Initialization Matters: Privacy-Utility Analysis of Overparameterized Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

We analytically investigate how overparameterization of models in randomized machine learning algorithms impacts the information leakage about their training data. Specifically, we prove a privacy bound for the KL divergence between model distributions on worst-case neighboring datasets, and explore its dependence on the initialization, width, and depth of fully connected neural networks. We find that this KL privacy bound is largely determined by the expected squared gradient norm relative to model parameters during training. Notably, for the special setting of linearized network, our analysis indicates that the squared gradient norm (and therefore the escalation of privacy loss) is tied directly to the per-layer variance of the initialization distribution. By using this analysis, we demonstrate that privacy bound improves with increasing depth under certain initializations (LeCun and Xavier), while degrades with increasing depth under other initializations (He and NTK). Our work reveals a complex interplay between privacy and depth that depends on the chosen initialization distribution. We further prove excess empirical risk bounds under a fixed KL privacy budget, and show that the interplay between privacy utility trade-off and depth is similarly affected by the initialization.

1 Introduction

Deep neural networks (DNNs) in the over-parameterized regime (i.e., more parameters than data) perform well in practice but the model predictions can easily leak private information about the training data under inference attacks such as membership inference attacks [38] and reconstruction attacks. [14, 5, 23] This leakage can be mathematically measured in terms of how much the algorithm's output distribution changes if it were trained on a neighboring dataset (that only differs in one record), following the differential privacy (DP) framework [18].

To train differential private model, a typical way is randomize each gradient update in neural networks training, e.g., stochastic gradient descent (SGD), which leads to the most widely applied differentially private training algorithm in the literature – DP-SGD [1]. In each step, DP-SGD employs gradient clipping and adds calibrated Gaussian noise, and thus it comes with a differential privacy guarantee that scales with the noise multiplier (i.e., per-dimensional Gaussian noise standard deviation divided by the clipping threshold) and number of training epochs. However, this privacy bound [1] is overly general due to its independence on the network properties (e.g., width and depth) and training schemes (e.g., initializations). Accordingly, a natural question arises in the community:

How does overparameterization (e.g., increasing width and depth) of neural networks affect the (worst-case) privacy bound of the training algorithm?

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Table 1: Our results for the privacy utility trade-off of training linearized network (3) via Langevin diffusion, under different width m, depth L and initializations. We set per-layer width $m_0 = d$, $m_1, \cdots, m_{L-1} = m$ and $m_L = o$. We prove privacy bound in KL divergence, and obtain excess empirical risk bounds given KL privacy budget ε. For the excess risk bounds, we assume the network width $m = \Omega(n)$ is sufficiently large, and the data and network satisfy regularity assumption Assumption 2.1. For NTK, He and LeCun initialization, we observe that the privacy utility trade-off improves with overparameterization (increasing depth).

Init	Variance β_l for layer l	Gradient norm constant B (9)	Approximate lazy training distance $\tilde{R}$ (12)	Excess Empirical risk under KL privacy bound ε (Corollary 5.4)
LeCun	$1/m_{l-1}$	$\frac{om(L-1+\frac{d}{m})}{2^{L-1}d}$	$\tilde{O}(\frac{n}{m(L-1)})$	$\tilde{O}\left(\frac{1}{n^2} + \sqrt{\frac{1}{2^L d\varepsilon}\left(1+\frac{d}{m(L-1)}\right)}\right)$
He	$2/m_{l-1}$	$\frac{om(L-1+\frac{d}{m})}{d}$	$\tilde{O}(\frac{n}{2^L m(L-1)})$	$\tilde{O}\left(\frac{1}{n^2} + \sqrt{\frac{1}{2^L d\varepsilon}\left(1+\frac{d}{m(L-1)}\right)}\right)$
NTK	$2/m_l, l<L$ $1/o, l=L$	$\frac{m(L-1)}{2} + o$	$\tilde{O}(\frac{n}{d\cdot 2^L\cdot(m(L-2)+1)})$	$\tilde{O}\left(\frac{1}{n^2} + \sqrt{\frac{1}{d\cdot 2^L\varepsilon}\cdot\frac{m(L-1)+2}{m(L-2)+1}}\right)$
Xavier	$\frac{2}{m_{l-1}+m_l}$	$\frac{L-1+\frac{d+o}{2m}}{2^{L-1}(1+\frac{d}{m})(1+\frac{o}{m})}$	-	-

- The Xavier initialization makes neural networks fall into non-lazy training regime [31], so we do not include the lazy training distance nor privacy-utility trade-off analysis here.

To answer this question, we would need new algorithmic framework and (or) new privacy analyses. In this paper, we focus on analyzing privacy for the Langevin diffusion algorithm [1]. This is to avoid artificially setting a sensitivity constraint on the gradient update and thus making the privacy bound insensitive to the network overparameterization (as in DP-SGD analysis). Instead, we prove a KL privacy bound for Langevin diffusion that scales with the expected gradient difference between the training on any two worst-case neighboring datasets (Theorem 3.1). [2] By proving precise upper bounds on the expected ℓ_2-norm of this gradient difference, we obtain KL privacy bounds for fully connected neural network (Lemma 3.2) and its linearized variant (Corollary 4.2) that changes with the network width, depth and per-layer variance for the initialization distribution. We summarized the details of our KL privacy bounds in Table 1, and highlight our key observations below.

- Width always worsen privacy, under all the considered initialization distributions. Meanwhile, the interplay between network depth and privacy is much more complex and crucially depends on what initialization distribution is used and how long the training time is.
- Specifically, when the initialization distribution has small per-layer variance (such as LeCun and Xavier initialization), our KL privacy bound for training fully connected network (with a small amount of time) and for training linearized network (with finite time) decay exponentially with increasing depth, as long as the depth is large enough. To the best of our knowledge, this is the first time that an improvement of privacy bound under overparameterization is observed for randomized training algorithm.

To further understand how the privacy utility trade-off is affected by overparameterization, we also analyze the excess empirical risk and excess population risk of training linearized network using Langevin diffusion. Our risk bounds scale with the lazy training distance R (i.e., how close is the initialization vector to an optimal solution for the empirical risk minimization problem), as well as a constant B for expected gradient norm in Langevin diffusion. By analyzing these two terms precisely under overparameterization, we prove that given any fixed KL privacy budget ε, our risk bounds strictly improves with increasing depth and width for linearized network under LeCun and He initialization. To our best knowledge, this is the first time that such a gain in privacy-utility trade-off

[1] A key difference between this paper and existing privacy utility analysis of Langevin diffusion [21] is that we analyze in the absence of gradient clipping or Lipschitz assumption on loss function. Our results also readily extend to discretized noisy GD with constant step-size (as discussed in Appendix E).

[2] We focus on KL privacy loss because it is a more relaxed distinguishability notion than standard (ε, δ)-DP, and therefore could be upper bounded even without gradient clipping. Moreover, KL divergence enables upper bound for the advantage (relative success) of various inference attacks, as studied in recent works [32, 22].

due to overparameterization (increasing depth) is shown. Meanwhile, prior results only prove (nearly) dimension-independent privacy utility trade-off for such linear models in the literature [39, 26, 30]. Our improvement demonstrates the unique benefits of our new KL privacy analysis in understanding the effect of overparameterization.

1.1 Related Works

Overparameterization in DNNs and NTK. Theoretical demonstration on the benefit of over-parameterization in DNNs occurs in global convergence [2, 17], generalization [3, 13]. Under proper initialization, the training dynamics of over-parameterized DNNs can be described by a kernel function, termed as neural tangent kernel (NTK) [25], which stimulates a series of analysis in DNNs. Accordingly, over-parameterization has been demonstrated to be beneficial/harmful to several topics in deep learning, e.g., robustness [12, 47], covariate shift [44]. However, the relationship between overparameterization and privacy (based on the differential privacy framework) remains largely an unsolved problem, as the training dynamics typically change [11] after adding new components in the privacy-preserving learning algorithm (such as DP-SGD [1]) to enforce privacy constraints.

Membership inference privacy risk under overparameterization. A recent line of works [42, 43] investigates how overparameterization affects the theoretical and empirical privacy in terms of membership inference advantage, and proves novel trade-off between privacy and generalization error. These are the closest works in the literature to our objective of investigating the interplay between privacy and overparameterization. However, Tan et al. [42, 43] focus on proving upper bounds for an average-case privacy risk defined by the advantage (relative success) of membership inference attack on models trained from randomly sampled training dataset from a population distribution. By contrast, our KL privacy bound is heavily based on the strongest adversary model in the differential privacy definition, and holds under an arbitrary *worst-case* pair of neighboring datasets that only differ in one record. Our setting for model (fully connected network) is also very different from that considered in Tan et al. [42, 43], thus requiring very different analysis tools.

Differentially private learning in high dimension. Standard results for private empirical risk minimization [6, 41] and private stochastic convex optimization [7, 9, 4] under ℓ_1 and ℓ_2 constraints suggest that there is an unavoidable factor d in the empirical risk and population risk that depends on the model dimension. However, for unconstrained optimization, it is possible to go across the dimension-dependency in proving risk bounds for certain class of problems (such as generalized linear model [39]). Recently, there is a growing line of works that prove dimension-independent excess risk bounds for differentially private learning, by utilizing the low-rank structure of data features [39] or gradient matrices [26, 30] in training. Several follow-up works [27, 10] further explore techniques to enforce the low-rank property (via random projection) and boost privacy utility trade-off. However, all the works focus on investigating a general high-dimensional problem for private learning, rather than separating the study for different network choices such as structure, width, depth and initializaiton. On the contrary, our study focus on the fully connected neural network and its linearized variant, which enables us to prove more precise privacy utility trade-off bounds for these particular networks under overparameterization.

2 Problem and Methodology

We consider the following standard multi-class supervised learning setting. Let $\mathcal{D} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n)$ be a finite input dataset of size n, where each input data record $\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$ contains a d-dimensional input feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$ and a label vector $\boldsymbol{y}_i \in \mathcal{Y} = \{0, 1\}^o$ on o possible classes. The goal of learning is to learn a neural network output function $\boldsymbol{f}_{\boldsymbol{W}}(\cdot) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\boldsymbol{W}$ that achieves high prediction performance on the training dataset $\mathcal{D}$. Formally, we consider the learning objective to be the empirical risk defined as follows.

$$\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}_i); \boldsymbol{y}_i)\,, \tag{1}$$

where $\ell(\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}_i); \boldsymbol{y}_i)$ is a loss function that reflects the approximation quality of model prediction $f_{\boldsymbol{W}}(\boldsymbol{x}_i)$ compared to the ground truth label $\boldsymbol{y}_i$. For simplicity, we assume that the cross-

entropy loss is used, i.e., $\ell(\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}); \boldsymbol{y}) = -\langle \boldsymbol{y}, \log \text{softmax}(\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}))\rangle$ for multi-output network, and $\ell(\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}); \boldsymbol{y}) = \log(1 + \exp(-\boldsymbol{y}\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}))$ for single-output network.

Network. We focus on the *multi-output, fully connected, deep neural network (DNN) with ReLU activation* with depth L (i.e., $L-1$ hidden layers). Denote the width of each hidden layer with $m_1, \cdots, m_{L-1}$. The output function $f_{\boldsymbol{W}}(\boldsymbol{x}) \coloneqq \boldsymbol{h}_L(\boldsymbol{x})$ is defined recursively as follows.

$$\boldsymbol{h}_0(\boldsymbol{x}) = \boldsymbol{x}; \quad \boldsymbol{h}_l(\boldsymbol{x}) = \phi(\boldsymbol{W}_l \boldsymbol{x}) \text{ for } l = 1, \cdots, L-1; \quad \boldsymbol{h}_L(\boldsymbol{x}) = \boldsymbol{W}_L \boldsymbol{h}_{L-1}(\boldsymbol{x}) \tag{2}$$

where $h_l(\boldsymbol{x})$ denotes the output after activation at the l-th layer, the parameter matrices of each layer of the neural network satisfy $\boldsymbol{W}_1 \in \mathbb{R}^{m_1 \times d}$, $\boldsymbol{W}_l \in \mathbb{R}^{m_l \times m_{l-1}}$, $l = 2, \ldots, L-1$ and $\boldsymbol{W}_L \in \mathbb{R}^{o \times m_{L-1}}$. The model parameter $\boldsymbol{W} \coloneqq (\text{Vec}(\boldsymbol{W}_1), \ldots, \text{Vec}(\boldsymbol{W}_L)) \in \mathbb{R}^{m_1 \cdot d + m_2 \cdot m_1 + \cdots + o \cdot m_{L-1}}$ consists of concatenation of vectorizations for parameters of all the layers. For consistency, we also denote $m_0 = d$ and $m_L = o$.

We also analyze the following ***linearized network***, which is used in prior works [28, 2, 34] as an important tool to (approximately and qualitatively) analyze the training dynamics of deep neural networks. More formally, the linearized network $\boldsymbol{f}_{\boldsymbol{W}}^{lin,0}(\boldsymbol{x})$ is a first-order Taylor expansion of the fully connected ReLU network at initialization parameter $\boldsymbol{W}_0^{lin}$, as follows.

$$\boldsymbol{f}_{\boldsymbol{W}}^{lin,0}(\boldsymbol{x}) \equiv \boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}) + \frac{\partial \boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x})}{\partial \boldsymbol{W}}\Big|_{\boldsymbol{W}=\boldsymbol{W}_0^{lin}} \left(\boldsymbol{W} - \boldsymbol{W}_0^{lin}\right), \tag{3}$$

where $\boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x})$ is the output function of the fully connected ReLU network (2) at initialization $\boldsymbol{W}_0^{lin}$. We denote $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D}) = \sum_{i=1}^n \ell\left(\boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_i) + \frac{\partial \boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x})}{\partial \boldsymbol{W}}|_{\boldsymbol{W}=\boldsymbol{W}_0^{lin}}(\boldsymbol{W} - \boldsymbol{W}_0^{lin}); \boldsymbol{y}_i\right)$ as the empirical loss function for training linearized network, by plugging (3) into (1).

Langevin Diffusion. In terms of optimization algorithm, we focus on the *Langevin diffusion* algorithm [29] with per-dimensional noise variance σ^2. Note that we aim to *avoid gradient clipping* while still proving KL privacy bounds. After initializing the model parameters $\boldsymbol{W}_0$ at time zero, the model parameters $\boldsymbol{W}_t$ at subsequent time t evolves as the below stochastic differential equation.

$$d\boldsymbol{W}_t = -\nabla \mathcal{L}(\boldsymbol{W}_t; \mathcal{D}) dt + \sqrt{2\sigma^2} d\boldsymbol{B}_t\,. \tag{4}$$

Initialization Distribution. The initialization of parameters $\boldsymbol{W}_0$ crucially affects the convergence of Langevin diffusion, as observed in prior literatures [46, 20, 19]. Moreover, when the network function depends on the initialization parameters (as in linearized network (3)), the stationary distribution of Langevin diffusion also depends on the initialization distribution (as discussed in Section 5). In this work, we investigate the following general class of Gaussian initialization distribution with (possibly depth-dependent) variance for the parameters in each layer. For any layer $l = 1, \cdots, L$, we have that

$$[\boldsymbol{W}^l]_{ij} \sim \mathcal{N}(0, \beta_l), \text{ for } (i,j) \in [m_l] \times [m_{l-1}] \tag{5}$$

where $\beta_1, \cdots, \beta_L > 0$ are the per-layer variance for Gaussian initialization. By choosing different variance, we recover many common initialization schemes in the literature, as summarized in Table 1.

