Can We Do Better Than Random Start? The Power of Data Outsourcing

YI CHEN, Hong Kong University of Science and Technology, China
JING DONG, Columbia University, USA
XIN T. TONG, National University of Singapore, Singapore
BO SHEN, New Jersey Institute of Technology, USA

Many organizations have access to abundant data but lack the computational power to process the data. While they can outsource the computational task to other facilities, there are various constraints on the amount of data that can be shared. It is natural to ask what can data outsourcing accomplish under such constraints. We address this question from a machine learning perspective. When training a model with optimization algorithms, the quality of the results often relies heavily on the points where the algorithms are initialized. We propose simulation-based algorithms that can utilize a small amount of outsourced data to find good initial points. Under suitable regularity conditions, we provide theoretical guarantees that the algorithms can find good initial points with a high probability. We also conduct numerical experiments to demonstrate that our algorithms perform significantly better than the random start approach.

Additional Key Words and Phrases: Non-convex optimization, initialization,

ACM Reference Format:
Yi Chen, Jing Dong, Xin T. Tong, and Bo Shen. 2023. Can We Do Better Than Random Start? The Power of Data Outsourcing. 1, 1 (June 2023), 27 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In this era, data is the new gold. Organizations of different sizes and sectors all realize the value of collecting data. However, it often requires substantial computational power to turn these data into valuable prediction models and not all organizations have such computational resources. One possible solution to this problem is outsourcing the data processing task to an external computing facility, where the computational power is substantially cheaper. However, the data organization may only be willing to share a small portion of data due to the following reasons: First, if the external computing facility has access to all the available data, it can obtain an accurate prediction model, which leads to potential competition risk. In addition, transferring data can be expensive especially when certain encryption is required.

Given the constraint that only part of the data is "shareable", the data organization can only expect sub-optimal solutions from the computing facility, and additional learning is required to improve these premature results. Since the data organization is assumed to have limited computational power, it is desirable if the computational cost of the additional learning can be minimized. In this context, we are interested in investigating the following two questions: 1) What type of computational task

Authors' addresses: Yi Chen, yichen@ust.hk, Hong Kong University of Science and Technology, Hong Kong, China; Jing Dong, jing.dong@gsb.columbia.edu, Columbia University, New York, NY, USA; Xin T. Tong, mattxin@nus.edu.sg, National University of Singapore, Singapore; Bo Shen, bo.shen@njit.edu, New Jersey Institute of Technology, USA.

XXXX-XXXX/2023/6-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

should be assigned to the external computing facility? 2) How much data should be outsourced? In this paper, we address these two questions from the perspective of machine learning.

Most machine learning models are trained using the risk minimization approach. That is, the unknown parameter θ is inferred by minimizing a loss function of the form $F(\theta) = \mathbb{E}[f(\theta, X)]$ where X is averaged over a population distribution or an empirical distribution of N data points (size of the full dataset), and $f(\theta, x)$ is the loss of using the model with parameter θ to explain data point x. Greedy local optimization algorithms are often applied to minimize F. If F is strongly convex, the computational cost of an algorithm $\mathcal{T}$, $c(\mathcal{T})$, depends on the accuracy requirement ϵ, i.e., how close to the optimal object value one wants to achieve, the initialization θ^0, and/or the number of data points N. In this setting, $c(\mathcal{T})$ can be large but is computationally manageable since $\mathcal{T}$ converges to the optimal parameter regardless of the initialization [22, 39]. However, if F is non-convex, the quality of the parameter learned from $\mathcal{T}$ can depend heavily on its initialization θ^0. In general, greedy algorithms converge to the local minimum that is close to θ^0. Thus, in order to find the global minimum θ^*, one needs to start $\mathcal{T}$ in an appropriate attraction region of the optimal θ^*, $\mathbb{B}_0^*$. In practice, the location and shape of $\mathbb{B}_0^*$ are unknown. A common way to deal with this issue is using *randomized initialization* where the initial points are sampled uniformly at random from the solution space. The idea is that by trying multiple, say M, random initial points, at least one of them will be in $\mathbb{B}_0^*$, and $\mathcal{T}$ applied to that point will find θ^*. Hence, the total computational cost, in this case, is $Mc(\mathcal{T})$, where $c(\mathcal{T})$ is the computational cost to find a stationary point (e.g., local or global minimum). Here, we assume the computational costs of $\mathcal{T}$ starting from different initial points are the same, i.e., $c(\mathcal{T})$ can be interpreted as the worst-case complexity among all possible initial points. In practice, when using smarter initialization, one can be closer to local/global minimums when applying $\mathcal{T}$, and achieve reduced computational cost when applying $\mathcal{T}$ to find a stationary point. However, in this paper, we mainly focus on using smarter initialization to increase the chance of finding B_0^*. In this case, fewer initialization points are required, and $\mathcal{T}$ only needs to be applied a few times.

From the above discussion, we note that when learning a non-convex loss function, the computational cost is the combined cost of two tasks: 1) Exploration: find an initial point within the attraction region of the global minimum and 2) Exploitation: running greedy algorithm starting from a given initial point. To achieve high accuracy, the exploitation task given a good starting point often requires a sufficiently large amount of data and is very well understood in the literature [5]. In contrast, the exploration task is less studied. The performance can be problem dependent and the computational cost can be very high. One important insight that we will leverage in our subsequent development is that the landscape of the empirical risk based on a random sample of size n, $\widehat{F}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} f(\theta, x_i)$, should resemble that of $F(\theta)$ reasonably well when n is large enough. Thus, in the data outsourcing context, it is natural to ask if we can assign the exploration task to the external computing facility. In other words, we split the computation tasks into two phases:

Exploration: The external computing facility is assigned to explore the energy landscape of $\widehat{F}_n(\theta)$, where n is much smaller than the size of the full dataset, and find a good initial point(s) θ^0 (or $\theta_1, \ldots, \theta_m$).

