Particle Dual Averaging: Optimization of Mean Field Neural Network with Global Convergence Rate Analysis

Atsushi Nitanda
Kyushu Institute of Technology
RIKEN Center for Advanced Intelligence Project
nitanda@ai.kyutech.ac.jp

Denny Wu
University of Toronto
Vector Institute for Artificial Intelligence
dennywu@cs.toronto.edu

Taiji Suzuki
University of Tokyo
RIKEN Center for Advanced Intelligence Project
taiji@mist.i.u-tokyo.ac.jp

Abstract

We propose the *particle dual averaging* (PDA) method, which generalizes the dual averaging method in convex optimization to the optimization over probability distributions with quantitative runtime guarantee. The algorithm consists of an inner loop and outer loop: the inner loop utilizes the Langevin algorithm to approximately solve for a stationary distribution, which is then optimized in the outer loop. The method can thus be interpreted as an extension of the Langevin algorithm to naturally handle *nonlinear* functional on the probability space. An important application of the proposed method is the optimization of neural network in the *mean field* regime, which is theoretically attractive due to the presence of nonlinear feature learning, but quantitative convergence rate can be challenging to obtain. By adapting finite-dimensional convex optimization theory into the space of measures, we analyze PDA in regularized empirical / expected risk minimization, and establish *quantitative* global convergence in learning two-layer mean field neural networks under more general settings. Our theoretical results are supported by numerical simulations on neural networks with reasonable size.

1 Introduction

Gradient-based optimization can achieve vanishing training error on neural networks, despite the apparent non-convex landscape. Among various works that explains the global convergence, one common ingredient is to utilize overparameterization to translate the training dynamics into function spaces, and then exploit the convexity of the loss function with respect to the function. Such endeavors usually consider models in one of the two categories: the *mean field* regime or the *kernel* regime.

On one hand, analysis in the kernel (lazy) regime connects gradient descent on wide neural network to kernel regression with respect to the neural tangent kernel (Jacot et al., 2018), which leads to global convergence at linear rate (Du et al., 2019; Allen-Zhu et al., 2019; Zou et al., 2020). However, key to the analysis is the *linearization* of the training dynamics, which requires appropriate scaling of the model such that distance traveled by the parameters vanishes (Chizat and Bach, 2018a). Such regime thus fails to explain the *feature learning* of neural networks (Yang and Hu, 2020), which is believed to be an important advantage of deep learning; indeed, it has been shown that deep learning can outperform kernel models due to this adaptivity (Suzuki, 2018; Ghorbani et al., 2019a).

In contrast, the mean field regime describes the gradient descent dynamics as Wasserstein gradient flow in the probability space (Nitanda and Suzuki, 2017; Mei et al., 2018; Chizat and Bach, 2018b),

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

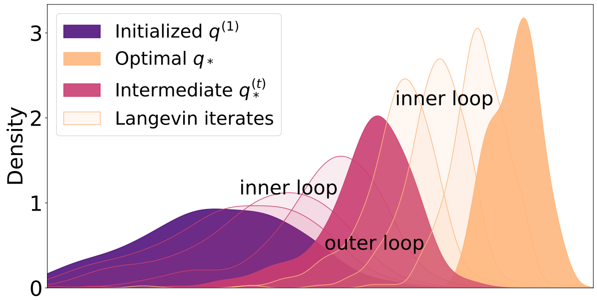

Figure 1: 1D visualization of parameter distribution of mean field two-layer neural network (tanh) optimized by PDA. The *inner loop* uses the Langevin algorithm to solve an approximate stationary distribution $q^{(t)}$, which is then optimized in the *outer loop* towards the true target q .

which captures the potentially *nonlinear* evolution of parameters travelling beyond the kernel regime. While the mean field limit is appealing due to the presence of "feature learning", its characterization is more challenging and quantitative analysis is largely lacking. Recent works established convergence rate in continuous time under modified dynamics (Rotskoff et al., 2019), strong assumptions on the target function (Javanmard et al., 2019), or regularized objective (Hu et al., 2019), but such result can be fragile in the discrete-time or finite-particle setting — in fact, the discretization error often scales exponentially with the time horizon or dimensionality, which limits the applicability of the theory. Hence, an important research problem that we aim to address is

Can we develop optimization algorithms for neural networks in the mean field regime with more accurate quantitative guarantees the kernel regime enjoys?

We address this question by introducing the *particle dual averaging* (PDA) method, which globally optimizes an entropic regularized nonlinear functional. For two-layer mean field network which is an important application, we establish polynomial runtime guarantee for the *discrete-time* algorithm; to our knowledge this is the first quantitative global convergence result under similar settings.

1.1 Contributions

We propose the PDA algorithm, which draws inspiration from the dual averaging method originally developed for finite-dimensional convex optimization (Nesterov, 2005, 2009; Xiao, 2009). We iteratively optimize a probability distribution in the form of a Boltzmann distribution, samples from which can be obtained from the Langevin algorithm (see Figure 1). The resulting algorithm has comparable per-iteration cost as gradient descent and can be efficiently implemented.

For optimizing two-layer neural network in the mean-field regime, we establish quantitative global convergence rate of PDA in minimizing an KL-regularized objective: the algorithm requires $O(\quad ^{3})$ steps and $O(\quad ^{2})$ particles to reach an -accurate solution, where O hides logarithmic factors. Importantly, our analysis does not couple the learning dynamics with certain continuous time limit, but directly handles the discrete update. This leads to a simpler analysis that covers more general settings. We also derive the generalization bound on the solution obtained by the algorithm. From the viewpoint of the optimization, PDA is an extension of Langevin algorithm to handle entropic-regularized nonlinear functionals on the probability space. Hence we believe our proposed method can also be applied to other distribution optimization problems beyond the training of neural networks.

1.2 Related Literature

Mean field limit of two-layer NNs. The key observation for the mean field analysis is that when the number of neurons becomes large, the evolution of parameters is well-described by a nonlinear partial differential equation (PDE), which can be viewed as solving an infinite-dimensional *convex* problem (Bengio et al., 2006; Bach, 2017). Global convergence can be derived by studying the limiting PDE (Mei et al., 2018; Chizat and Bach, 2018b; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020), yet quantitative convergence rate generally requires additional assumptions.

Javanmard et al. (2019) analyzed a particular RBF network and established linear convergence (up to certain error[1]) for strongly concave target functions. Rotskoff et al. (2019) provided a sublinear rate in continuous time for a modified gradient flow. In the regularized setting, Chizat (2019) obtained local linear convergence under certain non-degeneracy assumption on the objective. Wei et al. (2019) also proved polynomial rate for a perturbed dynamics under weak `$_2$ regularization.

[1] Note that such error yields sublinear rate with respect to arbitrarily small accuracy .

Our setting is most related to Hu et al. (2019), who studied the minimization of a nonlinear functional with KL regularization on the probability space, and showed linear convergence (in continuous time) of a particle dynamics named *mean field Langevin dynamics* when the regularization is sufficiently strong. Chen et al. (2020) also considered optimizing a KL-regularized objective in the infinite-width and continuous-time limit, and derived NTK-like convergence guarantee under certain parameter scaling. Compared to these prior works, we directly handle the discrete time update in the mean-field regime, and our analysis covers a wider range of regularization parameters and loss functions.

Langevin algorithm. Langevin dynamics can be viewed as optimization in the space of probability measures (Jordan and Kinderlehrer, 1996; Jordan et al., 1998); this perspective has been explored in Wibisono (2018); Durmus et al. (2019). It is known that the continuous-time Langevin diffusion converges exponentially fast to target distributions satisfying certain growth conditions (Roberts and Tweedie, 1996; Mattingly et al., 2002). The discretized *Langevin algorithm* has a sublinear convergence rate that depends on the numerical scheme (Li et al., 2019) and has been studied under various metrics (Dalalyan, 2014; Durmus and Moulines, 2017; Cheng and Bartlett, 2017).

The Langevin algorithm can also optimize certain non-convex objectives (Raginsky et al., 2017; Xu et al., 2018; Erdogdu et al., 2018), in which one finite-dimensional "particle" can attain approximate global convergence due to concentration of Boltzmann distribution around the true minimizer. However, such result often depends on the spectral gap that grows exponentially in dimensionality, which renders the analysis ineffective for neural net optimization in the high-dimensional parameter space

Very recently, convergence of Hamiltonian Monte Carlo in learning certain mean field models has been analyzed in Bou-Rabee and Schuh (2020); Bou-Rabee and Eberle (2021). Compared to these concurrent results, our formulation covers a more general class of potentials, and in the context of two-layer neural network, we provide optimization guarantees for a wider range of loss functions.

1.3 Notations

Let $\mathbb{R}_+$ denote the set of non-negative real numbers and $\|\cdot\|_2$ the Euclidean norm. Given a density function $q:\mathbb{R}^p\to\mathbb{R}_+$, we denote the expectation with respect to $q(\theta)d\theta$ by $\mathbb{E}_q[\cdot]$. For a function $f:\mathbb{R}^p\to\mathbb{R}$, we define $\mathbb{E}_q[f]=\int f(\theta)q(\theta)d\theta$ when f is integrable. KL is the Kullback-Leibler divergence: $KL(q\|q')\overset{\mathrm{def}}{=}\int q(\theta)\log\left(\frac{q(\theta)}{q'(\theta)}\right)d\theta$. Let $\mathcal{P}_2$ denote the set of positive densities q on $\mathbb{R}^p$ such that the second-order moment $\mathbb{E}_q[\|\theta\|_2^2]<\infty$ and entropy $-\infty<-\mathbb{E}_q[\log(q)]<+\infty$ are well defined. $N(0,I_p)$ is the Gaussian distribution on $\mathbb{R}^p$ with mean 0 and covariance matrix I_p.

2 Problem Setting

We consider the problem of risk minimization with neural networks in the mean field regime. For simplicity, we focus on supervised learning. We here formalize the problem setting and models. Let $\mathcal{X}\subset\mathbb{R}^d$ and $\mathcal{Y}\subset\mathbb{R}$ be the input and output spaces, respectively. For given input data $x\in\mathcal{X}$, we predict a corresponding output $y=h(x)\in\mathcal{Y}$ through a hypothesis function $h:\mathcal{X}\to\mathcal{Y}$.

2.1 Neural Network and Mean Field Limit

We adopt a neural network in the mean field regime as a hypothesis function. Let $\Omega=\mathbb{R}^p$ be a parameter space and $h_\theta:\mathcal{X}\to\mathcal{Y}$ $(\theta\in\Omega)$ be a bounded function which will be a component of a neural network. We sometimes denote $h(\theta,x)=h_\theta(x)$. Let $q(\theta)d\theta$ be a probability distribution on the parameter space Ω and $\Theta=\{\theta_r\}_{r=1}^M$ be the set of parameters θ_r sampled from $q(\theta)d\theta$. A hypothesis is defined as an ensemble of h_{θ_r} as follows:

$$h_\Theta(x)\overset{\mathrm{def}}{=}\frac{1}{M}\sum_{r=1}^{M}h_{\theta_r}(x). \tag{1}$$

A typical example in the literature of the above formulation is a two-layer neural network.

Example 1 (Two-layer Network)**.** Let $a_r\in\mathbb{R}$ and $b_r\in\mathbb{R}^d$ $(r\in\{1,2,\dots,M\})$ be parameters for output and input layers, respectively. We set $\theta_r=(a_r,b_r)$ and $\Theta=\{\theta_r\}_{r=1}^M$. Denote $h_{\theta_r}(x)\overset{\mathrm{def}}{=}\sigma_2(a_r\sigma_1(b_r^\top x))$ $(x\in\mathcal{X})$, where σ_1 and σ_2 are smooth activation functions. Then the hypothesis h_Θ is a two-layer neural network composed of neurons h_{θ_r}: $h_\Theta(x)=\frac{1}{M}\sum_{r=1}^{M}\sigma_2(a_r\sigma_1(b_r^\top x))$.

Remark. The purpose of σ_2 in the last layer is to ensure the boundedness of output (e.g., see Assumption 2 in Mei et al. (2018)); this nonlinearity can also be removed if parameters of output layer are fixed. In addition, although we mainly focus on the optimization of two-layer neural network, our proposed method can also be applied to ensemble of deep neural networks h_{Θ_r}.

Suppose the parameters θ_r follow a probability distribution $q(\theta)d\theta$, then h_Θ can be viewed as a finite-particle discretization of the following expectation,

$$h_q(x) = \mathbb{E}_q[h_\theta(x)]. \tag{2}$$

which we refer to as the *mean field limit* of the neural network h_Θ. As previously discussed, when h_Θ is overparameterized, optimizing h_Θ becomes “close” to directly optimizing the probability distribution on the parameter space Ω, for which convergence to the optimal solution may be established under appropriate conditions (Nitanda and Suzuki, 2017; Mei et al., 2018; Chizat and Bach, 2018b). Hence, the study of optimization of h_q with respect to the probability distribution $q(\theta)d\theta$ may shed light on important properties of overparameterized neural networks.

2.2 Regularized Empirical Risk Minimization

We briefly outline our setting for regularized expected / empirical risk minimization. The prediction error of a hypothesis is measured by the loss function $\ell(z,y)$ $(z, y \in \mathcal{Y})$, such as the squared loss $\ell(z,y) = 0.5(z-y)^2$ for regression, or the logistic loss $\ell(z,y) = \log(1+\exp(-yz))$ for binary classification. Let $\mathcal{D}$ be a data distribution over $\mathcal{X}\times\mathcal{Y}$. For expected risk minimization, the distribution $\mathcal{D}$ is set to the true data distribution; whereas for empirical risk minimization, we take $\mathcal{D}$ to be the empirical distribution defined by training data $\{(x_i,y_i)\}_{i=1}^n$ $(x_i \in \mathcal{X}, y_i \in \mathcal{Y})$ independently sampled from the data distribution. We aim to minimize the expected / empirical risk together with a regularization term, which controls the model complexity and also stabilizes the optimization. The regularized objective can be written as follows: for $\lambda_1, \lambda_2 > 0$,

$$\min_{q\in\mathcal{P}_2}\left\{\mathcal{L}(q) \overset{\text{def}}{=} \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(h_q(X),Y)] + \mathcal{R}_{\lambda_1,\lambda_2}(q)\right\}, \tag{3}$$

where $\mathcal{R}_{\lambda_1,\lambda_2}$ is a regularization term composed of the weighted sum of the second-order moment and negative entropy with regularization parameters λ_1, λ_2:

$$\mathcal{R}_{\lambda_1,\lambda_2}(q) \overset{\text{def}}{=} \lambda_1\mathbb{E}_q[\|\theta\|_2^2] + \lambda_2\mathbb{E}_q[\log(q)]. \tag{4}$$

Note that this regularization is the KL divergence of q from a Gaussian distribution. In our setting, such regularization ensures that the Gibbs distributions $q_*^{(t)}$ specified in Section 3 are well defined.

While our primary focus is the optimization of the objective (3), we can also derive a generalization error bound for the empirical risk minimizer of order $O(n^{-1/2})$ for both the regression and binary classification settings, following Chen et al. (2020). We defer the details to Appendix D.

2.3 The Langevin Algorithm

Before presenting our proposed method, we briefly review the Langevin algorithm. For a given smooth potential function $f : \mathbb{R}^p \to \mathbb{R}$, the Langevin algorithm performs the following update: given the initial $\theta^{(1)} \sim q^{(1)}(\theta)d\theta$, step size $\eta > 0$, and Gaussian noise $\zeta^{(k)} \sim \mathcal{N}(0, I_p)$,

$$\theta^{(k+1)} \leftarrow \theta^{(k)} - \eta\nabla f(\theta^{(k)}) + \sqrt{2\eta}\,\zeta^{(k)}. \tag{5}$$

Under appropriate conditions on f, it is known that $\theta^{(t)}$ converges to a stationary distribution proportional to $\exp(-f(\theta))$ in terms of KL divergence at a linear rate (e.g., Vempala and Wibisono (2019)) up to $O(\eta)$-error, where we hide additional factors in the big-O notation.

Alternatively, note that when the normalization constant $\int\exp(-f(\theta))d\theta$ exists, the Boltzmann distribution in proportion to $\exp(-f(\theta))$ is the solution of the following optimization problem,

$$\min_{q:\text{density}}\left\{\mathbb{E}_q[f] + \mathbb{E}_q[\log(q)]\right\}. \tag{6}$$

Hence we may interpret the Langevin algorithm as approximately solving an entropic regularized linear functional (i.e., free energy functional) on the probability space. This connection between sampling and optimization (see Dalalyan (2017); Wibisono (2018); Durmus et al. (2019)) enables us to employ the Langevin algorithm to obtain (samples from) the closed-form Boltzmann distribution which is the minimizer of (6); for example, many Bayesian inference problems fall into this category.

However, the objective (3) that we aim to optimize is beyond the scope of Langevin algorithm – due to the *nonlinearity* of loss $\ell(z, y)$ with respect to z, the stationary distribution cannot be described as a closed-form solution of (6). To overcome this limitation, we develop the particle dual averaging (PDA) algorithm which efficiently solves (3) with quantitative runtime guarantees.

3 Proposed Method

We now propose the *particle dual averaging* method to approximately solve the problem (3) by optimizing a two-layer neural network in the mean field regime; we also introduce the mean field limit of the proposed method to explain the algorithmic intuition and develop the convergence analysis.

3.1 Particle Dual Averaging

Our proposed particle dual averaging method (Algorithm 1) is an optimization algorithm on the space of probability measures. The algorithm consists of an inner loop and outer loop; we run Langevin algorithm in inner loop to approximate a Gibbs distribution, which is optimized in the outer loop so that it converges to the optimal distribution q. This outer loop update is designed to extend the classical dual averaging scheme (Nesterov, 2005, 2009; Xiao, 2009) to infinite dimensional optimization problems (described in Section 3.2). Below we provide a more detailed explanation.

- In the outer loop, the last iterate $\tilde{}^{(t)}$ of the previous inner loop is given. We compute $\partial_2\ell(h_{\tilde{}(t)}(x_t), y_t)$, which is a component of the Gibbs potential[2], and initialize a set of particles ${}^{(1)}$ at $\tilde{}^{(t)}$. In Appendix B we introduce a different "restarting" scheme for the initialization.
- In the inner loop, we run the Langevin algorithm (noisy gradient descent) starting from ${}^{(1)}$, where the gradient at the k-th inner step is given by $\nabla\bar{g}^{(t)}({}_r^{(k)})$, which is a sum of weighted average of $\partial_2\ell(h_{\tilde{}(s)}(x_s), y_s)\partial h({}_r^{(k)}; x_s)$ and the gradient of ℓ_2-regularization (see Algorithm 1).

