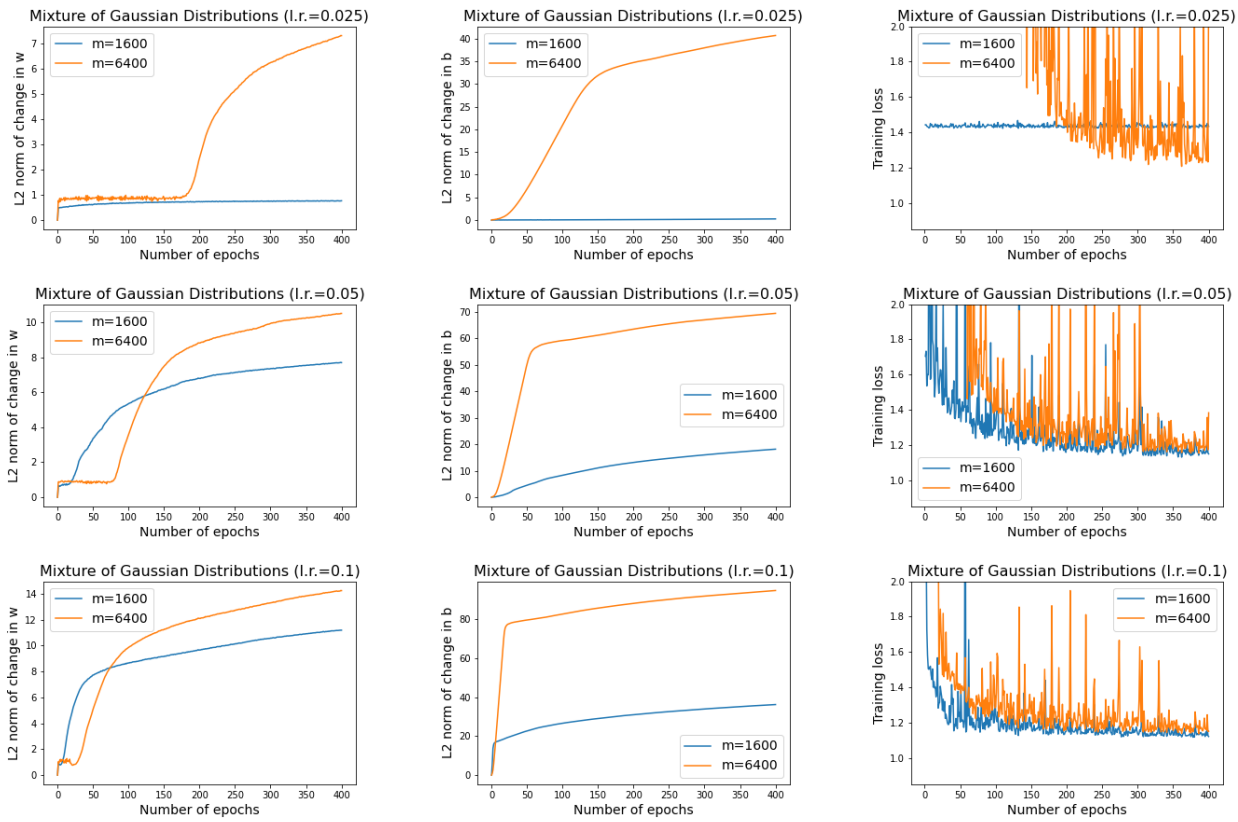

Figure 8: Effect of over-parameterization on training of CNF-NNWB on mixture of Gaussian dataset for number of hidden nodes $m = 1600, 6400$

I.5 Training curves for Constrained and Unconstrained Normalizing Flow

To provide a complete picture, we provide training error and L_2 distance of weights $W^{(t)}$ and biases $B^{(t)}$ from the initialization for all time step t during the training. We first discuss results for CNFs and then move our discussion to UNFs.

Constrained Normalizing Flow. In Figure 8, 9, 11, 10, 12 and 13, we plot number of epochs on x-axis and y-axis can be training error, L_2 distance of weights or L_2 distance of biases from the initialization.

In all figures, we see that for any fixed learning rate, curve of training error for smaller m is always below than curve of training for larger m, which proves our claim that increasing overparameterization hurts the training speed of CNF models. This phenomenon is consistent for all datasets, different initializations and various learning rates. Only exception to this phenomenon is results on mixture of Gaussian dataset for $m = 1600$ and $m = 6400$ and learning rate equal to 0.025 but note that in this case, the training of CNF for $m = 6400$ is very unstable and therefore, at some time steps, $m = 6400$ curve has slightly smaller training error than $m = 1600$ because of unstable training.

Apart from training error, we see that L_2 distance for biases (that is, L_2 norm of $B^{(t)}$) is always larger for large m. The difference is clearly visible and significant in comparison figures of large hidden layer nodes ($m = 1600$ and $m = 6400$). This is consistent across different initializations, datasets and learning rates. Only exception to this trend is results on mixture of Gaussian dataset for $m = 100$ and $m = 400$. Even in this case, L_2 distance is comparable for $m = 100$ and $m = 400$.