2.1 Our objective and methodology

We aim to understand the relation between privacy, utility and over-parameterization (depth and width) for the Langevin diffusion algorithm (under different initializaiton distributions). To understand privacy, we prove a KL divergence upper bound for running Langevin diffusion on any two *worst-case* neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ of size n that only differ in one record, denoted as $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ and $(\boldsymbol{x}, \boldsymbol{y}') \in \mathcal{D}'$. For brevity, in later sections, we denote $\boldsymbol{W}_t$ (with distribution p_t) and $\boldsymbol{W}_t'$ (with distribution p_t') as the trained model parameters after running Langevin diffusion (4) for time T on $\mathcal{D}$ and $\mathcal{D}'$ respectively. We make the following assumptions for privacy analysis in this paper.

Assumption 2.1 (Bounded Data)**.** We assume that all $\boldsymbol{x}$ in the data domain is bounded s.t. $\|\boldsymbol{x}\|_2 \leq 1$.

To understand utility (under a given KL privacy budget), we aim to prove upper bounds for excess empirical risk and excess population risk given an arbitrarily fixed KL divergence privacy budget ε. Finally, we also investigate how trade-off between our KL privacy bound and risk bounds is affected by the network width and depth. For utility analysis, we additionally make the following fair and attainable assumptions on data and network regularity.

Assumption 2.2 (Data and network regularity [33, Assumption 2.1])**.** For any training data $\boldsymbol{x}_i \in \mathcal{D}$, it satisfies that $\|\boldsymbol{x}_i\|_2 = 1$. Moreover, $\boldsymbol{x}_i \in \mathcal{D}$ are i.i.d. samples from a data distribution P_x that satisfies $\int \|\boldsymbol{x}\|_2^2 dP_x(\boldsymbol{x}) = 1$. We also assume that the network only has single output.

Note that Assumption 2.2 is only required for utility analysis, and is not need for our privacy bound.

3 KL Privacy for Training Fully Connected ReLU Neural Networks

In this section, we perform the composition-based privacy analysis in KL divergence for Langevin Diffusion on deep ReLU neural networks, under Gaussian initialization distribution specified by Eq. (5). More specifically, we prove upper bound for the KL divergence between distribution of output model parameters when running Langevin diffusion on an arbitrary pair of neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$.

Our first key insight is that by the joint convexity of KL divergence, it is possible to prove composition-based KL privacy bound under more relaxed condition regarding the sensitivity of gradient computation (i.e., *without* gradient clipping).

Theorem 3.1 (KL composition under possibly unbounded gradient difference)**.** *The KL divergence between running Langevin diffusion* (4) *for DNN* (2) *on neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ satisfies*

$$KL(\boldsymbol{W}_T, \boldsymbol{W}'_T) \leq \frac{1}{2\sigma^2} \int_0^T \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{W}_t; \mathcal{D}) - \nabla \mathcal{L}(\boldsymbol{W}_t; \mathcal{D}')\|_2^2\right] dt\,. \tag{6}$$

Proof sketch. We compute the partial derivative of KL divergence with regard to time t, and then compute integral to bound the KL divergence. During computing the limit of differentiation, we upper bound KL divergence at time $t+\eta$ for small enough step-size with the divergence on path $[t, t+\eta]$. Then we use Girsanov's theorem to compute the KL divergence between the path of coupled Langevin diffusion processes. The complete proof is in Appendix B.1.

Theorem 3.1 is an extension of the standard additivity [45] of KL divergence (also known as chain rule [40]) for a finite sequence of distributions to continuous time processes with (possibly) unbounded drift difference. The key extension is that Theorem 3.1 does not require bounded sensitivity between Langevin Diffusion on neighboring datasets. Instead, it only requires finite second-order moment of drift difference (in the ℓ_2-norm sense) between neighboring datasets $\mathcal{D}, \mathcal{D}'$. By using this extended KL composition Theorem 3.1, we prove KL privacy bound for running Langevin diffusion algorithm (without gradient clipping) on deep neural networks, by tracking the upper bound for ℓ_2 norm of the gradient difference throughout training (under mild assumptions) as follows.

Lemma 3.2 (Drift Difference in Noisy Training)**.** *Let M_T be the subspace spanned by gradients $\{\nabla \ell(\boldsymbol{f}_{\boldsymbol{W}_t}(\boldsymbol{x}_i; \boldsymbol{y}_i) : (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}, t \in [0, T]\}$ on each training data record throughout Langevin diffusion $(\boldsymbol{W}_t)_{t\in[0,T]}$. Denote $\|\cdot\|_{M_T}$ as the ℓ_2 norm of the projection of the input vector onto linear space M_T. Suppose that $\exists c, \beta > 0$ s.t. for any $\boldsymbol{W}, \boldsymbol{W}'$ and $\boldsymbol{x}, \boldsymbol{y}$ we have $\|\nabla \ell(f_{\boldsymbol{W}}(\boldsymbol{x}); \boldsymbol{y})) - \nabla \ell(f_{\boldsymbol{W}'}(\boldsymbol{x}); \boldsymbol{y})\|_2 < \max\{c, \beta \|\boldsymbol{W} - \boldsymbol{W}'\|_{M_T}\}$. Then over the randomness of the Brownian motion $\boldsymbol{B}_t$ and initialization distribution* (5) *in Langevin diffusion $(\boldsymbol{W}_t)_{t\in[0,T]}$, it satisfies that*

$$\int_0^T \mathbb{E}_{p_t}\left[\|\nabla \mathcal{L}(\boldsymbol{W}; \mathcal{D}) - \nabla \mathcal{L}(\boldsymbol{W}; \mathcal{D}')\|_2^2\right] dt \leq 2 \cdot T \cdot \underbrace{\mathbb{E}_{p_0}\left[\|\nabla \mathcal{L}(\boldsymbol{W}; \mathcal{D}) - \nabla \mathcal{L}(\boldsymbol{W}; \mathcal{D}')\|_2^2\right]}_{\textit{gradient difference at initialization}}$$

$$+ 2\left(\frac{e^{(2+\beta^2)T} - (2+\beta^2)T}{2+\beta^2}\right) \cdot \underbrace{\left(\mathbb{E}_{p_0}\left[\|\nabla \mathcal{L}(\boldsymbol{W}; \mathcal{D})\|_2^2\right] + \sigma^2 \textit{rank}(M_T) + c^2\right)}_{\textit{gradient difference fluctuation during training}} + \underbrace{2c^2 \cdot T}_{\textit{non-smoothness cost}}\,. \tag{7}$$

Remark 3.3. The assumption $\|\nabla \ell(f_{\boldsymbol{W}}(\boldsymbol{x}); \boldsymbol{y})) - \nabla \ell(f_{\boldsymbol{W}'}(\boldsymbol{x}); \boldsymbol{y})\|_2 < \max\{c, \beta \|\boldsymbol{W} - \boldsymbol{W}'\|_{M_T}\}$ is similar to smoothness condition for the loss function, but is more relaxed as it allows non-smoothness at places where the gradient is bounded. Therefore, the assumption holds under ReLU activation.

Remark 3.4 (Gradient difference at initialization). The first term and in our upper bound linearly scales with the difference between gradients on neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ at initialization. Under different initializations, this gradient difference exhibits different dependency on the network depth and width, as we will prove theoretically in Lemma 4.1.

Remark 3.5 (Gradient difference fluctuation during training). The second term in our upper bound is to bound the change of gradient difference norm during training, and is therefore proportional to the the rank of a subspace M_T spanned by gradients of all training data. Intuitively, this fluctuation is because Langevin diffusion adds per-dimensional noise with variance σ^2, thus perturbing the training parameters away from the initialization at a rate of $O(\sigma\sqrt{\text{rank}(M_T)})$ in expected ℓ_2 distance.

Growth of KL privacy bound with regard to training time T. The first term in the gradient difference bound Lemma 3.2 grows linearly with the training time T, while the second term grows exponentially with regard to T. Consequently, for learning tasks that requires a long training time to converge, the second term will become the dominating term and the KL privacy bound suffers from exponential growth with regard to the training time. Nevertheless, if the total amount of required training time (for convergence) is small enough e.g. $T \leq \frac{1}{2(2+\beta^2)}$, then we have that $\frac{e^{(2+\beta^2)T}-(2+\beta^2)T}{2+\beta^2} < T$ and therefore the second term in the gradient difference upper bound accumulates at a lower than linear rate with increasing training time.

Dependence of KL privacy bound on network overparameterization. Under a fixed training time T and noise scale σ^2, Lemma 3.2 predicts that the KL divergence upper bound in Theorem 3.1 is dependent on the gradient difference and gradient norm at initialization, and the rank of gradient subspace $\text{rank}(M_T)$ throughout training. We now discuss the how these two terms change under increasing width and depth, and whether there are possibilities to improve them under overparameterization.

1. The gradient norm at initialization crucially depend on how the per-layer variance in the Gaussian initialization distribution scales with the network width and depth. Therefore, it is possible to improve the KL privacy bound by using initialization distributions that enable smaller gradient difference at initialization, as we will theoretically show in Section 4.
2. Regarding the rank of gradient subspace $\text{rank}(M_T)$: when the gradients along the training trajectory span the whole optimization space, $\text{rank}(M)$ would equal the dimension of the learning problem. Consequently, the gradient fluctuation upper bound (and thus the KL privacy bound) worsens with increasing number of model parameters (overparameterization) in the worst-case. However, if the gradients are low-dimensional [39, 26, 37] or sparse [30], it is possible that $\text{rank}(M_T)$ will be dimension-independent and thus enable better bound for gradient fluctuation (and KL privacy bound). We leave this as an interesting open problem.

4 KL privacy bound for Linearized Network under overparameterization

In this section, we restrict ourselves to the training of linearized networks as described in (3), and investigate the interplay between KL privacy and overparameterization (increasing width and depth). The analysis of DNN via linearization is a commonly used technique in both theory and practice. In theory, DNN can work in the lazy training regime [16] (also called linear regime), under which the linearized network well approximates the training dynamics for deep neural entwork [28] and has been well studied by NTK. In practice, linearized network can still achieve decent performance, which provides a good justification of linearized networks. [37, 34]. We hope our analysis for linearized network serve as an initial attempt that would open a door to theoretically understanding the relationship between overparameterization and privacy.

To derive a composition-based KL privacy bound for training linearized network, we apply Theorem 3.1 which requires an upper bound for the norm of gradient difference between the training processes on neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ at any time t. Note that the empirical risk function for training linearized models enjoys convexity, and therefore requires a relatively short amount of training time for convergence. Therefore intuitively, the gradient difference between neighboring datasets does not change a lot during training, thus allowing us to prove tighter upper bound for the gradient difference norm for linearized networks (than Lemma 3.2).

In the following lemma, we prove that for linearized network, the gradient difference throughout training has a uniform upper bound that only depends on the network width, depth and initialization.

Lemma 4.1 (Gradient Difference throughout training linearized network)**.** *Under Assumption 2.1, taking over the randomness of the random initialization and the Brownian motion in Langevin*

diffusion, for any $t \in [0, T]$, it satisfies that

$$\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \mathcal{L}(\boldsymbol{W}_t;\mathcal{D}')\|^2\right] \leq \frac{4B}{n^2}\,, \tag{8}$$

where n is the training dataset size, and B is a constant that only depends on the network width, depth and initialization distribution as follows.

$$B := o\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right)\sum_{l=1}^{L}\frac{\beta_L}{\beta_l}\,, \tag{9}$$

where o is the number of output classes, $\{m_i\}_{i=1}^L$ are the per-layer network widths, and $\{\beta_i\}_{i=1}^L$ are the variances of Gaussian initialization at each layer.

Lemma 4.1 provides an precise analytical upper bound for the gradient difference during training linearized network, by tracking the gradient distribution under fully connected feed-forward ReLU network with Gaussian weight matrices. The full proof is in Appendix C.1 and is heavily based on similar techniques for computing the gradient distribution in the NTK literature [2, 47]. By plugging Eq. (8) into Theorem 3.1, we have the following KL privacy bound for training linearized network.

Corollary 4.2 (KL privacy bound for training linearized network)**.** *Under Assumption 2.1 and neural networks* (3) *initialized by Gaussian distribution with per-layer variance $\{\beta_i\}_{i=1}^L$, running Langevin diffusion for linearized network with time T on neighboring datasets satisfies that*

$$KL(\boldsymbol{W}_t^{lin}\|\boldsymbol{W'}_T^{lin}) \leq \frac{2BT}{n^2\sigma^2}\,. \tag{10}$$

where B is the constant that specifies the gradient norm upper bound, given by (9)*.*

Overparameterization affects privacy differently under different initialization. Corollary 4.2 and Lemma 4.1 suggest that the effect of network overparameterization on KL privacy bound crucially relies on how the per-layer Gaussian initialization variance β_i is scaled with the per-layer network width m_i and depth L. We summarize our KL privacy bound for linearized network under different width, depth and initialization schemes in Table 1, and elaborate the comparison below.

(1) LeCun initialization uses small, width-independent variance for initializing the first layer $\beta_1 = \frac{1}{d}$ (where d is the number of input features), and width-dependent variance $\beta_2 = \cdots = \beta_L = \frac{1}{m}$ for initializing all the subsequent layers. Therefore, the second term $\sum_{l=1}^{L}\frac{\beta_L}{\beta_l}$ in the constant B (9) increases linearly with the width m and depth L. However, due to $\frac{m_l\cdot\beta_l}{2} < 1$ for all $l = 2, \cdots, L$, the first product term $\prod_{l=1}^{L-1}\frac{\beta_l m_l}{2}$ in constant B decays with the increasing depth. Therefore, by combining the two terms, we prove that the KL privacy bound worsens with increasing width, but improves with increasing depth (as long as the depth is large enough). Similarly, under Xavier initialization $\beta_l = \frac{2}{m_{l-1}+m_l}$, we prove that the KL privacy bound (especially the constant B (9)) improves with increasing depth as long as the depth is large enough.

(2) NTK and He initializations user large per-layer variance $\beta_l = \begin{cases}\frac{2}{m_l} & l = 1, \cdots, L-1\\ \frac{1}{o} & l = L\end{cases}$ (for NTK) and $\beta_l = \frac{2}{m_{l-1}}$ (for He). Consequently, the gradient difference under NTK or He initialization is significantly larger than that under LeCun initialization. Specifically, the gradient norm constant B (9) grows linearly with the width m and the depth L under He and NTK initializations, thus suggesting a worsening of KL privacy bound under increasing width and depth.