Algorithm 1 Particle Dual Averaging (PDA)

```
Input: data distribution $D$, initial density $q^{(1)}$, number of outer-iterations $T$, learning rates $\{{}_t\}_{t=1}^T$,
number of inner-iterations $\{T_t\}_{t=1}^T$
Randomly draw i.i.d. initial parameters $\tilde{}_r^{(1)} \quad q^{(1)}(\ )d \quad (r \in \{1, 2, \ldots, M\})$
$\tilde{}^{(1)} \quad \{\tilde{}_r^{(1)}\}_{r=1}^M$
for $t = 1$ to $T$ do
  Randomly draw data $(x_t, y_t)$ from $D$
  ${}^{(1)} = \{{}_r^{(1)}\}_{r=1}^M \quad \tilde{}^{(t)}$
  for $k = 1$ to $T_t$ do
    Run inexact noisy gradient descent for $r \in \{1, 2, \ldots, M\}$
    $\nabla\bar{g}^{(t)}({}_r^{(k)}) \quad \frac{2}{{}_2(t+2)(\ t+1)}\sum_{s=1}^{t} s\,\partial_2\ell(h_{\tilde{}(s)}(x_s), y_s)\partial h({}_r^{(k)}; x_s) + \frac{2\ {}_1 t}{{}_2(t+2)}\ {}_r^{(k)}$
    ${}_r^{(k+1)} \quad {}_r^{(k)} \quad {}_t\nabla\bar{g}^{(t)}({}_r^{(k)}) + \sqrt{2\ {}_t}\ {}_r^{(k)}$ (i.i.d. Gaussian noise ${}_r^{(k)} \quad N(0, I_p)$)
  end for
  $\tilde{}^{(t+1)} \quad {}^{(T_t+1)} = \{{}_r^{(T_t+1)}\}_{r=1}^M$
end for
Randomly pick up $t \in \{2, 3, \ldots, T+1\}$ following the probability $P[t] = \frac{2t}{T(T+3)}$ and return $h_{\tilde{}(t)}$
```

Figure 1 provides a pictorial illustration of Algorithm 1. Note that this procedure is a slight modification of the normal gradient descent algorithm: the first term of $\nabla\bar{g}^{(t)}$ is similar to the gradient of the loss $\partial_{{}_r}\ell(h_{(k)}(x), y) \quad \partial_2\ell(h_{(k)}(x), y)\partial h({}_r^{(k)}; x)$ where ${}^{(k)} = \{{}_r^{(k)}\}_{r=1}^M$. Indeed, if we

[2] In Algorithm 1, the terms $\partial_2\ell(h_{\tilde{}(s)}(x_s), y_s)$ appear in inner loop; but note that these terms only need to be computed in outer loop because they are independent to the inner loop iterates.

set the number of inner-iterations $T_t = 1$ and replace the direction $\bar g^{(t)}(\theta_r^{(k)})$ with the gradient of the L_2-regularized loss, then PDA exactly reduces to the standard noisy gradient descent algorithm considered in Mei et al. (2018). Algorithm 1 can be extended to the minibatch variant in the obvious manner; for efficient implementation in the empirical risk minimization setting see Appendix E. 1.

3.2 Mean Field View of PDA

In this subsection we discuss the mean field limit of PDA and explain its algorithmic intuition. Note that the inner loop of Algorithm 1 is the Langevin algorithm with M particles, which optimizes the potential function given by the weighted sum:

$$\bar g^{(t)}(\theta) = \frac{2}{\lambda_2(t+2)(t+1)}\sum_{s=1}^{t} s\left(\partial_z\ell(h_{\tilde\Theta^{(s)}}(x_s); y_s)h(\theta; x_s) + \lambda_1\|\theta\|_2^2\right).$$

Due to the rapid convergence of Langevin algorithm outlined in Subsection 2.3, the particles $\theta_r^{(k+1)}$ $(r \in \{1, \ldots, M\})$ can be regarded as (approximate) samples from the Boltzmann distribution: $\exp\left(-\bar g^{(t)}\right)$. Hence, the inner loop of PDA returns an M-particle approximation of some stationary distribution, which is then modified in the outer loop. Importantly, the update on the stationary distribution is designed so that the algorithm converges to the optimal solution of the problem (3).

We now introduce the *mean field limit* of PDA, i.e., taking the number of particles $M \to \infty$ and directly optimizing the problem (3) over q. We refer to this mean field limit simply as the dual averaging (DA) algorithm. The dual averaging method was originally developed for the convex optimization in finite-dimensional spaces (Nesterov, 2005, 2009; Xiao, 2009), and here we adapt it to optimization on the probability space. The detail of the DA algorithm is described in Algorithm 2.

```
Algorithm 2 Dual Averaging (DA)
  Input: data distribution $\mathcal{D}$ and initial density $q^{(1)}$
  for $t = 1$ to $T$ do
    Randomly draw a data $(x_t, y_t)$ from $\mathcal{D}$
    $g^{(t)} \leftarrow \partial_z\ell(h_{q^{(t)}}(x_t); y_t)h(\cdot; x_t) + \lambda_1\|\cdot\|_2^2$
    Obtain an approximation $q^{(t+1)}$ of the density function $q^{(t+1)} \propto \exp\left(-\frac{\sum_{s=1}^{t} 2sg^{(s)}}{\lambda_2(t+2)(t+1)}\right)$
  end for
  Randomly pick up $\hat t \in \{2, 3, \ldots, T+1\}$ following the probability $P[t] = \frac{2t}{T(T+3)}$ and return $h_{q^{(\hat t)}}$
```

Algorithm 2 iteratively updates the density function $q^{(t+1)} \in \mathcal{P}_2$ which is a solution to the objective:

$$\min_{q\in\mathcal{P}_2}\left\{\mathbb{E}_q\left[\sum_{s=1}^{t} sg^{(s)}\right] + \frac{\lambda_2}{2}(t+2)(t+1)\,\mathbb{E}_q[\log(q)]\right\}, \qquad (7)$$

where the function $g^{(t)} = \partial_z\ell(h_{q^{(t)}}(x_t); y_t)h(\cdot; x_t) + \lambda_1\|\cdot\|_2^2$ is the functional derivative of $\ell(h_q(x_{i_t}); y_t) + \lambda_1\mathbb{E}_q[\|\theta\|_2^2]$ with respect to q at $q^{(t)}$. In other words, the objective (7) is the sum of weighted average of linear approximations of loss function and the entropic regularization in the space of probability distributions. In this sense, the DA method can be seen as an extension of the Langevin algorithm to handle entropic regularized nonlinear functionals on the probability space by *iteratively linearizing* the objective.

To sum up, we may interpret the DA method as approximating the optimal distribution q_* by iteratively optimizing $q^{(t)}$, which takes the form of a Boltzmann distribution. In the inner loop of the PDA algorithm, we obtain M (approximate) samples from $q^{(t)}$ via the Langevin algorithm. In other words, PDA can be viewed as a finite-particle approximation of DA – indeed, the stationary distributions obtained in PDA converges to $q^{(t+1)}$ by taking $M \to \infty$. In the following section, we present the convergence rate of the DA method, and also take into account the iteration complexity of the Langevin algorithm; we defer the finite-particle approximation error analysis to Appendix C.

4 Convergence Analysis

We now provide quantitative global convergence guarantee for our proposed method in discrete time. We first derive the outer loop complexity, assuming approximate optimality of the inner loop iterates, which we then verify in the inner loop analysis. The total complexity is then simply obtained by combining the outer- and inner-loop runtime.

4.1 Outer Loop Complexity

We first analyze the convergence rate of the dual averaging (DA) method (Algorithm 2). Our analysis will be made under the following assumptions.

Assumption 1.

- (A1) $\mathcal{Y} \subset [-1,1]$. $\ell(z,y)$ is a smooth convex function w.r.t. z and $|\partial_z \ell(z,y)| \le 2$ for $y, z \in \mathcal{Y}$.
- (A2) $|h(\theta;x)| \le 1$ and $h(\theta;x)$ is smooth with respect to θ for $x \in \mathcal{X}$.
- (A3) $\mathrm{KL}(\tilde{q}^{(t+1)} \| q_*^{(t+1)}) \le 1/t^2$.

Remark. (A2) is satisfied by smooth activation functions such as sigmoid and tanh. Many loss functions including the squared loss and logistic loss satisfy (A1) under the boundedness assumptions $\mathcal{Y} \subset [-1,1]$ and $|h_\theta(x)| \le 1$. Note that constants in (A1) and (A2) are defined for simplicity and can be relaxed to any value. (A3) specifies the precision of approximate solutions of sub-problems (7) to guarantee the global convergence of Algorithm 2, which we verify in our inner loop analysis.

We first introduce the following quantity for $q \in \mathcal{P}_2$,

$$e(q) \stackrel{\mathrm{def}}{=} \mathbb{E}_q[\log(q)] - \left(-\frac{4}{\lambda_2} - \frac{p}{2}\left(\exp\left(\frac{4}{\lambda_2}\right) + \log\frac{3\lambda_2}{\lambda_1}\right)\right).$$

Observe that the expression consists of the negative entropy minus its lower bound for $q^{(t)}$ under Assumption (A1), (A2); in other words $e(q^{(t)}) \ge 0$. We have the following convergence rate of DA[3].

Theorem 1 (Convergence of DA)**.** *Under Assumptions (A1), (A2), and (A3), for arbitrary $q_* \in \mathcal{P}_2$, iterates of the DA method (Algorithm 2) satisfies*

$$\frac{2}{T(T+3)}\sum_{t=2}^{T+1} t\left(\mathbb{E}[\mathcal{L}(q^{(t)})] - \mathcal{L}(q_*)\right)$$
$$\le O\left(\frac{1}{T^2}\left(1 + \lambda_1 \mathbb{E}_{q_*}\left[\|\theta\|_2^2\right]\right) + \frac{\lambda_2 e(q_*)}{T} + \frac{\lambda_2}{T}\left(1 + \exp(8/\lambda_2)\right)p^2\log^2(T+2)\right),$$

where the expectation $\mathbb{E}[\mathcal{L}(q^{(t)})]$ is taken with respect to the history of examples.

Theorem 1 demonstrates the convergence rate of Algorithm 2 to the optimal value of the regularized objective (3) in expectation. Note that $\frac{2}{T(T+3)}\sum_{t=2}^{T+1} t\mathbb{E}[\mathcal{L}(q^{(t)})]$ is the expectation of $\mathbb{E}[\mathcal{L}(q^{(t)})]$ according to the probability $\mathbb{P}[t] = \frac{2t}{T(T+3)}$ $(t \in \{2, \ldots, T+1\})$ as specified in Algorithm 2. If we take p, λ_1, λ_2 as constants and use $\tilde{O}$ to hide the logarithmic terms, we can deduce that after $\tilde{O}(\epsilon^{-1})$ iterations, an ϵ-accurate solution of the optimal distribution: $\mathcal{L}(q) \le \inf_{q \in \mathcal{P}_2} \mathcal{L}(q) + \epsilon$ is achieved in expectation. Importantly, this convergence rate applies to *any* choice of regularization parameters, in contrast to the strong regularization required in Hu et al. (2019); Jabir et al. (2019).

On the other hand, due to the exponential dependence on λ_2^{-1}, our convergence rate is not informative under weak regularization $\lambda_2 \to 0$. Such dependence follows from the classical LSI perturbation lemma (Holley and Stroock, 1987), which is likely unavoidable for Langevin-based methods in the most general setting (Menz and Schlichting, 2014), unless additional assumptions are imposed (e.g., a student-teacher setup); we intend to further investigate these conditions in future work.

[3] In Appendix B we introduce a more general version of Theorem 1 that allows for inexact $h_{q^{(t)}}(x)$, which simplifies the analysis of finite-particle discretization presented in Appendix C.

4.2 Inner Loop Complexity

In order to derive the total complexity (i.e., taking both the outer loop and inner loop into account) towards a required accuracy, we also need to estimate the iteration complexity of Langevin algorithm. We utilize the following convergence result under the log-Sobolev inequality (Definition A):

Theorem 2 (Vempala and Wibisono (2019)). Consider a probability density $q(\ ) /\ \exp(\ f(\ ))$ satisfying the log-Sobolev inequality with constant , and assume f is smooth and f is L-Lipschitz, i.e., $kr\ f(\ )\ r\ f(\ ^0)k_2\ \ Lk\ \ ^0k_2$. If we run the Langevin algorithm (5) with learning rate $0 < \frac{}{4L^2}$ and let $q^{(k)}(\ )d$ be a probability distribution that $^{(k)}$ follows, then we have,

$$KL(q^{(k)}kq)\ \ \exp(\ \ k\ )KL(q^{(1)}kq) + 8\ \ ^1 pL^2:$$

Theorem 2 implies that a -accurate solution in KL divergence can be obtained by the Langevin algorithm with $\ \frac{}{4L^2}\min\ 1; \frac{}{4p}$ and $\frac{1}{}\log\frac{2KL(q^{(1)}kq)}{}$ -iterations.

Since the optimal solution of a sub-problem in DA (Algorithm 2) takes the form of $q^{(t+1)}\ /\ \exp\ \frac{\ P_{s=1}^{t}\ 2sg^{(s)}}{_2(t+2)(\ t+1)}$, we can verify the LSI and determine the constant for $q^{(t+1)}(\ )d$ based on the LSI perturbation lemma from Holley and Stroock (1987) (see Lemma B and Example 2 in Appendix A.2). Consequently, we can apply Theorem 2 to $q^{(t+1)}$ for the inner loop complexity when $r\ \log q^{(t+1)}$ is Lipschitz continuous, which motivates us to introduce the following assumption.

Assumption 2.

(A4) $@h(\ ;x)$ is 1-Lipschitz continuous: $k@h(\ ;x)\ \ @h(\ ^0;x)k_2\ \ k\ \ \ ^0k_2$, $8x\ 2\ X,\ ;\ ^0 2\ $.

Remark. (A4) is parallel to (Mei et al., 2018, Assumption A3), and is satisfied by two-layer neural network in Example 1 when the output or input layer is fixed and the input space X is compact. We remark that this assumption can be relaxed to Hölder continuity of $@h(\ ;x)$ via the recent result of Erdogdu and Hosseinzadeh (2020), which allows us to extend Theorem 1 to general L_p-norm regularizer for $p > 1$. For now we work with (A4) for simplicity of the presentation and proof.

Set $_{t+1}$ to be the desired accuracy of an approximate solution $q^{(t+1)}$ specified in (A3): $_{t+1}\ = 1=(t+1)^2$, we have the following guarantee for the inner loop.

Corollary 1 (Inner Loop Complexity). Under (A1), (A2), and (A4), if we run the Langevin algorithm with step size $_t\ =\ O\ \frac{\ _1\ _2\ _{t+1}}{p(1+\ \ _1)^2\exp(8=\ _2)}$ on (7), then an approximate solution satisfying $KL(q^{(t+1)}kq^{(t+1)})\ \ \ _{t+1}$ can be obtained within $O\ \frac{_2\exp(8=\ _2)}{_1\ _t}\log\frac{2KL(q^{(t)}kq^{(t+1)})}{_{t+1}}$ -iterations. Moreover, $KL(q^{(t)}kq^{(t+1)})$ $(t\ 2\ f1;2;:::;T+1g)$ are uniformly bounded with respect to t as long as $q^{(1)}$ is a Gaussian distribution and (A3) is satisfied.

We comment that for the inner loop we utilized the overdamped Langevin algorithm, since it is the most standard and commonly used sampling method for the objective (7). Our analysis can easily incorporate other inner loop updates such as the underdamped Langevin algorithm (Cheng et al., 2018; Eberle et al., 2019) or the Metropolis-adjusted Langevin algorithm (Roberts and Tweedie, 1996; Dwivedi et al., 2018), which may improve the iteration complexity.

4.3 Total Complexity

Combining Theorem 1 and Corollary 1, we can now derive the total complexity of our proposed algorithm. For simplicity, we take $p;\ _1;\ \ _2$ as constants and hide logarithmic terms in $\tilde{O}$ and $\tilde{\ }$. The following corollary establishes a $\tilde{O}(\ \ ^3)$ total iteration complexity to obtain an -accurate solution in expectation because $T_t = \tilde{\ }(\ t^2) = \tilde{O}(\ \ ^2)$ for $t\ \ \ T$.

Corollary 2 (Total Complexity). Let $\ >\ 0$ be an arbitrary desired accuracy and $q^{(1)}$ be a Gaussian distribution. Under assumptions (A1), (A2), (A3), and (A4), if we run Algorithm 2 for $T\ =\ \tilde{\ }(\ \ ^1)$ iterations on the outer loop, and the Langevin algorithm with step size $_t = \frac{\ _1\ _2\ _{t+1}}{p(1+\ \ _1)^2\exp(8=\ _2)}$ for $T_t\ =\ \tilde{\ }(\ \ _t\ ^1)$ iterations on the inner loop, then an -accurate solution: $L(q)\ \ \ \inf_{q2P_2} L(q) +$ of the objective (3) is achieved in expectation.

Quantitative convergence guarantee. To translate the above convergence rate result to the finite-particle PDA (Algorithm 1), we also characterize the finite-particle discretization error in Appendix C. For the particle complexity analysis, we consider two versions of particle update: (i) the warm-start scheme described in Algorithm 1, in which $^{(1)}$ is initialized at the last iterate $\tilde{}^{(t)}$ of the previous inner loop, and (ii) the *resampling* scheme, in which $^{(1)}$ is initialized from the initial distribution $q^{(1)}(\ )d$ (see Appendix B for details). Remarkably, for the resampling scheme, we provide convergence rate guarantee in time- and space-discretized settings that is polynomial in both the iterations and particle size; specifically, the particle complexity of $\tilde{O}(^{-2})$, together with the total iteration complexity of $\tilde{O}(^{-3})$, suffices to obtain an -accurate solution to the objective (3) (see Appendix B and C for precise statement).

5 Experiments

5.1 Experiment Setup

We employ our proposed algorithm in both synthetic student-teacher settings (see Figure 2(a)(b)) and real-world dataset (see Figure 2(c)). For the student-teacher setup, the labels are generated as $y_i = f(x_i) + \varepsilon_i$, where f is the teacher model (target function), and $_i$ is zero-mean i.i.d. label noise. For the student model f, we follow Mei et al. (2018, Section 2.1) and parameterize a two-layer neural network with fixed second layer as:

$$f(x) = \frac{1}{M}\sum_{r=1}^{M}(w_r^\top x + b_r); \tag{8}$$

which we train to minimize the objective (3) using PDA. Note that $= 1$ corresponds to the mean field regime (which we are interested in), whereas setting $= 1/2$ leads to the kernel (NTK) regime[4].

Synthetic student-teacher setting. For Figure 2(a)(b) we design synthetic experiments for both regression and classification tasks, where the student model is a two-layer tanh network with $M = 500$. For regression, we take the target function f to be a multiple-index model with m neurons: $f(x) = \frac{1}{\sqrt{m}}\sum_{i=1}^{m}(\langle w_i; x\rangle)$, and the input is drawn from a unit Gaussian $N(0; I_p)$. For binary classification, we consider a simple two-dimensional dataset from sklearn.datasets.make_circles (Pedregosa et al., 2011), in which the goal is to separate two groups of data on concentric circles (red and blue in Figure 2(b)). We include additional experimental results in Appendix F.

PDA hyperparameters. We optimize the *squared loss* for regression and the *logistic loss* for binary classification. The model is trained by PDA with batch size 50. We scale the number of inner loop steps T_t with t, and the step size $_t$ with $1/\sqrt{t}$, where t is the outer loop iteration; this heuristic is consistent with the required inner-loop accuracy in Theorem 1 and Proposition 2.

(a) objective value (regression).

(b) parameter trajectory (classification).

(c) MNIST odd vs. even (classification).

Figure 2: (a) Iteration complexity of PDA: the $O(T^{-1})$ rate on the outer loop agrees with Theorem 1. (b) Parameter trajectory of PDA: darker color (purple) indicates earlier in training, and vice versa. (c) odd vs. even classification on MNIST; we report the training loss (red) as well as the train and test accuracy (blue and green).

[4] We use the term *kernel regime* only to indicate the parameter scaling; this does not necessarily imply that the NTK linearization is an accurate description of the trained model.

5.2 Empirical Findings

Convergence rate. In Figure 2(a) we verify the $O(T^{1})$ iteration complexity of the outer loop in Theorem 1. We apply PDA to optimize the expected risk (analogous to one-pass SGD) in the regression setting, in which the input dimensionality $d = 1$ and the target function is a single-index model ($m = 1$) with tanh activation. We employ the *resampled* update (i.e., without warm-start; see Appendix B) with hyperparameters $_1 = 10^{2}$; $_2 = 10^{3}$. To compute the entropy in the objective (3), we adopt the k-nearest neighbors estimator (Kozachenko and Leonenko, 1987) with $k = 10$.