Unconstrained Normalizing Flow. Similar to Constrained Normalizing Flows, we study the effect of overparameterization on convergence speed and L_2-norm of $W^{(t)}$ and $B^{(t)}$. The first row of Figure 14 contains

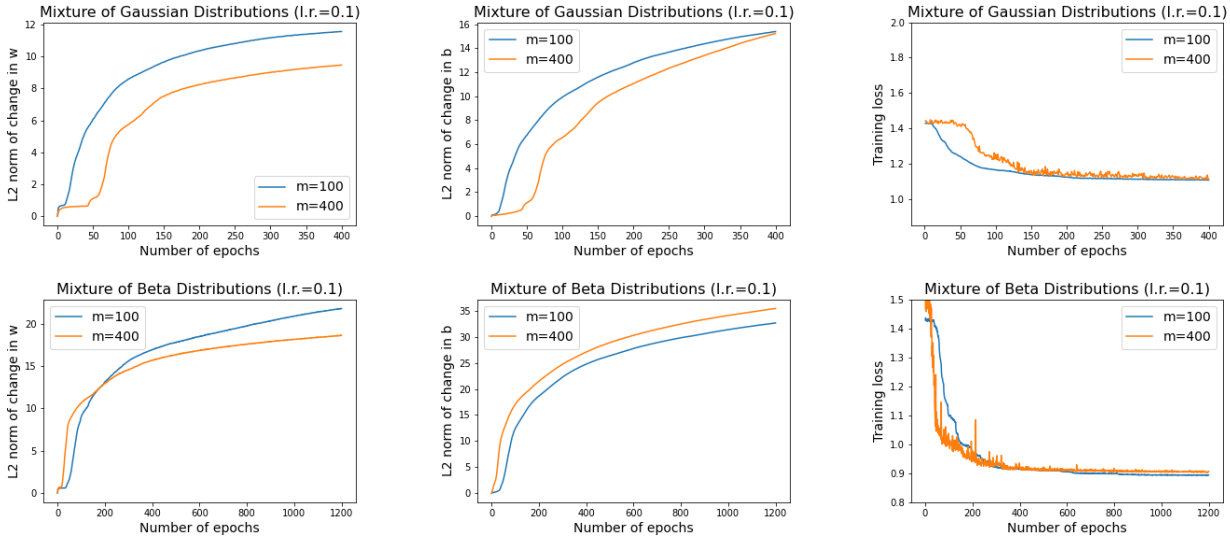

Figure 9: Effect of over-parameterization on training of small sized CNF-NNWB