5 Utility guarantees for Training Linearized Network

Our privacy analysis suggest that training linearized network under certain initialization schemes (such as LeCun initialization) enable significantly better privacy bounds under overparameterization by increasing depth. In this section, we further prove utility bounds for Langevin diffusion under initialization schemes, and investigate the effect of overparameterization on the privacy utility trade-off. In other words, we aim to understand whether there are any utility degradation for training linearized network when using the more privacy-preserving initialization schemes.

Convergence of training linearized network. We now prove convergence of excess empirical risk in training linearized network via Langevin diffusion. This is well-studied problem in the literature for noisy gradient descent. We extend the convergence theorem to continuous-time Langevin diffusion below, and investigate factors that affect the convergence under overparameterization.

Proposition 5.1 (Extension of [36, Theorem 2] and [39, Theorem 3.1])**.** *Let $\boldsymbol{W}_0^{lin}$ be a randomly initialized parameter vector by* (5)*. Let the empirical NTK feature mapping matrix for dataset training $\mathcal{D}$ at initialization be $M_0 = \begin{pmatrix} \nabla \boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_1) & \cdots & \nabla \boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_n) \end{pmatrix}$. Let $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D})$ be the empirical loss for linearized network* (3) *expanded at initialization vector $\boldsymbol{W}_0^{lin}$. Then running Langevin diffusion* (4) *under empirical loss $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D})$ and initialization $\boldsymbol{W}_0^{lin}$ for time T satisfies the following excess empirical risk bound*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})] - \mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^*; \mathcal{D})] \leq \frac{2R}{T} + \frac{1}{2}\sigma^2 \mathbb{E}[rank(M_0)] \left(1 + \log \frac{2BT^2}{R}\right) \tag{11}$$

where $\boldsymbol{W}_0^*$ is an (exact or approximate) solution for the ERM problem on $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D})$, and $R = \mathbb{E}[\|\boldsymbol{W}_0^{lin} - \boldsymbol{W}_0^*\|_{M_0}^2]$ is the expected gap between initialization parameters $\boldsymbol{W}_0$ and solution $\boldsymbol{W}_0^*$.

Remark 5.2. The excess empirical risk bound Proposition 5.1 is smaller if data is low-rank, e.g., image data, then $rank(M_0)$ is small. This is consistent with the prior dimension-independent private learning literature [26, 27, 30] and show benefit of low-dimensional gradients on private learning.

Proposition 5.1 highlights that the excess empirical risk scales with the expected gap R between initialization and optima (which we refer as the lazy training distance), the rank of the gradient subspace rank rank(M_0), and the constant B that specifies upper bound for expected gradient norm during training. Specifically, the smaller is the lazy training distance R, the better is the excess risk bound for Langevin diffusion given fixed training time T and noise variance σ^2. We have discussed how overparameterization affects the the gradient norm constant B and the gradient subspace rank rank(M_0) in Section 3. Therefore, we only still need to investigate how the lazy training distance R changes with the network width, depth and initialization, as follows.

Lazy training distance R decreases with increasing depth. It is widely observed in the literature [16, 48, 31] that under appropriate choices of initializations, gradient descent on fully connected neural network falls under a lazy training regime. That is, with high probability, there exists a (nearly) optimal solution for the ERM problem that is close to the initialization parameters in terms of l_2 distance. Moreover, this lazy training distance R is closely related to the smallest eigenvalue of the NTK matrix. In the following proposition, we compute a near optimal solution via the pseudo inverse of the NTK matrix, and prove that it has small distance to the initialization parameters via existing lower bounds for the smallest eigenvalue of the NTK matrix [33].

Proposition 5.3 (Bounding lazy training distance via smallest eigenvalue of the NTK matrix)**.** *Under the data and network regularity Assumption 2.1, if the width $m_1 = \cdots = m_{L-1} = \Omega(n)$ is sufficiently large, then there exists an optimal solution $\boldsymbol{W}_0^{\frac{1}{n^2}}$ that satisfies $\mathcal{L}_0^{lin}(\boldsymbol{W}_0^{\frac{1}{n^2}}) \leq \frac{1}{n^2}$ and satisfies*

$$\tilde{R} = \mathbb{E}[\|\boldsymbol{W}_0^{\frac{1}{n}} - \boldsymbol{W}_0\|_2^2] \leq \begin{cases} \tilde{O}(\frac{n}{d \cdot 2^L \cdot (m(L-2)+1)}) & \textit{for NTK initialization} \\ \tilde{O}(\frac{n}{2^L m(L-1)}) & \textit{for He initialization} \\ \tilde{O}(\frac{n}{m(L-1)}) & \textit{for LeCun initialization} \end{cases} \tag{12}$$

We refer to $\tilde{R}$ as the approximate lazy training distance because $\boldsymbol{W}_0^{\frac{1}{n^2}}$ is only a nearly optimal solution for the ERM problem. Proposition 5.3 shows that this approximate lazy training distance improves with overparameterization (width and depth) under LeCun, He and NTK initializations.

Privacy Excess empirical risk Tradeoff for training linearized network via Langevin diffusion. We now use the approximate lazy training distance $\tilde{R}$ to prove empirical risk bound, and combine it with our KL privacy bound Section 4 to show the privacy utility trade-off under overparameterization.

Corollary 5.4 (Privacy utility trade-off for last iterate)**.** *Assume that the data and network regularity Assumption 2.2 holds. Assume that all the conditions and definition for constants in Proposition 5.1*

holds. Then by setting $\sigma^2 = \frac{2BT}{\varepsilon n^2}$ *and* $T = \sqrt{\frac{2\varepsilon n\tilde{R}}{B}}$*, we have that running Langevin diffusion for time* T *satisfies bound KL divergence* ϵ*, and has ecess empirical risk upper bounded by*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})] \leq \mathcal{O}\left(\frac{1}{n^2} + \sqrt{\frac{B\tilde{R}}{\varepsilon n}}\log(\varepsilon n)\right) \tag{13}$$

where B *is the gradient norm constant Eq.* (9)*, and* $\tilde{R}$ *is the approximate lazy training distance in Eq.* (12)*. A summary of* B *and* $\tilde{R}$ *under different initializations is in Table 1.*

Corollary 5.4 suggests that the excess empirical risk worsens in the presence of a stronger privacy constraint, i.e., under a small privacy budget ε, thus contributing to a trade-off between privacy and utility. However, the excess empirical risk also scales with constants such as the approximate lazy training distance $\tilde{R}$ and the gradient norm constant B. These constants depend on network width, depth and initialization distributions, and therefore we prove privacy utility trade-offs for training linearized network that changes with overparameterization, as sumarized in Table 1.

We would like to highlight that our privacy utility trade-off bound under LeCun and He initialization strictly improves with increasing width and depth as long as the data and network satisfy regularity Assumption 2.2 and the network width is large enough. To our best knowledge, this is the first time that a strictly improving privacy utility trade-off under overparameterization is shown in literature. This shows the benefits of our precise KL privacy analysis under overparameterization.

Extension of privacy utility results to excess population risk. Our privacy utility trade-off results can be generalized to to excess population risk, by additionally bounding the generalization error with standard stability-based arguments [35, 24, 8]. We elaborate the details below.

Proposition 5.5 (Extended excess population risk bound for training linearized network)**.** *Denote* $R_0(\boldsymbol{W}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in pop}[\ell(f_{\boldsymbol{W}_0}(\boldsymbol{x}) + \frac{\partial \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x})}{\partial \boldsymbol{W}_0}(\boldsymbol{W} - \boldsymbol{W}_0); \boldsymbol{y})]$ *as the population risk of linearized network expanded at initialization vector* $\boldsymbol{W}_0$ *over population data distribution pop. Then under the conditions of Corollary 5.4, we have that for any dataset* $\mathcal{D}$ *of size* n*, the following excess population risk upper bound holds.*

$$\mathbb{E}[R_0(\boldsymbol{W}_T)] - \mathbb{E}[\mathcal{L}(\boldsymbol{W}_{pop,0}^*; \mathcal{D})] \leq O\left(\frac{1}{n^2} + \sqrt{\frac{B\tilde{R}}{\varepsilon n}}(\log(\varepsilon n) + \varepsilon)\right) \tag{14}$$

where the expectation is over the randomness of sampling the training dataset $\mathcal{D} \sim pop^n$ *from the data population and the random coins for the Langevin diffusion training algorithm, and* $\boldsymbol{W}_{pop,0}^* = \operatorname{argmin}_{\boldsymbol{W}} R_0(\boldsymbol{W})$ *is the optimal solution for the population risk minimization problem.*

Proposition 5.5 shows that the excess population risk bound is almost the same as excess empirical risk, except that there is an additional generalization term that scales with the privacy budget ε. Intuitively, this is because the generalization error is proportional to the stability of model prediction function under different training dataset, which is smaller when the KL privacy loss ε is small.

6 Conclusion

We prove new KL privacy bound for training fully connected ReLu network (and its linearized variant) using the Langevin diffusion algorithm, and investigate how privacy is affected by the network width, depth and initialization. Our results suggest that there is a complex interplay between privacy and overparameterization (width and depth) that crucially relies on what initialization distribution is used and the how much the gradient fluctuates during training. To this end, we show that for training a linearized variant of fully connected network with finite time, it is possible to prove a KL privacy bound that improves with depth, as long as the initialization distribution is set appropriately (such as LeCun). We also study the excess empirical and population risk bounds for linearized network, and prove that the privacy-utility trade-off similarly improves as depth increases under LeCun initialization. This shows the gain of our new privacy analysis for capturing the effect of overparameterization. We leave it as an important open problem as to whether our privacy utility trade-off results for linearized network could be generalized to deep neural networks.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[3] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

[4] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in l1 geometry. In *International Conference on Machine Learning*, pages 393–403. PMLR, 2021.

[5] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1138–1156. IEEE, 2022.

[6] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[7] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

[8] Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33:4381–4391, 2020.

[9] Raef Bassily, Cristóbal Guzmán, and Michael Menart. Differentially private stochastic optimization: New results in convex and non-convex settings. *Advances in Neural Information Processing Systems*, 34:9317–9329, 2021.

[10] Raef Bassily, Mehryar Mohri, and Ananda Theertha Suresh. Differentially private learning with margin guarantees. *arXiv preprint arXiv:2204.10376*, 2022.

[11] Zhiqi Bu, Hua Wang, and Qi Long. On the convergence and calibration of deep learning with differential privacy. *arXiv preprint arXiv:2106.07830*, 2021.

[12] Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *Journal of the ACM*, 70(2):1–18, 2023.

[13] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[14] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.

[15] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

[16] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

[17] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

[18] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[19] Yoav Freund, Yi-An Ma, and Tong Zhang. When is the convergence time of langevin algorithms dimension independent? a composite optimization viewpoint. *Journal of Machine Learning Research*, 23(214):1–32, 2022.

[20] Arun Ganesh and Kunal Talwar. Faster differentially private samplers via rényi divergence analysis of discretized langevin mcmc. *Advances in Neural Information Processing Systems*, 33:7222–7233, 2020.

[21] Arun Ganesh, Abhradeep Thakurta, and Jalaj Upadhyay. Langevin diffusion: An almost universal algorithm for private euclidean (convex) optimization. *arXiv preprint arXiv:2204.01585*, 2022.

[22] Chuan Guo, Alexandre Sablayrolles, and Maziar Sanjabi. Analyzing privacy leakage in machine learning via multiple hypothesis testing: A lesson from fano. *arXiv preprint arXiv:2210.13662*, 2022.

[23] Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. *arXiv preprint arXiv:2206.07758*, 2022.

[24] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

[25] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[26] Peter Kairouz, Monica Ribero Diaz, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private erm with adagrad rates via publicly estimated subspaces. In *Conference on Learning Theory*, pages 2717–2746. PMLR, 2021.

[27] Shiva Prasad Kasiviswanathan. Sgd with low-dimensional gradients with applications to private and distributed learning. In *Uncertainty in Artificial Intelligence*, pages 1905–1915. PMLR, 2021.

[28] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

[29] DS Lemons, A Gythiel, and Paul Langevin's. Sur la théorie du mouvement brownien [on the theory of brownian motion]. *CR Acad. Sci.(Paris)*, 146:530–533, 1908.

[30] Xuechen Li, Daogao Liu, Tatsunori B Hashimoto, Huseyin A Inan, Janardhan Kulkarni, Yin-Tat Lee, and Abhradeep Guha Thakurta. When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems*, 35:28616–28630, 2022.

[31] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *The Journal of Machine Learning Research*, 22(1): 3327–3373, 2021.

[32] Saeed Mahloujifar, Alexandre Sablayrolles, Graham Cormode, and Somesh Jha. Optimal membership inference bounds for adaptive composition of sampled gaussian mechanisms. *arXiv preprint arXiv:2204.06106*, 2022.

[33] Quynh Nguyen, Marco Mondelli, and Guido F Montufar. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. In *International Conference on Machine Learning*, pages 8119–8129. PMLR, 2021.

[34] Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems*, 34:8998–9010, 2021.

[35] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, volume 2, page 5, 2009.

[36] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79. PMLR, 2013.

[37] Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Jonathan Ragan-Kelley, Ludwig Schmidt, and Benjamin Recht. Neural kernels without tangents. In *International Conference on Machine Learning*, pages 8614–8623. PMLR, 2020.

[38] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[39] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics*, pages 2638–2646. PMLR, 2021.

[40] Thomas Steinke. Adaptive data analysis. 2016.

[41] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *arXiv preprint arXiv:1411.5417*, 2014.

[42] Jasper Tan, Blake Mason, Hamid Javadi, and Richard Baraniuk. Parameters or privacy: A provable tradeoff between overparameterization and membership inference. *Advances in Neural Information Processing Systems*, 35:17488–17500, 2022.

[43] Jasper Tan, Daniel LeJeune, Blake Mason, Hamid Javadi, and Richard G Baraniuk. A blessing of dimensionality in membership inference through regularization. In *International Conference on Artificial Intelligence and Statistics*, pages 10968–10993. PMLR, 2023.

[44] Nilesh Tripuraneni, Ben Adlam, and Jeffrey Pennington. Covariate shift in high-dimensional random feature regression. *arXiv preprint arXiv:2111.08234*, 2021.

[45] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[46] Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *Advances in neural information processing systems*, 32, 2019.

[47] Zhenyu Zhu, Fanghui Liu, Grigorios G Chrysos, and Volkan Cevher. Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization). *arXiv preprint arXiv:2209.07263*, 2022.

[48] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109:467–492, 2020.

Contents

A Symbols and definitions

Vecorization $\mathrm{Vec}(\cdot)$ denotes the transformation that takes an input matrix $\boldsymbol{A} = (a_{ij})_{i\in[r],j\in[c]} \in \mathbb{R}^{r\times c}$ (with r rows and c columns) and outputs a rc-dimensional column vector: $\mathrm{Vec}(\boldsymbol{A}) = (a_{1,1},\cdots,a_{r,1},a_{1,2},\cdots,a_{r,2},\cdots,a_{1,c},\cdots,a_{r,c})^\top$.