Presence of feature learning. In Figure 2(b) we visualize the evolution of neural network parameters optimized by PDA in a 2-dimensional classification problem. Due to structure of the input data (concentric rings), we expect that for a two-layer neural network to be a good separator, its parameters should also distribute on a circle. Indeed the converged solution of PDA (bright yellow) agrees with this intuition and demonstrates that PDA learns useful features beyond the kernel regime.

Binary classification on MNIST. In Figure 2(c) we report the training and test performance of PDA in separating odd vs. even digits from the MNIST dataset. We subsample $n = 2500$ training examples with binary labels, and learn a two-layer tanh network with width $M = 2500$. We use the resampled update of PDA to optimize the cross entropy loss, with hyperparameters $_1 = 10^{2}$; $_2 = 10^{4}$. Observe that the algorithm achieves good generalization performance (green) and roughly maintains[5] the $O(T^{1})$ iteration complexity (red) in optimizing the training objective (3).

Conclusion

We proposed the particle dual averaging (PDA) algorithm for optimizing two-layer neural networks in the mean field regime. Leveraging tools from finite-dimensional convex optimization developed in the original dual averaging method, we established *quantitative* convergence rate of PDA for regularized empirical and expected risk minimization. We also provided particle complexity analysis and generalization bounds for both regression and classification problems. Our theoretical findings are aligned with experimental results on neural network optimization. Looking forward, we plan to investigate specific problem instances in which convergence rate can be obtained under vanishing regularization. It is also important to consider accelerated variants of PDA to further improve the convergence rate in the empirical risk minimization setting. Another interesting direction would be to explore other applications of PDA beyond two-layer neural networks, such as deep models (Araújo et al., 2019; Nguyen and Pham, 2020; Lu et al., 2020; Pham and Nguyen, 2021), as well as other optimization problems for entropic regularized nonlinear functional.

Acknowledgment

The authors would like to thank Murat A. Erdogdu and anonymous NeurIPS reviewers for their helpful feedback. AN was partially supported by JSPS Kakenhi (19K20337) and JST-PRESTO (JPMJPR1928). DW was partially supported by NSERC and LG Electronics. TS was partially supported by JSPS KAKENHI (18H03201), Japan Digital Design and JST CREST.

References

Allen-Zhu, Z. and Li, Y. (2019). What can resnet learn efficiently, going beyond kernels? In *Advances in Neural Information Processing Systems*, pages 9017–9028.

Allen-Zhu, Z. and Li, Y. (2020). Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *Proceedings of International Conference on Machine Learning*, 36, pages 242–252.

[5] Note that the estimated training objective (red) slightly deviates from the $1/T$ rate; this may be due to inaccuracy in the entropy estimation, or non-convergence of the algorithm (i.e., overestimation of $L(q)$)

Araújo, D., Oliveira, R. I., and Yukimura, D. (2019). A mean-field limit for certain deep neural networks. arXiv preprint arXiv:1906.00193.

Bach, F. (2017). Breaking the curse of dimensionality with convex neural networks. The Journal of Machine Learning Research, 18(1):629–681.

Bai, Y. and Lee, J. D. (2019). Beyond linearization: On quadratic and higher-order approximation of wide neural networks. arXiv preprint arXiv:1910.01619.

Bakry, D. and Émery, M. (1985). Diffusions hypercontractives in sem. probab. xix lnm 1123.

Bengio, Y., Le Roux, N., Vincent, P., Delalleau, O., and Marcotte, P. (2006). Convex neural networks. Advances in neural information processing systems, 18:123.

Bou-Rabee, N. and Eberle, A. (2021). Mixing time guarantees for unadjusted hamiltonian monte carlo. arXiv e-prints, pages arXiv–2105.

Bou-Rabee, N. and Schuh, K. (2020). Convergence of unadjusted hamiltonian monte carlo for mean-field models. arXiv preprint arXiv:2009.08735.

Cao, Y. and Gu, Q. (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. In Advances in Neural Information Processing Systems, pages 10836–10846.

Chen, Z., Cao, Y., Gu, Q., and Zhang, T. (2020). A generalized neural tangent kernel analysis for two-layer neural networks. arXiv preprint arXiv:2002.04026.

Cheng, X. and Bartlett, P. (2017). Convergence of langevin mcmc in kl-divergence. arXiv preprint arXiv:1705.09048

Cheng, X., Chatterji, N. S., Bartlett, P. L., and Jordan, M. I. (2018). Underdamped langevin mcmc: A non-asymptotic analysis. In Conference on Learning Theory, pages 300–323. PMLR.

Chizat, L. (2019). Sparse optimization on measures with over-parameterized gradient descent. arXiv preprint arXiv:1907.10300.

Chizat, L. (2021). Convergence rates of gradient methods for convex optimization in the space of measures. arXiv preprint arXiv:2105.08368.

Chizat, L. and Bach, F. (2018a). A note on lazy training in supervised differentiable programming. arXiv preprint arXiv:1812.07956.

Chizat, L. and Bach, F. (2018b). On the global convergence of gradient descent for over-parameterized models using optimal transport. In Advances in Neural Information Processing Systems 31, pages 3040–3050.

Chu, C., Blanchet, J., and Glynn, P. (2019). Probability functional descent: A unifying perspective on gans, variational inference, and reinforcement learning. In Proceedings of International Conference on Machine Learning 36, pages 1213–1222.

Dai, B., He, N., Dai, H., and Song, L. (2016). Provable bayesian inference via particle mirror descent. In Proceedings of International Conference on Artificial Intelligence and Statistics 19, pages 985–994.

Dalalyan, A. S. (2014). Theoretical guarantees for approximate sampling from smooth and log-concave densities. arXiv preprint arXiv:1412.7392

Dalalyan, A. S. (2017). Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. arXiv preprint arXiv:1704.04752.

Daniely, A. and Malach, E. (2020). Learning parities with neural networks. arXiv preprint arXiv:2002.07400

Ding, Z. and Li, Q. (2019). Ensemble kalman sampling: Mean-field limit and convergence analysis. arXiv preprint arXiv:1910.12923.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. International Conference on Learning Representations. 7

Durmus, A., Majewski, S., and Miasojedow, B. (2019). Analysis of langevin monte carlo via convex optimization. Journal of Machine Learning Research, 20(73):1–46.

Durmus, A. and Moulines, E. (2017). Nonasymptotic convergence analysis for the unadjusted langevin algorithm. The Annals of Applied Probability, 27(3):1551–1587.

Dwivedi, R., Chen, Y., Wainwright, M. J., and Yu, B. (2018). Log-concave sampling: Metropolis-hastings algorithms are fast! In Conference on Learning Theory, pages 793–797. PMLR.

Eberle, A., Guillin, A., Zimmer, R., et al. (2019). Couplings and quantitative contraction rates for langevin dynamics. Annals of Probability, 47(4):1982–2010.

Erdogdu, M. A. and Hosseinzadeh, R. (2020). On the convergence of langevin monte carlo: The interplay between tail growth and smoothness. arXiv preprint arXiv:2005.13097.

Erdogdu, M. A., Mackey, L., and Shamir, O. (2018). Global non-convex optimization with discretized diffusions. In Advances in Neural Information Processing Systems, pages 9671–9680.

Garbuno-Inigo, A., Hoffmann, F., Li, W., and Stuart, A. M. (2020). Interacting langevin diffusions: Gradient structure and ensemble kalman sampler. SIAM Journal on Applied Dynamical Systems, 19(1):412–441.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. (2019a). Limitations of lazy training of two-layers neural network. In Advances in Neural Information Processing Systems, pages 9111–9121.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. (2019b). Linearized two-layers neural networks in high dimension. arXiv preprint arXiv:1904.12191.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. (2020). When do neural networks outperform kernel methods? arXiv preprint arXiv:2006.13409.

Holley, R. and Stroock, D. (1987). Logarithmic sobolev inequalities and stochastic ising models. Journal of statistical physics, 46(5-6):1159–1194.

Hsieh, Y.-P., Liu, C., and Cevher, V. (2019). Finding mixed nash equilibria of generative adversarial networks. In Proceedings of International Conference on Machine Learning, pages 2810–2819. 36

Hu, K., Ren, Z., Siska, D., and Szpruch, L. (2019). Mean-field langevin dynamics and energy landscape of neural networks. arXiv preprint arXiv:1905.07769.

Imaizumi, M. and Fukumizu, K. (2020). Advantage of deep neural networks for estimating functions with singularity on curves. arXiv preprint arXiv:2011.02256.

Jabir, J.-F., Siska, D., and Szpruch, . (2019). Mean-field neural odes via relaxed optimal control. arXiv preprint arXiv:1912.05475.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Advances in Neural Information Processing Systems 31, pages 8580–8589.

Javanmard, A., Mondelli, M., and Montanari, A. (2019). Analysis of a two-layer neural network via displacement convexity. arXiv preprint arXiv:1901.01375.

Ji, Z. and Telgarsky, M. (2019). Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. arXiv preprint arXiv:1909.12292.

Jordan, R. and Kinderlehrer, D. (1996). 18. an extended variational. Partial differential equations and applications: collected papers in honor of Carlo Pucci, 177:187.

Jordan, R., Kinderlehrer, D., and Otto, F. (1998). The variational formulation of the fokker–planck equation. SIAM journal on mathematical analysis, 29(1):1–17.

Kent, C., Blanchet, J., and Glynn, P. (2021). Frank-wolfe methods in probability space. arXiv preprint arXiv:2105.05352.

Kozachenko, L. and Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. Problemy Peredachi Informatsii, 23(2):9–16.

Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. The Annals of statistics, 28(5):1302–1338.

Li, X., Wu, Y., Mackey, L., and Erdogdu, M. A. (2019). Stochastic runge-kutta accelerates langevin monte carlo and beyond. In Advances in Neural Information Processing Systems, pages 7748–7760.

Li, Y., Ma, T., and Zhang, H. R. (2020). Learning over-parametrized two-layer neural networks beyond ntk. In Proceedings of Conference on Learning Theory, 33, pages 2613–2682.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In Advances in neural information processing systems, pages 2378–2386.

Lu, J., Lu, Y., and Nolen, J. (2019). Scaling limit of the stein variational gradient descent: The mean field regime. SIAM Journal on Mathematical Analysis, 51(2):648–671.

Lu, Y., Ma, C., Lu, Y., Lu, J., and Ying, L. (2020). A mean-field analysis of deep resnet and beyond: Towards provable optimization via overparameterization from depth. arXiv preprint arXiv:2003.05508.

Mattingly, J. C., Stuart, A. M., and Higham, D. J. (2002). Ergodicity for sdes and approximations: locally lipschitz vector fields and degenerate noise. Stochastic processes and their applications, 101(2):185–232.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. Proceedings of the National Academy of Sciences, 115(33):E7665–E7671.

Menz, G. and Schlichting, A. (2014). Poincaré and logarithmic sobolev inequalities by decomposition of the energy landscape. The Annals of Probability, 42(5):1809–1884.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). Foundations of Machine Learning. The MIT Press.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. Mathematical programming, 103(1):127–152.

Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. Mathematical programming, 120(1):221–259.

Nguyen, P.-M. and Pham, H. T. (2020). A rigorous framework for the mean field limit of multilayer neural networks. arXiv preprint arXiv:2001.11443.

Nitanda, A., Chinot, G., and Suzuki, T. (2019). Gradient descent can learn less over-parameterized two-layer neural networks on classification problems. arXiv preprint arXiv:1905.09870.

Nitanda, A. and Suzuki, T. (2017). Stochastic particle gradient descent for infinite ensembles. arXiv preprint arXiv:1712.05438.

Nitanda, A. and Suzuki, T. (2020). Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime. arXiv preprint arXiv:2006.12297.

Otto, F. and Villani, C. (2000). Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. Journal of Functional Analysis, 173(2):361–400.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.

Pham, H. T. and Nguyen, P.-M. (2021). Global convergence of three-layer neural networks in the mean field regime. In International Conference on Learning Representations.

Raginsky, M., Rakhlin, A., and Telgarsky, M. (2017). Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. arXiv preprint arXiv:1702.03849.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of langevin distributions and their discrete approximations. Bernoulli, 2(4):341–363.

Rotskoff, G. M., Jelassi, S., Bruna, J., and Vanden-Eijnden, E. (2019). Global convergence of neuron birth-death dynamics. In Proceedings of International Conference on Machine Learning, 36, pages 9689–9698.

Rotskoff, G. M. and Vanden-Eijnden, E. (2018). Trainability and accuracy of neural networks: An interacting particle system approach. arXiv preprint arXiv:1805.00915.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with relu activation function. The Annals of Statistics, 48(4):1875–1897.

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

Sirignano, J. and Spiliopoulos, K. (2020). Mean field analysis of neural networks: A central limit theorem. Stochastic Processes and their Applications, 130(3):1820–1852.

Suzuki, T. (2018). Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. arXiv preprint arXiv:1810.08033.

Suzuki, T. (2020). Generalization bound of globally optimal non-convex neural network training: Transportation map estimation by infinite dimensional langevin dynamics. arXiv preprint arXiv:2007.05824.

Suzuki, T. and Nitanda, A. (2019). Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic besov space. arXiv preprint arXiv:1910.12799.

Vempala, S. and Wibisono, A. (2019). Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. In Advances in Neural Information Processing Systems, pages 8094–8106.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. (2019). Regularization matters: Generalization and optimization of neural nets vs their induced kernel. In Advances in Neural Information Processing Systems, pages 9712–9724.

Wibisono, A. (2018). Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In Proceedings of Conference on Learning Theory, 31, pages 2093–3027.

Xiao, L. (2009). Dual averaging method for regularized stochastic learning and online optimization. In Advances in Neural Information Processing Systems, 22, pages 2116–2124.

Xu, P., Chen, J., Zou, D., and Gu, Q. (2018). Global convergence of langevin dynamics based algorithms for nonconvex optimization. In Advances in Neural Information Processing Systems, pages 3122–3133.

Yang, G. and Hu, E. J. (2020). Feature learning in infinite-width neural networks. arXiv preprint arXiv:2011.14522.

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

Ying, L. (2020). Mirror descent algorithms for minimizing interacting free energy. Journal of Scientific Computing, 84(3):1–14.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2020). Gradient descent optimizes over-parameterized deep relu networks. Machine Learning, 109(3):467–492.

Table of Contents

MISSING PROOFS

A Preliminaries

A.1 Entropic Regularized Linear Functional

In this section, we explain the property of the optimal solution of the entropic regularized linear functional. We here define the gradient of the negative entropy $\mathbb{E}_q[\log(q)]$ with respect to q over the probability space as $\nabla_q \mathbb{E}_q[\log(q)] = \log(q)$. Note that this gradient is well defined up to constants as a linear operator on the probability space: $q' \mapsto \int (q' - q)(\theta)\log(q(\theta))d\theta$. The following lemma shows the strong convexity of the negative entropy.

Lemma A. Let q, q' be probability densities such that the entropy and Kullback-Leibler divergence $KL(q'\|q) = \int q'(\theta)\log\frac{q'(\theta)}{q(\theta)}d\theta$ are well defined. Then, we have

$$\mathbb{E}_q[\log(q)] + \int (q' - q)(\theta)\nabla_q \mathbb{E}_q[\log(q)]d\theta + KL(q'\|q) = \mathbb{E}_{q'}[\log(q')];$$

$$\mathbb{E}_q[\log(q)] + \int (q' - q)(\theta)\nabla_q \mathbb{E}_q[\log(q)]d\theta + \frac{1}{2}\|q' - q\|^2_{L_1(d\theta)} \le \mathbb{E}_{q'}[\log(q')].$$

The first equality of this lemma can be shown by the direct computation of the entropy, and the second inequality can be obtained by Pinsker's inequality $\frac{1}{2}\|q' - q\|^2_{L_1(d\theta)} \le KL(q'\|q)$.

Recall that $\mathcal{P}_2$ is the set of positive densities on $\mathbb{R}^p$ such that the second moment $\mathbb{E}_q[\|\theta\|_2^2] < \infty$ and entropy $-\infty < -\mathbb{E}_q[\log(q)] < +\infty$ are well defined. We here consider the minimization problem of entropic regularized linear functional on $\mathcal{P}_2$. Let $\lambda_1, \lambda_2 > 0$ be positive real numbers and $H : \mathbb{R}^p \to \mathbb{R}$ be a bounded continuous function.

$$\min_{q \in \mathcal{P}_2}\left\{ F(q) \overset{\text{def}}{=} \mathbb{E}_q[H(\theta)] + \lambda_1 \mathbb{E}_q[\|\theta\|_2^2] + \lambda_2 \mathbb{E}_q[\log(q(\theta))] \right\}. \tag{9}$$

Then, we can show $q \propto \exp\left(-\frac{H(\theta)+\lambda_1\|\theta\|_2^2}{\lambda_2}\right)$ is an optimal solution of the problem (9) as follow. Clearly, $q \in \mathcal{P}_2$ and the assumption on q in Lemma A with $q' \in \mathcal{P}_2$ holds. Hence, for $\forall q' \in \mathcal{P}_2$,

$$\begin{aligned}
F(q) &= \mathbb{E}_q[H(\theta)] + \lambda_1 \mathbb{E}_q[\|\theta\|_2^2] + \lambda_2 \mathbb{E}_q[\log(q(\theta))] \\
&= \mathbb{E}_{q'}[H(\theta)] + \lambda_1 \mathbb{E}_{q'}[\|\theta\|_2^2] + \lambda_2 \mathbb{E}_{q'}[\log(q'(\theta))] \\
&\quad + \int (q - q')(\theta)\left(H(\theta) + \lambda_1\|\theta\|_2^2\right)d\theta + \lambda_2\left(\mathbb{E}_q[\log(q(\theta))] - \mathbb{E}_{q'}[\log(q'(\theta))]\right) \\
&= F(q') + \int (q - q')(\theta)\left(H(\theta) + \lambda_1\|\theta\|_2^2\right)d\theta + \lambda_2\left(\mathbb{E}_q[\log(q(\theta))] - \mathbb{E}_{q'}[\log(q'(\theta))]\right) \\
&\le F(q') + \int (q - q')(\theta)\left(H(\theta) + \lambda_1\|\theta\|_2^2\right)d\theta - \lambda_2\left(\int (q' - q)(\theta)\nabla_q \mathbb{E}_q[\log(q)]d\theta + \frac{1}{2}\|q' - q\|^2_{L_1(d\theta)}\right) \\
&= F(q') + \int (q - q')(\theta)\left(H(\theta) + \lambda_1\|\theta\|_2^2 + \lambda_2\log(q(\theta))\right)d\theta - \frac{\lambda_2}{2}\|q' - q\|^2_{L_1(d\theta)} \\
&= F(q') - \frac{\lambda_2}{2}\|q' - q\|^2_{L_1(d\theta)}.
\end{aligned} \tag{10}$$

For the inequality we used Lemma A and for the last equality we used $q \propto \exp\left(-\frac{H(\theta)+\lambda_1\|\theta\|_2^2}{\lambda_2}\right)$. Therefore, we conclude that q is a minimizer of F on $\mathcal{P}_2$ and the strong convexity of F holds at q with respect to $L_1(d\theta)$-norm. This crucial property is used in the proof of Theorem 1.

A.2 Log-Sobolev and Talagrand's Inequalities

The log-Sobolev inequality is useful in establishing the convergence rate of Langevin algorithm.