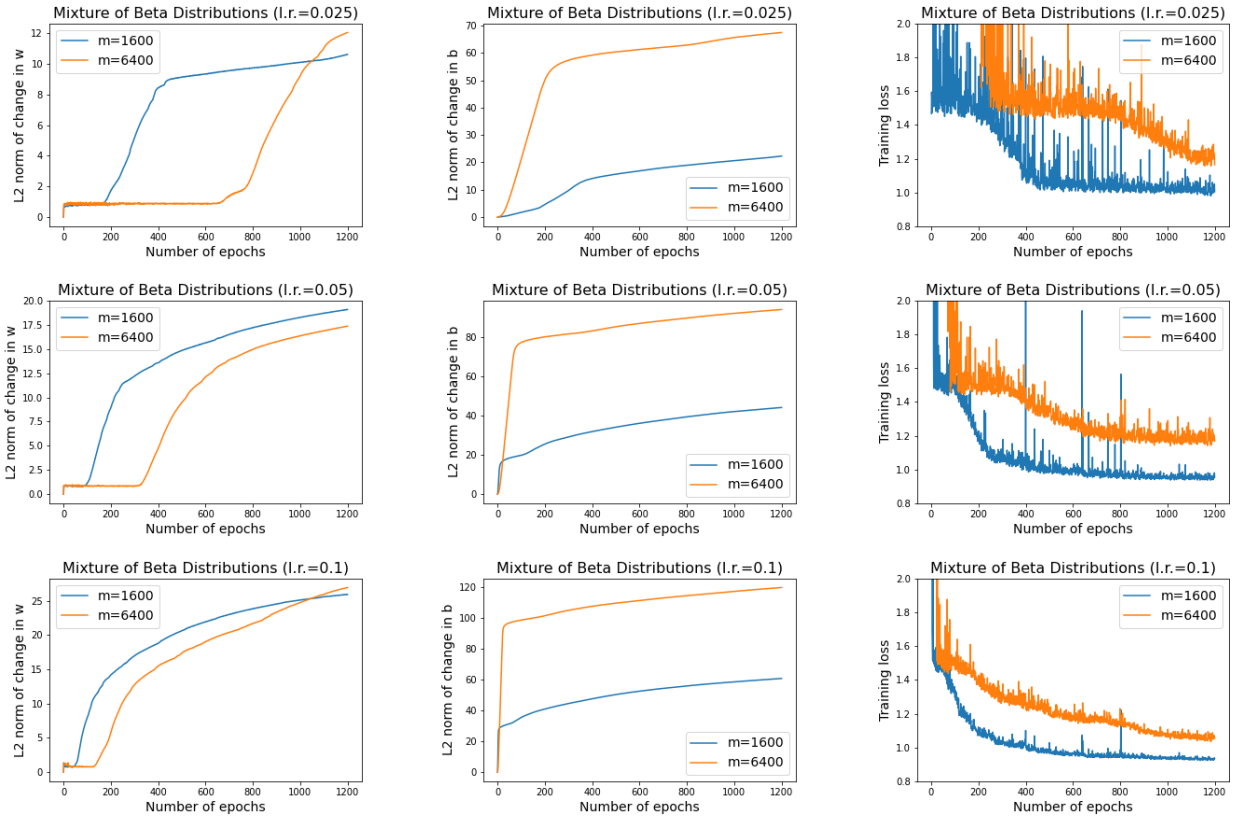

Figure 10: Effect of over-parameterization on training of CNF-NNWB on mixture of beta distribution dataset for number of hidden nodes $m = 1600, 6400$

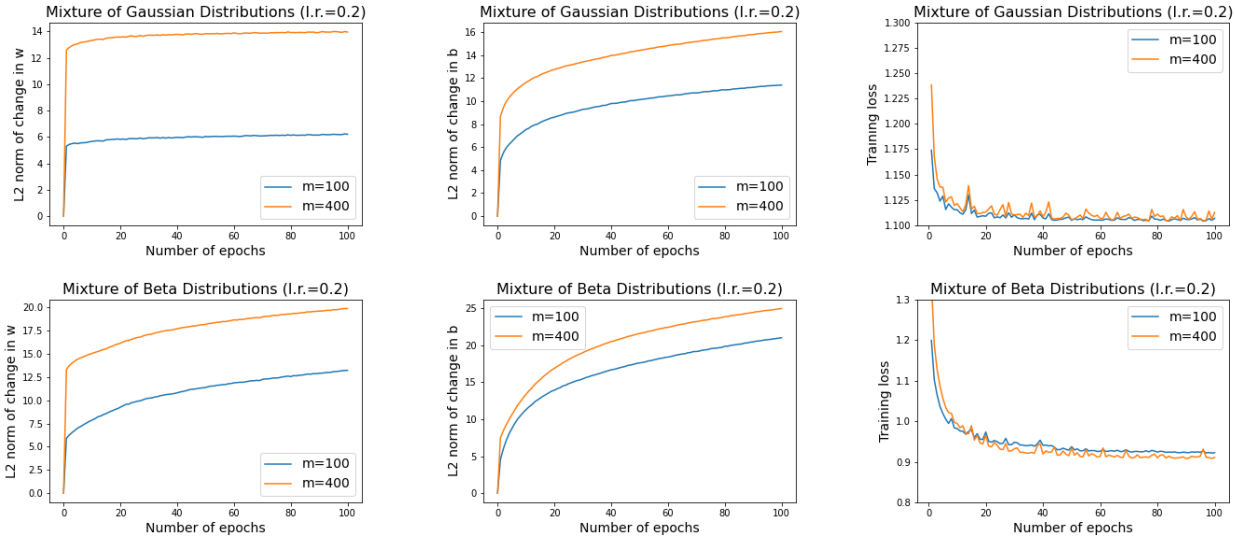

Figure 11: Effect of over-parameterization on training of small sized CNF-SNWB of weights and biases