Exploitation: The data organization can run more refined exploitation starting from θ^0 (or $\theta_1, \ldots, \theta_m$). In this case, the computational cost, from the data organization's perspective, can be reduced from $Mc(\mathcal{T})$ to $c(\mathcal{T})$ (or $mc(\mathcal{T})$). Such a reduction can be substantial if M needs to be a very large number to achieve the desired performance.

Similar computational strategies can also be applied outside the data outsourcing context. The idea is that we can first use a less accurate loss function $\widehat{F}_n(\theta)$ with a smaller amount of data to

find good initializations. We then employ greedy optimization algorithms on $F(\theta)$ starting from these carefully selected initial points.

Our contribution. First, we propose simulation-based algorithms for the external computing facility to obtain good initializations for $F(\theta)$-optimization with outsourced data. Particularly, we design two types of procedures, sampling and optimization, depending on whether the optimization cost $c(\mathcal{T})$ is moderate or large: *If $c(\mathcal{T})$ is moderate,* multiple instances of $\mathcal{T}$ can be implemented starting from different initial points. In this scenario, we suggest sampling multiple initial points from a distribution $\pi_\beta(\theta) \propto \exp(-\beta \widehat{F}_n(\theta))$ with a properly chosen β. The distribution $\pi_\beta(\theta)$ tends to concentrate in regions where the loss is small, and the degree of concentration is determined by β. *If $c(\mathcal{T})$ is large,* only one instance of $\mathcal{T}$ can be implemented. In this scenario, we suggest starting from the global minimum of $\widehat{F}_n(\theta)$, i.e., a single initial point. This minimizer can be obtained by applying a selection procedure on samples from $\pi_\beta(\theta)$ as we will explain in detail in Section 2.2.

Second, our analytical results provide a rigorous justification of these procedures and guide how much data should be outsourced and how to choose the sampling parameter β. In particular, we show that under proper regularity conditions, for $n = \Omega(d \log(1/\rho)\delta^{-2})$, with probability $(1-\rho)$, both methods can find a good initial point, i.e., when $\mathcal{T}$ is initialized from this point, it will find the global minimum of $F(\theta)$. Here, d is the dimension of θ, and δ is a parameter for the required approximation accuracy of the loss function, i.e., how well $\widehat{F}_n(\theta)$ approximates $F(\theta)$, and it may depend on the structure/geometry of $F(\theta)$.

To sum up the data outsourcing idea, the data organization only needs to share n data points with the external computing facility. The external computing facility will carry out either the sampling or the optimization procedure on $\widehat{F}_n(\theta)$ to generate a small set of good initial point(s). The data organization can then run a greedy optimization algorithm starting from these point(s) to optimize F. The data organization saves in-house computational effort by running much fewer copies of greedy optimization algorithms.

Related literature. Data outsourcing has become an interesting problem due to the emergence of big data and cloud computing. Most existing work focuses on data management policies and encryption [8, 15, 38]. To the best of our knowledge, this work is the first to study data outsourcing from the perspective of finding better initializations for machine learning.

Our problem can be viewed as a special non-convex stochastic optimization problem. How to efficiently solve smooth but non-convex problems is a fast developing area [2, 18, 46]. Our contribution is the development of a new initialization method. While finding good initial points is an important problem, the related literature is rather limited. The most common approach is using crude uniform sampling, which can be costly as many such initial points need to be tested to find a global minimum. Our approach provides a computationally feasible refined solution to this problem. Finding good initializations in more specific settings has been studied in the literature. For example, [7] studies the efficacy of gradient descent with random initialization for solving systems of quadratic equations. Weight initialization for neural networks has been investigated in [3, 21, 52]. Spectral initialization has been proposed for generalized linear sensing models in the high dimensional regime [28]. The key advantages of our proposed method are its general applicability and theoretical performance guarantee.

Our proposed algorithm is closely related to stochastic adaptive search and the two-phase methods in global optimization [51]. One popular stochastic adaptive search algorithm is simulated annealing [1, 24, 32]. The key difference between our approach and simulated annealing is that we completely separate the exploration task from the exploitation task. This allows us to use a smaller sample size for the exploration task, and when using the Boltzmann machine type of procedure for exploration, we do not need to set a cooling schedule. Our method can be viewed as a special

case of the two-phase methods (see [48] for an overview). In the context of data outsourcing, we advocate outsourcing the exploration phase to the external computing facility using a smaller set of data. We analyze how the Boltzmann machine type of procedures can be applied by the external computing facility to achieve good exploration, i.e., finding good initializations for the in-house exploitation phase.

Our problem is related to but different from federated learning. Federated learning is a special form of distributed learning where the central learning agent does not have access to distributed agents' devices and data [26]. Most existing developments in federated learning try to address two main challenges: i) the communication cost between distributed agents and the central agent, and ii) heterogeneous distributed agents where the data owned by the individual agent may not be a representative sample of the full data (see, e.g., [23, 27, 53]). In contrast, our setting assumes the data organization (central agent) owns all the data and can decide what to distribute to external computing facilities (distributed agents). In this case, we can ensure that the data sent to distributed agents are representative. The task we assign to distributed agents is also fundamentally different from federated learning.