Definition A (Log-Sobolev inequality). Let $d\mu = p(\theta)d\theta$ be a probability distribution with a positive smooth density $p > 0$ on $\mathbb{R}^p$. We say that μ satisfies the log-Sobolev inequality with constant $\alpha > 0$ if for any smooth function $f : \mathbb{R}^p \to \mathbb{R}$,

$$\mathbb{E}_\mu[f^2\log f^2] - \mathbb{E}_\mu[f^2]\log \mathbb{E}_\mu[f^2] \le \frac{2}{\alpha}\mathbb{E}_\mu[\|\nabla f\|_2^2].$$

This inequality is analogous to strong convexity in optimization. Let $d = q( )d$ be a probability distribution on $\mathbb{R}^p$ such that q is smooth and positive. Then, if satisfies the log-Sobolev inequality with , it follows that

$$\mathrm{KL}(\|) \quad \frac{1}{2} E [\|\nabla \quad \log q\|_2^2]:$$

The above relation is directly obtained by setting $f = \sqrt{q}$ in the definition of log-Sobolev inequality. Note that the right hand side is nothing else but the squared norm of functional gradient of $\mathrm{KL}(\|)$ with respect to a transport map for .

It is well-known that strong log-concave densities satisfy the LSI with a dimension-free constant (up to the spectral norm of the covariance).

Example 2 (Bakry and Émery (1985)). Let $q \propto \exp(f)$ be a probability density, where $f : \mathbb{R}^p \to \mathbb{R}$ is a smooth function. If there exists $c > 0$ such that $\nabla^2 f \quad cI_p$, then $q( )d$ satisfies Log-Sobolev inequality with constant c.

In addition, the LSI is preserved under bounded perturbation, as originally shown in Holley and Stroock (1987). We also provide a proof for completeness.

Lemma B (Holley and Stroock (1987)). Let $q( )d$ be a probability distribution on $\mathbb{R}^p$ satisfying the log-Sobolev inequality with a constant . For a bounded function $B : \mathbb{R}^p \to \mathbb{R}$, we define a probability distribution $q_B( )d$ as follows:

$$q_B( )d \quad = \quad \frac{\exp(B( ))q( )}{E_q[\exp(B( ))]} d :$$

Then, $q_B d$ satisfies the log-Sobolev inequality with a constant $= \exp(4\|B\|_\infty)$.

Proof. Taking an expectation E_{q_B} of the Bregman divergence defined by a convex function $x \log x$, for $\forall a > 0$,

$$\begin{aligned} 0 \quad & E_{q_B} \quad f^2( )\log(f^2( )) \quad (a\log(a) + (\log(a) + 1)(f^2( ) \quad a)) \\ = \ & E_{q_B} \quad f^2( )\log(f^2( )) \quad (f^2( )\log(a) + f^2( ) \quad a) \ : \end{aligned}$$

Since the minimum is attained at $a = E_{q_B}[f^2( )]$,

$$\begin{aligned} 0 \quad & E_{q_B} \quad f^2( )\log(f^2( )) \quad E_{q_B}[f^2( )]\log E_{q_B}[f^2( )] \\ = \ & \inf_{a>0} E_{q_B} \quad f^2( )\log(f^2( )) \quad (f^2( )\log(a) + f^2( ) \quad a) \\ & \exp(2\|B\|_\infty) \inf_{a>0} E_q \quad f^2( )\log(f^2( )) \quad (f^2( )\log(a) + f^2( ) \quad a) \\ = \ & \exp(2\|B\|_\infty) \quad E_q \quad f^2( )\log(f^2( )) \quad E_q[f^2( )]\log E_q[f^2( )] \\ & \frac{2\exp(2\|B\|_\infty)}{} E_q \quad \|\nabla \ f\|_2^2 \\ = \ & \frac{2\exp(2\|B\|_\infty)}{} E_{q_B} \quad \frac{E_q[\exp(B( ))]}{\exp(B( ))} \|\nabla \ f\|_2^2 \\ & \frac{2\exp(4\|B\|_\infty)}{} E_{q_B} \quad \|\nabla \ f\|_2^2 \ ; \end{aligned}$$

where we used the non-negativity of the integrand for the second inequality. □

We next introduce Talagrand's inequality.

Definition B (Talagrand's inequality). We say that a probability distribution $q( )d$ satisfies Talagrand's inequality with a constant > 0 if for any probability distribution $q'( )d$,

$$\frac{}{2} W_2^2(q', q) \quad \mathrm{KL}(q'\|q);$$

where $W_2(q', q)$ denotes the 2-Wasserstein distance between $q( )d$ and $q'( )d$.

The next theorem gives a relationship between KL divergence and 2-Wasserstein distance.

Theorem C (Otto and Villani (2000)). If a probability distribution $q( )d$ satisfies the log-Sobolev inequality with constant > 0, then $q( )d$ satisfies Talagrand's inequality with the same constant.

B Proof of Main Results

B.1 Extension of Algorithm

In this section, we prove the main theorem that provides the convergence rate of the dual averaging method. We first introduce a slight extension of PDA (Algorithm 1) which incorporates two different initializations at each outer loop step. We refer to the two versions as the *warm-start* and the *resampled* update, respectively. Note that Algorithm 1 in the main text only includes the warm-start update. In Appendix C we provide particle complexity analysis for both updates. We remark that the benefit of resampling strategy is the simplicity of estimation of approximation error $|h_x^{(t)} - h_{q^{(t)}}(x_t)|$, because $h_x^{(t)}$ is composed of i.i.d particles and a simple concentration inequality can be applied to estimate this error.

```
Algorithm 3 Particle Dual Averaging (general version)
  Input: data distribution $\mathcal{D}$, initial density $q^{(1)}$, number of outer-iterations $T$, learning rates $\{\eta_t\}_{t=1}^T$,
  number of inner-iterations $\{T_t\}_{t=1}^T$
  Randomly draw i.i.d. initial parameters $\tilde\theta_r^{(1)} \sim q^{(1)}(\theta)d\theta$ $(r \in \{1,2,\ldots,M\})$
  $\tilde\Theta^{(1)} \leftarrow \{\tilde\theta_r^{(1)}\}_{r=1}^M$
  for $t = 1$ to $T$ do
    Randomly draw a data $(x_t, y_t)$ from $\mathcal{D}$
    Either $\Theta^{(1)} = \{\theta_r^{(1)}\}_{r=1}^M \leftarrow \tilde\Theta^{(t)}$ (warm-start)
    Or randomly initialize $\Theta^{(1)}$ from $q^{(1)}(\theta)d\theta$ (resampling)
    for $k = 1$ to $T_t$ do
      Run an inexact noisy gradient descent for $r \in \{1,2,\ldots,M\}$
      $\nabla_\theta \bar g^{(t)}(\theta_r^{(k)}) \leftarrow \frac{2}{\lambda_2(t+2)(t+1)}\sum_{s=1}^t s\,\partial_z\ell(h_{\tilde\Theta^{(s)}}(x_s), y_s)\,\partial_\theta h(\theta_r^{(k)}; x_s) + \frac{2\lambda_1 t}{\lambda_2(t+2)}\theta_r^{(k)}$
      $\theta_r^{(k+1)} \leftarrow \theta_r^{(k)} - \eta_t \nabla_\theta \bar g^{(t)}(\theta_r^{(k)}) + \sqrt{2\eta_t}\,\xi_r^{(k)}$ (i.i.d. Gaussian noise $\xi_r^{(k)} \sim \mathcal{N}(0, I_p)$)
    end for
    $\tilde\Theta^{(t+1)} \leftarrow \Theta^{(T_t+1)} = \{\theta_r^{(T_t+1)}\}_{r=1}^M$
  end for
  Randomly pick up $t \in \{2,3,\ldots,T+1\}$ following the probability $\mathbb{P}[t] = \frac{2t}{T(T+3)}$ and return $h_{\tilde\Theta^{(t)}}$
```

We also extend the mean field limit (Algorithm 2) to take into account the inexactness in computing $h_{q^{(t)}}(t)$. This relaxation is useful in convergence analysis of Algorithm 3 with resampling because it allows us to regard this method as an instance of the generalized DA method (Algorithm 4) by setting an inexact estimate $h_x^{(t)} = h_{\tilde\Theta^{(t)}}(x_t)$, instead of the exact value of $h_{q^{(t)}}(t)$, which is actually used to defined the potential for which Langevin algorithm run in Algorithm 3. This means convergence analysis of Algorithm 4 (Theorem D) immediately provides a convergence guarantee for Algorithm 3 if the discretization error $|h_x^{(t)} - h_{q^{(t)}}(x_t)|$ can be estimated (as in the resampling scheme).

On the other hands, the convergence analysis of warm-start scheme requires the convergence of mean field limit due to certain technical difficulties, that is, we show the convergence of Algorithm 3 with warm-start by coupling the update with its mean field limit (Algorithm 2) and taking into account the discretization error which stems from finite-particle approximation.

We now present generalized version of the outer loop convergence rate of DA. We highlight the tolerance factor in the generalized assumption (A3') in blue.

Assumption C. Let $\epsilon > 0$ be a given accuracy.

(A1') $\mathcal{Y} \subset [-1, 1]$. $\ell(z, y)$ is a smooth convex function w.r.t. z and $|\partial_z \ell(z, y)| \le 2$ for $y, z \in \mathcal{Y}$ and $\partial_z\ell(\cdot, y)$ is 1-Lipschitz continuous for $y \in \mathcal{Y}$.

(A2') $|h_\theta(x)| \le 1$ and $h(\theta, x)$ is smooth w.r.t. θ for $x \in \mathcal{X}$.

(A3') $\mathrm{KL}(q^{(t+1)} \| q^{(t+1)}) \le 1/t^2$ and $|h_x^{(t)} - h_{q^{(t)}}(x_t)| \le \epsilon$ for $t \ge 1$.

```
Algorithm 4 Dual Averaging (general version)
  Input: data distribution D and initial density $q^{(1)}$
  for $t = 1$ to $T$ do
    Randomly draw a data $(x_t; y_t)$ from D
    Compute an approximation $h_x^{(t)}$ of $h_{q^{(t)}}(x_t)$
    $g^{(t)} \quad @_2`(h_x^{(t)}; y_t)h( \ ; x_t) + \ _1 k \ k_2^2$
    Obtain an approximation $q^{(t+1)}$ of the density function $q^{(t+1)} \ / \ \exp \ \frac{P_{s=1}^{t} \ 2sg^{(s)}}{_2(t+2)( \ t+1)}$
  end for
  Randomly pick up $t \ 2 \ f \ 2; 3; :::; T + 1 \ g$ following the probability $P[t] = \frac{2t}{T(T+3)}$ and return $h_{q^{(t)}}$
```

Remark. The new condition of (A3') allows for inexactness of computing $h_{q^{(t)}}(x_t)$. When showing solely the convergence of the Algorithm 2 which is the exact mean-field limit, the original assumptions (A1), (A2), and (A3) are sufficient, in other words, we can take $= 0$ and Lipschitz continuity of $@_2`(\ ; y)$ in (A1') can be relaxed.

Theorem D (Convergence of general DA). Under Assumptions (A1'), (A2'), and (A3') with $\quad 0$, for arbitrary $q \ \ 2 \ P_2$, iterates of the general DA method (Algorithm 4) satisfies

$$\frac{2}{T(T+3)} \sum_{t=2}^{T+1} t \quad E[L(q^{(t)})] \quad L \ (q \ )$$

$$2 + O \ \frac{1}{T^2} \quad 1 + \ _1 E_q \quad k \ k_2^2 \quad + \frac{_2 e(q \ )}{T} + \frac{2}{T}(1 + \exp(8 = \ _2)) \, p^2 \log^2(T+2) \quad ;$$

where the expectation $E[L(q^{(t)})]$ is taken with respect to the history of examples.

Notation. In the proofs, we use the following notations which are consistent with the description of Algorithm 3 and 4:

$$g^{(t)} = @_2`(h_x^{(t)}; y_t)h(\ ; x_t) + \ _1 k \ k_2^2;$$

$$\bar{g}^{(t)} = \frac{2}{_2(t+2)(t+1)} \sum_{s=1}^{t} sg^{(s)}$$

$$= \frac{2}{_2(t+2)(t+1)} \sum_{s=1}^{t} s@_2`(h_x^{(s)}; y_s)h(\ ; x_s) + \frac{_1 t}{_2(t+2)} k \ \ j_2^2;$$

$$q^{(t+1)} \ / \ \exp \quad \bar{g}^{(t)}$$

$$= \exp \quad \frac{P_{s=1}^{t} \ 2sg^{(s)}}{_2(t+2)(t+1)} \ ! \quad :$$

When considering the resampling scheme, $h_x^{(t)}$ is set to the approximation $h_{\sim(t)}(x_t)$, whereas when considering the warm-start scheme, $h_x^{(t)}$ is set to $h_{q^{(t)}}(x_t)$ with the mean field limit $M \ ! \ 1$ and without tolerance ($= 0$).

B. 2 Auxiliary Lemmas

We introduce several auxiliary results used in the proof of Theorem 1 (Theorem D) and Corollary 1. The following lemma provides a tail bound for Chi-squared variables (Laurent and Massart, 2000).

Lemma C (Tail bound for Chi-squared variable). Let $\quad N \ (0; \ ^2 I_p)$ be a Gaussian random variable on R^p. Then, we get for $c \quad p \ ^2$,

$$P \ k \ k_2^2 \quad 2c \quad \exp \quad \frac{c}{10 \ ^2} \quad :$$

Based on Lemma C, we get the following bound.

Lemma D. Let $\xi \sim N(0, \sigma^2 I_p)$ be Gaussian random variable on $\Theta = \mathbb{R}^p$. Then, we get for $\forall R \geq p\sigma^2$,

$$E\left[\|\xi\|_2^2 1[\|\xi\|_2^2 > 2R]\right] = \frac{1}{Z}\int_{\|\xi\|_2^2 > 2R} \|\xi\|_2^2 \exp\left(-\frac{\|\xi\|_2^2}{2\sigma^2}\right) d\xi \leq 2(R + 10\sigma^2)\exp\left(-\frac{R}{10\sigma^2}\right),$$

where $Z = \int \exp\left(-\frac{\|\xi\|_2^2}{2\sigma^2}\right) d\xi$.

Proof. We set $p(\xi) = \exp(-\|\xi\|_2^2/2\sigma^2)/Z$. Then,

$$\begin{aligned}
\int_{\|\xi\|_2^2 > 2R} \|\xi\|_2^2 p(\xi) d\xi &= \int p(\xi) 1[\|\xi\|_2^2 > 2R] \int_0^\infty 1[\|\xi\|_2^2 > r] dr d\xi \\
&= \int\int_0^\infty p(\xi) 1\left[\|\xi\|_2^2 > \max\{2R, r\}\right] dr d\xi \\
&\leq 2R \int p(\xi) 1\left[\|\xi\|_2^2 > 2R\right] d\xi + \int\int_{2R}^\infty p(\xi) 1\left[\|\xi\|_2^2 > r\right] dr d\xi \\
&= 2R P[\|\xi\|_2^2 > 2R] + \int_{2R}^\infty P[\|\xi\|_2^2 > r] dr \\
&\leq 2R \exp\left(-\frac{R}{10\sigma^2}\right) + \int_{2R}^\infty \exp\left(-\frac{r}{20\sigma^2}\right) dr \\
&\leq 2(R + 10\sigma^2) \exp\left(-\frac{R}{10\sigma^2}\right).
\end{aligned}$$

□

Proposition A (Continuity). Let $q_\theta(\xi) \propto \exp\left(-H_\theta(\xi) - \lambda\|\xi\|_2^2\right)$ $(\lambda > 0)$ be a density on $\mathbb{R}^p$ such that $\|H_\theta\|_\infty \leq c$. Then, for $\forall \gamma > 0$ and a density $\forall q \in \mathcal{P}_2$,

$$\int \|\xi\|_2^2 (q - q_\theta)(\xi) d\xi \leq \frac{(2 + \gamma + 1/\gamma)\exp(4c)}{\lambda} KL(q\|q_\theta) + \frac{(1+\gamma)p\exp(2c)}{2\lambda},$$

$$\begin{aligned}
\int q(\xi)\log(q(\xi))d\xi - \int q_\theta(\xi)\log(q_\theta(\xi))d\xi &\leq (1 + (2 + \gamma + 1/\gamma)\exp(4c)) KL(q\|q_\theta) + c\sqrt{2KL(q\|q_\theta)} \\
&\quad + \frac{(1+\gamma)p\exp(2c)}{2}.
\end{aligned}$$

Proof. Let π be an optimal coupling between $q d\xi$ and $q_\theta d\xi$. Using Young's inequality, we have

$$\begin{aligned}
\int \|\xi\|_2^2 q(\xi) d\xi &= \int \|\xi\|_2^2 d\pi(\xi, \xi') \\
&= \int \left(\|\xi - \xi'\|_2^2 + \|\xi'\|_2^2 + 2(\xi - \xi')^\top \xi'\right) d\pi(\xi, \xi') \\
&\leq \int \left(\|\xi - \xi'\|_2^2 + \|\xi'\|_2^2 + \frac{1}{\gamma}\|\xi - \xi'\|_2^2 + \gamma\|\xi'\|_2^2\right) d\pi(\xi, \xi') \\
&= (1 + 1/\gamma)\int \|\xi - \xi'\|_2^2 d\pi(\xi, \xi') + (1+\gamma)\int \|\xi'\|_2^2 q_\theta(\xi') d\xi' \\
&= (1 + 1/\gamma) W_2^2(q, q_\theta) + (1+\gamma)\int \|\xi'\|_2^2 q_\theta(\xi') d\xi'. \qquad (11)
\end{aligned}$$

The last term can be bounded as follows:

$$\begin{aligned}
\int \|\xi\|_2^2 q_\theta(\xi) d\xi &= \int \|\xi\|_2^2 \frac{\exp\left(-H_\theta(\xi) - \lambda\|\xi\|_2^2\right)}{\int \exp(-H_\theta(\xi) - \lambda\|\xi\|_2^2) d\xi} d\xi \\
&\leq \exp(2c) \int \|\xi\|_2^2 \frac{\exp\left(-\lambda\|\xi\|_2^2\right)}{\int \exp(-\lambda\|\xi\|_2^2) d\xi} d\xi \\
&= \frac{p\exp(2c)}{2\lambda}, \qquad (12)
\end{aligned}$$

where the last equality comes from the variance of Gaussian distribution.