Distribution p_t and p'_t: we denote p_t as the distribution of model parameters after running Langevin diffusion on dataset $\mathcal{D}$ with time t, and similarly denote p'_t as the distribution of model parameters after running Langevin diffusion on dataset $\mathcal{D}'$ with time t.

Softmax function: $\mathrm{softmax}(\boldsymbol{y}) = \frac{e^{\boldsymbol{y}[j]}}{\sum_{j=1}^{o} e^{\boldsymbol{y}[j]}}$ where o is the number of output classes.

Neighboring datasets D and D': two dataset with the same number of data records that differ in one record. We also denote the differing records as $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}$ and $(\boldsymbol{x}',\boldsymbol{y}') \in \mathcal{D}'$.

o: number of output classes for the neural network.

B Deferred proofs for Section 3

B.1 Deferred proofs for Theorem 3.1

To prove the new composition theorem, we will use the Girsanov's Theorem. Here we follow the presentation of [15, Theorem 6].

Theorem B.1 (Implication of Girsanov's theorem [15, Theorem 6])**.** *Let $(\tilde{X}_t)_{t\in[0,\eta]}$ and $(X_t)_{t\in[0,\eta]}$ be two continuous-time processes over $\mathbb{R}^r$. Let P_T be the probability measure that corresponds to the trajectory of $(\tilde{X}_t)_{t\in[0,\eta]}$, and let Q_T be the probability measure that corresponds to the trajectory of $(X_t)_{t\in[0,\eta]}$. Suppose that the process $(\tilde{X}_t)_{t\in[0,\eta]}$ follows*

$$d\tilde{X}_t = \tilde{b}_t dt + \sigma_t d\tilde{B}_t,$$

where $\tilde{B}$ is a Brownian motion, and the process $(X_t)_{t\in[0,\eta]}$ follows

$$dX_t = b_t dt + \sigma_t dB_t,$$

where B is a Brownian motion. We assume that for each $t > 0$, σ_t is a $r \times r$ symmetric positive definite matrix. Then, provided that Novikov's condition holds,

$$\mathbb{E}_{Q_T} \exp\left(\frac{1}{2}\int_0^\eta \|\sigma_t^{-1}(\tilde{b}_t - b_t)\|_2^2 dt\right) < \infty\,, \tag{15}$$

we have that

$$\frac{dP_T}{dQ_T} = exp\left(\int_0^\eta \sigma_t^{-1}(\tilde{b}_t - b_t)dB_t - \frac{1}{2}\int_0^\eta \|\sigma_t^{-1}(\tilde{b}_t - b_t)\|_2^2 dt\right).$$

Now we apply Girsanov's theorem on the coupled Langevin diffusion processes on neighboring datasets, and obtain the following new composition theorem for KL divergence in the context of privacy.

Theorem 3.1 (KL composition under possibly unbounded gradient difference)**.** The KL divergence between running Langevin diffusion (4) for DNN (2) on neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ satisfies

$$KL(\boldsymbol{W}_T, \boldsymbol{W}'_T) \le \frac{1}{2\sigma^2}\int_0^T \mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}')\|_2^2\right] dt\,. \tag{16}$$

Proof. Denote p_t as the distribution of model parameters after running Langevin diffusion on dataset $\mathcal{D}$ with time t, and similarly denote p'_t as the distribution of model parameters after running Langevin diffusion on dataset $\mathcal{D}'$ with time t. Then by definition,

$$\begin{aligned}\frac{\partial KL(p_t, p'_t)}{\partial t} &= \lim_{\eta\to 0}\frac{KL(p_{t+\eta}, p'_{t+\eta}) - KL(p_t, p'_t)}{\eta}\\ &\le \lim_{\eta\to 0}\frac{KL(p_{t,t+\eta}, p'_{t,t+\eta}) - KL(p_t, p'_t)}{\eta},\end{aligned} \tag{17}$$

where the last inequality is by the data processing inequality for KL divergence [45, Theorem 9] (with the data processing operation given by $(\boldsymbol{W}_t, \boldsymbol{W}_{t+\eta}) \to \boldsymbol{W}_t$). Now we compute the term $KL(p_{t:t+\eta}, p'_{t:t+\eta})$ as follows.

$$\begin{aligned}KL(p_{t,t+\eta}, p'_{t,t+\eta}) &= \mathbb{E}_{p_{t,t+\eta}(\boldsymbol{W}_t, \boldsymbol{W}_{t+\eta})}\left[\log\left(\frac{p_{t+\eta|t}(\boldsymbol{W}_{t+\eta}|\boldsymbol{W}_t)p_t(\boldsymbol{W}_t)}{p'_{t+\eta|t}(\boldsymbol{W}_{t+\eta}|\boldsymbol{W}_t)p'_t(\boldsymbol{W}_t)}\right)\right]\\ &= \mathbb{E}_{p_{t,t+\eta}(\boldsymbol{W}_t, \boldsymbol{W}_{t+\eta})}\left[\log\left(\frac{p_{t+\eta|t}(\boldsymbol{W}_{t+\eta}|\boldsymbol{W}_t)}{p'_{t+\eta|t}(\boldsymbol{W}_{t+\eta}|\boldsymbol{W}_t)}\right)\right] + \mathbb{E}_{p_t(\boldsymbol{W}_t)}\left[\log\left(\frac{p_t(\boldsymbol{W}_t)}{p'_t(\boldsymbol{W}_t)}\right)\right]\\ &= \mathbb{E}_{p_t(\boldsymbol{W}_t)}\left[KL(p_{t+\eta|t}, p'_{t+\eta|t})\right] + KL(p_t, p'_t)\end{aligned} \tag{18}$$

Now we want to apply Girsanov's theorem to the following langevin diffusion processes $(\boldsymbol{W}_{t+s|t})_{s\in[0,\eta]}$ and $(\boldsymbol{W}'_{t+s|t})_{s\in[0,\eta]}$.

$$d\boldsymbol{W}_{t+s|t} = -\nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D})dt + \sqrt{2\sigma^2}dB_s$$
$$d\boldsymbol{W}'_{t+s|t} = -\nabla\mathcal{L}(\boldsymbol{W}'_{t+s};\mathcal{D}')dt + \sqrt{2\sigma^2}dB_s$$

where we have the boundary condition that $\boldsymbol{W}_{t|t} = \boldsymbol{W}'_{t|t}$ due to the conditioning at time t. Note that when η is small enough, we have that the Novikov's condition in Eq. (15) holds because the exponent inside integration $\frac{1}{2}\int_0^\eta \|\sigma_t^{-1}(\tilde{b}_t - b_t)\|_2^2 dt$ scales linearly with η and is small when η is small enough. Therefore, by applying Girsanov's theorem, we have that

$$\begin{aligned}KL(p_{t+\eta|t}, p'_{t+\eta|t}) \leq& KL(p_{t:t+\eta|t}, p'_{t:t+\eta|t})\\ =&\mathbb{E}_{p_{t:t+\eta|t}}\left[\int_0^\eta \sigma^{-1}(\tilde{b}_s - b_s)dB_s - \frac{1}{2}\int_0^T \|\sigma^{-1}(\tilde{b}_s - b_s)\|_2^2 ds\right]\end{aligned}$$

where $\tilde{b}_s - b_s = -\nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}) + \nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}')$. By Itô integration with regard to standard Brownian motion, we have that

$$KL(p_{t+\eta|t}, p'_{t+\eta|t}) \leq \frac{1}{2\sigma^2}\mathbb{E}_{p_{t:t+\eta|t}}\left[\int_0^\eta \|\nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}')\|_2^2\, ds\right] \tag{19}$$

By plugging Eq. (19) into Eq. (18), we have that

$$KL(p_{t,t+\eta}, p'_{t,t+\eta}) \leq \frac{1}{2\sigma^2}\mathbb{E}_{p_{t:t+\eta}}\left[\int_0^\eta \|\nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}')\|_2^2\, ds\right] + KL(p_t, p'_t) \tag{20}$$

By plugging Eq. (20) into Eq. (17), and by exchanging the order of expectation and integration, we have that

$$\begin{aligned}\frac{\partial KL(p_t, p'_t)}{\partial t} \leq& \frac{1}{2\sigma^2}\lim_{\eta\to 0}\frac{\int_0^\eta \mathbb{E}_{p_{t+s}}\left[\|\nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_{t+s};\mathcal{D}')\|_2^2\right] ds}{\eta}\\ =& \frac{1}{2\sigma^2}\mathbb{E}_{p_t}\left[\|\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}')\|_2^2\right]\end{aligned} \tag{21}$$

Integrating Eq. (21) on $t \in [0, T]$ finishes the proof. □

B.2 Deferred proofs for Lemma 3.2

Lemma 3.2. Let M_T be the subspace spanned by gradients $\{\nabla\ell(\boldsymbol{f}_{\boldsymbol{W}_t}(\boldsymbol{x}_i;\boldsymbol{y}_i) : (\boldsymbol{x}_i,\boldsymbol{y}_i) \in \mathcal{D}, t \in [0,T]\}$ on each training data record throughout Langevin diffusion $(\boldsymbol{W}_t)_{t\in[0,T]}$. Denote $\|\cdot\|_{M_T}$ as the ℓ_2 norm of the projection of the input vector onto linear space M_T. Suppose that $\exists c, \beta > 0$ s.t. for any $\boldsymbol{W}, \boldsymbol{W}'$ and $\boldsymbol{x}, \boldsymbol{y}$ we have $\|\nabla\ell(f_{\boldsymbol{W}}(\boldsymbol{x});\boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}'}(\boldsymbol{x});\boldsymbol{y})\|_2 < \max\{c, \beta\|\boldsymbol{W} - \boldsymbol{W}'\|_{M_T}\}$. Then over the randomness of the Brownian motion $\boldsymbol{B}_t$ and initialization distribution (5) in Langevin diffusion $(\boldsymbol{W}_t)_{t\in[0,T]}$, it satisfies that

$$\begin{aligned}\int_0^T \mathbb{E}_{p_t}\left[\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}')\|_2^2\right] dt \leq 2\cdot T\cdot \underbrace{\mathbb{E}_{p_0}\left[\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}')\|_2^2\right]}_{\text{gradient difference at initialization}}\\ + 2\left(\frac{e^{(2+\beta^2)T} - (2+\beta^2)T}{2+\beta^2}\right)\cdot \underbrace{\left(\mathbb{E}_{p_0}\left[\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D})\|_2^2\right] + \sigma^2\text{rank}(M_T) + c^2\right)}_{\text{gradient difference fluctuation during training}} + \underbrace{2c^2\cdot T}_{\text{non-smoothness cost}}.\end{aligned} \tag{22}$$

Proof. By definition of the neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, we have that

$$\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}')\|_2^2 = \|\ell(f_{\boldsymbol{W}}(\boldsymbol{x});\boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}}(\boldsymbol{x}');\boldsymbol{y}')\|_2^2 \tag{23}$$

where $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}', \boldsymbol{y}')$ are the differing records between two datasets. By the assumption that $\|\nabla\ell(f_{\boldsymbol{W}}(\boldsymbol{x}); \boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}'}(\boldsymbol{x}); \boldsymbol{y})\|_2 < \max\{c, \beta\|\boldsymbol{W} - \boldsymbol{W}'\|_{M_T}\}$, and by the Cauchy-Schwarz inequality, we have that

$$\begin{aligned}\|\nabla\ell(f_{\boldsymbol{W}_t}(\boldsymbol{x}); \boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}_t}(\boldsymbol{x}'); \boldsymbol{y}')\|_2^2 \leq & 2\|\nabla\ell(f_{\boldsymbol{W}_0}(\boldsymbol{x}); \boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}_0}(\boldsymbol{x}'); \boldsymbol{y}')\|_2^2 \\ & + 2\beta^2\|\boldsymbol{W}_t - \boldsymbol{W}_0\|_{M_T}^2 + 2c^2 \end{aligned} \tag{24}$$

The first term $\|\nabla\ell(f_{\boldsymbol{W}_0}(\boldsymbol{x}); \boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}_0}(\boldsymbol{x}'); \boldsymbol{y}')\|_2^2$ is constant during training (as it only depends on the initialization). Therefore, we only need to bound the second term $\|\boldsymbol{W}_t - \boldsymbol{W}_0\|_{M_T}^2$. For brevity, we denote the term inside expectation as $d(\boldsymbol{W}) = \|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_T}^2$. Then by definition we have that

$$\frac{\partial}{\partial t}\mathbb{E}_{p_t}[d(\boldsymbol{W})] = \lim_{\eta\to 0}\frac{\mathbb{E}_{p_{t+\eta}}[d(\boldsymbol{W})] - \mathbb{E}_{p_t}[d(\boldsymbol{W})]}{\eta}. \tag{25}$$

Denote Γ_s as the following random operator on model parameters θ.

$$\Gamma_s(\boldsymbol{W}) = \theta - s\nabla\mathcal{L}(\boldsymbol{W}; \mathcal{D}) + \sqrt{2\sigma^2 s}Z$$

where $Z \sim \mathcal{N}(0, \mathbb{I})$. We first claim that the following equation holds.

$$\lim_{\eta\to 0}\frac{\mathbb{E}_{p_{t+\eta}}[d(\boldsymbol{W})] - \mathbb{E}_{p_t}[d(\Gamma_\eta(\boldsymbol{W}))]}{\eta} = 0 \tag{26}$$

This is by using Euler-Maruyama discretization method to approximate the solution $\boldsymbol{W}_t$ of SDE Eq. (4). More specifically, the approximation error $\mathbb{E}_{p_{t+\eta}}[d(\boldsymbol{W})] - \mathbb{E}_{p_t}[d(\Gamma_\eta(\boldsymbol{W}))]$ is of size $O(r\eta^2)$, where r is the dimension of $\boldsymbol{W}$.