Our theoretical analysis relies on detailed finite-sample performance quantification when using $\widehat{F}_n(\theta)$ to approximate $F(\theta)$. Empirical process theory can be utilized to establish uniform convergence of $\widehat{F}_n(\theta)$ to $F(\theta)$ and consistency of M-estimators [44]. In stochastic programming, bounds on sample size have been developed to ensure that the set of δ-optimal solutions of the sample average approximation is contained in the set of ϵ-optimal solutions of the true objective (Chapter 5.3.1 in [40]). Our development leverages a stronger notion of convergence recently developed in [30], which ensures uniform convergence of not only the empirical loss $\widehat{F}_n(\theta)$, but also its gradient and Hessian, i.e., $\nabla\widehat{F}_n(\theta)$ and $\nabla^2\widehat{F}_n(\theta)$. These convergence results allow us to establish the accuracy and complexity of the sampling and optimization procedures on $\widehat{F}_n(\theta)$ using some recent complexity bounds developed in the literature [17, 45].

Notation. We use $\|\theta\| = \sqrt{\theta^T\theta}$ to denote the Euclidean norm of a vector θ, and $\|A\|_{\text{op}} = \sup\{\|Av\|/\|v\| : v \neq 0\}$ to denote the operator norm of a matrix A in the Euclidean space. For real numbers a, b, let $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. Given two sequences of real numbers $\{a_n\}_{n\geq 1}$ and $\{b_n\}_{n\geq 1}$, $a_n = O(b_n)$ denotes that there exists a constant $C > 0$, such that $|a_n| \leq C|b_n|$, and $a_n = \Omega(b_n)$ denotes that $|a_n| \geq C|b_n|$. For two probability measures ν and $\tilde{\nu}$ on the same sigma algebra, we denote $\|\nu - \tilde{\nu}\|_{TV}$ as the total variation distance between ν and $\tilde{\nu}$.

2 METHODOLOGY

We consider minimizing a smooth but non-convex function $F(\theta)$, which takes the form

$$F(\theta) = \mathbb{E}_{X\sim\xi}\big[f(\theta, X)\big],$$

over a d-dimensional unit ball $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\| \leq 1\}$. We allow F to be the empirical loss function, in which case $F(\theta) = \frac{1}{N}\sum_{i=1}^{N} f(\theta, x_i)$ where $\{x_1, \ldots, x_N\}$ is the full dataset, i.e., ξ is the empirical distribution. The restriction that $\theta \in \Theta$ can be relaxed to any bounded convex domains, which is commonly assumed in the literature [30]. In machine learning applications, the range of θ can often be moderated by scaling the elements of x.

Since $F(\theta)$ is non-convex, the performance of any greedy deterministic optimization algorithm relies heavily on the choice of initial points. Specifically, a deterministic optimization algorithm $\mathcal{T}$ such as gradient descent (GD) or Newton's method can be trapped in a suboptimal local minimum instead of converging to the desired global minimum if initialized inappropriately. In this work, we design data outsourcing and exploration mechanisms to find good initial points for the optimization

algorithm $\mathcal{T}$. The objective is to increase the chance that $\mathcal{T}$ initialized from these points finds the global minimum.

Assume the outsourced data $\{x_1, \ldots, x_n\}$ follow the same distribution as ξ. We can construct a sample average approximation of $F(\theta)$ as $\widehat{F}_n(\theta) = \frac{1}{n}\sum_{i=1}^n f(\theta, x_i)$, which is a loss function of n data points. Evaluating $\widehat{F}_n(\theta)$ or $\nabla\widehat{F}_n(\theta)$ has a much smaller cost (linear in n) than evaluating $F(\theta)$ or $\nabla F(\theta)$ if the sample size n is not too large. This makes exploring the energy landscape of $\widehat{F}_n(\theta)$ using a sampling-based method (e.g., Langevin dynamics) more computationally friendly. Note that $\widehat{F}_n(\theta)$ captures certain structural information of $F(\theta)$. We are interested in effectively utilizing this information. More specifically, the work of [30] has shown that the energy landscape of $\widehat{F}_n(\theta)$ bears a close similarity to that of $F(\theta)$ when n surpasses a certain threshold. This indicates that the global minimum of $\widehat{F}_n(\theta)$ is likely to be closer to that of $F(\theta)$ than a random guess. Let $\hat{\theta}^*$ denote the global minimum of $\widehat{F}_n(\theta)$ and θ^* denote the global minimum of $F(\theta)$. Intuitively, if we use $\hat{\theta}^*$ as the initial point when applying the optimization algorithm $\mathcal{T}$, we are more likely to converge to θ^*. We refer to this approach as the *optimization approach*. It is quite computationally friendly to the data organization, since only one instance of in-house $\mathcal{T}$ is needed. However, it also comes with certain costs: 1) $\widehat{F}_n(\theta)$ is a noisy realization of $F(\theta)$, especially when n is small. Using just the global minimizer of $\widehat{F}_n$, which is a single point, can be risky. 2) $\widehat{F}_n$ is likely to be nonconvex as well and optimizing it can be expensive for the external computing facility.