From (11) and (12),

$$
\begin{aligned}
Z\ k\ k_2^2(q\quad q\ )(\ )d &\quad (1+1=\ )W_2^2(q;q\ )+\ Z\ k\ k_2^2 q\ (\ )d\\
&\quad (1+1=\ )W_2^2(q;q\ )+\frac{p\exp(2c)}{2}:
\end{aligned}
$$

From the symmetry of (11), and applying (11) again with (12),

$$
\begin{aligned}
Z\ k\ k_2^2(q\quad q)(\ )d &\quad (1+1=\ )W_2^2(q;q\ )+\ Z\ k\ k_2^2 q(\ )d\\
&\quad (2+\quad +1=\ )W_2^2(q;q\ )+\ (1+\ )\ Z\ k\ k_2^2 q\ (\ )d\\
&\quad (2+\quad +1=\ )W_2^2(q;q\ )+\frac{(1+\ )p\exp(2c)}{2}:
\end{aligned}
$$

From Lemma B and Example 2, we see q satisfies the log-Sobolev inequality with a constant 2 = exp(4c). As a result, q satisfies Talagrand's inequality with the same constant from Theorem C. Hence, by combining the above two inequalities, we have

$$
\begin{aligned}
Z\ k\ k_2^2(q\quad q\ )(\ )d &\quad (2+\quad +1=\ )W_2^2(q;q\ )+\frac{(1+\ )p\exp(2c)}{2}\\
&\quad \frac{(2+\quad +1=\ )\exp(4c)}{}KL(\ qkq\ )+\frac{(1+\ )p\exp(2c)}{2}
\end{aligned}
$$

Therefore, we know that

$$
\begin{aligned}
&Z\ q(\ )\log(q(\ ))d \qquad Z\ q\ (\ )\log(q\ (\ ))d\\
&KL(\ qkq\ )+\ Z\ (q\quad q)(\ )\quad H(\ )+\ k\ k_2^2\ d\\
&KL(\ qkq\ )+\ ckq\quad q\ k_{L^1(d\ )}+(2+\quad +1=\ )\exp(4c)KL(\ qkq\ )+\frac{(1+\ )p\exp(2c)}{2}\\
&KL(\ qkq\ )+\ c\ \sqrt[p]{2KL(\ qkq\ )}+(2+\quad +1=\ )\exp(4c)KL(\ qkq\ )+\frac{(1+\ )p\exp(2c)}{2}:
\end{aligned}
$$

where we used Pinsker's theorem for the last inequality. This finishes the proof. □

Proposition B (Maximum Entropy). Let q ( ) / exp H ( ) k k_2^2 (> 0) on R^p be a density such that $kH k_1$ c. Then,

$$
E_q\ [\log(q\ )]\quad 2c+\frac{p}{2}\ \exp(2c)+\log\ \ -\ \ :
$$

Proof. It follows that

$$
\begin{aligned}
E_q\ [\log(q\ )] &= E_q\ [H(\ )+\ k\ k_2^2]+\log\ Z\ \exp(\ H(\ )\quad k\ k_2^2)d\\
&\quad c+\ E_q\ [k\ k_2^2]+\log\ Z\ \exp(c\quad k\ k_2^2)d\\
&= 2\,c+\ E_q\ [k\ k_2^2]+\log\ Z\ \exp(\ k\ k_2^2)d\\
&\quad 2c+\frac{p\exp(2c)}{2}+\frac{p}{2}\log\ -\ ;
\end{aligned}
$$

where we used (12) and Gaussian integral for the last inequality. □

Proposition C (Boundedness of KL-divergence). Let $q\ (\ ) /\ \exp\ \ H\ (\ )\ \ \ k\ k_2^2\ \ (\ \ > 0)$ be a density on R^p such that $kH\ k_1\ \ c$, and $q_]\ (\ ) /\ \exp\ \ H_]\ (\ )\ \ {}_]k\ k_2^2\ \ (\ {}_] > 0)$ be a density on R^p such that $kH_] k_1\ \ c_]$. Then, for any density q,

$$KL(\ qkq\ )\ \ \ 4c\ \ + 2\,c_]\ + \frac{3}{2}\ \ 1 + \frac{}{]}\ \ p\exp(2c_])+\frac{p}{2}\log\ \frac{]}{}$$
$$+\ \ 1 + 4\ \ 1 + \frac{}{]}\ \ \exp(4c_])\ \ KL(\ qkq_])+\ c_]\ \sqrt{2KL(\ qkq_])}:$$

Proof. Applying Proposition A with $\ = 1$,

$$KL(\ qkq\ ) = \int q(\ )\log\ \frac{q(\ )}{q\ (\ )}\ \ d$$
$$= \int q_]\ (\ )\log\ \frac{q_]\ (\ )}{q\ (\ )}\ \ d\ \ + \int (q_]\ (\ )\ \ \ q(\ ))\log(q\ (\ ))d$$
$$+ \int q(\ )\log(q(\ ))d\ \ \ \ \int q_]\ (\ )\log(q_]\ (\ ))d$$
$$\int q_]\ (\ )\log\ \frac{q_]\ (\ )}{q\ (\ )}\ \ d\ \ + \int (q(\ )\ \ \ q_]\ (\ ))(H\ (\ )+\ \ \ k\ k_2^2)d$$
$$+(1+4\exp(4\ c_]))KL(\ qkq_])+\ c_]\ \sqrt{2KL(\ qkq_])+\ p\exp(2c_])}$$
$$\int q_]\ (\ )\log\ \frac{q_]\ (\ )}{q\ (\ )}\ \ d\ \ + 2\,c\ \ + \frac{4\ \ \exp(4c_])}{]}KL(\ qkq_])+\frac{p\ \ \ \exp(2c_])}{]}$$
$$+(1+4\exp(4\ c_]))KL(\ qkq_])+\ c_]\ \sqrt{2KL(\ qkq_])+\ p\exp(2c_])}:$$

We next bound the first term in the last equation as follows.

$$\int q_]\ (\ )\log\ \frac{q_]\ (\ )}{q\ (\ )}\ \ d\ \ = \int q_]\ (\ )\log\ \frac{\exp(\ \ H_]\ (\ )\ \ \ \ {}_]k\ k_2^2)}{\exp(\ \ H\ (\ )\ \ \ \ k\ k_2^2)}\ \ d\ \ +\log\ \frac{\int \exp(\ \ H\ (\ )\ \ \ \ k\ k_2^2)d}{\int \exp(\ \ H_]\ (\ )\ \ \ {}_]k\ k_2^2)d}$$
$$= \int q_]\ (\ )\ \ H\ (\ )\ \ \ H_]\ (\ )+(\ \ \ \ \ {}_])k\ k_2^2\ \ d$$
$$+\log\int \exp(\ \ H\ (\ )\ \ \ \ k\ k_2^2)d\ \ \ \ \log\int \exp(\ \ H_]\ (\ )\ \ \ {}_]k\ k_2^2)d$$
$$c\ \ + c_]\ + \frac{1}{2}\ \ 1 + \frac{}{]}\ \ p\exp(2c_])$$
$$+\log\int \exp(c\ \ \ \ \ \ k\ k_2^2)d\ \ \ \ \log\int \exp(\ \ c_]\ \ \ \ {}_]k\ k_2^2)d$$
$$2c\ \ + 2\,c_]\ + \frac{1}{2}\ \ 1 + \frac{}{]}\ \ p\exp(2c_])+\frac{p}{2}\log\ \frac{]}{}\ \ ;$$

where for the first inequality we used a similar inequality as in (12) and for the second inequality we used the Gaussian integral. Hence, we get

$$KL(\ qkq\ )\ \ \ 4c\ \ + 2\,c_]\ + \frac{3}{2}\ \ 1 + \frac{}{]}\ \ p\exp(2c_])+\frac{p}{2}\log\ \frac{]}{}$$
$$+\ \ 1 + 4\ \ 1 + \frac{}{]}\ \ \exp(4c_])\ \ KL(\ qkq_])+\ c_]\ \sqrt{2KL(\ qkq_])}:$$

□

Lemma E. Suppose Assumption (A1') and (A2') hold. If $KL(\ q^{(t)}kq^{(t)})\ \ \ \frac{1}{t^2}$ for $t\ \ \ 2$, then

$$t\ \ \ \int g^{(t)}(\ )(q^{(t)}(\ )\ \ \ \ q^{(t)}(\ ))d\ \ \ ;\ \ {}_2t\ \ e(q^{(t)})\ \ \ \ e(q^{(t)})\ \ = O(1+\ \ {}_2+\ p\ \ {}_2\exp(8=\ {}_2)):$$

Proof. Recall the definition of $g^{(t)}$; $\overline{g}^{(t)}$ and $q^{(t)}$ (see notations in subsection B.1). We set ${}_{t+1} = \frac{\sum_{s=1}^{t} s}{2\sum_{s=1}^{t+1} s} = \frac{t}{2(t+2)}$. Note that for $t \ 1$,

$$2 + {}_1\| \|_2^2 \quad g^{(t)}(\ ) \quad 2 + {}_1\| \|_2^2; \tag{13}$$

$${}_{t+1}(\ 2 + {}_1\| \|_2^2) \quad \overline{g}^{(t)}(\ ) \quad {}_{t+1}(2 + {}_1\| \|_2^2); \tag{14}$$

$$\frac{1}{3\ {}_2} \quad {}_{t+1} \quad \frac{1}{{}_2}: \tag{15}$$

Therefore, we have for $t \ 2$ from Proposition A with $= 1/t < 1$,

$$t\int g^{(t)}(\ )(q^{(t)}(\ ) \ q^{(t)}(\ ))d$$

$$2t\|q^{(t)} \ q^{(t)}\|_{L_1(d\ )} + {}_1 t\int \| \|_2^2(q^{(t)}(\ ) \ q^{(t)}(\ ))d$$

$$2t\sqrt{2KL(q^{(t)}\|q^{(t)})}$$

$$+ {}_2(t+2) \ (2 + \ + 1/\ )\exp(8/{}_2)KL(q^{(t)}\|q^{(t)}) + \frac{(1+\ )\sqrt{\ }\exp(4/{}_2)}{2}$$

$$2\sqrt{2} + 3\ {}_2(4\exp(8/{}_2) + \sqrt{\ }\exp(4/{}_2))$$

$$= O(1 + \sqrt{\ }{}_2\exp(8/{}_2)):$$

Moreover, we have for $t \ 2$,

$${}_2 t \ e(q^{(t)}) \ e(q^{(t)})$$

$${}_2 t \ (1 + (2 + \ + 1/\ )\exp(8/{}_2))KL(q^{(t)}\|q^{(t)}) + \frac{2}{{}_2}\sqrt{2KL(q^{(t)}\|q^{(t)})} + \frac{(1+\ )\sqrt{\ }\exp(4/{}_2)}{2}$$

$${}_2 t \ (1 + (3 + \ t)\exp(8/{}_2))\frac{1}{(t \ 1)^2} + \frac{2\sqrt{2}}{{}_2(t \ 1)} + \frac{\sqrt{\ }\exp(4/{}_2)}{t}\Bigg)$$

$$= O(1 + {}_2 + \sqrt{\ }{}_2\exp(8/{}_2)):$$

This finishes the proof. □

B.3 Outer Loop Complexity

Based on the auxiliary results and the convex optimization theory developed in Nesterov (2009); Xiao (2009), we now prove Theorem D which is an extension of Theorem 1.

Proof of Theorem D. For $t \ 1$ we define,

$$V_t(q) = E_q\left[\sum_{s=1}^{t} s g^{(s)}\right] \ {}_2 e(q)\sum_{s=1}^{t+1} s:$$

From the definition, the density $q^{(t+1)} \in P_2$ calculated in Algorithm 4 maximizes $V_t(q)$. We denote $V_t = V(q^{(t+1)})$. Then, for $t \ 2$, we get

$$V_t = E_{q^{(t+1)}}\left[\sum_{s=1}^{t \ 1} s g^{(s)}\right] \ {}_2 e(q^{(t+1)})\sum_{s=1}^{t} s \quad E_{q^{(t+1)}}\left[t g^{(t)}\right] \ {}_2(t+1)e(q^{(t+1)})$$

$$V_{t \ 1} \ \frac{{}_2\sum_{s=1}^{t} s}{2}\|q^{(t+1)} \ q^{(t)}\|_{L_1(d\ )}^2 \quad E_{q^{(t)}}\left[t g^{(t)}\right] \ {}_2(t+1)e(q^{(t+1)})$$

$$+ t\int(q^{(t)} \ q^{(t+1)})(\ )g^{(t)}(\ )d$$

$$V_{t \ 1} \ \frac{{}_2\sum_{s=1}^{t} s}{2}\|q^{(t+1)} \ q^{(t)}\|_{L_1(d\ )}^2 \quad E_{q^{(t)}}\left[t g^{(t)}\right] \ {}_2(t+1)e(q^{(t+1)})$$

$$+ \ t \int g^{(t)}(\ )(q^{(t)}(\ ) \quad q^{(t)}(\ ))d \quad + \ t \int (q^{(t)} \quad q^{(t+1)})(\ )g^{(t)}(\ )d$$

$$V_{t \ 1} \quad \frac{\ _2 \sum_{s=1}^{t} \ _s}{2} kq^{(t+1)} \quad q^{(t)} k^2_{L_1(d \ )} \quad E_{q^{(t)}} \left[tg^{(t)} \right] \quad _2(t+1)\, e(q^{(t+1)} \,)$$

$$+ \ t \int (q^{(t)} \quad q^{(t+1)})(\ )g^{(t)}(\ )d \quad + O(1 + \ _2 + p \ _2 \exp(8= \ _2)); \qquad (16)$$

where for the first inequality we used the optimality of $q^{(t)}$ and the strong convexity (10) of $q^{(t)}$, and for the final inequality we used Lemma E.

We set $R_t = \ \frac{3}{2}p + 15 \ \frac{\ _2}{\ _1} \log(1 + t)$ and also $_{t+1} = \frac{\sum_{s=1}^{t} \ _s}{\ _2 \sum_{s=1}^{t+1} \ _s} = \frac{t}{\ _2(t+2)}$, as done in the proof of Lemma E.

From Assumptions (A1'), (A2') and $q^{(t)} = \exp\left(\frac{\sum_{s=1}^{t-1} \ _s g^{(s)}}{\ _2 \sum_{s=1}^{t} \ _s}\right) \Big/ \int \exp\left(\frac{\sum_{s=1}^{t-1} \ _s g^{(s)}(\ )}{\ _2 \sum_{s=1}^{t} \ _s}\right) d \quad (t \ 2)$, we have for $t \ 2$,

$$q^{(t)}(\ ) \quad \exp(\ _t(2 \quad _1 k \ k_2^2)) = \int \exp(\ _t(\ 2 \quad _1 k \ k_2^2))d$$

$$\exp(4 \ _t) \exp(\quad _t \ _1 k \ k_2^2) = \int \exp(\quad _t \ _1 k \ k_2^2)d$$

$$\exp(4= \ _2) \exp(\quad _t \ _1 k \ k_2^2) = \int \exp(\quad _t \ _1 k \ k_2^2)d \ : \qquad (17)$$

Using (17) and applying Lemma D with $^2 = \frac{1}{2 \ _t \ _1}; \frac{1}{2 \ _{t+1} \ _1}$ and $R = R_t$, we have for $t \ 2$,

$$\int (q^{(t)} \quad q^{(t+1)})(\ )g^{(t)}(\ )d$$

$$2kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )} + \ _1 \int k \ k_2^2 j(q^{(t)} \quad q^{(t+1)})(\ )jd$$

$$(2 + 2 \ _1 R_t)kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )} + \ _1 \int_{k \ k_2^2 > 2R_t} k \ k_2^2 (q^{(t)} + q^{(t+1)})(\ )d$$

$$(2 + 2 \ _1 R_t)kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )} + \ _1 \exp(4= \ _2) \int_{k \ k_2^2 > 2R_t} k \ k_2^2 \frac{\exp(\quad _t \ _1 k \ k_2^2)}{R \exp(\quad _t \ _1 k \ k_2^2)d}d$$

$$+ \ _1 \exp(4= \ _2) \int_{k \ k_2^2 > 2R_t} k \ k_2^2 \frac{\exp(\quad _{t+1} \ _1 k \ k_2^2)}{R \exp(\quad _{t+1} \ _1 k \ k_2^2)d}d$$

$$(2 + 2 \ _1 R_t)kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )} + 2 \ _1 \exp(4= \ _2) \quad R_t + \frac{5}{\ _1 \ _t} \quad \exp \quad \frac{\ _1 R_t \ _t}{5}$$

$$+ 2 \ _1 \exp(4= \ _2) \quad R_t + \frac{5}{\ _1 \ _t} \quad \exp \quad \frac{\ _1 R_t \ _{t+1}}{5}$$

$$(2 + 2 \ _1 R_t)kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )} + 4 \ _1 \exp(4= \ _2) \quad R_t + 15 \frac{\ _2}{\ _1} \quad \exp \quad \frac{R_t \ _1}{15 \ _2}$$

$$2 + 2 \ \frac{3}{2}p + 15 \quad _2 \log(1 + t) \quad kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )} + 8 \exp(4= \ _2) \quad \frac{3}{2}p + 15 \quad \frac{\ _2 \log(1 + t)}{(1 + t)^{1+ \frac{p}{10}}};$$

where for the fifth inequality we used (15) and for the sixth inequality we used $15 \ _2 = \ _1 \quad R_t$.

Applying Young's inequality $ab \quad \frac{a^2}{2} + \frac{b^2}{2}$ with $a = \quad 2 + 2 \ \frac{3}{2}p + 15 \quad _2 \log(1 + t)$, $b = kq^{(t)} \quad q^{(t+1)} k_{L_1(d \ )}$, and $= \frac{\ _2}{2}(t + 1)$, we get

$$\int (q^{(t)} \quad q^{(t+1)})(\ )g^{(t)}(\ )d \qquad \frac{2 + 2 \ \frac{3}{2}p + 15 \quad _2 \log(1 + t) \ ^2}{\ _2(t + 1)} + \frac{\ _2(t + 1)\, kq^{(t)} \quad q^{(t+1)} k^2_{L_1(d \ )}}{4}$$

$$+ 8 \exp(4= \ _2) \quad \frac{3}{2}p + 15 \quad \frac{\ _2 \log(1 + t)}{(1 + t)^{1+ \frac{p}{10}}}: \qquad (18)$$

Combining (16) and (18), we have for t 2,

$$
\begin{aligned}
V_t \quad V_{t\ 1} \quad & E_{q^{(t)}}\left[t g^{(t)} \right] \quad {}_2 (t+1)\, e(q^{(t+1)}) + O(1 + \ {}_2 + p\ {}_2 \exp(8= {}_2)) \\
& + \frac{1}{\ {}_2} \quad 2 + 2 \quad \frac{3}{2}p + 15 \quad {}_2 \log(1+t) \quad ^{2} + 8\exp(4= {}_2) \quad \frac{3}{2}p + 15 \quad \frac{{}_2 \log(1+t)}{(1+t)^{\frac{p}{10}}} \\
& = V_{t\ 1} \quad E_{q^{(t)}}\left[t g^{(t)} \right] \quad {}_2 (t+1)\, e(q^{(t+1)}) + O(1 + \ {}_2 + p\ {}_2 \exp(8= {}_2)) \\
& + O \quad \frac{1}{\ {}_2} + p^2 \ {}_2 \log^2(1+t) + p\ {}_2 \exp(4= {}_2) \\
& = V_{t\ 1} \quad E_{q^{(t)}}\left[t g^{(t)} \right] \quad {}_2 (t+1)\, e(q^{(t+1)}) + O \quad p\ {}_2 \exp(8= {}_2) + p^2 \ {}_2 \log^2(1+t) \\
& = V_{t\ 1} \quad E_{q^{(t)}}\left[t g^{(t)} \right] \quad {}_2 (t+1)\, e(q^{(t+1)}) + O \quad (1 + \exp(8= {}_2))\, p^2 \ {}_2 \log^2(1+t) \\
& = V_{t\ 1} \quad E_{q^{(t)}}\left[t g^{(t)} \right] \quad {}_2 (t+1)\, e(q^{(t+1)}) + \ {}_t ; \qquad (19)
\end{aligned}
$$

where we set ${}_t = O \quad (1 + \exp(8= {}_2))\, p^2 \ {}_2 \log^2(1+t)$.