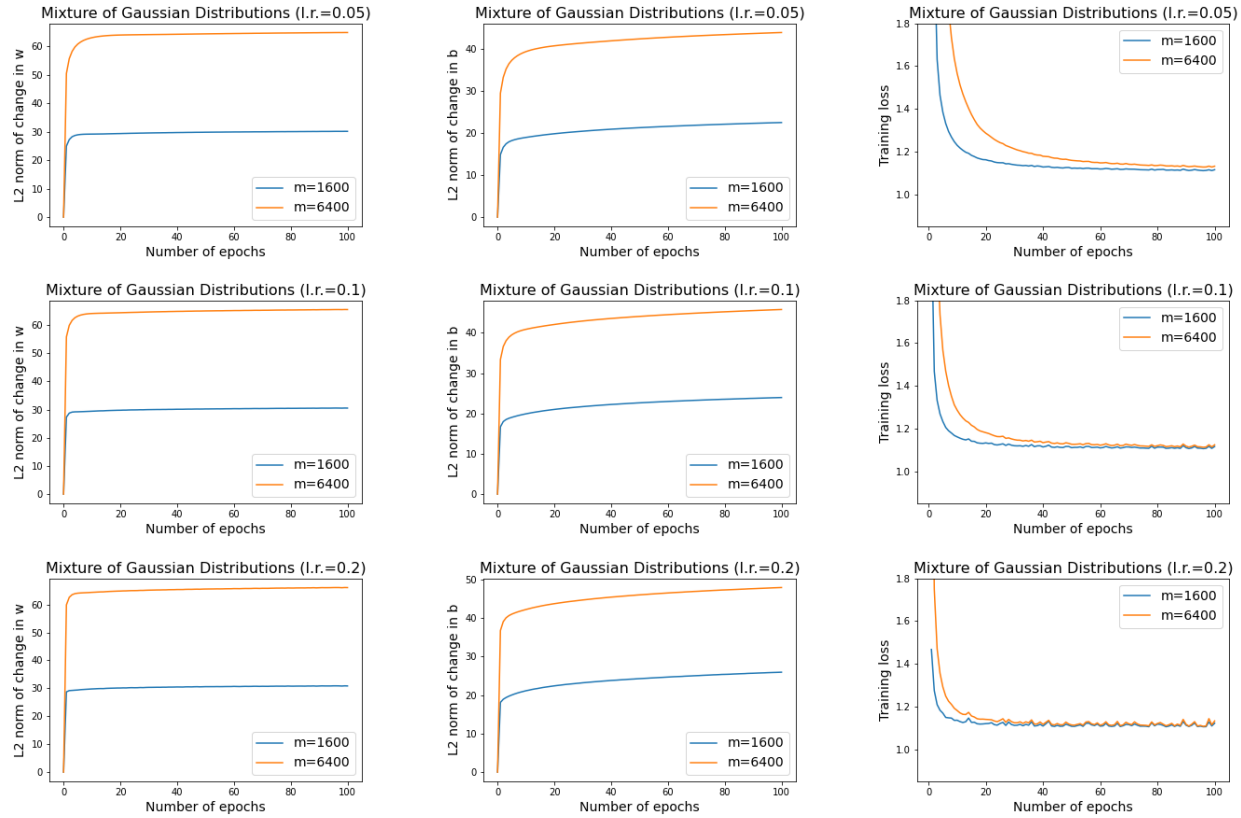

Figure 12: Effect of over-parameterization on training of CNF-SNWB on mixture of Gaussian dataset for number of hidden nodes $m = 1600, 6400$

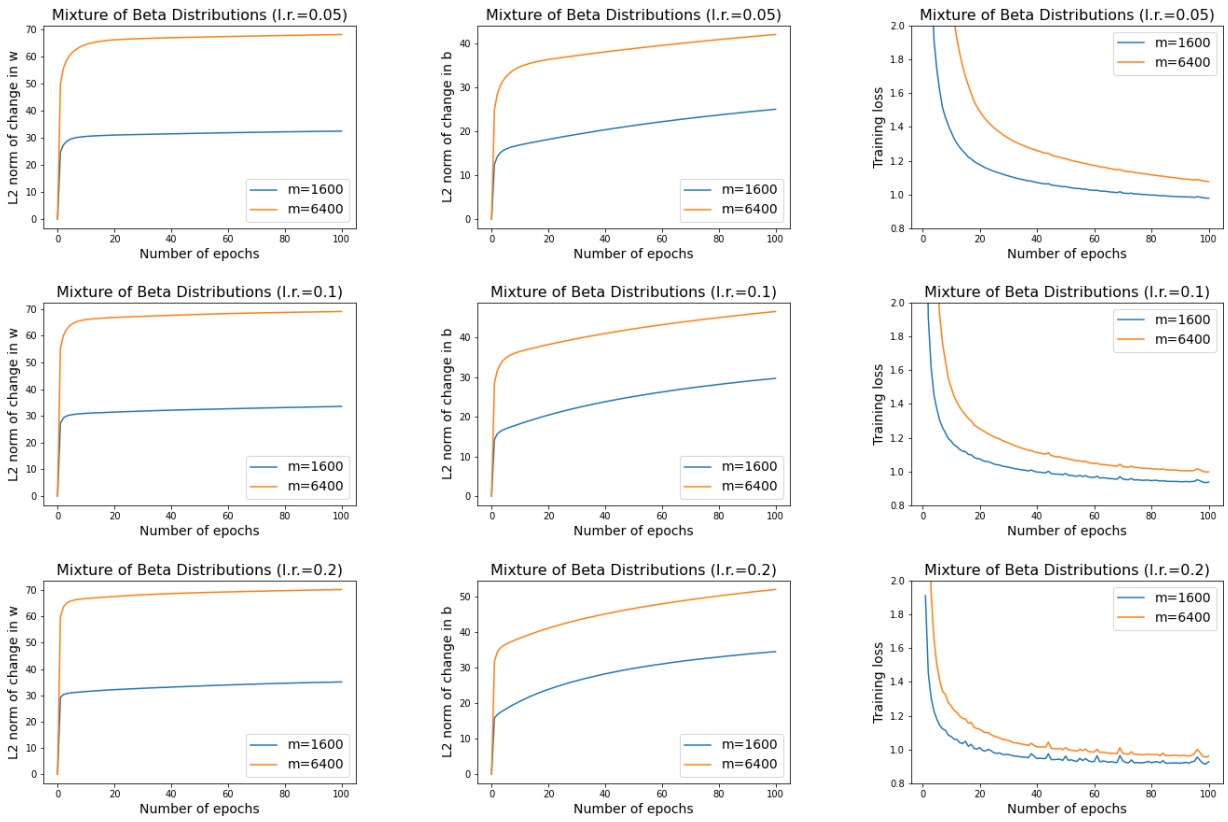

Figure 13: Effect of over-parameterization on training of CNF-SNWB on mixture of Beta distribution dataset for number of hidden nodes $m = 1600, 6400$