An alternative approach is to sample initial points from a distribution

$$\pi_\beta(\theta) \propto \exp(-\beta\widehat{F}_n(\theta)) \cdot 1_{\{\theta\in\Theta\}}. \tag{1}$$

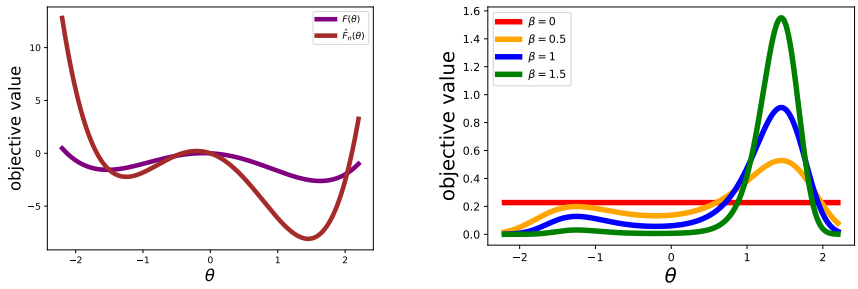
The parameter $\beta > 0$ is often referred to as the inverse temperature [50], which determines how much $\pi_\beta(\theta)$ concentrates around the global minimum of $\widehat{F}_n(\theta)$. A larger β leads to a higher concentration around $\hat{\theta}^*$. When $\beta = \infty$, we get $\hat{\theta}^*$ with probability one. On the other hand, when $\beta = 0$, π_β is simply the uniform distribution over Θ, which is equivalent to random initialization. Sampling from π_β with $\beta \in (0, \infty)$ can be viewed as an interpolation of the two extreme cases (see Figure 1 for a pictorial illustration). In general, the probability that a sample from $\pi_\beta(\theta)$ is far away from $\hat{\theta}^*$ decays exponentially fast as β increases. However, the cost of sampling from $\pi_\beta(\theta)$ may increase as β increases, due to the slow rate of convergence of the underlying Markov chain in the sampling algorithm (See Section 3.3 for more details). Compared to the optimization approach, this *sampling approach* takes into account that $\widehat{F}_n(\theta)$ is a noisy estimate of $F(\theta)$. Thus, instead of outputting a single point, we draw several initial points from $\pi_\beta(\theta)$.

We next provide more details of these two approaches. In practice, the implementation of the optimization approach requires sampling tools due to the non-convexity of $\widehat{F}_n$. Thus, we start with the sampling approach.

2.1 Procedures with the sampling approach

There is rich literature on how to sample from $\pi_\beta(\theta)$. When $\pi_\beta(\theta)$ is close to some simple reference distributions, independent samples can be obtained through rejection sampling, though this method can be highly inefficient for high-dimensional θ. For more complicated target distributions, Markov Chain Monte Carlo (MCMC) algorithms are typically applied. The idea is to simulate a stochastic process for which $\pi_\beta(\theta)$ is its invariant distribution. We next provide a well-known MCMC algorithm, called unadjusted Langevin algorithm (ULA) [11], to sample approximately from $\pi_\beta(\theta)$.

Other popular MCMC algorithms include random walk Metropolis, Metropolis adjusted Langevin algorithm (MALA) [37], etc. Recent studies have shown that these MCMC algorithms are efficient



(a) $F(\theta)$ (b) $\pi_\beta(\theta)$

Fig. 1. Density function $\pi_\beta(\theta)$ for different values of β

Algorithm 1 Unadjusted Langevin Algorithm (ULA)

Input: Outsourced data sample $\{x_1, \ldots, x_n\}$, inverse temperature parameter β, step size h, sampling budget K.
Initialization: Initial point θ^0.
for $k = 1$ to K **do**
Given θ^{k-1}, sample θ^k distributed as $\mathcal{N}(\theta - h\nabla\widehat{F}_n(\theta^{k-1}), 2\beta hI)$ where $\mathcal{N}$ denotes a Normal distribution and I is the $d \times d$ identity matrix.
end for
Output: θ^K.

when the target distribution is log-concave with perturbations [13, 29]. When $\nabla\widehat{F}_n$ is too expensive to evaluate, one may use online versions of these algorithms where the gradient is replaced by a stochastic gradient [47]. When $\widehat{F}_n(\theta)$ is not differentiable, one may use random walk Metropolis. When $\widehat{F}_n$ is non-convex with well-separated local minima, $\pi_\beta(\theta)$ is a multimodal distribution, and it can be difficult to sample $\pi_\beta(\theta)$ using these algorithms due to the slow rate of convergence to stationarity. This is particularly the case if β is large, since the stochastic algorithm may stick to one mode for many iterations before visiting the other modes. This issue can often be alleviated using methods such as parallel tempering or simulated tempering [9, 17, 41, 49]. The papers [17, 25] show that simulated tempering algorithms can sample a multimodal distribution with polynomial complexity.

The exploration algorithm is summarized in Algorithm 2

Algorithm 2 Sampling-based Initial Point Selection (SIPS)

Input: Outsourced data sample $\{x_1, \ldots, x_n\}$, inverse temperature parameter β, sampling algorithm $\mathcal{M}$, exploration sample size m.
Initialization: Construct the empirical average $\widehat{F}_n(\theta) = \frac{1}{n}\sum f(\theta, x_i)$ and the target density $\pi_\beta(\theta) \propto \exp(-\beta\widehat{F}_n(\theta)) \cdot 1_{\{\theta\in\Theta\}}$.
Sampling: Apply $\mathcal{M}$ to draw samples $\{\theta_1, \ldots, \theta_m\}$ from distribution π_β.
Output: Candidate initial points $\{\theta_1, \ldots, \theta_m\}$.

Given the samples $\theta_1, \ldots, \theta_m$ obtained by the external computing facility, the data organization can then implement $\mathcal{T}$ starting from each θ_i. Let $\mathcal{T}(\theta)$ denote the output of the optimization algorithm $\mathcal{T}$ starting from θ. Our theoretical analysis in the next section gives a rigorous justification of this procedure assuming β is large enough (see Theorem 3.5). In practice, this approach is more efficient than the naive random start even with a moderate β. As we will demonstrate through numerical experiments in Section 4, $\beta = 2$ already generates good initial points with a high probability in many examples.