From Proposition B, (14), and (15),

$$
E_{q^{(t)}}[\log(q^{(t)})] \quad \frac{4}{\ {}_2} + \frac{p}{2} \quad \exp \quad \frac{4}{\ {}_2} \quad + \log \quad \frac{3 \ \ {}_2}{\ {}_1} \quad ;
$$

meaning $e(q^{(t)})$ 0. Hence,

$$
V_1 = \quad E_{q^{(2)}}[g^{(1)}] \quad 3\ {}_2 e(q^{(2)}) \quad 2 \quad 3\ {}_2 e(q^{(2)}) \quad 2 \quad 2\ {}_2 e(q^{(2)}):
$$

Summing the inequality (19) over t 2 f 2; : : : ; T + 1 g,

$$
\begin{aligned}
V_{T+1} \quad 2 \quad 2\ {}_2 e(q^{(2)}) + \sum_{t=2}^{T+1} \Big\{ & E_{q^{(t)}}\left[t g^{(t)} \right] \quad {}_2 (t+1)\, e(q^{(t+1)}) + \ {}_t \Big\} \\
= 2 \quad \sum_{t=2}^{T+1} \Big\{ & t \ E_{q^{(t)}}\left[g^{(t)} \right] + \ {}_2 e(q^{(t)}) \Big\} + \sum_{t=2}^{T+1} \ {}_t \quad {}_2 (T+2)\, e(q^{(T+2)}) \\
2 \quad \sum_{t=2}^{T+1} \Big\{ & t \ E_{q^{(t)}}\left[g^{(t)} \right] + \ {}_2 e(q^{(t)}) \Big\} + \sum_{t=2}^{T+1} \ {}_t ; \qquad (20)
\end{aligned}
$$

where we used ${}_2 t \ e(q^{(t)}) \quad e(q^{(t)}) \ = \ {}_t$ (Lemma E), $2\ {}_t = O(\ {}_t)$, and $e(q^{(T+2)})$ 0.

On the other hand, for 8 q 2 P ${}_2$,

$$
V_{T+1} = \max_{q 2 P\ {}_2} \Big(E_q \Big[\sum_{t=1}^{T+1} t g^{(t)} \Big] \quad {}_2 e(q) \quad \sum_{t=1}^{T+2} t \Big) \quad E_q \Big[\sum_{t=1}^{T+1} t g^{(t)} \Big] \quad {}_2 e(q\ ) \quad \sum_{t=1}^{T+2} t: \qquad (21)
$$

Using (A1'), (A2'), and (A3'), we have for any density function q,

$$
(@ `(h_{q^{(t)}}(x_t); y_t) \quad @ `(h_x^{(t)}; y_t))\, E_q[h(\ ; x_t)] \qquad : \qquad (22)
$$

Hence, from (20), (21), (22), and the convexity of the loss,

$$\begin{aligned}
&\frac{2}{T(T+3)}\sum_{t=2}^{T+1} t\Big\{\ell(h_{q^{(t)}}(x_t);y_t)+\lambda_1\mathbb{E}_{q^{(t)}}[\|\theta\|_2^2]+\lambda_2\mathbb{E}_{q^{(t)}}[\log(q^{(t)})] \\
&\qquad -\ell(h_{q_*}(x_t);y_t)-\lambda_1\mathbb{E}_{q_*}[\|\theta\|_2^2]-\lambda_2\mathbb{E}_{q_*}[\log(q_*)]\Big\} \\
&\le \frac{2}{T(T+3)}\sum_{t=2}^{T+1} t\Big\{\partial_z\ell(h_{q^{(t)}}(x_t);y_t)\left(\mathbb{E}_{q^{(t)}}[h(\theta;x_t)]-\mathbb{E}_{q_*}[h(\theta;x_t)]\right) \\
&\qquad +\lambda_1\left(\mathbb{E}_{q^{(t)}}[\|\theta\|_2^2]-\mathbb{E}_{q_*}[\|\theta\|_2^2]\right)+\lambda_2\left(\mathbb{E}_{q^{(t)}}[\log(q^{(t)})]-\mathbb{E}_{q_*}[\log(q_*)]\right)\Big\} \\
&\le \frac{2}{T(T+3)}\sum_{t=2}^{T+1} t\Big\{2\epsilon+\mathbb{E}_{q^{(t)}}[g^{(t)}]-\mathbb{E}_{q_*}[g^{(t)}]+\lambda_2\left(e(q^{(t)})-e(q_*)\right)\Big\} \\
&\le 2\epsilon+\frac{2}{T(T+3)}\left(-2V_{T+1}+\sum_{t=2}^{T+1}\delta_t-\sum_{t=2}^{T+1}t\left(\mathbb{E}_{q_*}[g^{(t)}]+\lambda_2 e(q_*)\right)\right) \\
&\le 2\epsilon+\frac{2}{T(T+3)}\left(2+\mathbb{E}_{q_*}\left[g^{(1)}\right]+\lambda_2(T+3)e(q_*)+\sum_{t=2}^{T+1}\delta_t\right) \\
&\le 2\epsilon+\frac{2}{T(T+3)}\left(4+\lambda_1\mathbb{E}_{q_*}\left[\|\theta\|_2^2\right]\right)+\frac{2\lambda_2 e(q_*)}{T}+\frac{2}{T}O\left(\frac{(1+\exp(8/\lambda_2))p^2}{\lambda_2}\log^2(T+2)\right).
\end{aligned}$$

Taking the expectation with respect to the history of examples, we have

$$\begin{aligned}
&\frac{2}{T(T+3)}\sum_{t=2}^{T+1} t\left(\mathbb{E}[\mathcal{L}(q^{(t)})]-\mathcal{L}(q_*)\right) \\
&=2\epsilon+O\left(\frac{1}{T^2}\left(1+\lambda_1\mathbb{E}_{q_*}\left[\|\theta\|_2^2\right]\right)+\frac{\lambda_2}{T}\left(e(q_*)+(1+\exp(8/\lambda_2))p^2\log^2(T+2)\right)\right).
\end{aligned}$$

□

B.4 Inner Loop Complexity

We next prove Corollary 1 which gives an estimate of inner loop iteration complexity. This result is derived by utilizing the convergence rate of the Langevin algorithm under LSI developed in Vempala and Wibisono (2019). We here consider the ideal Algorithm 2 (i.e., warm-start and exact mean field limit ($\epsilon = 0$)).

Proof of Corollary 1. We verify the assumptions required in Theorem 2. We recall that $q^{(t+1)}$ takes the form of Boltzmann distribution: for $t \ge 1$,

$$\begin{aligned}
q^{(t+1)} &\propto \exp\left(-\frac{\sum_{s=1}^{t}\beta_s g^{(s)}}{\lambda_2\sum_{s=1}^{t+1}\beta_s}\right) \\
&=\exp\left(-\frac{1}{\lambda_2\sum_{s=1}^{t+1}\beta_s}\sum_{s=1}^{t}\beta_s\partial_z\ell(h_x^{(t)};y_t)h(\theta;x_t)-\frac{\lambda_1 t}{\lambda_2(t+2)}\|\theta\|_2^2\right).
\end{aligned}$$

Note that $\frac{\lambda_1}{\lambda_2}\ge\frac{\lambda_1 t}{\lambda_2(t+2)}\ge\frac{\lambda_1}{3\lambda_2}$ $(t \ge 1)$ and $\left|\frac{1}{\lambda_2\sum_{s=1}^{t+1}\beta_s}\sum_{s=1}^{t}\beta_s\partial_z\ell(h_x^{(t)};y_t)h(\theta;x_t)\right|\le\frac{2t}{\lambda_2(t+2)}\le\frac{2}{\lambda_2}$. Therefore, from Example 2 and Lemma B, we know that $q^{(t+1)}$ satisfies the log-Sobolev inequality with a constant $\frac{2\lambda_1}{3\lambda_2\exp(8/\lambda_2)}$; in addition, the gradient of $\log(q^{(t+1)})$ is $\frac{2}{\lambda_2}(1+\lambda_1)$-Lipschitz continuous. Therefore, from Theorem 2 we deduce that Langevin algorithm with learning rate $\eta_t \le \frac{\lambda_1\lambda_2\delta_{t+1}}{96p(1+\lambda_1)^2\exp(8/\lambda_2)}$ yields q^{t+1} satisfying $\mathrm{KL}(\tilde{q}^{(t+1)}\|q^{(t+1)}) \le \delta_{t+1}$ within $\frac{3\lambda_2\exp(8/\lambda_2)}{2\lambda_1\eta_t}\log\frac{2\mathrm{KL}(\tilde{q}^{(t)}\|q^{(t+1)})}{\delta_{t+1}}$-iterations.

We next bound $\mathrm{KL}(q^{(t)} \| q^{(t+1)})$. Apply Proposition C with $q = q^{(t)}$, $q = q^{(t+1)}$, and $q_{]} = q^{(t)}$. Note that in this setting, constants $c; c_{]};$; and $_{]}$ satisfy

$$c \quad \frac{2}{_{2}}; \ \frac{1}{3 \ _{2}} \qquad \frac{1}{_{2}};$$
$$c_{]} \quad \frac{2}{_{2}}; \ \frac{1}{3 \ _{2}} \quad _{]} \quad \frac{1}{_{2}}:$$

Then, we get

$$\mathrm{KL}(q^{(t)} \| q^{(t+1)}) \quad \frac{12}{_{2}} + 6\,p\,\exp \ \frac{4}{_{2}} \ + \frac{p}{2} \log 3 + \ 1 + 16 \exp \ \frac{8}{_{2}} \quad \mathrm{KL}(q^{(t)} \| q^{(t)})$$
$$+ \frac{2}{_{2}} \sqrt{\overline{2\mathrm{KL}(q^{(t)} \| q^{(t)})}}:$$

Hence, we can conclude $\mathrm{KL}(q^{(t)} \| q^{(t+1)})$ are uniformly bounded with respect to $t \, 2 \, f \, 1; : : : ; T \, g$ as long as $\mathrm{KL}(q^{(t)} \| q^{(t)})$ $_{t}$ and $q^{(1)}$ is a Gaussian distribution. □

Case of resampling. We note that for resampling scheme, the similar inner loop complexity of $O \ \frac{_{2} \exp(8 = \ _{2})}{1 \ \ t} \log \frac{2\mathrm{KL}(\ q^{(1)} \ \| q^{(t+1)} \ )}{t+1}$ can be immediately obtained by replacing the initial distribution of Langevin algorithm with $q^{(1)}(\ )d$. Moreover, the uniform boundedness of $\mathrm{KL}(q^{(1)} \| q^{(t+1)})$ with respect to t is also guaranteed by applying Proposition C with $q = q_{]} = q^{(1)}$ and $q = q^{(t+1)}$ as long as $q^{(1)}(\ )d$ is a Gaussian distribution.

ADDITIONAL RESULTS AND DISCUSSIONS

C Discretization Error of Finite Particles

C. 1 Case of Resampling

As discussed in subsection B. 1, to establish the finite-particle convergence guarantees of Algorithm 3 with resampling up to $O(\ )$-error, we need to show that $h_x^{(t)} = h_{\sim^{(t)}}(x_t)$ satisfies the condition $j h_x^{(t)} \quad h_{q^{(t)}}(x_t) j$ in (A3'). Hence, we are interested in characterizing the discretization error that stems from using finitely many particles.

For the resampling scheme, we can easily derive that the required number of particles is $O(\ ^{2} \log(T = \ ))$ with high probability 1 , because i.i.d. particles are obtained by the Langevin algorithm and Hoeffding's inequality is applicable.

Lemma F (Hoeffding's inequality). Let $Z; Z_1; : : : ; Z_m$ be i.i.d. random variables taking values in $[\ a; a]$ for $a > 0$. Then, for any > 0, we get

$$P \ " \ \frac{1}{M} \sum_{r=1}^{M} Z_r \quad E[Z] \ > \ \# \quad 2 \exp \ \frac{^{2} M}{2a^2} \ :$$

C. 2 Case of Warm-start

We next consider the warm-start scheme. Note that the convergence of PDA with warm-start is guaranteed by coupling it with its mean-field limit $M \ ! \ 1$ and applying Theorem 1 without tolerance (i.e., $= 0$). To analyze the particle complexity, we make an additional assumption regarding the regularity of the loss function and the model.

Assumption D.

(A5) $h(\ ; x)$ is 1-Lipschitz continuous[6] for $8x \, 2 \, X$.

[6] WLOG the Lipschitz constant is set to 1, since the same analysis works for any fixed constant.

Remark. The above regularity assumption is common in the literature and cover many important problem settings in the optimization of two-layer neural network in the mean field regime. Indeed, (A5) is satisfied for two-layer network in Example 1 when the output or input layer is fixed and when the activation function is Lipschitz continuous.

The following proposition shows the convergence of Algorithm 1 to Algorithm 2 as $M \to \infty$.

Proposition D (Finite Particle Approximation). For training examples $\{x_t\}_{t=1}^{T}$ and any example x, define

$$_{T,M} = \max_{\substack{s\in\{1,\ldots,T\}\\ t\in\{1,\ldots,T+1\}}} h_{q^{(t)}}(x_s) \quad h_{\tilde{}^{(t)}}(x_s) \vee h_{q^{(t)}}(x) \quad h_{\tilde{}^{(t)}}(x) .$$

Under (A1'), (A2), (A4), and (A5), if we run PDA (Algorithm 1) on $\tilde{}$ and the corresponding mean field limit DA (Algorithm 2) on q, then with high probability $\lim_{M\to\infty} {}_{T,M} = 0$. Moreover, if we set $_t \quad \frac{2}{2_1}$, $_1 \quad \frac{3}{2}$, and $T_t \quad \frac{3 _2 \log(4)}{(2_1 \quad 1)_t}$, then with probability at least 1 ,

$$_{T,M} \quad 1 + \frac{4}{2_1 \quad 1} \quad \sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}} .$$

Remark. Proposition D together with Corollary 2 imply that under appropriate regularization, a prediction on any point with an -gap from an -accurate solution of the regularized objective (4) can be achieved with high probability by running PDA with warm-start (Algorithm 1) for poly(^{1}) steps using poly(^{1}) particles, where we omit dependence on hyperparameters and logarithmic factors. Note that specific choices of hyper-parameters in Proposition D are consistent with those in Corollary 2. We also remark that under weak regularization (vanishing $_1$), our current derivation suggests that the required particle size could be exponential in the time horizon, due to the particle correlation in the warm-start scheme. Finally, we remark that for the empirical risk minimization, the term $\log(2(T+1)^2/\ )$ could be changed to $\log(2n(T+1)/\ )$ in the obvious way.

Proof of Proposition D. We analyze an error of finite particle approximation for a fixed history of data $\{x_t\}_{t=1}^{T}$. To Algorithm 2 with the corresponding particle dynamics (Algorithm 1), we construct an semi particle dual averaging update, which is an intermediate of these two algorithms. In particular, the semi particle dual averaging method is defined by replacing $h_{\tilde{}^{(t)}}$ in Algorithm 1 with $h_{q^{(t)}}$ for $q^{(t)}$ in Algorithm 2. Let $\tilde{}'^{(t)} = \{\tilde{}'^{(t)}_r\}_{r=1}^{M}$ be parameters obtained in outer loop of the semi particle dual averaging. We first estimate the gap between Algorithm 2 and the semi particle dual averaging.

Note that there is no interaction among $\tilde{}'^{(t)}$; in other words these are i.i.d. particles sampled from $q^{(t)}$, and we can thus apply Hoeffding's inequality (Lemma F) to $h_{\tilde{}'^{(t)}}(x)$ and $h_{\tilde{}'^{(t)}}(x_s)$ $(s \in \{1,\ldots,T\}, t \in \{1,\ldots,T+1\})$. Hence, for $\forall \ > 0$, $\forall s \in \{1,\ldots,T\}$, and $\forall t \in \{1,\ldots,T+1\}$, with the probability at least 1 ,

$$h_{\tilde{}'^{(t)}}(x_s) \quad h_{q^{(t)}}(x_s) = \frac{1}{M}\sum_{r=1}^{M} h_{\tilde{}'^{(t)}_r}(x_s) \quad h_{q^{(t)}}(x_s) \quad \sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}} ; \tag{23}$$

$$h_{\tilde{}'^{(t)}}(x) \quad h_{q^{(t)}}(x) = \frac{1}{M}\sum_{r=1}^{M} h_{\tilde{}'^{(t)}_r}(x) \quad h_{q^{(t)}}(x) \quad \sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}} . \tag{24}$$

We next bound the gap between the semi particle dual averaging and Algorithm 1 sharing a history of Gaussian noises and initial particles. That is, $\tilde{}^{(1)}_r = \tilde{}'^{(1)}_r$. Let $^{(k)} = \{^{(k)}_r\}_{r=1}^{M}$ and $'^{(k)} = \{'^{(k)}_r\}_{r=1}^{M}$ denote inner iterations of these methods.

(i) Here we show the first statement of the proposition. We set $_1 = 0$ and $\bar{}_1 = 0$. We define $_t$ and $\bar{}_t$ recursively as follows.

$$_{t+1} \overset{\mathrm{def}}{=} \left(1 + \frac{2(1+_1)t_t}{_2(t+2)}\right)^{T_t} \bar{}_t + \frac{t_t}{_2(t+2)}\left(\bar{}_t + \sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\right)\sum_{s=0}^{T_t-1}\left(1 + \frac{2(1+_1)t_t}{_2(t+2)}\right)^{s} ; \tag{25}$$

and $\bar{\ }_{t+1} = \max_{s\in\{1,\dots,t+1\}} {}_s$. We show that for any event where (23) and (24) hold, $\|\tilde{\ }^{(t)}_r \quad \tilde{\bar{\ }}^{(t)}_r\|_2 \quad {}_t$ $(\forall t\in\{1,\dots,T+1\}, \forall r\in\{1,\dots,M\})$ by induction. Suppose $\|\tilde{\ }^{(s)}_r \quad \tilde{\bar{\ }}^{(s)}_r\|_2 \quad {}_s$ $(\forall s\in\{1,\dots,t\}, \forall r\in\{1,\dots,M\})$ holds. Then, for any x and $s\in\{1,\dots,t\}$

$$|h_{\tilde{\ }^{(s)}}(x) \quad h_{\tilde{\bar{\ }}^{(s)}}(x)| \quad \frac{1}{M}\sum_{r=1}^{M} \quad h(\tilde{\ }^{(s)}_r;x) \quad h(\tilde{\bar{\ }}^{(s)}_r;x) \quad \frac{1}{M}\sum_{r=1}^{M} \quad \tilde{\ }^{(s)}_r \quad \tilde{\bar{\ }}^{(s)}_r \quad {}_2 \quad {}_s. \tag{26}$$

Consider the inner loop at the outer step. Then, for an event where (23) holds,

$$\begin{aligned}
&\| {}^{(k+1)}_r \quad \bar{\ }^{(k+1)}_r\|_2 \\
&\quad {}^{(k)}_r \quad \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s \quad \partial_1\ell(h_{\tilde{\ }^{(s)}}(x_s);y_s)\partial h({}^{(k)}_r;x_s) + 2\ {}_1\ {}^{(k)}_r \\
&\quad \bar{\ }^{(k)}_r + \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s \quad \partial_1\ell(h_{\bar{q}^{(s)}}(x_s);y_s)\partial h(\bar{\ }^{(k)}_r;x_s) + 2\ {}_1\ \bar{\ }^{(k)}_r \quad {}_2 \\
&\quad 1 + \frac{2\ {}_1 t\ {}_t}{{}_2(t+2)} \quad \| {}^{(k)}_r \quad \bar{\ }^{(k)}_r\|_2 \\
&+ \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s\|\partial_1\ell(h_{\tilde{\ }^{(s)}}(x_s);y_s)\partial h({}^{(k)}_r;x_s) \quad \partial_1\ell(h_{\bar{q}^{(s)}}(x_s);y_s)\partial h(\bar{\ }^{(k)}_r;x_s)\|_2 \\
&\quad 1 + \frac{2\ {}_1 t\ {}_t}{{}_2(t+2)} \quad \| {}^{(k)}_r \quad \bar{\ }^{(k)}_r\|_2 \\
&+ \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s \quad (\partial_1\ell(h_{\tilde{\ }^{(s)}}(x_s);y_s) \quad \partial_1\ell(h_{\bar{q}^{(s)}}(x_s);y_s))\partial h({}^{(k)}_r;x_s) \quad {}_2 \\
&+ \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s \quad \partial_1\ell(h_{\bar{q}^{(s)}}(x_s);y_s)(\partial h(\bar{\ }^{(k)}_r;x_s) \quad \partial h({}^{(k)}_r;x_s)) \quad {}_2 \\
&\quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad \| {}^{(k)}_r \quad \bar{\ }^{(k)}_r\|_2 + \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s \quad h_{\tilde{\ }^{(s)}}(x_s) \quad h_{\bar{q}^{(s)}}(x_s) \\
&\quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad \| {}^{(k)}_r \quad \bar{\ }^{(k)}_r\|_2 + \frac{2\ {}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t} s \quad {}_s + \sqrt{\frac{2}{M}\log\frac{2(T+1)}{\ }}\ {}^2 \Bigg)\\
&\quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad \| {}^{(k)}_r \quad \bar{\ }^{(k)}_r\|_2 + \frac{t\ {}_t}{{}_2(t+2)} \quad \bar{\ }_t + \sqrt{\frac{2}{M}\log\frac{2(T+1)}{\ }}\ {}^2 \Bigg) \quad :
\end{aligned}$$

Expanding this inequality,

$$\begin{aligned}
&\|\tilde{\ }^{(t+1)}_r \quad \tilde{\bar{\ }}^{(t+1)}_r\|_2 \\
&\quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad ^{T_t} \quad \bar{\ }_t + \frac{t\ {}_t}{{}_2(t+2)} \quad \bar{\ }_t + \sqrt{\frac{2}{M}\log\frac{2(T+1)}{\ }}\ {}^2 \Bigg) \sum_{s=0}^{T_t\ 1} \quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad ^s \\
&= \quad {}_{t+1}:
\end{aligned}$$

Hence, $\|\tilde{\ }^{(t)}_r \quad \tilde{\bar{\ }}^{(t)}_r\|_2 \quad \bar{\ }_{T+1}$ for $\forall t\in\{1,\dots,T+1\}$.