Therefore, by plugging Eq. (26) into Eq. (25), we have that

$$\frac{\partial}{\partial t}\mathbb{E}_{p_t}[d(\boldsymbol{W})] = \lim_{\eta\to 0}\frac{\mathbb{E}_{p_t}[d(\Gamma_\eta(\boldsymbol{W}))] - \mathbb{E}_{p_t}[d(\boldsymbol{W})]}{\eta}$$

Recall that $\nabla^2 d(\boldsymbol{W})$ exists almost everywhere with regard to $\boldsymbol{W} \sim p_t$. Therefore we could approximate the term $\mathbb{E}_{p_t}[d(\Gamma_\eta(\boldsymbol{W}); \mathcal{D}, \mathcal{D}')]$ via its second-order Taylor expansion at $\boldsymbol{W}$ as follows.

$$\begin{aligned}\frac{\partial}{\partial t}\mathbb{E}_{p_t}[d(\boldsymbol{W})] &= \lim_{\eta\to 0}\frac{\mathbb{E}_{p_t}[\langle\nabla d(\boldsymbol{W}), -\eta\nabla\mathcal{L}(\boldsymbol{W}; \mathcal{D}) + \sqrt{2\sigma^2\eta}Z\rangle + \sigma^2\eta Z^\top\nabla^2 d(\boldsymbol{W})Z + o(\eta)]}{\eta} \\ &= -\mathbb{E}_{p_t}[\langle\nabla d(\boldsymbol{W}), \nabla\mathcal{L}(\boldsymbol{W}; \mathcal{D})\rangle] + \sigma^2\mathbb{E}_{p_t}[\mathrm{Tr}\left(\nabla^2 d(\boldsymbol{W})\right)] \end{aligned} \tag{27}$$

By plugging $d(\boldsymbol{W}) = \|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_T}^2$ into the above equation, we have that

$$\begin{aligned}\frac{\partial}{\partial t}\mathbb{E}_{p_t}[\|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_t}^2] &\leq -2\mathbb{E}_{p_t}[\langle\boldsymbol{W} - \boldsymbol{W}_0, \nabla\mathcal{L}(\boldsymbol{W}; \mathcal{D})\rangle] + \sigma^2\mathrm{rank}(M_T) && (28) \\ &= -2\mathbb{E}_{p_t}[\langle\boldsymbol{W} - \boldsymbol{W}_0, \nabla\mathcal{L}(\boldsymbol{W}_0; \mathcal{D})\rangle] + \sigma^2\mathrm{rank}(M_T) \\ &\quad -2\mathbb{E}_{p_t}[\langle\boldsymbol{W} - \boldsymbol{W}_0, \nabla\mathcal{L}(\boldsymbol{W}; \mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_0; \mathcal{D})\rangle] \\ &\leq \mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_0; \mathcal{D})\|_2^2] + \mathbb{E}_{p_t}[\|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_t}^2] + \sigma^2\mathrm{rank}(M_T) && (29) \\ &\quad + \mathbb{E}_{p_t}[\|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_T}^2] + \mathbb{E}_{p_t}[\|\nabla\mathcal{L}(\boldsymbol{W}; \mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_0; \mathcal{D})\|_2^2] && (30)\end{aligned}$$

where the last inequality is by using the Cauchy-schwartz inequality. By plugging the assumption that $\|\nabla\ell(f_{\boldsymbol{W}}(\boldsymbol{x}); \boldsymbol{y})) - \nabla\ell(f_{\boldsymbol{W}'}(\boldsymbol{x}); \boldsymbol{y})\|_2 < \max\{c, \beta\|\boldsymbol{W} - \boldsymbol{W}'\|_2\}$ into the above inequality, we have that

$$\frac{\partial}{\partial t}\mathbb{E}_{p_t}[\|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_T}^2] \leq (2 + \beta^2)\mathbb{E}_{p_t}[\|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_T}^2] + \mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_0; \mathcal{D})\|_2^2] + c^2 + \sigma^2\mathrm{rank}(M_T) \tag{31}$$

By solving the above ordinary differential inequality on $t \in [0, T]$, we have that

$$\mathbb{E}_{p_t}[\|\boldsymbol{W} - \boldsymbol{W}_0\|_{M_T}^2] \leq \frac{e^{(2+\beta^2)t} - 1}{\beta^2}\left(\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_0; \mathcal{D})\|_2^2] + \sigma^2\mathrm{rank}(M_T) + c^2\right) \tag{32}$$

By plugging Eq. (32) into Eq. (24) and integrating over time $t \in [0, T]$, we have that

$$\int_0^T \mathbb{E}_{p_t}\left[\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}')\|_2^2\right] dt \leq 2T \cdot \mathbb{E}_{p_0}\left[\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W};\mathcal{D}')\|_2^2\right] \quad (33)$$

$$+ 2\left(\frac{e^{(2+\beta^2)T} - (2+\beta^2)T}{2+\beta^2}\right) \cdot \left(\mathbb{E}_{p_0}\left[\|\nabla\mathcal{L}(\boldsymbol{W};\mathcal{D})\|_2^2\right] + \sigma^2 \text{rank}(M_T) + c^2\right) + 2c^2T. \quad (34)$$

□

C Deferred proofs for Section 4

C.1 Bounding the gradient norm at initialization

To bound the moment of ℓ_2 norm of the gradient $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{W}_l}$ of network output function, we need the following (extended) lemmas from Zhu et al. [47].

Lemma C.1 ([47, Lemma 1]). *Let $\boldsymbol{w} \sim \mathcal{N}(0, \sigma^2\mathbb{I}_n)$, then for two fixed non-zero vectors $\boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^n$ whose correlation is unknown, define two random variables $X = (\boldsymbol{w}^\top \boldsymbol{h}_1 1_{\{\boldsymbol{w}^\top \boldsymbol{h}_2 \geq 0\}})^2$ and $Y = s(\boldsymbol{w}^\top \boldsymbol{h}_1)^2$, where $s \sim Ber(1, 1/2)$ follows a Bernoulli distribution with 1 trial and $\frac{1}{2}$ success rate, and s and $\boldsymbol{w}$ are independent random variables. Then X and Y have the same distribution.*

Lemma C.2 (Extension of [47, Lemma 2]). *Given a fixed non-zero matrix $\boldsymbol{H}_1 \in \mathbb{R}^{p\times r}$ and a fixed non-zero vector $\boldsymbol{h}_2 \in \mathbb{R}^p$ and , let $\boldsymbol{W} \in \mathbb{R}^{q\times p}$ be a random matrix with i.i.d. entries $W_{ij} \sim \mathcal{N}(0, \beta)$ and a matrix (or vector) $\boldsymbol{V} = \phi'(\boldsymbol{W}\boldsymbol{h}_2)\boldsymbol{W}\boldsymbol{H}_1 \in \mathbb{R}^{q\times r}$, then, we have $\mathbb{E}[\frac{\|\boldsymbol{V}\|_F^2}{\|\boldsymbol{H}_1\|_F^2}] = \frac{q\beta}{2}$.*

Proof. According to the definition of $\boldsymbol{V} = \phi'(\boldsymbol{W}\boldsymbol{h}_2)\boldsymbol{W}\boldsymbol{H}_1 \in \mathbb{R}^{q\times r}$, we have:

$$\|\boldsymbol{V}\|_F^2 = \sum_{i=1}^q \sum_{j=1}^r \left(\boldsymbol{D}_{i,i}\langle \boldsymbol{W}^{[i;]}, \boldsymbol{H}_1^{[;j]}\rangle\right)^2,$$

where $\boldsymbol{D}_{i,i} = 1_{\{\langle \boldsymbol{W}^{[i]}, \boldsymbol{h}_2\rangle \geq 0\}}$, $\boldsymbol{W}^{[i;]}$ is the i-th row of $\boldsymbol{W}$, and $\boldsymbol{H}_1^{[;j]}$ is the j-th column vector of $\boldsymbol{H}_1$. Therefore by Lemma C.1, with i.i.d. Bernoulli random variable $\rho_1, \cdots, \rho_q \sim \text{Ber}(1, 1/2)$, we have

$$\|\boldsymbol{V}\|_F^2 \overset{d}{=} \sum_{i=1}^q \sum_{j=1}^r \rho_i \langle \boldsymbol{W}^{[i;]}, \boldsymbol{H}_1^{[;j]}\rangle^2 = \sum_{i=1}^q \sum_{j=1}^r \rho_i \beta \|\boldsymbol{H}_1^{[;j]}\|_2^2 \tilde{w}_{ij}^2 .$$

where $\tilde{w}_{ij} = \langle \boldsymbol{W}^{[i;]}, \boldsymbol{H}_1^{[;j]}\rangle / \left(\sqrt{\beta\|\boldsymbol{H}_1^{[;j]}\|_2^2}\right)$. By the fact that $\boldsymbol{W}^{[i;]}$ has i.i.d. Gaussian entries, for any fixed j, we have that $\tilde{w}_{ij} \sim \mathcal{N}(0, 1), i = 1, \cdots, q$ independently. Therefore, we have

$$\mathbb{E}\left[\|\boldsymbol{V}\|_F^2\right] = \sum_{i=1}^q \sum_{j=1}^r \mathbb{E}[\rho_i]\beta\|\boldsymbol{H}_1^{[;j]}\|_2^2 \mathbb{E}[\tilde{w}_{ij}^2] = \frac{q\beta}{2}\mathbb{E}[\|\boldsymbol{H}_1\|_F^2] .$$

□

Now, we are ready to prove output gradient expectation at random initialization as follows.

Lemma C.3 (Output Gradient Expectation Bound at Random Initialization). *Fix any data record $\boldsymbol{x}$, then over the randomness of the initialization distributions for $\boldsymbol{W}_1, \cdots, \boldsymbol{W}_L$, i.e., $\boldsymbol{W}_l \sim \mathcal{N}(0, \beta_l\mathbb{I})$ for $l = 1, \cdots, L-1$, it satisfies that*

$$\mathbb{E}_{\boldsymbol{W}}\left[\|\frac{\partial f(\boldsymbol{x})}{\partial Vec(\boldsymbol{W})}\|_F^2\right] = \|\boldsymbol{x}\|_2^2 o\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right)\sum_{l=1}^L \frac{\beta_L}{\beta_l} . \quad (35)$$

Proof. We use $\text{Vec}(\boldsymbol{W}_l)$ to denote the concatenation of all row vector of the parameter matrix $\boldsymbol{W}_l$. By chain rule, for $l = 1, \cdots, L-1$, we have that

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \text{Vec}(\boldsymbol{W}_l)} = \frac{\partial h_L(\boldsymbol{x})}{\partial h_{L-1}(\boldsymbol{x})} \left(\prod_{i=1}^{L-1-l} \frac{\partial h_{L-i}(\boldsymbol{x})}{\partial h_{L-1-i}(\boldsymbol{x})} \right) \frac{\partial h_l}{\text{Vec}(\boldsymbol{W}_l)} \tag{36}$$

$$= \boldsymbol{W}_L \left(\prod_{i=1}^{L-1-l} \sigma'_{L-i} \boldsymbol{W}_{L-i} \right) \sigma'_l \begin{pmatrix} h_{l-1}^\top & 0 & \cdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & h_{l-1}^\top \end{pmatrix}_{m_l \times m_l m_{l-1}} . \tag{37}$$

Similarly, for the L-th layer, we have that

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \text{Vec}(\boldsymbol{W}_L)} = \begin{pmatrix} h_{L-1}^\top & 0 & \cdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & h_{L-1}^\top \end{pmatrix}_{o \times o m_{L-1}} . \tag{38}$$

By properties of ReLU activation ϕ, we have $\phi'_{L-i} = \text{diag}[\text{sgn}(W_{L-i} h_{L-1-i})]$, where $\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$ operates coordinate-wise with regard to the input matrix. Therefore, we have that for $l = 1, \cdots, L-1$

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \text{Vec}(\boldsymbol{W}_l)} = \boldsymbol{W}_L \left(\prod_{i=1}^{L-1-l} \text{diag}[\text{sgn}(W_{L-i} h_{L-1-i})] \boldsymbol{W}_{L-i} \right) \cdot \text{diag}[\text{sgn}(W_l h_{l-1})] \begin{pmatrix} h_{l-1}^\top & 0 & \cdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & h_{l-1}^\top \end{pmatrix}_{m_l \times m_l m_{l-1}} .$$

For notational simplicity, we introduce the notation of $\boldsymbol{t}_l^{l'}$ for $l = 1, \cdots, L-1$ and $l \leq l' < L$ as follows.

$$\boldsymbol{t}_l^{l'} := \left(\prod_{i=L-l'}^{L-1-l} \text{diag}[\text{sgn}(W_{L-i} h_{L-1-i})] \boldsymbol{W}_{L-i} \right) \cdot \text{diag}[\text{sgn}(W_l h_{l-1})] \begin{pmatrix} h_{l-1}^\top & 0 & \cdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & h_{l-1}^\top \end{pmatrix}_{m_l \times m_l m_{l-1}} .$$

Then by definition, we have that

$$\mathbb{E}_{\boldsymbol{W}} \left[\| \frac{\partial f(\boldsymbol{x})}{\partial \text{Vec}(\boldsymbol{W}_l)} \|_F^2 \right] = \mathbb{E}_{\boldsymbol{W}} \left[\| \boldsymbol{W}_L \boldsymbol{t}_l^{L-1} \|_F^2 \right] = \mathbb{E}_{\boldsymbol{W}} \left[\frac{\| \boldsymbol{W}_L \boldsymbol{t}_l^{L-1} \|_F^2}{\| \boldsymbol{t}_l^{L-1} \|_F^2} \cdot \frac{\| \boldsymbol{t}_l^{L-1} \|_F^2}{\| \boldsymbol{t}_l^{L-2} \|_F^2} \cdots \frac{\| \boldsymbol{t}_l^{l+1} \|_F^2}{\| \boldsymbol{t}_l^{l} \|_F^2} \cdot \| \boldsymbol{t}_l^{l} \|_F^2 \right]$$

$$= \mathbb{E}_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_l} \left[\| \boldsymbol{t}_l^l \|_F^2 \cdot \mathbb{E}_{\boldsymbol{W}_{l+1}} \left[\frac{\| \boldsymbol{t}_l^{l+1} \|_F^2}{\| \boldsymbol{t}_l^{l} \|_F^2} \cdots \mathbb{E}_{\boldsymbol{W}_L} \left[\frac{\| \boldsymbol{W}_L \boldsymbol{t}_l^{L-1} \|_F^2}{\| \boldsymbol{t}_l^{L-1} \|_F^2} \right] \right] \right] .$$

By rotational invariance of Gaussian column vectors, we prove that for any possible value of $\boldsymbol{t}_l^{L-1}$ (which is completely determined by $\boldsymbol{W}_1, \cdots, \boldsymbol{W}_{L-1}$ and $\boldsymbol{x}$), for any $l = 1, \cdots, L-1$, we have that

$$\mathbb{E}_{\boldsymbol{W}_L} \left[\frac{\| \boldsymbol{W}_L \boldsymbol{t}_l^{L-1} \|_F^2}{\| \boldsymbol{t}_l^{L-1} \|_F^2} \right] = \mathbb{E}_{\boldsymbol{W}_L} \left[\frac{\| \boldsymbol{W}_L \boldsymbol{e}_1 \|_2^2}{\| \boldsymbol{e}_1 \|_2^2} \right] = \beta_L o . \tag{39}$$

By Lemma C.2, for any $l = 1, \cdots, L-2$ and $l \leq l' \leq L-2$, we have that

$$\mathbb{E}_{\boldsymbol{W}_{l+1}} \left[\frac{\| \boldsymbol{t}_l^{l'+1} \|_F^2}{\| \boldsymbol{t}_l^{l'} \|_F^2} \right] = \frac{\beta_{l'+1}}{2} m_{l'+1} . \tag{40}$$

We now bound the last term $\mathbb{E}_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_l} \left[\| \boldsymbol{t}_l^l \|_F^2 \right]$ for $l = 1, \cdots, L-1$. By definition, we have that

$$\mathbb{E}_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_l} \left[\| \boldsymbol{t}_l^l \|_F^2 \right] = \mathbb{E}_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_l} \left[\sum_{i=1}^{m_l} 1_{\{ \boldsymbol{W}_l^{[i;]} h_{l-1} \leq 0 \}} \cdot \| h_{l-1} \|_2^2 \right] = \frac{m_l}{2} \mathbb{E}_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_{l-1}} \mathbb{E}[\| h_{l-1} \|_2^2] . \tag{41}$$