2.2 Procedures with the optimization approach

When the in-house computational resource is limited, we can utilize the external computing facility to do further optimization and only pass the global minimizer of $\widehat{F}_n(\theta)$ to the data organization. In this case, the data organization only needs to implement one instance of $\mathcal{T}$. When $\widehat{F}_n(\theta)$ is non-convex, there is no consensus on how to find its global minimizer. Typical choices include either using meta-heuristic algorithms or sampling-based algorithms. Here we consider using sampling-based algorithms due to their connection to the sampling approach.

One popular way to find the global minimum of $\widehat{F}_n(\theta)$ involves generating samples $\theta_1, \ldots, \theta_m$ from the distribution $\pi_\beta(\theta)$ with a large β. This approach is investigated by [6, 36, 50] when ULA or its online version is used to sample from $\pi_\beta(\theta)$. As mentioned earlier, the parameter β determines how much $\pi_\beta(\theta)$ concentrates around the global minimum of $\widehat{F}_n(\theta)$, and a larger β leads to a higher concentration. When the samples $\theta_1, \ldots, \theta_m$ are available as candidate solutions, we can choose the one with the lowest objective value, i.e., θ_{i^*}, where

$$i^* = \text{argmin}_{i \in \{1,\ldots,m\}} \widehat{F}_n(\theta_i). \tag{2}$$

This procedure is summarized as the *annealing* approach in Algorithm 3. For this approach to be effective at finding the global minimum of $\widehat{F}_n$, β needs to be large enough. This usually increases the difficulty of sampling from $\pi_\beta(\theta)$ (see, e.g., Lemma 3.8).

A refinement of (2) can be applied to improve the quality of the initial point. In particular, if we apply a deterministic optimization algorithm $\widehat{\mathcal{T}}$, e.g., gradient descent, to $\widehat{F}_n$ initialized at θ_i, we may achieve a lower $\widehat{F}_n$-value. We then pick $\widehat{\mathcal{T}}(\theta_i)$ with the lowest $\widehat{F}_n$-value as the initial point, i.e., $\widehat{\mathcal{T}}(\theta_{i^*})$, where

$$i^* = \text{argmin}_{i \in \{1,\ldots,m\}} \widehat{F}_n\big(\widehat{\mathcal{T}}(\theta_i)\big). \tag{3}$$

This procedure is summarized as the *sampling-assisted-optimization (SAO) approach* in Algorithm 3. The SAO approach is similar to GDxLD developed in [10]. Comparing SAO to the annealing approach, sampling for SAO can often be done more efficiently with a smaller value of β. However, we incur the extra cost of invoking $\widehat{\mathcal{T}}$. We provide more detailed discussions about the computational cost for the external computing facility in Section 3.3.

3 THEORETICAL GUARANTEE IN FINDING THE GLOBAL MINIMUM

In this section, we analyze the performance of Algorithms 2 and 3. The key to the successful implementation of the algorithms is to set the appropriate outsourcing sample size n, inverse temperature β, and sample size of initial points m. Our performance analysis provides guidance on how to choose these parameters.

Conditions on the energy landscape. We start with some assumptions on the energy landscape of $F(\theta)$ and the randomness when evaluating $f(\theta, x)$. Since we run an optimization algorithm $\mathcal{T}$ that converges to a stationary point in the second phase (in-house optimization phase), the following assumption regularizes the configuration of the stationary points:

Algorithm 3 Optimization-based Initial Point Selection (OIPS)

Input: Outsourced data sample $\{x_1, \ldots, x_n\}$, inverse temperature parameter β, exploration sample size m, sampling algorithm $\mathcal{M}$, optimization algorithm $\hat{\mathcal{T}}$.
Initialization: Construct the empirical average $\widehat{F}_n(\theta) = \frac{1}{n}\sum f(\theta, x_i)$ and the target density $\pi_\beta(\theta) \propto \exp\{-\beta\widehat{F}_n(\theta)\} \cdot 1_{\{\theta\in\Theta\}}$
Sampling: Apply $\mathcal{M}$ to draw a sample $\{\theta_1, \ldots, \theta_m\}$ from distribution π_β.
if Annealing **then**
  Set $\theta^0 = \theta_{i^*}$ where $i^* = \text{argmin}_{i\in\{1,\ldots,m\}}\widehat{F}_n(\theta_i)$.
end if
if Sampling-assisted-optimize (SAO) **then**
  Set $\theta^0 = \widehat{\mathcal{T}}(\theta_{i^*})$ where $i^* = \text{argmin}_{i\in\{1,\ldots,m\}}\widehat{F}_n(\widehat{\mathcal{T}}(\theta_i))$.
end if
Output: Candidate initial point θ^0

Assumption 1. *$F(\theta) : \Theta \to \mathbb{R}$ is (σ, η)-strongly Morse, that is, $\|\nabla F(\theta)\| \geq \sigma$ for $\|\theta\| = 1$, and $\lambda_{\min}(\nabla^2 F(\theta)) \geq \eta$ if $\|\nabla F(\theta)\| \leq \sigma$, where $\lambda_{\min}(A)$ is the minimum eigenvalue of A. Moreover $L^* := \sup_{\theta\in\Theta}\|\nabla^3 F(\theta)\|_{op} < \infty$.*

One consequence of Assumption 1 is that all the stationary points of $F(\theta)$ in Θ are finite and well-separated [30]. We denote these stationary points as $(\theta_0^*, \theta_1^*, \ldots \theta_K^*)$. Without loss of generality, let θ_0^* be the global minimum of $F(\theta)$.