Noting that $\bar{\ }_1 = 0$ and

$$ {}_{t+1} = \quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad ^{T_t} + \frac{t\ {}_t}{{}_2(t+2)} \sum_{s=0}^{T_t\ 1} \quad 1 + \frac{2(1+\ {}_1)t\ {}_t}{{}_2(t+2)} \quad ^s \Bigg) \quad \bar{\ }_t$$

$$+\frac{t\,{}_t}{{}_2(t+2)}\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\sum_{s=0}^{T\ \ 1}\left(1+\frac{2(1+{}_1)t\,{}_t}{{}_2(t+2)}\right)^s;$$

we see $\bar{}_{T+1}\to 0$ as $M\to+\infty$. Then, the proof is finished because for $\forall t\in\{1,\dots,T+1\}$ and $\forall s\in\{1,\dots,T\}$ with high probability 1 ,

$$\begin{aligned}
h_{\tilde{}(t)}(x_s)\quad h_{q(t)}(x_s)\ &|h_{\tilde{}(t)}(x_s)\quad h_{\tilde{\alpha}(t)}(x_s)|+|h_{\tilde{\alpha}(t)}(x_s)\quad h_{q(t)}(x_s)|\\
&\bar{}_{T+1}+\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\ ;\\
h_{\tilde{}(t)}(\bar{x})\quad h_{q(t)}(\bar{x})\ &|h_{\tilde{}(t)}(\bar{x})\quad h_{\tilde{\alpha}(t)}(\bar{x})|+|h_{\tilde{\alpha}(t)}(\bar{x})\quad h_{q(t)}(\bar{x})|\\
&\bar{}_{T+1}+\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\ :
\end{aligned}$$

(ii) We next show the second statement of the proposition. We change the definition (25) of ${}_{t+1}$ as follows:

$${}_{t+1}\overset{\text{def}}{=}\frac{3}{4}\bar{}_t+\frac{1}{2\,{}_1\quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\ :$$

We prove that for any event where (23) and (24) hold, $\|\tilde{}^{(t)}_r\quad\tilde{\alpha}^{(t)}_r\|_2\quad{}_t$ ($\forall t\in\{1,\dots,T+1\}$, $\forall r\in\{1,\dots,M\}$) by induction. Suppose $\|\tilde{}^{(s)}_r\quad\tilde{\alpha}^{(s)}_r\|_2\quad{}_s$ ($\forall s\in\{1,\dots,t\}$, $\forall r\in\{1,\dots,M\}$) holds. Consider the inner loop at t-step. Note that ${}_t\ \frac{\ 2}{2\,{}_1}$ implies $1\ \frac{2\,{}_1t\,{}_t}{{}_2(t+2)}>0$. Therefore, by the similar argument as above, we get

$$\begin{aligned}
&\|{}^{(k+1)}_r\quad\alpha^{(k+1)}_r\|_2\\
&\quad{}^{(k)}_r\quad\frac{2\,{}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t}{}_s\,\partial_1\ell(h_{\tilde{}(s)}(x_s);y_s)\partial h({}^{(k)}_r;x_s)+2\,{}_1\,{}^{(k)}_r\\
&\quad{}^{(k)}_r+\frac{2\,{}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t}{}_s\,\partial_1\ell(h_{q(s)}(x_s);y_s)\partial h(\alpha^{(k)}_r;x_s)+2\,{}_1\,\alpha^{(k)}_r\qquad{}_2\\
&\quad 1\quad\frac{2\,{}_1t\,{}_t}{{}_2(t+2)}\quad\|{}^{(k)}_r\quad\alpha^{(k)}_r\|_2\\
&+\frac{2\,{}_t}{{}_2(t+2)(t+1)}\sum_{s=1}^{t}{}_s\|\partial_1\ell(h_{\tilde{}(s)}(x_s);y_s)\partial h({}^{(k)}_r;x_s)\quad\partial_1\ell(h_{q(s)}(x_s);y_s)\partial h(\alpha^{(k)}_r;x_s)\|_2\\
&\quad\left(1+\frac{(1\quad 2\,{}_1)t\,{}_t}{{}_2(t+2)}\right)\|{}^{(k)}_r\quad\alpha^{(k)}_r\|_2+\frac{t\,{}_t}{{}_2(t+2)}\left(\bar{}_t+\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\right):
\end{aligned}$$

Expanding this inequality,

$$\begin{aligned}
&\|\tilde{}^{(t+1)}_r\quad\tilde{\alpha}^{(t+1)}_r\|_2\\
&\quad\left(1+\frac{(1\quad 2\,{}_1)t\,{}_t}{{}_2(t+2)}\right)^{T_t}\bar{}_t+\frac{t\,{}_t}{{}_2(t+2)}\left(\bar{}_t+\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\right)\sum_{s=0}^{T_t\ \ 1}\left(1+\frac{(1\quad 2\,{}_1)t\,{}_t}{{}_2(t+2)}\right)^s\\
&\quad\left(\left(1+\frac{(1\quad 2\,{}_1)t\,{}_t}{{}_2(t+2)}\right)^{T_t}+\frac{1}{2\,{}_1\quad 1}\right)\bar{}_t+\frac{1}{2\,{}_1\quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\\
&\quad\left(\left(1+\frac{(1\quad 2\,{}_1)t\,{}_t}{{}_2(t+2)}\right)^{T_t}+\frac{1}{2}\right)\bar{}_t+\frac{1}{2\,{}_1\quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)^2}{}}\ ;
\end{aligned}$$

where we used $0<1+\frac{(1\quad 2\,{}_1)t\,{}_t}{{}_2(t+2)}<1$ and ${}_1\quad\frac{3}{2}$.

Noting that $(1 \quad x)^{1=x} \quad \exp(\quad 1)$ for $8x \; 2 \; (0;1]$, we see that

$$1 \quad \frac{(2 \;_1 \quad 1)t \;_t}{_2(t+2)}^{T_t} \quad 1 \quad \frac{(2 \;_1 \quad 1)t \;_t}{_2(t+2)}^{\frac{3 \;\; _2}{(2 \;_1 \;\; 1) \;_t}\log(4)}$$

$$= \quad 1 \quad \frac{(2 \;_1 \quad 1)t \;_t}{_2(t+2)}^{\frac{_2(t+2)}{(2 \;_1 \;\; 1)\,t \;_t}\frac{3t}{t+2}\log(4)}$$

$$\exp \quad \frac{3t}{t+2}\log(4)$$

$$\exp(\quad \log(4))$$

$$= \frac{1}{4};$$

where we used $T_t \quad \frac{3 \;_2 \log(4)}{(2 \;_1 \;\; 1) \;_t}$. Hence, we know that for t,

$$k\tilde{}^{(t+1)}_r \quad \tilde{}^{(t+1)}_r k_2 \quad \frac{3}{4}\bar{}_t + \frac{1}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}} : \qquad (27)$$

This means that $k\tilde{}^{(t+1)}_r \quad \tilde{}^{(t+1)}_r k_2 \quad _{t+1}$ and nishes the induction.

Next, we show

$$\bar{}_t \quad \frac{4}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}} : \qquad (28)$$

This inequality obviously holds for $t = 1$ because $\bar{}_1 = 0$. We suppose it is true for $t \quad T$. Then,

$$_{t+1} = \frac{3}{4}\bar{}_t + \frac{1}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}}$$

$$\frac{4}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}} :$$

Hence, the inequality (28) holds for $8t \; 2 \; f\,1; : : : ; T+1\,g$, yielding

$$k\tilde{}^{(t+1)}_r \quad \tilde{}^{(t+1)}_r k_2 \quad \frac{4}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}} :$$

In summary, it follows that for $8t \; 2 \; f\,1; : : : ; T+1\,g$ and $8s \; 2 \; f\,1; : : : ; T\,g$ with high probability $1 \quad$,

$$h_{\sim(t)}(x_s) \quad h_{q(t)}(x_s) \quad j\; h_{\sim(t)}(x_s) \quad h_{\sim\alpha(t)}(x_s)j + \; h_{\sim\alpha(t)}(x_s) \quad h_{q(t)}(x_s)$$

$$1 + \frac{4}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}} ;$$

$$h_{\sim(t)}(\tilde{x}) \quad h_{q(t)}(\tilde{x}) \quad j\; h_{\sim(t)}(\tilde{x}) \quad h_{\sim\alpha(t)}(\tilde{x})j + \; h_{\sim\alpha(t)}(x_s) \quad h_{q(t)}(x_s)$$

$$1 + \frac{4}{2 \;_1 \quad 1}\sqrt{\frac{2}{M}\log\frac{2(T+1)}{}^{2}} ;$$

where we used (26). This completes the proof. □

D Generalization Bounds for Empirical Risk Minimization

In this section, we give generalization bounds for the problem (3) in the context of **empirical** risk minimization, by using techniques developed by Chen et al. (2020). We consider the smoothed hinge loss and squared loss for binary classification and regression problems, respectively.

D.1 Auxiliary Results

For a set $\mathcal{F}$ of functions from a space $\mathcal{Z}$ to $\mathbb{R}$ and a set $S = \{z_i\}_{i=1}^n \subset \mathcal{Z}$, the empirical Rademacher complexity $\hat{\Re}_S(\mathcal{F})$ is defined as follows:

$$\hat{\Re}_S(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right],$$

where $\sigma = (\sigma_i)_{i=1}^n$ are i.i.d random variables taking -1 or 1 with equal probability.

We introduce the uniform bound using the empirical Rademacher complexity (see Mohri et al. (2012)).

Lemma G (Uniform bound)**.** Let $\mathcal{F}$ be a set of functions from $\mathcal{Z}$ to $[-C, C]$ ($C \in \mathbb{R}$) and $\mathcal{D}$ be a distribution over $\mathcal{Z}$. Let $S = \{z_i\}_{i=1}^n \subset \mathcal{Z}$ be a set of size n drawn from $\mathcal{D}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the choice of S, we have

$$\sup_{f\in\mathcal{F}}\left\{\mathbb{E}_{Z\sim\mathcal{D}}[f(Z)] - \frac{1}{n}\sum_{i=1}^n f(z_i)\right\} \leq 2\hat{\Re}_S(\mathcal{F}) + 3C\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}.$$

The contraction lemma (see Shalev-Shwartz and Ben-David (2014)) is useful in estimating the Rademacher complexity.

Lemma H (Contraction lemma)**.** Let $\phi_i : \mathbb{R} \to \mathbb{R}$ ($i \in \{1, \ldots, n\}$) be ρ-Lipschitz functions and $\mathcal{F}$ be a set of functions from $\mathcal{Z}$ to $\mathbb{R}$. Then it follows that for any $\{z_i\}_{i=1}^n \subset \mathcal{Z}$,

$$\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i \phi_i \circ f(z_i)\right] \leq \rho\,\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right].$$

Let $p_0(\theta)d\theta$ be a distribution in proportion to $\exp\left(-\frac{1}{2}\|\theta\|_2^2\right)d\theta$. We define a family of mean field neural networks as follows: for $R > 0$,

$$\mathcal{F}_{KL}(R) = \{h_q : \mathcal{X} \to \mathbb{R} \mid q \in \mathcal{P}_2,\ KL(q\|p_0) \leq R\}.$$

The Rademacher complexity of this function class is obtained by Chen et al. (2020).

Lemma I (Chen et al. (2020))**.** Suppose $|h_\theta(x)| \leq 1$ holds for $\forall\theta \in \Theta$ and $\forall x \in \mathcal{X}$. We have for any constant $R \geq \frac{1}{2}$ and set $S \subset \mathcal{X}$ of size n,

$$\hat{\Re}_S(\mathcal{F}_{KL}(R)) \leq 2\sqrt{\frac{R}{n}}.$$

D.2 Generalization Bound on the Binary Classification Problems

We here give a generalization bound for the binary classification problems. Hence, we suppose $\mathcal{Y} = \{-1, 1\}$ and consider the problem (3) with the smoothed hinge loss defined below.

$$\ell(z, y) = \begin{cases} 0 & \text{if } zy \geq 1/2; \\ (1-2zy)^2 & \text{if } 0 \leq zy < 1/2; \\ 1-4zy & \text{else} \end{cases}$$

We also define the 0-1 loss as $\ell_{01}(z, y) = 1[zy < 0]$.

Theorem E. Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Suppose there exists a true distribution $q_\circ \in \mathcal{P}_2$ satisfying $h_{q_\circ}(x)y \geq 1/2$ for $\forall(x, y) \in \mathrm{supp}(\mathcal{D})$ and $KL(q_\circ\|p_0) \leq 1/2$. Let $S = \{(x_i, y_i)\}_{i=1}^n$ be training examples independently sampled from $\mathcal{D}$. Suppose $|h_\theta(x)| \leq 1$ holds for $\forall(\theta, x) \in \Theta \times \mathcal{X}$. Then, for the minimizer $q_* \in \mathcal{P}_2$ of the problem (3), it follows that with probability at least $1-\delta$ over the choice of S,

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell_{01}(h_{q_*}(X), Y)] \leq \lambda_2 KL(q_\circ\|p_0) + 16\sqrt{\frac{KL(q_\circ\|p_0)}{n}} + 15\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}.$$

Proof. We first estimate a radius R to satisfy $q_\lambda \in \mathcal{F}_{KL}(R)$. Note that the regularization term of objective $\mathcal{L}(q)$ is $\lambda_2 KL(q\|p_0)$ and that $\ell(h_{q_*}(x_i); y_i) = 0$ from the assumption on q_* and the definition of the smoothed hinge loss. Since $\mathcal{L}(q_\lambda) \le \mathcal{L}(q_*)$, we get

$$KL(q_\lambda\|p_0) \le \frac{1}{\lambda_2}\mathcal{L}(q_*) = KL(q_*\|p_0); \tag{29}$$

$$\frac{1}{n}\sum_{i=1}^{n} \ell(h_{q_\lambda}(x_i); y_i) \le \mathcal{L}(q_*) = \lambda_2 KL(q_*\|p_0). \tag{30}$$

Especially, setting $R = KL(q_*\|p_0)$, we see $q_\lambda \in \mathcal{F}_{KL}(R)$.

We next define the set of composite functions of loss and mean field neural networks as follows:

$$\mathcal{F}(R) = \{(x;y) \in \mathcal{X}\times\mathcal{Y} \mapsto \ell(h(x);y) \mid h \in \mathcal{F}_{KL}(R)\}. \tag{31}$$

Since $\ell(z;y)$ is 4-Lipschitz continuous with respect to z, we can estimate the Rademacher complexity $\hat{\Re}_S(\mathcal{F}(R))$ by using Lemma H with $\phi_i(\cdot) = \ell(\cdot\,; y_i)$ as follows:

$$\begin{aligned}\hat{\Re}_S(\mathcal{F}(R)) &= \mathbb{E}\left[\sup_{h\in\mathcal{F}_{KL}(R)} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(h(x_i); y_i)\right]\\ &\le 4\mathbb{E}\left[\sup_{h\in\mathcal{F}_{KL}(R)} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(x_i)\right]\\ &= 4\hat{\Re}_{\{x_i\}_{i=1}^n}(\mathcal{F}_{KL}(R))\\ &\le 8\sqrt{\frac{R}{n}},\end{aligned} \tag{32}$$

where we used Lemma I for the last inequality.

From the boundedness assumption on h_q, we have $0 \le \ell(h_q(x);y) \le 5$ for $\forall q \in \mathcal{P}_2$. Applying Lemma G with $\mathcal{F} = \mathcal{F}(R)$, we have with probability at least $1-\delta$,

$$\begin{aligned}\mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell_{01}(h_{q_\lambda}(X);Y)] &\le \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(h_{q_\lambda}(X);Y)]\\ &\le \frac{1}{n}\sum_{i=1}^{n}\ell(h_{q_\lambda}(x_i);y_i) + 2\hat{\Re}_S(\mathcal{F}(R)) + 15\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}\\ &\le \lambda_2 KL(q_*\|p_0) + 16\sqrt{\frac{R}{n}} + 15\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}\\ &= \lambda_2 KL(q_*\|p_0) + 16\sqrt{\frac{KL(q_*\|p_0)}{n}} + 15\sqrt{\frac{1}{2n}\log\frac{2}{\delta}},\end{aligned}$$

where we used $\ell_{01}(z;y) \le \ell(z;y)$, (30) and (32). □

This theorem results in the following corollary:

Corollary C. *Suppose the same assumptions in Theorem E hold. Moreover, we set $\lambda_1 = \lambda\sqrt{n}$ ($\lambda > 0$) and $\lambda_2 = 1/\sqrt{n}$. Then, the following bound holds with the probability at least $1-\delta$ over the choice of training examples,*

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell_{01}(h_{q_\lambda}(X);Y)] \le \frac{KL(q_*\|p_0')}{\sqrt{n}} + 16\sqrt{\frac{KL(q_*\|p_0')}{n}} + 15\sqrt{\frac{1}{2n}\log\frac{2}{\delta}},$$

where p_0' is the Gaussian distribution in proportion to $\exp(-\lambda\|\theta\|_2^2)$.

D.3 Generalization Bound on the Regression Problem

We here give a generalization bound for the regression problems. We consider the squared loss $\ell(z;y) = 0.5(z-y)^2$ and the bounded label $\mathcal{Y} \subset [-1;1]$.

Theorem F. Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Suppose there exists a true distribution $q^\circ \in \mathcal{P}_2$ satisfying $y = h_{q^\circ}(x)$ for $\forall (x, y) \in \mathrm{supp}(\mathcal{D})$ and $KL(q^\circ \| p_0) \le 1/2$. Let $S = \{(x_i, y_i)\}_{i=1}^n$ be training examples independently sampled from $\mathcal{D}$. Suppose $|h(\theta; x)| \le 1$ holds for $\forall (\theta, x) \in \Omega \times \mathcal{X}$. Then, for the minimizer $q_* \in \mathcal{P}_2$ of the problem (3), it follows that with probability at least $1 - \delta$ over the choice of S,

$$\mathbb{E}_{(X,Y) \sim \mathcal{D}}[\ell(h_{q_*}(X), Y)] \le \lambda_2 KL(q^\circ \| p_0) + 8\sqrt{\frac{KL(q^\circ \| p_0)}{n}} + 6\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}.$$

Proof. The proof is very similar to that of Theorem E. Note that $\ell(h_{q^\circ}(x_i), y_i) = 0$ from the assumption on q° and that inequalities (29) and (30) hold in this case too. Hence, setting $R = KL(q^\circ \| p_0)$, we see $q_* \in \mathcal{F}_{KL}(R)$.