To bound the term $\mathbb{E}_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_l}[\|h_{l-1}\|_2^2]$, note that by Lemma C.2, we have that for any $l = 1, \cdots, L-1$

$$\mathbb{E}_{\boldsymbol{W}_{l-1}}\left[\frac{\|h_{l-1}(\boldsymbol{x})\|_2^2}{\|h_{l-2}(\boldsymbol{x})\|_2^2}\right] = \frac{\beta_{l-1}}{2} m_{l-1}\,. \tag{42}$$

Therefore, for any $l = 1, \cdots, L$, we have that

$$\mathbb{E}_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{l-1}}\left[\|h_{l-1}(\boldsymbol{x})\|_2^2\right] = \mathbb{E}_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{l-1}}\left[\frac{\|h_{l-1}(\boldsymbol{x})\|_2^2}{\|h_{l-2}(\boldsymbol{x})\|_2^2}\cdots\frac{\|h_1(\boldsymbol{x})\|_2^2}{\|\boldsymbol{x}\|_2^2}\right]\cdot\|\boldsymbol{x}\|_2^2 \tag{43}$$

$$= \left(\prod_{i=1}^{l-1}\frac{\beta_i}{2}m_i\right)\|\boldsymbol{x}\|_2^2\,. \tag{44}$$

By plugging (44) into (41), we have that

$$\mathbb{E}_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_l}\left[\|\boldsymbol{t}_l^l\|_2^2\right] = \frac{m_l}{2}\left(\prod_{i=1}^{l-1}\frac{\beta_i}{2}m_i\right)\|\boldsymbol{x}\|_2^2\,. \tag{45}$$

By combining (39), (40) and (45), we have for any $l = 1, \cdots, L-1$

$$\mathbb{E}_{\boldsymbol{W}}\left[\|\frac{\partial f(\boldsymbol{x})}{\partial \mathrm{Vec}(\boldsymbol{W}_l)}\|_F^2\right]$$
$$= \frac{m_l}{2}\left(\prod_{i=1}^{l-1}\frac{\beta_i}{2}m_i\right)\cdot\left(\prod_{i=l+1}^{L-1}\frac{\beta_i}{2}m_i\right)\cdot\beta_L o\cdot\|\boldsymbol{x}\|_2^2 = \frac{\beta_L}{\beta_l}\|\boldsymbol{x}\|_2^2 o\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right).$$

On the other hand, by plugging Eq. (44) (under $\ell = L$) into Eq. (38), we have that

$$\mathbb{E}_{\boldsymbol{W}}\left[\|\frac{\partial f(\boldsymbol{x})}{\partial \mathrm{Vec}(\boldsymbol{W}_L)}\|_2^2\right] = o\left(\prod_{i=1}^{L-1}\frac{\beta_i}{2}m_i\right)\|\boldsymbol{x}\|_2^2$$

Therefore,

$$\mathbb{E}_{\boldsymbol{W}}\left[\|\frac{\partial f(\boldsymbol{x})}{\partial \mathrm{Vec}(\boldsymbol{W})}\|_F^2\right] = \sum_{l=1}^{L}\|\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{W}_l}\|_F^2 = \|\boldsymbol{x}\|_2^2 o\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right)\sum_{l=1}^{L}\frac{\beta_L}{\beta_l}\,,$$

which suffices to prove Eq. (35). □

C.2 Deferred proof for Lemma 4.1

Finally, we prove that the gradient difference between two training datasets under linearized network is bounded by a constant throughout training (which only depends on the network width, depth and initialization distribution).

Lemma 4.1. Under Assumption 2.1, taking over the randomness of the random initialization and the Brownian motion in Langevin diffusion, for any $t \in [0, T]$, it satisfies that

$$\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \mathcal{L}(\boldsymbol{W}_t;\mathcal{D}')\|^2\right] \le \frac{4B}{n^2}\,, \tag{46}$$

where n is the training dataset size, and B is a constant that only depends on the network width, depth and initialization distribution as follows.

$$B := o\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right)\sum_{l=1}^{L}\frac{\beta_L}{\beta_l}\,, \tag{47}$$

where o is the number of output classes, $\{m_i\}_{i=1}^L$ are the per-layer network widths, and $\{\beta_i\}_{i=1}^L$ are the variances of Gaussian initialization at each layer.

Proof. Denote $\boldsymbol{W}$ as the initialization parameters and denote $\boldsymbol{W}_t^{lin}$ as the parameters for linearized network after training time t. Then the gradient difference under linearized network and cross-entropy loss function is as follows.

$$\begin{aligned}
&\|\nabla\mathcal{L}(W_t;\mathcal{D})-\nabla\mathcal{L}(W_t;\mathcal{D}')\|_F^2\\
=&\left\|\frac{\nabla\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x})^\top\left(\text{softmax}(\boldsymbol{f}_{\boldsymbol{W}_t}(\boldsymbol{x}))-\boldsymbol{y}\right)}{n}-\frac{\nabla\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}')^\top\left(\text{softmax}(\boldsymbol{f}_{\boldsymbol{W}_t}(\boldsymbol{x}'))-\boldsymbol{y}'\right)}{n}\right\|_F^2\\
\leq&\frac{2}{n^2}\left(\|\nabla\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x})\|_F^2+\|\nabla\boldsymbol{f}_{\boldsymbol{W}}(\boldsymbol{x}')\|_F^2\right).
\end{aligned}$$

Plugging Lemma 4.1 into the above equation with data Assumption 2.1 suffice to prove the result. □

D Deferred proofs for Section 5

D.1 Excess empirical risk for training linearized network (average iterate)

To prove empirical risk bound for the last iterate of training linearized network, we will first need to prove the following intermediate result of excess empirical risk bound for average iterate.

Lemma D.1 (Excess empirical risk for average iterate (Extension of [39, Theorem 3.1]))**.** *Let $\boldsymbol{W}_0^{lin}$ be a randomly initialized parameter vector by* (5)*. Let the empirical NTK feature matrix for dataset training $\mathcal{D}$ at initialization be $M_0=\left(\nabla\boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_1)\quad\cdots\quad\nabla\boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_n)\right)$. Let $\mathcal{L}_0^{lin}(\boldsymbol{W};\mathcal{D})$ be the empirical loss for linearized network* (3) *expanded at initialization vector $\boldsymbol{W}_0^{lin}$. Then running Langevin diffusion* (4) *under empirical loss $\mathcal{L}_0^{lin}(\boldsymbol{W};\mathcal{D})$ and initialization $\boldsymbol{W}_0^{lin}$ for time T satisfies the following excess empirical risk bound.*

$$\mathbb{E}[\mathcal{L}(\bar{\boldsymbol{W}}_T^{lin})]-\mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^*;\mathcal{D})]\leq\frac{R}{2T}+\frac{1}{2}\sigma^2\mathbb{E}[rank(M_0)]$$

where $\bar{\boldsymbol{W}}_T^{lin}=\frac{1}{T}\int\bar{\boldsymbol{W}}_t^{lin}dt$ is the average of all iterates, $\boldsymbol{W}_0^*$ is an (exact or approximate) solution for the ERM problem on $\mathcal{L}_0^{lin}(\boldsymbol{W};\mathcal{D})$, and $R=\mathbb{E}[\|\boldsymbol{W}_0^{lin}-\boldsymbol{W}_0^*\|_{M_0}^2]$ is the expected gap between initialization parameters $\boldsymbol{W}_0$ and solution $\boldsymbol{W}_0^*$.

Proof. Our proofs are heavily based on the idea in [39, Theorem 3.1] to work only in the parameter space spanned by the input feature vectors. And our proof serves as an extension of their bound to the continuous-time Langevin diffusion algorithm. We begin by using convexity of the empirical loss function $\mathcal{L}^{lin}(\boldsymbol{W};\mathcal{D})$ for linearized network to prove the following standard results

$$\mathcal{L}^{lin}(\bar{\boldsymbol{W}}_T^{lin};\mathcal{D})-\mathcal{L}^{lin}(\boldsymbol{W}_0^*;\mathcal{D})\leq\langle\bar{\boldsymbol{W}}_T^{lin}-\boldsymbol{W}_0^*,\nabla\mathcal{L}^{lin}(\bar{\boldsymbol{W}}_T^{lin};\mathcal{D})\rangle\tag{48}$$

Denote $M_0=\left(\nabla\boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_1)\quad\cdots\quad\nabla\boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_n)\right)$. By computing the gradient under cross entropy loss and linearized network, we have $\nabla\mathcal{L}^{lin}(\boldsymbol{W}_T^{lin};\mathcal{D})$ lies in the column space of M_0. Denote Π_{M_0} as the projection operator to the column space of M_0, then (48) can be rewritten as

$$\mathcal{L}^{lin}(\bar{\boldsymbol{W}}_T^{lin};\mathcal{D})-\mathcal{L}^{lin}(\boldsymbol{W}_0^*;\mathcal{D})\leq\langle\Pi_{M_0}(\bar{\boldsymbol{W}}_T^{lin}-\boldsymbol{W}_0^*),\nabla\mathcal{L}^{lin}(\bar{\boldsymbol{W}}_T^{lin};\mathcal{D})\rangle.\tag{49}$$

By taking expectation over training randomness and initialization distribution, we have

$$\mathbb{E}[\mathcal{L}(\bar{\boldsymbol{W}}_T^{lin})]-\mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^*;\mathcal{D})]\leq\frac{1}{T}\int_0^T\mathbb{E}\left[\langle\Pi_{M_0}(\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*),\nabla\mathcal{L}^{lin}(\boldsymbol{W}_t^{lin};\mathcal{D})\rangle\right]dt\tag{50}$$

We now rewrite $\mathbb{E}\left[\langle\Pi_{M_0}(\bar{\boldsymbol{W}}_t^{lin}-\boldsymbol{W}_0^*),\nabla\mathcal{L}^{lin}(\boldsymbol{W}_t^{lin};\mathcal{D})\rangle\right]$ by computing $\frac{\partial}{\partial t}\mathbb{E}[\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*\|_{M_0}^2]$, where $\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*\|_{M_0}^2=\Pi_{M_0}\left(\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*\right)^\top\Pi_{M_0}\left(\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*\right)$. By applying (27), we have

$$\frac{\partial}{\partial t}\mathbb{E}[\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*\|_{M_0}^2]=-2\mathbb{E}[\langle\Pi_{M_0}\left(\boldsymbol{W}_t^{lin}-\boldsymbol{W}_0^*\right),\nabla\mathcal{L}^{lin}(\boldsymbol{W}_t^{lin};\mathcal{D})\rangle]+\sigma^2\mathbb{E}[rank(M_0)]\tag{51}$$

Therefore by plugging (51) into (50), we have that

$$\begin{aligned}
\mathbb{E}[\mathcal{L}^{lin}(\boldsymbol{W};\mathcal{D})-\mathcal{L}^{lin}(\boldsymbol{W}_0^*;\mathcal{D})]&\leq-\frac{1}{2T}\int_0^T\frac{\partial}{\partial t}\mathbb{E}[\|\boldsymbol{W}_t-\boldsymbol{W}_0^*\|_{M_0}^2]dt+\frac{1}{2}\sigma^2\mathbb{E}[rank(M_0)]&(52)\\
&\leq\frac{1}{2T}\mathbb{E}[\|\boldsymbol{W}_0^{lin}-\boldsymbol{W}_0^*\|_{M_0}^2]+\frac{1}{2}\sigma^2\mathbb{E}[rank(M_0)]&(53)
\end{aligned}$$

□

D.2 Deferred proof for Proposition 5.1

We are now ready to prove the last iterate excess empirical risk bound for training linearized network.

Proposition 5.1 (Excess empirical risk for training linearized network (last iterate))**.** Let $\boldsymbol{W}_0^{lin}$ be a randomly initialized parameter vector by (5). Let the empirical NTK feature mapping matrix for dataset training $\mathcal{D}$ at initialization be $M_0 = \big(\nabla \boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_1) \quad \cdots \quad \nabla \boldsymbol{f}_{\boldsymbol{W}_0^{lin}}(\boldsymbol{x}_n)\big)$. Let $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D})$ be the empirical loss for linearized network (3) expanded at initialization vector $\boldsymbol{W}_0^{lin}$. Then running Langevin diffusion (4) under empirical loss $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D})$ and initialization $\boldsymbol{W}_0^{lin}$ for time T satisfies the following excess empirical risk bound

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})] - \mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^*; \mathcal{D})] \leq \frac{2R}{T} + \frac{1}{2}\sigma^2 \mathbb{E}[rank(M_0)] \left(1 + \log \frac{2BT^2}{R}\right) \tag{54}$$

where $\boldsymbol{W}_0^*$ is an (exact or approximate) solution for the ERM problem on $\mathcal{L}_0^{lin}(\boldsymbol{W}; \mathcal{D})$, and $R = \mathbb{E}[\|\boldsymbol{W}_0^{lin} - \boldsymbol{W}_0^*\|_{M_0}^2]$ is the expected gap between initialization parameters $\boldsymbol{W}_0$ and solution $\boldsymbol{W}_0^*$.

Proof. We first define the following potential function.

$$\Phi(t) = \frac{1}{T-t} \int_t^T \mathbb{E}[\mathcal{L}(\boldsymbol{W}_\tau^{lin}; \mathcal{D})] d\tau \tag{55}$$

By definition we have that the boundary values are $\Phi(0) = \frac{1}{T}\int_0^T \mathbb{E}[\mathcal{L}(\boldsymbol{W}_\tau^{lin}; \mathcal{D})] d\tau$ and $\Phi(T) = \lim_{t \to T} \frac{1}{T-t}\int_t^T \mathbb{E}[\mathcal{L}(\boldsymbol{W}_\tau^{lin}; \mathcal{D})] d\tau = \mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin}; \mathcal{D})]$. Since we have proved upper bound for $\Phi(0)$ in the excess empirical risk bound for the average iterate Appendix D.1, to analyze $\Phi(T)$ the loss difference between last iterate and average iterate, we only need to prove upper bound for $\Phi(T) - \Phi(0)$.