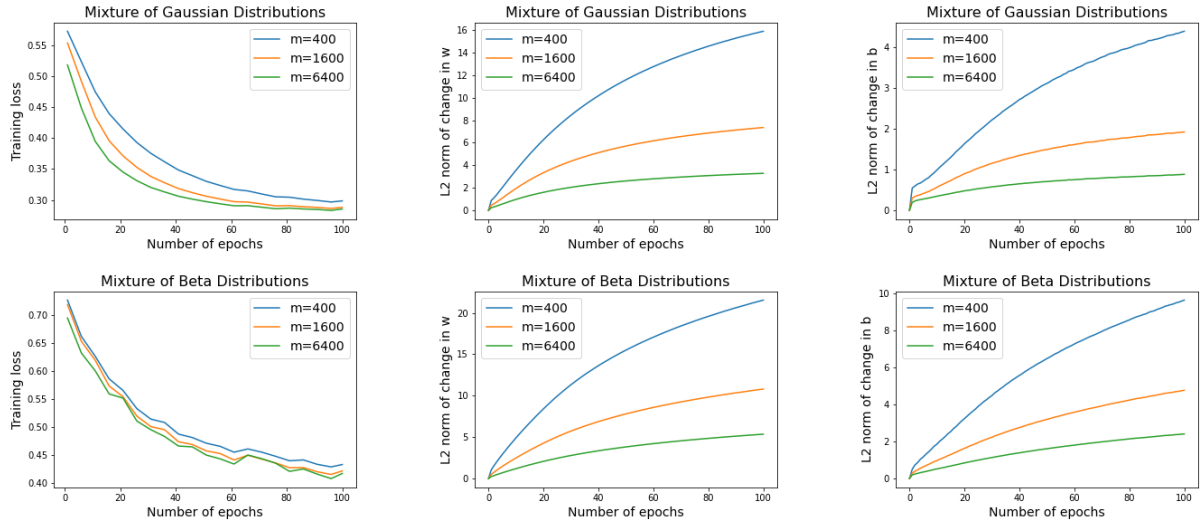

Figure 14: Effect of over-parameterization on training of UNF on mixture of Gaussian and mixture of beta distributions

results for mixture of Gaussians dataset and the second row contains results for mixtures of beta distributions dataset. From the first column of Fig. 14, we see that the training speed for larger m is better or comparable to smaller m. Additionally, we see that L_2-norm of $W^{(t)}$ and $B^{(t)}$ decreases significantly with increasing m. This results validate our theoretical finding that L_2 distance of parameters from the initialization decreases with increasing m.

J Related Work

Previous work on normalizing flows has studied different variants such as planar and radial flows in Rezende and Mohamed [2015], Sylvester flow in van den Berg et al. [2018], Householder flow in Tomczak and Welling [2016], masked autoregressive flow in Papamakarios et al. [2017]. Most variants of normalizing flows are specific to certain applications, and the expressive power (i.e., which base and data distributions they can map between) and complexity of normalizing flow models have been studied recently, e.g. Kong and Chaudhuri [2020] and Teshima et al. [2020]. Invertible transformations defined by monotonic neural networks can be combined into autoregressive flows that are universal density approximators of continuous probability distributions; see Masked Autoregressive Flows (MAF) Papamakarios et al. [2017], UNMM-MAF by Wehenkel and Louppe [2019], Neural Autoregressive Flows (NAF) by Huang et al. [2018], Block Neural Autoregressive Flow (B-NAF) by Cao et al. [2019a]. Unconstrained Monotonic Neural Network (UMNN) models proposed by Wehenkel and Louppe [2019] are particularly relevant to the technical part of our paper.

Koehler et al. [2020] theoretically study representation ability of affine couplings (a type of normalizing flow) and particularly analyze several aspects such as depth of normalizing flows. Lei et al. [2020], Balaji et al. [2021] show that when the generator is a two-layer tanh, sigmoid or leaky ReLU network, Wasserstein GAN trained with stochastic gradient descent-ascent converges to a global solution with polynomial time and sample complexity. Using the moments method and a learning algorithm motivated by tensor decomposition, Li and Dou [2020] show that GANs can efficiently learn a large class of distributions including those generated by two-layer networks. Nguyen et al. [2019a] show that two-layer autoencoders with ReLU or threshold activations can be trained with normalized gradient descent over the reconstruction loss to provably learn the parameters of any generative bilinear model (e.g., mixture of Gaussians, sparse coding model). Nguyen et al. [2019b] extend the work of Du et al. [2018] on supervised learning mentioned earlier to study weakly-trained (i.e., only encoder is trained) and jointly-trained (i.e., both encoder and decoder are trained) two-layer autoencoders, and show joint training requires less overparameterization and converges to a global optimum. The effect of overparameterization in unsupervised learning has also been of recent interest. Buhai et al. [2020] do an empirical study to show that across a variety of latent variable models and training algorithms, overparameterization can significantly increase

the number of recovered ground truth latent variables. Radhakrishnan et al. [2020] show that overparameterized autoencoders and sequence encoders essentially implement associative memory by storing training samples as attractors in a dynamical system.