For simplicity of discussion, we assume that $\mathcal{T}$ is a deterministic optimization algorithm that is guaranteed to converge to a stationary point, and where it convergence to is determined by the initial point. Recall that $\mathcal{T}(\theta^0)$ denotes the stationary point to which $\mathcal{T}$ converges starting from θ^0. Then, $\mathcal{T}$ can be viewed as a deterministic mapping from the parameter space Θ to the set of stationary points $\{\theta_0^*, \theta_1^*, \ldots, \theta_K^*\}$. Our goal is to find a θ^0 such that $\mathcal{T}(\theta^0) = \theta_0^*$.

Given the deterministic optimization algorithm $\mathcal{T}$, the attraction region of the global minimum θ_0^* can be defined as

$$\mathbb{B}_0^* = \{\theta \in \Theta : \mathcal{T}(\theta) = \theta_0^*\}.$$

In general, $\mathbb{B}_0^*$ cannot be characterized without $\mathcal{T}$. On the other hand, it is well-known that for many optimization algorithms, $\mathcal{T}(\theta^0) = \theta_0^*$ if θ^0 is in a neighborhood of θ_0^* within which $F(\theta)$ is strongly convex. This indicates that a proper neighborhood of θ_0^* can be used as a substitution of $\mathbb{B}_0^*$. We formalize this idea as follows.

Assumption 2. *There exists a ball centered at θ_0^* with radius r, $\mathcal{B}_r(\theta_0^*) = \{\theta : \|\theta - \theta_0^*\| \leq r\}$, such that $\mathcal{B}_r(\theta_0^*) \subseteq \mathbb{B}_0^*$ and $F(\theta)$ is μ-strongly convex in $\mathcal{B}_r(\theta_0^*)$.*

Note that Assumption 2 may come as a consequence of Assumption 1. In particular, $F(\theta)$ is $\eta/2$-strongly convex in $\mathcal{B}_r(\theta_0^*)$ when $r \leq \eta/(2L^*)$.

The assumptions above allow us to derive an upper bound for the failure rate of the benchmark random start algorithm. Let $\mathcal{F}_b$ denote the random event that among the M initial points drawn uniformly at random from Θ, none of them leads to θ_0^* [1]. Then, $\mathbb{P}(\mathcal{F}_b) \leq (1 - \mathbb{P}(\theta \in \mathcal{B}_r(\theta_0^*)))^M$, where θ follows a Uniform distribution on Θ. Since $\mathbb{P}(\theta \in \mathcal{B}_r(\theta_0^*)) = \Omega(r^d)$, in order for $\mathbb{P}(\mathcal{F}_b)$ to be lower than a user-specified confidence level ρ, we need $M = \Omega(|\log\rho| r^{-d})$, which has an exponential dependence on d.

Our next assumption concerns the uniqueness of the global minimum.

[1] In $\mathcal{F}_b$, b stands for baseline and random start is the baseline algorithm.

Assumption 3. *There exists a constant $\alpha > 0$, such that for all $\theta \notin \mathcal{B}_r(\theta_0^*)$, $F(\theta) - F(\theta_0^*) \geq \alpha$.*

We refer to α as the optimality gap. In Section 3.4, we will discuss what can be achieved if Assumption 3 does not hold.

The basic idea of our data outsourcing and exploration scheme is to approximate $F(\theta)$ via its sample average $\widehat{F}_n(\theta)$ and then use the global minimum of $\widehat{F}_n(\theta)$ as the initial point to optimize $F(\theta)$. A key question is that in order for $\widehat{F}_n(\theta)$ to be a good approximation of $F(\theta)$, how many data points need to be outsourced? A similar estimation problem has been studied in [30]. We adapt some of their results to our setting, which involves the following regularity conditions on the loss function and the data variability.

Assumption 4. *The following hold for some τ, c_h:*

(1) *The loss function for each data point is τ^2-sub-Gaussian. Namely, for any $\lambda \in \mathbb{R}$, and $\theta \in \Theta$,*

$$\mathbb{E}\Big[\exp\big(\lambda\big(f(\theta;X) - \mathbb{E}_{X\sim\xi}[f(\theta;X)]\big)\big) \Big] \leq \exp\Big\{\frac{\tau^2\|\lambda\|^2}{2}\Big\}.$$

(2) *The gradient of the loss is τ^2-sub-Gaussian. Namely, for any $\lambda \in \mathbb{R}^d$, and $\theta \in \Theta$,*

$$\mathbb{E}\Big[\exp\big\langle \lambda, \nabla_\theta f(\theta;X) - \mathbb{E}_{X\sim\xi}[\nabla_\theta f(\theta;X)]\big\rangle \Big] \leq \exp\Big\{\frac{\tau^2\|\lambda\|^2}{2}\Big\}.$$

(3) *The Hessian of the loss, evaluated on a unit vector, is τ^2-sub-exponential. Namely, for any $\lambda \in \mathbb{R}^d$ with $\|\lambda\| \leq 1$, and $\theta \in \Theta$,*

$$\mathbb{E}\Big[\exp\Big\{\frac{1}{\tau^2}\big|\mathcal{Z}_{\lambda,\theta}(X) - \mathbb{E}_{X\sim\xi}[\mathcal{Z}_{\lambda,\theta}(X)]\big|\Big\} \Big] \leq 2,$$

where $\mathcal{Z}_{\lambda,\theta}(X) = \langle \lambda, \nabla^2_\theta f(\theta;X)\lambda\rangle$.