By definition, we compute the partial derivative of the function $\Phi(t)$ with regard to time t as follows.

$$\begin{aligned}
\frac{\partial}{\partial t}\Phi(t) =& \frac{1}{(T-t)^2}\int_t^T \mathbb{E}[\mathcal{L}(\boldsymbol{W}_\tau^{lin}; \mathcal{D})] d\tau - \frac{1}{T-t}\mathbb{E}[\mathcal{L}(\boldsymbol{W}_t^{lin}; \mathcal{D})] \\
=& \frac{1}{(T-t)^2}\int_t^T \mathbb{E}[\mathcal{L}(\boldsymbol{W}_\tau^{lin}; \mathcal{D}) - \mathcal{L}(\boldsymbol{W}_t^{lin}; \mathcal{D})] d\tau \\
\leq& \frac{1}{(T-t)^2}\int_t^T \mathbb{E}[\langle \nabla \mathcal{L}(\boldsymbol{W}_\tau^{lin}), \Pi_{M_0}(\boldsymbol{W}_\tau^{lin} - \boldsymbol{W}_t^{lin})\rangle] d\tau
\end{aligned} \tag{56}$$

where the last inequality is by the convexity of loss function for linearized network. Now to control the intergral in (56), we use a similar argument to (51) as follows.

$$\frac{\partial}{\partial \tau}\mathbb{E}\left[\|\boldsymbol{W}_\tau^{lin} - \boldsymbol{W}_t^{lin}\|_{M_0}^2\right] = -2\mathbb{E}[\langle \nabla \mathcal{L}(\boldsymbol{W}_\tau^{lin}), \Pi_{M_0}(\boldsymbol{W}_\tau^{lin} - \boldsymbol{W}_t^{lin})\rangle] + \sigma^2 \mathbb{E}[rank(M_0)] \tag{57}$$

By plugging (57) into (56), we have that

$$\frac{\partial}{\partial t}\Phi(t) \leq \frac{\int_t^T -\frac{\partial}{\partial \tau}\mathbb{E}[\|\boldsymbol{W}_\tau^{lin} - \boldsymbol{W}_t^{lin}\|_{M_0}^2] + \sigma^2 \mathbb{E}[rank(M_0)] d\tau}{2(T-t)^2} \leq \frac{\sigma^2 \mathbb{E}[rank(M_0)]}{2(T-t)} \tag{58}$$

By intergrating the above equation over $t \in [0, T - \Delta T]$ where $\Delta \in (0, 1)$ is a tuning parameter that we will determine later, we have that

$$\Phi(T - \Delta T) \leq \Phi(0) + \sigma^2 \mathbb{E}[rank(M_0)] \ln \frac{1}{\Delta} \tag{59}$$

Now we proceed to bound $\Phi(T) - \Phi(T - \Delta T)$. By definition of $\Phi(t)$, we have that

$$\begin{aligned}
\Phi(T) - \Phi(T - \Delta T) =& \frac{1}{\Delta T}\int_{T-\Delta T}^T \mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin}; \mathcal{D}) - \mathcal{L}(\boldsymbol{W}_\tau^{lin}; \mathcal{D})] d\tau \\
\leq& \frac{1}{\Delta T}\int_{T-\Delta T}^T \mathbb{E}[\langle \nabla \mathcal{L}(\boldsymbol{W}_T^{lin}; \mathcal{D}), \Pi_{M_0}(\boldsymbol{W}_T^{lin} - \boldsymbol{W}_\tau^{lin})\rangle] d\tau
\end{aligned} \tag{60}$$

where (60) is by convexity of empirical loss function and by that $\nabla\mathcal{L}^{lin}(\boldsymbol{W}_T^{lin};\mathcal{D})$ is in the linear space spanned by the network output gradient at initialization.

By applying the Cauchy-Schwartz inequality, we have that for any $\alpha \in (0,+\infty)$, the following inequality holds.

$$\Phi(T)-\Phi(T-\Delta T) \leq \frac{1}{\Delta T}\int_{T-\Delta T}^{T} \frac{1}{\alpha}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_T^{lin};\mathcal{D})\|_2^2 + \frac{\alpha}{4}\mathbb{E}[\|\boldsymbol{W}_T^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2]d\tau \quad (61)$$

$$\leq \frac{1}{\alpha}\max_{\tau\in[T-\Delta T,T]}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_T^{lin};\mathcal{D})\|_2^2 + \frac{\alpha}{4}\max_{\tau\in[T-\Delta T,T]}\mathbb{E}[\|\boldsymbol{W}_T^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2] \quad (62)$$

By Eq. (8), we have that when the data is normalized,

$$\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_T^{lin};\mathcal{D})\|_2^2 \leq 2B \quad (63)$$

where $B = |\mathcal{Y}|\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right)\sum_{l=1}^{L}\frac{\beta_L}{\beta_l}$. We now only need to bound $\mathbb{E}[\|\boldsymbol{W}_T^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2]$. By similar argument as (57), for any $t \geq \tau$, we have that

$$\begin{aligned}
\frac{\partial}{\partial t}\mathbb{E}\left[\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2\right] &= -2\mathbb{E}[\langle\nabla\mathcal{L}(\boldsymbol{W}_t^{lin}), \Pi_{M_0}(\boldsymbol{W}_t^{lin}-\boldsymbol{W}_\tau^{lin})\rangle] + \sigma^2\mathbb{E}[rank(M_0)] \\
&\leq \frac{1}{\gamma}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_t^{lin})\|_2^2] + \gamma\mathbb{E}[\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2] + \sigma^2\mathbb{E}[rank(M_0)] \\
&\leq \gamma\mathbb{E}[\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2] + \frac{2B}{\gamma} + \sigma^2\mathbb{E}[rank(M_0)]
\end{aligned}$$

This is a linear ODE and can be solve closed formly as follows for $t \geq \tau$.

$$\mathbb{E}\left[\|\boldsymbol{W}_t^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2\right] \leq (\frac{2B}{\gamma^2} + \frac{\sigma^2\mathbb{E}[rank(M_0)]}{\gamma})(e^{\gamma(t-\tau)}-1)$$

Since the above equation holds for any $\tau \leq t$, by setting $\tau = T-\Delta T$ and $t = T$ we have that

$$\max_{\tau\in[T-\Delta T,T]}\mathbb{E}[\|\boldsymbol{W}_T^{lin}-\boldsymbol{W}_\tau^{lin}\|_{M_0}^2] \leq (\frac{2B}{\gamma^2} + \frac{\sigma^2\mathbb{E}[rank(M_0)]}{\gamma})(e^{\gamma\Delta T}-1) \quad (64)$$

$$\leq 4B(\Delta T)^2 + 2\sigma^2\mathbb{E}[rank(M_0)]\Delta T \quad (65)$$

where (65) is by setting $\gamma = \frac{1}{\Delta T}$ and by $e^1 - 1 \leq 2$. By combining (62), (63) and (65), we have that

$$\begin{aligned}
\Phi(T) - \Phi(T-\Delta T) &\leq \frac{2B}{\alpha} + \alpha B(\Delta T)^2 + \frac{\alpha\sigma^2\mathbb{E}[rank(M_0)]\Delta T}{2} \\
&= \frac{3R}{2T} + \frac{\sigma^2\mathbb{E}[rank(M_0)]}{2} \quad (66)
\end{aligned}$$

where the last equation (66) is by setting $\alpha = \frac{2B}{R}T$ and $\Delta = \frac{R}{2BT^2}$ (note that here B and R are as given in the proposition statement). By combining (59) and (66) and using that $\Delta = \frac{R}{2BT^2}$, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})] = \Phi(T) &= \Phi(T) - \Phi(T-\Delta T) + \Phi(T-\Delta T) \\
&\leq \Phi(0) + \frac{\sigma^2\mathbb{E}[rank(M_0)]}{2}\left(1+\log\frac{2BT^2}{R}\right) + \frac{3R}{2T}
\end{aligned}$$

Observe that $\Phi(0) = \mathbb{E}[\mathcal{L}(\bar{\boldsymbol{W}}_T^{lin})]$, therefore by using Lemma D.1 we have that

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})] \leq \mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^{*};\mathcal{D})] + \frac{2R}{T} + \frac{\sigma^2\mathbb{E}[rank(M_0)]}{2}\left(1+\log\frac{2BT^2}{R}\right)$$

□

D.3 Deferred proof for Proposition 5.3

Proposition 5.3 (Bounding lazy training distance via smallest eigenvalue of the NTK matrix)**.** Under the data and network regularity Assumption 2.1, if the width $m_1 = \cdots = m_{L-1} = \Omega(n)$ is sufficiently large, then there exists an optimal solution $\boldsymbol{W}_0^{\frac{1}{n^2}}$ that satisfies $\mathcal{L}_0^{lin}(\boldsymbol{W}_0^{\frac{1}{n^2}}) \leq \frac{1}{n^2}$ and satisfies

$$\tilde{R} = \mathbb{E}[\|\boldsymbol{W}_0^{\frac{1}{n}} - \boldsymbol{W}_0\|_2^2] \leq \begin{cases} \tilde{O}(\frac{n}{d \cdot 2^L \cdot (m(L-2)+1)}) & \text{for NTK initialization} \\ \tilde{O}(\frac{n}{2^L m(L-1)}) & \text{for He initialization} \\ \tilde{O}(\frac{n}{m(L-1)}) & \text{for LeCun initialization} \end{cases} \tag{67}$$

Proof. Given arbitrary initialization parameters $\boldsymbol{W}_0$, we first construct an solution $\boldsymbol{W}_0^{\frac{1}{n^2}}$ that is nearly optimal for the ERM problem over $\mathcal{L}_0^{lin}(\boldsymbol{W})$. Specifically, let $\boldsymbol{W}_0^{\frac{1}{n^2}}$ have the following expression.

$$\boldsymbol{W}_0^{\frac{1}{n^2}} - \boldsymbol{W}_0 = M_0^\dagger \begin{pmatrix} 2\ln n \cdot y_1 - \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1) \\ \vdots \\ 2\ln n \cdot y_n - \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_n) \end{pmatrix} \tag{68}$$

where $M_0 = \begin{pmatrix} \nabla \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1)^\top \\ \vdots \\ \nabla \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_n)^\top \end{pmatrix}$ is the NTK feature matrix at initialization and $\dagger$ denotes the pseudo-inverse. By random Gaussian initialization, and by the data regularity assumption Assumption 2.2, we have that $rank(M_0) = n$ with probability one, therefore $M_0^\dagger = M_0^\top (M_0 M_0^\top)^{-1}$ and $\begin{pmatrix} \boldsymbol{f}_{\boldsymbol{W}_0^{\frac{1}{n^2}}}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{f}_{\boldsymbol{W}_0^{\frac{1}{n^2}}}(\boldsymbol{x}_n) \end{pmatrix} = \begin{pmatrix} 2\ln n \cdot y_1 \\ \vdots \\ 2\ln n \cdot y_n \end{pmatrix}$ with probability one. By further using the definition of cross-entropy loss for the single-output setting, we have that the solution $\boldsymbol{W}_0^{\frac{1}{n^2}}$ satisfies the following inequality.

$$\mathcal{L}_0^{lin}(\boldsymbol{W}_0^{\frac{1}{n^2}}) = \log(1 + \exp(-2\ln n)) < \frac{1}{n^2} \tag{69}$$

We now only need to prove that the solution $\boldsymbol{W}_0^{\frac{1}{n^2}}$ is close to the initialization parameters $\boldsymbol{W}_0$ in expected ℓ_2 norm. By applying the holder inequality on (68), we have that

$$\tilde{R} = \mathbb{E}[\|\boldsymbol{W}_0^{\frac{1}{n^2}} - \boldsymbol{W}_0\|_2^2] \leq \mathbb{E}\left[\|M_0^\dagger\|_2^2 \cdot \left\| \begin{pmatrix} 2\ln n \cdot y_1 - \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1) \\ \vdots \\ 2\ln n \cdot y_n - \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_n) \end{pmatrix} \right\|_2^2 \right] \tag{70}$$

$$\leq \mathbb{E}\left[\frac{1}{\lambda_{min}(M_0 M_0^\top)} \cdot \left\| \begin{pmatrix} 2\ln n \cdot y_1 - \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1) \\ \vdots \\ 2\ln n \cdot y_n - \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_n) \end{pmatrix} \right\|_2^2 \right] \tag{71}$$

We now prove bounds for the two terms on the right hand side separately. For the first term, when the width is sufficiently large $m = \Omega(n)$, by existing bound for the smallest eigenvalue of the NTK matrix $M_0 M_0^\top$ in [33, Theorem 4.1], we have that with high probability

$$\frac{1}{\lambda_{min}(M_0 M_0^\top)} \leq O\left(\frac{1}{\left(d \prod_{l=1}^{L-1} m_l\right) \cdot \left(\prod_{l=1}^{L} \beta_l\right) \cdot \left(\sum_{l=2}^{L} \beta_l^{-1}\right)} \right) \tag{72}$$

For the second term, by Cauchy-Schwarz inequality, we have that

$$\mathbb{E}\left[\left\|\begin{pmatrix}2\ln n\cdot y_1-\boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1)\\ \vdots\\ 2\ln n\cdot y_n-\boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_n)\end{pmatrix}\right\|_2^2\right]\leq 2\cdot(2\ln n)^2\sum_{i=1}^n y_i^2+2\sum_{i=1}^n\mathbb{E}[\|\boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1)\|_2^2]$$
$$=8n(\ln n)^2+2\sum_{i=1}^n\mathbb{E}[\|\boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1)\|_2^2]$$

By further using Eq. (44), we have that

$$\mathbb{E}\left[\left\|\begin{pmatrix}2\ln n\cdot y_1-\boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_1)\\ \vdots\\ 2\ln n\cdot y_n-\boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x}_n)\end{pmatrix}\right\|_2^2\right]=O\left(n(\ln n)^2+n\prod_{i=1}^L\frac{\beta_i m_i}{2}\right)\tag{73}$$

Therefore, by plugging Eq. (72) and Eq. (73) into Eq. (71), and by considering input dimension d as const, we have that

$$\tilde{R}=O\left(\frac{n(\ln n)^2+n\prod_{i=1}^L\frac{\beta_i m_i}{2}}{\left(d\prod_{l=1}^{L-1}\beta_l m_l\right)\cdot\beta_L\cdot\left(\sum_{l=2}^L\beta_l^{-1}\right)}\right)$$

By plugging the choice of initialization variance $\beta_1,\cdots,\beta_L$ for NTK, He and LeCun initialization into the above equation, for single output network with $L\geq 2$, we have that

$$\tilde{R}=\begin{cases}\tilde{O}(\frac{n}{d\cdot 2^L\cdot(m(L-2)+1)}) & \text{for NTK initialization}\\ \tilde{O}(\frac{n}{2^L m(L-1)}) & \text{for He initialization}\\ \tilde{O}(\frac{n}{m(L-1)}) & \text{for LeCun initialization}\end{cases}$$

□

D.4 Deferred proof for Corollary 5.4

Corollary 5.4 (Privacy utility trade-off for last iterate)**.** Assume that the data and network regularity Assumption 2.2 holds. Assume that all the conditions and definition for constants in Proposition 5.1 holds. Then by setting $\sigma^2=\frac{2BT}{\varepsilon n^2}$ and $T=\sqrt{\frac{2\varepsilon n\tilde{R}}{B}}$, we have that running Langevin diffusion for time T satisfies bound KL divergence ϵ, and has empirical excess risk upper bounded by

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})]\leq\mathcal{O}\left(\frac{1}{n^2}+\sqrt{\frac{B\tilde{R}}{\varepsilon n}}\log(\varepsilon n)\right)\tag{74}$$

where B is the gradient norm constant Eq. (9), and $\tilde{R}$ is the approximate lazy training distance in Eq. (12). A summary of B and $\tilde{R}$ under different initializations is in Table 1.