K Useful facts

Fact K.1. *For any $i \geq 0$, let h_i denote the degree−i probabilists' Hermite polynomial*

$$h_i(x) = i! \sum_{m=0}^{\lfloor \frac{i}{2} \rfloor} \frac{(-1)^m}{m!(i-2m)!} \frac{x^{i-2m}}{2^m}.$$

The Hermite polynomials satisfy following summation and multiplication formulas.

$$h_i(x+y) = \sum_{k=0}^{i} \binom{i}{k} x^{i-k} h_k(y),$$
$$h_i(xy) = \sum_{k=0}^{\lfloor \frac{i}{2} \rfloor} y^{i-2k}(y^2-1)^k \binom{i}{2k} \frac{(2k)!}{k!} 2^{-k} h_{i-2k}(x).$$

Fact K.2. *Let h_i denote the degree−i probabilists' Hermite polynomial, then for $i > 0$, we have*

$$\mathbb{E}_{\beta \sim \mathcal{N}(0,1)} \left[h_i(\beta) \right] = 0.$$

Lemma K.3. *Suppose $Z_k \sim \mathcal{N}(0, \sigma^2)$ and $Y = \sum_{k=1}^n Z_k^2$ is chi-squared distribution with following property for all $t \in (0,1)$.*

$$Pr\left[\left| \frac{1}{n} \sum_{k=1}^{n} Z_k^2 - \sigma^2 \right| \geq t \right] \leq 2 \exp\left(-\frac{nt^2}{8\sigma^4} \right)$$

Proof. From example 2.11 from Wainwright [2019], for $Z'_k \sim \mathcal{N}(0,1)$ and $Y = \sum_{k=1}^n Z_k'^2$ is chi-squared distribution with following property for all $t \in (0,1)$.

$$\Pr\left[\left| \frac{1}{n} \sum_{k=1}^{n} Z_k'^2 - 1 \right| \geq t \right] \leq 2 \exp\left(-\frac{nt^2}{8} \right)$$

Using above equation for $\frac{Z_k}{\sigma}$,

$$\Pr\left[\left| \frac{1}{n} \sum_{k=1}^{n} \frac{Z_k^2}{\sigma^2} - 1 \right| \geq \frac{t}{\sigma^2} \right] \leq 2 \exp\left(-\frac{nt^2}{8\sigma^4} \right)$$
$$\Pr\left[\left| \frac{1}{n} \sum_{k=1}^{n} Z_k^2 - \sigma^2 \right| \geq t \right] \leq 2 \exp\left(-\frac{nt^2}{8\sigma^4} \right)$$

□

Lemma K.4. *Let $X_1, X_2, ..., X_n$ be independent random variables from $\mathcal{N}(0, \sigma^2)$, then with at least $1 - \frac{1}{c_1}$ probability, following holds.*

$$\max_{i \in \{1,2,\ldots,n\}} |X_i| \leq 2c_1 \sigma \sqrt{2 \log n}$$

Proof. From Romberg [2012],

$$\mathbb{E}\left[\max_{i \in \{1,2,\ldots,n\}} |X_i| \right] \leq \sigma\left(\sqrt{2 \log n} + 1 \right) \leq 2\sigma\left(\sqrt{2 \log n} \right)$$

Assuming $n \geq 2$, the last inequality follows. Using Markov's inequality,

$$\Pr\left(\max_{i\in\{1,2,\ldots,n\}} |X_i| \geq 2c_1\sigma\left(\sqrt{2\log n}\right)\right) \leq \frac{1}{c_1}$$
$$\Pr\left(\max_{i\in\{1,2,\ldots,n\}} |X_i| \leq 2c_1\sigma\left(\sqrt{2\log n}\right)\right) \geq 1 - \frac{1}{c_1}$$

s

□

Lemma K.5. *For standard Gaussian random variable X from $\mathcal{N}(0,\sigma^2)$, the following anti-concentration inequality holds:*

$$\Pr\big(|X| \leq R\big) \leq \frac{2R}{\sigma\sqrt{2\pi}}.$$

Proof. (From Du et al. [2018]) For the standard Gaussian random variable $\frac{X}{\sigma}$,

$$\Pr\left(\left|\frac{X}{\sigma}\right| \leq R\right) \leq \frac{2R}{\sqrt{2\pi}}$$