(4) *There exists J_*, satisfying $J_* \leq \tau^3 d^{c_h}$, such that*

$$\mathbb{E}_{X\sim\xi}\Big[\sup_{\theta_1,\theta_2\in\Theta,\theta_1\neq\theta_2} \frac{\|\nabla_\theta f(\theta_1;X) - \nabla_\theta f(\theta_2;X)\|}{\|\theta_1-\theta_2\|} \Big] \leq J_*,$$

$$\mathbb{E}_{X\sim\xi}\Big[\sup_{\theta_1,\theta_2\in\Theta,\theta_1\neq\theta_2} \frac{\|\nabla^2_\theta f(\theta_1;X) - \nabla^2_\theta f(\theta_2;X)\|_{op}}{\|\theta_1-\theta_2\|} \Big] \leq J_*.$$

(5) *There exists $\theta^* \in \Theta$, such that $\|\nabla F(\theta^*)\|, \|\nabla^2 F(\theta^*)\|_{op} \leq H \leq \tau^3 d^{c_h}$.*

Assumption 4 allows us to find a close approximation of F, which is formally defined as follows.

Definition 3.1. We say $\widehat{F}_n(\theta)$ is a δ-approximation of $F(\theta)$, if both F and $\widehat{F}_n$ have $K+1$ stationary points, denoted by $\{\theta_i^*\}_{i=0,\ldots,K}$ and $\{\hat{\theta}_i^*\}_{i=0,\ldots,K}$, and the following inequalities hold

$$\sup_{\theta\in\Theta}|F(\theta) - \widehat{F}_n(\theta)| \leq \delta, \ \sup_{\theta\in\Theta}\|\nabla F(\theta) - \nabla\widehat{F}_n(\theta)\| \leq \delta,$$

$$\sup_{\theta\in\Theta}\|\nabla^2 F(\theta) - \nabla^2\widehat{F}_n(\theta)\|_{\text{op}} \leq \delta, \ \text{ and } \max_{0\leq i\leq K}\|\theta_i^* - \hat{\theta}_i^*\| \leq \delta.$$

The next lemma characterizes the minimal sample size required to achieve a δ-approximation.

Lemma 3.2. *Assume that Assumptions 1 and 4 hold. Consider a given confidence level $\rho \in (0,1)$ and a given approximation accuracy δ. Let $C = C_0(c_h \vee 1 \vee \log(\tau/\rho)) = O(|\log\rho|)$, where C_0 is some constant, and $\eta_* = (\sigma^3/\tau^2) \wedge (\eta^2/\tau^4) \wedge (\eta^4/(L^*\tau)^2) = \Omega(1)$. Then, when*

$$n \geq \max\Big\{\frac{Cd\tau^2\log n}{\delta^2}, 4Cd\log n\Big(\frac{\tau^2}{\sigma^2}\wedge\frac{\tau^4}{\eta^2}\Big), \frac{4Cd\log n}{\eta_*^2}, Cd\log d\Big\} := n(\delta,\rho,d),$$

with probability at least $1-\rho$, $\widehat{F}_n(\theta)$ is a δ-approximation of $F(\theta)$.

The proofs of Lemma 3.2 and all subsequent results are provided in Appendix A. Lemma 3.2 shows that to achieve a δ-approximation of $F(\theta)$ with probability $1-\rho$, the required sample size is

$$n(\delta, \rho, d) = \tilde{\Omega}\Big(d \log(1/\rho)\delta^{-2}\Big). \tag{4}$$

Here, $\tilde{\Omega}$ means that we ignore the $\log n$ terms.

3.1 Performance of the sampling approach

Let $\mathcal{F}_0$ denote the event that using the initial point(s) constructed based on Algorithm 2, the in-house optimization algorithm $\mathcal{T}$ fails to find θ_0^*. In this section, we establish an upper bound for $\mathbb{P}(\mathcal{F}_0)$. Recall that samples of initial points are drawn from $\pi_\beta(\theta)$ defined in (1). We first show that when β is large enough, a random sample $\tilde{\theta}_\beta$ from $\pi_\beta(\theta)$ has a high chance to fall into $\mathcal{B}_r(\theta_0^*)$.

Proposition 3.3. *Suppose Assumptions 1-4 hold and the approximation accuracy δ satisfies $\delta < \mu \wedge r \wedge \alpha/4$. If $\widehat{F}_n(\theta)$ is a δ-approximation of $F(\theta)$ and $\beta = \Omega(r^{-2})$, for a random sample $\tilde{\theta}_\beta$ from π_β,*

$$\mathbb{P}\big(\tilde{\theta}_\beta \notin \mathcal{B}_r(\theta_0^*)\big) = \exp(-\beta\alpha/2 + d\log\beta + C_d),$$

where $C_d := d\log d + d\log(H + L^* + \delta) + 3d$.

Proposition 3.3 shows that $\mathbb{P}\big(\tilde{\theta}_\beta \notin \mathcal{B}_r(\theta_0^*)\big)$ decays exponentially in β. This probability is also affected by α, the optimality gap, as well as the dimension d. In practice, we cannot choose β arbitrarily large as we have to consider the computational cost of the associated sampling algorithm (e.g., the rate of convergence to stationary of the MCMC algorithm). In general, when β increases, the difficulty of sampling from $\pi_\beta(\theta)$ increases. In practice, we want to find a β that balances the estimation accuracy and the sampling efficiency. We discuss this further in Section 3.3.