Proof. By setting $\boldsymbol{W}_0^*=\boldsymbol{W}_0^{\frac{1}{n^2}}$ in Proposition 5.3, we have that

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})]-\mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^{\frac{1}{n^2}};\mathcal{D})]\leq\frac{2\tilde{R}}{T}+\frac{\sigma^2\mathbb{E}[rank(M_0)]}{2}\left(1+\log\frac{2BT^2}{\tilde{R}}\right)$$

By Proposition 5.3, we have that $\mathbb{E}[\mathcal{L}(\boldsymbol{W}_0^{\frac{1}{n^2}};\mathcal{D})]\leq\frac{1}{n^2}$, therefore

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})]\leq\frac{1}{n^2}+\frac{2\tilde{R}}{T}+\frac{\sigma^2\mathbb{E}[rank(M_0)]}{2}\left(1+\log\frac{2BT^2}{\tilde{R}}\right)\tag{75}$$

By plugging $\sigma^2=\frac{2BT}{\varepsilon n^2}$ and $T=\sqrt{\frac{2\varepsilon n\tilde{R}}{B}}$ into (75), and by $\text{rank}(M_0)\leq n$, we have that

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W}_T^{lin})]\leq\frac{1}{n^2}+\sqrt{\frac{2B\tilde{R}}{\varepsilon n}}\left(2+\log(4\varepsilon n)\right)\tag{76}$$

□

D.5 Deferred proofs for Proposition 5.5

We will first need to prove following lemma to bound the uniform stability for the model parameters during training linearized network.

Lemma D.2 (Uniform stability for training linearized network)**.** *Assume that the data and network regularity Assumption 2.2 holds. Then it satisfies that*

$$\mathbb{E}\left[\|\boldsymbol{W}_T - \boldsymbol{W}_T'\|^2\right] \leq \frac{2B}{n^2}T^2 \tag{77}$$

where $B = |\mathcal{Y}|\left(\prod_{i=1}^{L-1}\frac{\beta_i m_i}{2}\right)\sum_{l=1}^{L}\frac{\beta_L}{\beta_l}$ *is the constant defined in* (37) *for network output gradient norm bound at initialization.*

Proof. By definition, and by coupling the choice of Gaussian noise in the two Langevin diffusion processes $(\boldsymbol{W}_t)_{t\in[0,T]}$ and $(\boldsymbol{W}_t')_{t\in[0,T]}$, we have that

$$\begin{aligned}
&\frac{\partial\mathbb{E}\left[\|\boldsymbol{W}_t - \boldsymbol{W}_t'\|^2\right]}{\partial t} = \lim_{\eta\to 0}\frac{\mathbb{E}[\|\boldsymbol{W}_{t+\eta} - \boldsymbol{W}_{t+\eta}'\|^2] - \mathbb{E}[\|\boldsymbol{W}_t - \boldsymbol{W}_t'\|^2]}{\eta}\\
&= \lim_{\eta\to 0}\frac{\mathbb{E}[\|\boldsymbol{W}_t - \eta\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \boldsymbol{W}_t' + \eta\nabla\mathcal{L}(\boldsymbol{W}_t';\mathcal{D}')\|^2] - \mathbb{E}[\|\boldsymbol{W}_t - \boldsymbol{W}_t'\|^2]}{\eta}\\
&= \lim_{\eta\to 0}\frac{\eta^2\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_t';\mathcal{D}')\|^2] - 2\eta\mathbb{E}[\langle\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_t';\mathcal{D}'), \boldsymbol{W}_t - \boldsymbol{W}_t'\rangle]}{\eta}\\
&= -2\mathbb{E}[\langle\nabla\mathcal{L}(\boldsymbol{W}_t;\mathcal{D}) - \nabla\mathcal{L}(\boldsymbol{W}_t';\mathcal{D}'), \boldsymbol{W}_t - \boldsymbol{W}_t'\rangle] \leq 2\sqrt{\frac{2B}{n^2}\cdot\mathbb{E}\left[\|\boldsymbol{W}_T - \boldsymbol{W}_T'\|^2\right]}
\end{aligned}$$

where the last inequality is by holder's inequality and by using Lemma 4.1. By solving the ordinary differential equation with boundary condition $\mathbb{E}\left[\|\boldsymbol{W}_0 - \boldsymbol{W}_0'\|^2\right] = 0$, we have that

$$\mathbb{E}\left[\|\boldsymbol{W}_T - \boldsymbol{W}_T'\|^2\right] \leq \frac{2B}{n^2}T^2 \tag{78}$$

□

We are now ready to prove our excess population risk bound for training linearized network.

Proposition 5.5. Denote $R_0(\boldsymbol{W}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in pop}[\ell(f_{\boldsymbol{W}_0}(\boldsymbol{x}) + \frac{\partial \boldsymbol{f}_{\boldsymbol{W}_0}(\boldsymbol{x})}{\partial \boldsymbol{W}_0}(\boldsymbol{W} - \boldsymbol{W}_0); \boldsymbol{y})]$ as the population risk of linearized network expanded at initialization vector $\boldsymbol{W}_0$ over population data distribution pop. Then under the conditions of Corollary 5.4, we have that for any dataset $\mathcal{D}$ of size n, the following excess population risk upper bound holds.

$$\mathbb{E}[R_0(\boldsymbol{W}_T)] - \mathbb{E}[\mathcal{L}(\boldsymbol{W}_{pop,0}^*;\mathcal{D})] \leq O\left(\frac{1}{n^2} + \sqrt{\frac{B\tilde{R}}{\varepsilon n}}(\log(\varepsilon n) + \varepsilon)\right) \tag{79}$$

where the expectation is over the randomness of sampling the training dataset $\mathcal{D}\sim pop^n$ from the data population and the random coins for the Langevin diffusion training algorithm, and $\boldsymbol{W}_{pop,0}^* = \operatorname{argmin}_{\boldsymbol{W}} R_0(\boldsymbol{W})$ is the optimal solution for the population risk minimization problem.

Proof. By the uniform stability method [24, Theorem 2.2], we have the following generalization error upper bound holds.

$$\alpha_{gen} = |\mathbb{E}[R_0(\boldsymbol{W}_T)] - \mathbb{E}_{\mathcal{D}\sim pop^n}[\mathcal{L}(\boldsymbol{W}_T;\mathcal{D})]| \leq \max_{z,D,D'}\mathbb{E}\left[\ell(f_{\boldsymbol{W}_T}(\boldsymbol{x}_z);\boldsymbol{y}_z) - \ell(f_{\boldsymbol{W}_T'}(\boldsymbol{x}_z);\boldsymbol{y}_z)\right] \tag{80}$$

where $\boldsymbol{z} = (\boldsymbol{x}_z, \boldsymbol{y}_z)$ is an arbitrary data point in the population data distribution pop. By convexity, we further have the following bound for the uniform stability.

$$\begin{aligned}
\ell(f_{\boldsymbol{W}_T}(\boldsymbol{x}_z);\boldsymbol{y}_z) - \ell(f_{\boldsymbol{W}_T'}(\boldsymbol{x}_z);\boldsymbol{y}_z) &\leq \nabla\ell(f_{\boldsymbol{W}_T}(\boldsymbol{x}_z);\boldsymbol{y}_z)^\top(\boldsymbol{W}_T - \boldsymbol{W}_T') &(81)\\
&\leq \sqrt{\|\nabla\ell(f_{\boldsymbol{W}_T}(\boldsymbol{x}_z);\boldsymbol{y}_z)\|^2\|\boldsymbol{W}_T - \boldsymbol{W}_T'\|^2} &(82)
\end{aligned}$$

By Eq. (37) and Lemma D.2, we further have that

$$\mathbb{E}[\ell(f_{\boldsymbol{W}_T}(\boldsymbol{x}_z);\boldsymbol{y}_z)-\ell(f_{\boldsymbol{W}'_T}(\boldsymbol{x}_z);\boldsymbol{y}_z)]\leq\sqrt{2B}\cdot\sqrt{\frac{2B}{n^2}T^2}=\frac{2B}{n}T \tag{83}$$

Therefore, by plugging the above equation into (80), we have that the generalization error satisfies

$$\alpha_{gen}=|\mathbb{E}_{\mathcal{D}\sim pop^n}[\mathcal{L}(\boldsymbol{W}_T;\mathcal{D})]-\mathbb{E}[R(\boldsymbol{W}_T)]|\leq\frac{2BT}{n}\leq 2\sqrt{\frac{2\varepsilon BR}{n}} \tag{84}$$

where the last inequality is by plugging our choice of $T=\sqrt{\frac{2\varepsilon nR}{B}}$ into the equation. On the other hand, by Proposition 5.1, we have that the empirical risk is upper bounded as follows.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}\sim pop^n}[\mathcal{L}(\boldsymbol{W}_T;\mathcal{D})]\leq&O\left(\sqrt{\frac{B\tilde{R}}{\varepsilon n}}\log(\varepsilon n)\right)\\ \leq&\mathbb{E}[\mathcal{L}(\boldsymbol{W}^*_{pop,0};\mathcal{D})]+O\left(\sqrt{\frac{B\tilde{R}}{\varepsilon n}}\log(\varepsilon n)\right)\end{aligned} \tag{85}$$

Combining the generalization error term (84) and the excess empirical risk term (85) suffice to prove the equation in the statement. □

E Discussion on extending our results to Noisy GD with constant step-size

In this section, we discuss how to extend our privacy analyses to noisy GD with constant step-size. Specifically, we only need to extend the KL composition theorem under possibly unbounded gradient difference, i.e., Theorem 3.1, to the noisy GD algorithm.

Theorem E.1 (KL composition for noisy GD under possibly unbounded gradient difference)**.** *Let the iterative update in noisy GD algorithm be defined by:* $\boldsymbol{W}_{(k+1)}=\boldsymbol{W}_{(k)}-\eta\nabla\mathcal{L}(\boldsymbol{W}_{(k)};\mathcal{D})+\sqrt{2\eta\sigma^2}Z_k$*, where* $Z_k\sim\mathcal{N}(0,\mathbb{I})$*. Then the KL divergence between running noisy GD for DNN* (2) *on neighboring datasets* $\mathcal{D}$ *and* $\mathcal{D}'$ *satisfies*

$$KL(\boldsymbol{W}_{(K)},\boldsymbol{W}'_{(K)})\leq\frac{1}{2\sigma^2}\sum_{k=0}^{K-1}\eta\cdot\mathbb{E}\left[\left\|\nabla\mathcal{L}(\boldsymbol{W}_{(k)};\mathcal{D})-\nabla\mathcal{L}(\boldsymbol{W}_{(k)};\mathcal{D}')\right\|_2^2\right]. \tag{86}$$

Proof. Denote $p_{(k)}$ as the distribution of model parameters after running noisy GD on dataset $\mathcal{D}$ with k steps, and similarly denote $p'_{(k)}$ as the distribution of model parameters after running noisy GD on dataset $\mathcal{D}'$ with k steps. Then by the data processing inequality for KL divergence [45, Theorem 9] (with the data processing operation given by $(\boldsymbol{W}_{(k)},\boldsymbol{W}_{(k+1)})\to\boldsymbol{W}_{(k)}$), we have that

$$KL(p_{(k+1)},p'_{(k+1)})\leq KL(p_{(k),(k+1)},p'_{(k),(k+1)}), \tag{87}$$

where $p_{(k),(k+1)}$ denotes the joint distribution of $(\boldsymbol{W}_{(k)},\boldsymbol{W}_{(k+1)})$, and $p'_{(k),(k+1)}$ denotes the joint distribution of $(\boldsymbol{W}'_{(k)},\boldsymbol{W}'_{(k+1)})$. Now we expand the term $KL(p_{(k),(k+1)},p'_{(k),(k+1)})$ by the Bayes rule as follows.

$$\begin{aligned}&KL(p_{(k),(k+1)},p'_{(k),(k+1)}) &(88)\\ =&\mathbb{E}_{p_{(k),(k+1)}(\boldsymbol{W}_{(k)},\boldsymbol{W}_{(k+1)})}\left[\log\left(\frac{p_{(k+1)|(k)}(\boldsymbol{W}_{(k+1)}|\boldsymbol{W}_{(k)})p_{(k)}(\boldsymbol{W}_{(k)})}{p'_{(k+1)|(k)}(\boldsymbol{W}_{(k+1)}|\boldsymbol{W}_{(k)})p'_{(k)}(\boldsymbol{W}_{(k)})}\right)\right]\\ =&\mathbb{E}_{p_{(k),(k+1)}(\boldsymbol{W}_{(k)},\boldsymbol{W}_{(k+1)})}\left[\log\left(\frac{p_{(k+1)|(k)}(\boldsymbol{W}_{(k+1)}|\boldsymbol{W}_{(k)})}{p'_{(k+1)|(k)}(\boldsymbol{W}_{t+\eta}|\boldsymbol{W}_t)}\right)\right]+\mathbb{E}_{p_{(k)}(\boldsymbol{W}_{(k)})}\left[\log\left(\frac{p_{(k)}(\boldsymbol{W}_t)}{p'_{(k)}(\boldsymbol{W}_{(k)})}\right)\right]\\ =&\mathbb{E}_{p_{(k)}(\boldsymbol{W}_{(k)})}\left[KL(p_{(k+1)|(k)},p'_{(k+1)|(k)})\right]+KL(p_{(k)},p'_{(k)}) &(89)\end{aligned}$$

Observe that $p_{(k+1)|(k)}, p'_{(k+1)|(k)}$ are two Gaussian distributions with per-dimensional variance σ^2, due to the conditioning on the same model parameters $\boldsymbol{W}_{(k)}$ at iteration k. Therefore, by computing the KL divergence between two multivariate Gaussians, we have that

$$KL(p_{(k),(k+1)}, p'_{(k),(k+1)}) = \frac{1}{2\sigma^2} \cdot \eta \cdot \left\| \nabla \mathcal{L}(\boldsymbol{W}_{(k)}; \mathcal{D}) - \nabla \mathcal{L}(\boldsymbol{W}_{(k)}; \mathcal{D}') \right\|_2^2 \tag{90}$$

Therefore, by plugging Eq. (90) into Eq. (89) and Eq. (87), we have that

$$KL(p_{(k+1)}, p'_{(k+1)}) \leq \frac{\eta}{2\sigma^2} \mathbb{E}\left[\left\| \nabla \mathcal{L}(\boldsymbol{W}_{(k)}; \mathcal{D}) - \nabla \mathcal{L}(\boldsymbol{W}_{(k)}; \mathcal{D}') \right\|_2^2 \right] + KL(p_{(k)}, p'_{(k)}) \tag{91}$$

By summing (91) over $k = 0, \cdots, K-1$ and observing that $KL(p_{(0)}, p'_{(0)}) = 0$ (as the initialization distribution is the same between noisy GD on $\mathcal{D}$ and $\mathcal{D}'$), we finish the proof for Eq. (86). $\square$