Using $R = \frac{R'}{\sigma}$, we get the required result. □

Lemma K.6. *Suppose function $f : \mathbb{R}^d \to \mathbb{R}$ is L_g-Lipschitz continuous and L_i-coordinate wise Lipschitz continuous i.e.*

$$\begin{aligned}\big|f(\mathbf{a}) - f(\mathbf{b})\big| \leq& L_g\|\mathbf{a}-\mathbf{b}\| \\ & \forall \mathbf{a},\mathbf{b} \in \mathbb{R}^d \quad \textit{(Standard Lipschitz continuity)}\\ \big|f(a_1,a_2,...,a_i,...,a_d) - f(a_1,a_2,...,b_i,...,a_d)\big| \leq& L_i|a_i - b_i| \\ & \forall a_1,a_2,...,a_i,...,a_d,b_i \in \mathbb{R} \textit{ and } \forall i \in [d] \quad \textit{(Coordinate-wise Lipschitz continuity)}\end{aligned}$$

If a function f satisfies L_i-coordinate wise Lipschitz continuity for all i, then function f follows following inequality.

$$\big|f(a_1,a_2,...,a_d) - f(b_1,b_2,...,b_d)\big| \leq \sum_{i=1}^{n} L_i|a_i - b_i|$$

Moreover, the function f also satisfies standard Lipschitz continuity with L_g Lipschitz constant where inequality between L_g and L_i is as follows.

$$L_g \leq \sqrt{\sum_{i=1}^{d} L_i^2}$$

Proof. Define $\mathbf{a} = (a_1,a_2,...,a_d)$ and $\mathbf{b} = (b_1,b_2,...,b_d)$.

$$\begin{aligned}\big|f(a_1,a_2,...,a_d) - f(b_1,b_2,...,b_d)\big| \leq& \big|f(a_1,a_2,...,a_d) - f(b_1,a_2,...,a_d)\big| \\ &+\big|f(b_1,a_2,a_3,...,a_d) - f(b_1,b_2,a_3,...,a_d)\big| \\ &+\big|f(b_1,b_2,a_3,...,a_d) - f(b_1,b_2,b_3,...,a_d)\big| \\ &+ ... +\big|f(b_1,b_2,...,b_{d-1},a_d) - f(b_1,b_2,b_3,...,b_d)\big| \\ \leq& L_1|a_1-b_1| + L_2|a_2-b_2| + ... + L_d|a_d-b_d| \\ \leq& \sqrt{\sum_{i=1}^{d} L_i^2}\|\mathbf{a}-\mathbf{b}\|_2\end{aligned}$$

where last inequality follows from Cauchy-Schwarz inequality. □

Fact K.7. *(Hoeffding's inequality on Binomial random variable) If we have a binomial random variable with parameters n (total number of trials) and p (probability of success). For $k \geq np$, following inequality holds.*

$$Pr(X \geq k) \leq \exp\left(-2n\left(\frac{k}{n} - p\right)^2\right)$$

Fact K.8. *(Hoeffding's inequality) Let $X_1, X_2, \ldots, X_n$ be independent random variables where X_i is bounded in the interval $[a_i, b_i]$. Then, for any $t \geq 0$, we have*

$$\Pr\left(\left|(X_1 + X_2 + \ldots + X_n) - \mathbb{E}\left[X_1 + X_2 + \ldots + X_n\right]\right| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n (a_i - b_i)^2}\right).$$

Fact K.9. *(Half-normal distribution) If X follows a normal distribution with with mean 0 and variance σ^2, $\mathcal{N}\left(0, \sigma^2\right)$, then $Y = |X| = X sign(X)$ follows a half-normal distribution with mean $\mathbb{E}[Y] = \frac{\sigma\sqrt{2}}{\sqrt{\pi}}$.*

Fact K.10. *For a gaussian random variable $X \sim \mathcal{N}(0, \sigma^2)$, $\forall t \in (0, \sigma)$, we have*

$$Pr(|X| \geq t) \geq 1 - \frac{4t}{5\sigma}$$

Fact K.11. *The sum of reciprocals of the squares of the natural numbers is given by*

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \leq 2$$

Fact K.12. *(Theorem 3.1($r_5^{'}$) of Li and Yeh [2013]) For any $\alpha > 1$ and $x \in \left[0, \frac{1}{\alpha - 1}\right)$,*

$$(1 + x)^{\alpha} \leq \frac{1}{1 - \frac{\alpha x}{1+x}} = 1 + \frac{\alpha x}{1 - (\alpha - 1)\, x}$$

Fact K.13. *(McDiarmid's Inequality) Let V be some set and let $f : V^m \mapsto \mathbb{R}$ be a function such that for some $c_i > 0$, for all $i \in [m]$ and for all $x_1, \ldots, x_m, x_i' \in V$, we have*

$$\left|f\left(x_1, \ldots, x_i, \ldots, x_m\right) - f\left(x_1, \ldots, x_i', \ldots, x_m\right)\right| \leq c_i.$$