Since $\ell(z, y)$ is 2-Lipschitz continuous with respect to $z \in [-1, 1]$ for any $y \in \mathcal{Y} \subset [-1, 1]$, we can estimate the Rademacher complexity $\hat{\Re}_S(\mathcal{F}(R))$ of $\mathcal{F}(R)$ (defined in (31)) in the same way as Theorem E:

$$\hat{\Re}_S(\mathcal{F}(R)) \le 4\sqrt{\frac{R}{n}}. \tag{33}$$

From the boundedness assumption on h_q and $\mathcal{Y}$, we have $0 \le \ell(h_q(x), y) \le 2$ for $\forall q \in \mathcal{P}_2$. Hence, applying Lemma G with $\mathcal{F} = \mathcal{F}(R)$, we have with probability at least $1 - \delta$,

$$\begin{aligned}\mathbb{E}_{(X,Y) \sim \mathcal{D}}[\ell(h_{q_*}(X), Y)] &\le \frac{1}{n}\sum_{i=1}^n \ell(h_{q_*}(x_i), y_i) + 2\hat{\Re}_S(\mathcal{F}(R)) + 6\sqrt{\frac{1}{2n}\log\frac{2}{\delta}} \\ &\le \lambda_2 KL(q^\circ \| p_0) + 8\sqrt{\frac{KL(q^\circ \| p_0)}{n}} + 6\sqrt{\frac{1}{2n}\log\frac{2}{\delta}},\end{aligned}$$

where we used (30) and (33). □

This theorem results in the following corollary:

Corollary D. Suppose the same assumptions in Theorem F hold. Moreover, we set $\lambda_1 = \lambda/\sqrt{n}$ ($\lambda > 0$) and $\lambda_2 = 1/\sqrt{n}$. Then, the following bound holds with the probability at least $1 - \delta$ over the choice of training examples,

$$\mathbb{E}_{(X,Y) \sim \mathcal{D}}[\ell(h_{q_*}(X), Y)] \le \frac{KL(q^\circ \| p_0')}{\sqrt{n}} + 8\sqrt{\frac{KL(q^\circ \| p_0')}{n}} + 6\sqrt{\frac{1}{2n}\log\frac{2}{\delta}},$$

where p_0' is the Gaussian distribution in proportion to $\exp(-\lambda\|\theta\|_2^2)$.

E Additional Discussions

E.1 Efficient Implementation of PDA

Note that similar to SGD, Algorithm 1 only requires gradient queries (and additional Gaussian noise); in particular, a weighted average $\bar{g}^{(t)}$ of functions $g^{(t)}$ is updated and its derivative with respect to parameters is calculated. In the case of empirical risk minimization, this procedure can be implemented as follows. We use $\{w_i\}_{i=1}^n$ (initialized as zeros) to store the weighted sums of $\partial_z \ell(h_{\tilde{q}^{(t)}}(x_{i_t}), y_{i_t})$. At step t in the outer loop, w_{i_t} is updated as

$$w_{i_t} \leftarrow w_{i_t} + t\,\partial_z \ell(h_{\tilde{q}^{(t)}}(x_{i_t}), y_{i_t}).$$

The average $\partial_\theta \bar{g}^{(t)}(\theta_r^{(k)})$ can then be computed as

$$\frac{2}{\lambda_2 (t+2)(t+1)}\sum_{i=1}^n w_i\,\partial_\theta h(\theta_r^{(k)}; x_i) + \frac{2\lambda_1 t}{\lambda_2 (t+2)}\theta_r^{(k)};$$

where we use $\{\theta_r^{(k)}\}_{k=1}^M$ to denote parameters $\Theta^{(k)}$ at step k of the inner loop. This formulation makes Algorithm 1 straightforward to implement.

In addition, the PDA algorithm can also be implemented with mini-batch update, in which a set of data indices $I_t = \{i_{t,1}, \ldots, i_{t,b}\} \subset \{1, 2, \ldots, n\}$ is selected per outer loop step instead of one single index i_t. Due to the reduced variance, mini-batch update can stabilize the algorithm and lead to faster convergence. Our theoretical results in the sequel trivially extends to the mini-batch setting.

E.2 Extension to Multi-class Classification

We give a natural extension of PDA method to multi-class classification settings. Let $\mathcal{C}$ denote the finite set of all class labels and $|\mathcal{C}|$ denote its cardinality. For multi-class classification problems, we define a component $h(\theta; x)$ of an ensemble as follows. Let $a_r \in \mathbb{R}^{|\mathcal{C}|}$ and $b_r \in \mathbb{R}^d$ $(r \in \{1, \ldots, M\})$ be parameters for output and input layers, respectively, and set $\theta_r = (a_r, b_r)$ and $\Theta = \{\theta_r\}_{r=1}^M$. Then, we define $h_{\theta_r}(x) = h(\theta_r; x) = \sigma_2(a_r \sigma_1(b_r^\top x))$[7] which is a neural network with one hidden neuron, and denote

$$h_\Theta(x) = \frac{1}{M}\sum_{r=1}^M h_{\theta_r}(x).$$

Note that $h_\Theta(x)$ is a natural two-layer neural network with multiple outputs. Suppose that each parameter θ_r follows $q(\theta)d\theta$. Then the mean field limit can be defined as

$$h_q(\cdot) = \mathbb{E}_{\theta \sim q}[h(\theta; \cdot)] : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{C}|}.$$

Let $\ell(z, y)$ $(z = \{z_y\}_{y \in \mathcal{C}} \in \mathbb{R}^{|\mathcal{C}|}, y \in \mathcal{C})$ be the loss for multi-class classification problems. A typical choice is the cross-entropy loss with the soft-max activation, that is

$$\ell(z, y) = -\log \frac{\exp(z_y)}{\sum_{y' \in \mathcal{C}} \exp(z_{y'})} = -z_y + \log \sum_{y' \in \mathcal{C}} \exp(z_{y'}).$$

In this case, the functional derivative of $\ell(h_q(x), y)$ with respect to q is

$$-h_y(\theta; x) + \frac{\sum_{y' \in \mathcal{C}} \exp(h_{q,y'}(x))\, h_{y'}(\theta; x)}{\sum_{y' \in \mathcal{C}} \exp(h_{q,y'}(x))}$$

where we supposed the outputs of h and h_q are also indexed by $\mathcal{C}$. Hence, the counterpart of $g^{(t)}$ in Algorithm 2 in this setting is

$$g^{(t)} = -h_{y_t}(\cdot; x_t) + \frac{\sum_{y' \in \mathcal{C}} \exp(h_{q^{(t)},y'}(x_t))\, h_{y'}(\cdot; x_t)}{\sum_{y' \in \mathcal{C}} \exp(h_{q^{(t)},y'}(x_t))} + \lambda_1 \|\cdot\|_2^2.$$

Using this function, the DA method for multi-class classification problems can be obtained in the same manner as Algorithm 2. Moreover, its discretization can be also immediately derived by replacing the function $\bar{g}^{(t)}$ used in Algorithm 1 with

$$\bar{g}^{(t)} = \frac{2}{\lambda_2 (t+2)(t+1)} \sum_{s=1}^t s \left(-h_{y_s}(\cdot; x_s) + \frac{\sum_{y' \in \mathcal{C}} \exp(h_{\tilde{\Theta}^{(s)},y'}(x_s))\, h_{y'}(\cdot; x_s)}{\sum_{y' \in \mathcal{C}} \exp(h_{\tilde{\Theta}^{(s)},y'}(x_s))} + \lambda_1 \|\cdot\|_2^2 \right).$$

In the case of empirical risk minimization, we can adopt an efficient implementation as done in Section E.1. We use $\{w_{i,y}\}_{i \in \{1,\ldots,n\}, y \in \mathcal{C}}$ (initialized as zeros) to store the coefficients of $h_y(\cdot; x_i)$. At step t in the outer loop, $w_{i_t,y}$ $(y \in \mathcal{C})$ are updated as

$$w_{i_t,y} \leftarrow \begin{cases} w_{i_t,y} + t\left(-1 + \frac{\exp(h_{\tilde{\Theta}^{(t)},y}(x_{i_t}))}{\sum_{y' \in \mathcal{C}} \exp(h_{\tilde{\Theta}^{(t)},y'}(x_{i_t}))}\right) & y = y_{i_t}, \\ w_{i_t,y} + t\frac{\exp(h_{\tilde{\Theta}^{(t)},y}(x_{i_t}))}{\sum_{y' \in \mathcal{C}} \exp(h_{\tilde{\Theta}^{(t)},y'}(x_{i_t}))} & y \neq y_{i_t}. \end{cases}$$

[7] Here, $a_r \sigma_1(b_r^\top x)$ is a scalar $\sigma_1(b_r^\top x)$ times a vector a_r.

Then, $\nabla_{\theta_r}\bar{g}^{(t)}(\Theta^{(k)})$ can be computed as

$$\frac{2}{\lambda_2(t+2)(t+1)}\sum_{i=1}^n\sum_{y\in C} w_{i,y}\,\partial h_y(\theta_r^{(k)}, x_i) + \frac{2\lambda_1 t}{\lambda_2(t+2)}\theta_r^{(k)},$$

where we use $\Theta^{(k)} = \{\theta_r^{(k)}\}_{k=1}^M$ to denote parameters $\Theta^{(k)}$ at step k of the inner loop.

Finally, we remark that while we here utilize a simple network $h_\theta(x)$ to recover a normal two-layer neural network, it is also possible to use deep narrow networks or narrow convolutional neural networks as a component $h_\theta(x)$; in other words h_Θ can represent an ensemble of various types of small network. While such extensions are not covered by our current theoretical analysis, they may achieve better practical performance.

E.3 Correspondence with Finite-dimensional Dual Averaging Method

We explain the correspondence between the finite-dimensional dual averaging method developed by Nesterov (2005, 2009); Xiao (2009) and our proposed method (Algorithm 2); our goal here is to provide an intuitive understanding of the derivation of Algorithm 2 in the context of the classical dual averaging method.

First, we introduce the (regularized) dual averaging method (Nesterov, 2009; Xiao, 2009) in a more general form for solving the regularized optimization problem on the finite-dimensional space. Let $w \in \mathbb{R}^m$ be a parameter, $l(w, z)$ be a convex loss in w, where z is a random variable which represents an example, and $\psi(w)$ is a regularization function. Then, the problem solved by the dual averaging method is given as

$$\min_{w\in\mathbb{R}^m}\{\mathbb{E}_z[l(w, z)] + \psi(w)\}.$$

Let $\{w^{(s)}\}_{s=1}^t$ and $\{\mathbf{g}^{(s)}\}_{s=1}^t = \{\partial_w l(w^{(s)}, z_s)\}_{s=1}^t$ be histories of iterates and stochastic gradients. The subproblems to produce the next iterate in the dual averaging method is designed by using the strongly convex function $d(w)$ and positive hyperparameters $\{\eta_s\}_{s=1}^\infty$ and $\{\beta_s\}_{s=2}^\infty$. Specifically, the next iterate $w^{(t+1)}$ is defined as the minimizer of the following problem in which the loss function is linearized and weighted sum of which is taken over the history:

$$\min_{w\in\mathbb{R}^m}\left\{\sum_{s=1}^t \eta_s \mathbf{g}^{(s)\top} w + \sum_{s=1}^t \eta_s \psi(w) + \beta_{t+1} d(w)\right\}. \tag{34}$$

Next, we consider our problem setting of optimizing the probability distribution and reformulate the subproblem (7) solved in Algorithm 2 as follows:

$$\min_{q\in\mathcal{P}_2}\left\{\mathbb{E}_q\left\langle\sum_{s=1}^t s g^{(s)}\right\rangle + \sum_{s=1}^t s\lambda_2\mathbb{E}_q[\log(q)] + (t+1)\lambda_2\mathbb{E}_q[\log(q)]\right\}, \tag{35}$$

By comparing (34) and (35), we arrive at the following correspondence: $\beta_s = \eta_s = s$, $\mathbf{g}^{(s)} \leftrightarrow g^{(s)}$, $d(w) = \psi(w) \leftrightarrow \lambda_2\mathbb{E}_q[\log(q)]$. We note that in our problem setting the expectation by q can be seen as an inner product with the integrand and $\lambda_2\mathbb{E}_q[\log(q)]$ is also set to $d(w)$ because the negative entropy acts as a strongly convex function (Lemma A).

F Additional Experiments

F.1 Comparison of Generalization Error

We provide additional experimental results on the generalization performance of PDA. We consider empirical risk minimization for a regression problem (squared loss): the input $x_i \sim N(0, I_p)$, and f^* is a single index model: $f^*(x) = \mathrm{sign}(\langle w^*, x\rangle)$. We set $n = 1000$, $p = 50$, $M = 200$, and implement both noisy gradient descent (Mei et al., 2018) using full-batch gradient and our proposed Algorithm 1 (PDA) using mini-batch update with batch size 50.

Figure 3 we compare the generalization performance of different training methods: noisy GD and PDA in the mean field regime, and also noisy GD in the kernel regime. We fix the $_2$ and entropy regularization to be the same across all settings: $_1 = 10^{2}$, $_2 = 5 \quad 10^{4}$. We set the total number of iterations (outer + inner loop steps) in PDA to be the same as GD, and tuned the learning rate for optimal generalization. Observe that

Model with the NTK scaling (green) generalizes worse than the mean field models (red and blue). This is consistent with observations in Chizat and Bach (2018a).

For the mean field scaling, PDA (under early stopping) leads to slightly lower test error than noisy GD. We intend to further investigate this difference in the generalization performance. (see Appendix D for generalization bounds of the PDA solution)

Figure 3: Test error of mean field neural networks (= 1) trained with noisy GD (red) and PDA (blue), and network in the kernel regime (= 1=2) optimized by GD (green).

F. 2 PDA Beyond $`_2$ Regularization

Note that our current formulation (4) considers $`_2$ regularization, which allows us to establish polynomial runtime guarantee for the inner loop via the Log-Sobolev inequality. As remarked in Section 4, our global convergence analysis can easily be extended to Hölder-smooth gradient via the convergence rate of Langevin algorithm given in Erdogdu and Hosseinzadeh (2020). Although we do not provide details for this extension in the current work (due to the use of Vempala and Wibisono (2019)), we empirically demonstrate one of its applications in handling $`_p$-regularized objectives for $p > 1$ in the following form,

$$R^p_{{}_1; {}_2}(q) \overset{\text{def}}{=} {}_1 E_q[k \;\; k_p^p] + {}_2 E_q[\log(q)]: \tag{36}$$

Erdogdu and Hosseinzadeh (2020) cannot directly cover the non-smooth $`_1$ regularization, but we can still obtain relatively sparse solution by setting p close to 1. Intuitively speaking, when the underlying task exhibits certain low-dimensional or sparse structure, we expect a sparsity-promoting regularization to achieve better generalization performance.

Figure 4(a) demonstrates the advantage of L_p-norm regularization for $p < 2$ in empirical risk minimization, when the target function exhibits sparse structure. We set $n = 1000; p = 50$; the teacher is a multiple-index model ($m = 2$) with binary activation, and parameters of each neuron are 1-sparse. We optimize the student model with PDA (warm-start), where we set $_1 = 10^{2}$, $_2 = 10^{4}$, and vary the norm penalty p from 1.01 to 2. Note that smaller p results in favorable generalization due to the induced sparsity. On the other hand, we expect the benefit of sparse regularization to diminish when the target function is not sparse. This intuition is confirmed in 4(b), where we control the target sparsity by randomly selecting r parameters to be non-zero, and we define $s = r=d$ to be the sparsity level. Observe that the benefit of sparsity-inducing regularization (smaller p) is more prominent under small s (brighter color), which indicates a sparse target function.

(a) Impact of L_p regularization. (b) Generalization under sparse teacher.

Figure 4: PDA with general $`_p$ regularizer (objective (36)). (a) Generalization error vs. training time in learning a 1-sparse target function. (b) generalization error vs. sparsity of the target function s.

F. 3 On the Role of Entropy Regularization

Our objective (3) includes an entropy regularization with magnitude $_2$. In this section we illustrate the impact of this regularization term. In Figure 5(a) we consider a synthetic 1D dataset ($n = 15$) and plot the output of a two-layer tanh network with 200 neurons trained by SGD and PDA to minimize the squared loss till convergence. We use the same ℓ_2 regularization ($_1 = 10^{3}$) for both algorithms, and for PDA we set the entropic term $_2 = 10^{4}$. Observe that SGD with weak regularization (red) almost interpolates the noisy training data, whereas PDA with entropy regularization finds low-complexity solution that is smoother (blue).

We therefore expect entropy regularization to be beneficial when the labels are noisy and the underlying target function (teacher) is "simple". We verify this intuition in Figure 5(b). We set $n = 500$, $d = 50$ and $M = 500$, and the teacher model is a linear function on the input features. We employ SGD or PDA to optimize the squared error. For both algorithms we use the same ℓ_2 regularization $_1 = 10^{2}$, but PDA includes a small entropy term $_2 = 5 \ 10^{4}$. We plot the generalization error of the converged model under varying amount of label noise. Note that as the labels becomes more corrupted, PDA (blue) results in lower test error due to the entropy regularization[8]. On the other hand, model under the kernel scaling (green) generalizes poorly compared to the mean field models. Furthermore, Figure 5(c) demonstrates that entropy regularization can be beneficial under low noise (or even noiseless) cases as well. We construct the teacher model to be a multiple-index model with binary activation. Note that in this setting PDA achieves lower stationary risk across all noise level, and the advantage amplifies as labels are further corrupted.

(a) Impact of entropy regularization (one-dimensional).

(b) Stationary risk vs. label noise (linear teacher).

(c) Stationary risk vs. label noise (multiple index teacher).

Figure 5: (a) 1D illustration of the impact of entropy regularization in two-layer tanh network: PDA (blue) finds a smoother solution that does not interpolate the training data due to entropy regularization. (b)(c) Test error of two-layer tanh network trained till convergence. PDA (blue) becomes advantageous compared to SGD (red) when labels become noisy, and the NTK model (green, note that the y-axis is on different scale) generalizes considerably worse than the mean field models.

F. 4 Adaptivity of Mean Field Neural Networks

Recall that one motivation to study the mean field regime (instead of the kernel regime) is the presence of feature learning. We illustrate this behavior in a simple student-teacher setup, where the target function is a single-index model with tanh activation. We set $n = 500; d = 50$, and optimize a two-layer tanh network ($M = 1000$), either in the mean field regime using PDA, or in the kernel regime using SGD. For both methods we choose $_1 = 10^{3}$, and for PDA we choose $_2 = 10^{4}$.

In Figure 6 we plot the the evolution of the cosine similarity between the target vector w and the top-5 singular vectors (PC1-5) of the weight matrix during training. In Figure 6(a) we observe that the mean field model trained with PDA "adapts" to the low-dimensional structure of the target function; in particular, the leading singular vector (bright yellow) aligns with the target direction. In contrast, we do not observe such alignment on the network in the kernel regime (Figure 6(b)), because the parameters do not travel away from the initialization. This comparison demonstrates the benefit of the mean field parameterization.

[8] Note that entropy regularization is not the only way to reduce overfitting – such capacity control can also be achieved by proper early stopping or other types of explicit regularization.