Let $X_1, X_2, \ldots X_m$ are independent random variables taking values in V. Then,

$$\Pr\left[f(X_1, X_2, \ldots, X_m) - \mathbb{E}\left(X_1, X_2, \ldots, X_m\right) \geq \epsilon\right] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^m c_i^2}\right).$$

Fact K.14. *If Arithmetic-Geometric Progression(AGP) is as follows.*

$$a, (a + d)r, (a + 2d)r^2, (a + 3d)r^3, ..., \left[a + (n - 1)d\right] r^{n-1}$$

where a is the initial term, d is the common difference and r is the common ratio. The sum of the first n terms of the AGP (S_n) is given by

$$S_n = \frac{a - \left[a + (n - 1)d\right] r^n}{1 - r} + \frac{dr\left(1 - r^{n-1}\right)}{(1 - r)^2}$$

Definition K.15. *Let $\mathcal{F}$ be a set of functions $\mathbb{R}^d \to \mathbb{R}$ and $\mathcal{X} = (x_1, x_2, ..., x_n)$ be a finite set of samples. The empirical Rademacher complexity of $\mathcal{F}$ with respect to $\mathcal{X}$ is defined by*

$$\hat{\mathcal{R}}\left(\mathcal{X}; \mathcal{F}\right) = \mathbb{E}_{\xi \sim \{\pm 1\}^n}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \xi_i f(x_i)\right].$$

The following results are standard and can be found, e.g., in Allen-Zhu et al. [2019].

Lemma K.16. *Rademacher complexity has the following properties:*

a. *For any $d \in \mathbb{R}$ and $x \in \mathbb{R}^d$ with $\|x\|_2 \leq 1$. The function class $\mathcal{F} = \{x \mapsto \langle w, x\rangle + b \mid \|w\|_2 \leq B, |b| \leq B\}$ has Rademacher complexity $\hat{\mathcal{R}}(\mathcal{X}, \mathcal{F}) \leq \frac{2B}{\sqrt{n}}$.*

b. *Given classes $\mathcal{F}_1, \mathcal{F}_2$ functions, $\hat{\mathcal{R}}(\mathcal{X}; \mathcal{F}_1 + \mathcal{F}_2) = \hat{\mathcal{R}}(\mathcal{X}; \mathcal{F}_1) + \hat{\mathcal{R}}(\mathcal{X}; \mathcal{F}_2)$.*

c. *Given classes $\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_m$ of functions of type $\mathcal{X} \to \mathbb{R}$ and suppose $w \in \mathbb{R}^m$ is a fixed vector, then $\mathcal{F}' = \{x \mapsto \sum_{r=1}^m w_r \sigma\left(f_r(x)\right) \mid f_r \in \mathcal{F}_r\}$ satisfies $\hat{\mathcal{R}}\left(\mathcal{X}; \mathcal{F}'\right) \leq 2\|w\|_1 \max_{r\in[m]} \hat{\mathcal{R}}(\mathcal{X}; \mathcal{F}_r)$ where σ is a 1-Lipschitz continuous function.*

Proof. These are standard results and can be found in Allen-Zhu et al. [2019] and Shalev-Shwartz and Ben-David [2014]. □

Fact K.17. *(Rademacher Complexity) If $\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_k$ are classes of functions of type $\mathbb{R}^d \to \mathbb{R}$ and $L_x : \mathbb{R}^d \to [-b, b]$ is a L_g-Lipschitz-continuous function for every x in the support of $\mathcal{D}$, then*

$$\sup_{f_1\in\mathcal{F}_1,\ldots,f_k\in\mathcal{F}_k} \left| \mathbb{E}_{x\in\mathcal{D}}\left[L_x\left(f_1(x), ..., f_k(x)\right)\right] - \frac{1}{n}\sum_{i=1}^n L_x\left(f_1(x_i), ..., f_k(x_i)\right)\right| \leq 2\hat{\mathcal{R}}(\mathcal{X}; \mathcal{L}) + b\sqrt{\frac{\log\frac{1}{\delta}}{2n}}$$

where $\mathcal{L}$ is set of functions obtained by composing L_x with $\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_k$, that is $\mathcal{L} := \{L_x \circ (f_1, \ldots, f) \mid f_1 \in \mathcal{F}_1, \ldots, f_k \in \mathcal{F}_k\}$. Using vector contraction inequality from Maurer [2016], we get

$$\begin{aligned}\sup_{f_1\in\mathcal{F}_1,\ldots,f_k\in\mathcal{F}_k} &\left| \mathbb{E}_{x\in\mathcal{D}}\left[L_x\left(f_1(x), ..., f_k(x)\right)\right] - \frac{1}{n}\sum_{i=1}^n L_x\left(f_1(x_i), ..., f_k(x_i)\right)\right| \\ &\leq 2\sqrt{2}L_g\left(\sum_{i=1}^k \hat{\mathcal{R}}(\mathcal{X}; \mathcal{F}_i)\right) + b\sqrt{\frac{\log\frac{1}{\delta}}{2n}}.\end{aligned}$$