One challenge when applying Proposition 3.3 to the sampling approach is that in practice we may not be able to sample from $\pi_\beta(\theta)$ exactly. Many MCMC algorithms can only draw samples from a distribution that is "close" to $\pi_\beta(\theta)$ (e.g., ULA). To handle this issue, we impose the following assumption as a relaxation to the requirement of sampling from $\pi_\beta(\theta)$ exactly.

Assumption 5. *There is a Markov chain based sampler $\widehat{\mathcal{M}}$ such that for any fixed $\delta_\beta \in [0, 1)$, starting from any $\theta_0 \in \Theta$, $\widehat{\mathcal{M}}$ can draw samples that satisfy $\|\mathbb{P}(\theta_i \in \cdot|\theta_{i-1}) - \pi_\beta(\cdot)\|_{TV} \le \delta_\beta$, where θ_i's are consecutive samples from $\widehat{\mathcal{M}}$.*

In addition, when we draw multiple samples from a Markov chain induced by the underlying MCMC algorithm, the samples are correlated. The following lemma justifies the quality of the sampler $\widehat{\mathcal{M}}$ under Assumption 5 even when the output θ_i's are correlated.

Lemma 3.4. *Given a measurable set B and a distribution π_β with $\pi_\beta(B) > 0$, suppose there exists a sampler $\widehat{\mathcal{M}}$ satisfying Assumption 5. If we draw m samples from $\widehat{\mathcal{M}}$, then*

$$\mathbb{P}(\theta_1 \notin B, \ldots, \theta_m \notin B) \le (\pi_\beta(B^c) + \delta_\beta)^m.$$

The following theorem then comes as a consequence of Proposition 3.3 and Lemma 3.4. Recall that $C_d = d\log d + d\log(H + L^* + \delta) + 3d$.

Theorem 3.5. *Consider Algorithm 2. Suppose Assumptions 1-5 hold. For an arbitrary confidence level $\rho \in (0, 1)$, let $\delta = \mu \wedge r \wedge \alpha/4$. If the sample size $n \ge n(\delta, \rho, d) = \Omega(d\log(1/\rho)\delta^{-2})$ and the inverse temperature $\beta = \Omega(r^{-2})$, then*

$$\mathbb{P}(\mathcal{F}_0) \le \rho + \big(\exp(-\beta\alpha/2 + d\log\beta + C_d) + \delta_\beta\big)^m. \tag{5}$$

Theorem 3.5 shows that $\mathbb{P}(\mathcal{F}_0) - \rho$ decays exponentially fast as the inverse temperature β or sample size m increases.

3.2 Performance of the optimization approaches

We first provide an analysis of the SAO approach in Algorithm 3. Let $\mathcal{F}_1$ denote the random event that the output of Algorithm 3-SAO fails to find θ_0^*. The result is largely the same as Theorem 3.5, although the proof is slightly more difficult.

Theorem 3.6. *Consider Algorithm 3-SAO. Suppose Assumptions 1-5 hold. For an arbitrary confidence level $\rho \in (0,1)$, let $\delta = \mu \wedge r \wedge \alpha/4$. If the sample size $n \geq n(\delta, \rho, d) = \Omega(d\log(1/\rho)/\delta^2)$ and the inverse temperature $\beta = \Omega(r^{-2})$, then*

$$\mathbb{P}(\mathcal{F}_1) \leq \rho + \left(\exp(-\beta\alpha/2 + d\log\beta + C_d) + \delta_\beta\right)^m.$$

We next analyze the annealing approach in Algorithm 3. Let $\mathcal{F}_2$ be the random event that Algorithm 3-annealing fails to find θ_0^*. The annealing approach needs more restrictions than the SAO approach. This is because: in order to generate a good starting point, one of the samples needs to be close to θ_0^*. Moreover, its $\widehat{F}_n$-value needs to be lower than the other samples. This can be formulated as requiring a smaller radius r_0 for the attraction neighborhood:

Theorem 3.7. *Consider Algorithm 3-annealing. Suppose Assumptions 1-5 hold. For an arbitrary confidence level $\rho \in (0,1)$, let r_0 be chosen such that $r_0^2 \sup_{\theta\in\Theta} \|\nabla^2 F(\theta)\|_{op} < \alpha$ and $\delta = \mu \wedge r_0 \wedge (\alpha/4)$. If the sample size $n \geq n(\delta, \rho, d) = \Omega(d\log(1/\rho)/\delta^2)$ and the inverse temperature $\beta = \Omega(r_0^{-2})$, then*

$$\mathbb{P}(\mathcal{F}_2) \leq \rho + \left(\exp(-\beta\alpha/2 + d\log\beta + C_d) + \delta_\beta\right)^m.$$

3.3 Choosing n and β and the computational costs

Theorems 3.5 – 3.7, combined with Lemma 3.2, provide us with insights on how to choose n and β to achieve a user specified confidence level $1-\rho$. In particular, we highlight how the required n and β depend on some important problem-specific parameters. Recall the following problem-specific parameters: 1) μ quantifies the convexity of the function around the global minimum and r measures the size of the neighborhood around the global minimum in which F is strongly convex. They are introduced in Assumption 2. α is the optimality gap, i.e., it characterizes how well separated is the global minimum from other local minima, and is introduced in Assumption 3. 2) d is the dimension of θ. 3) τ quantifies the data variability and is introduced in Assumption 4. The required sample size n satisfies

$$n = \tilde{\Omega}\left(\log(1/\rho)\tau^4 d(\mu \wedge r \wedge (\alpha/4))^{-2}\right)$$

We note that the required sample size n does not depend on the population size N. The smaller the value of μ, α, or r is, the larger n needs to be. The general intuition is that if the global minimum has a very small attraction neighborhood, or is hard to differentiate from other local minima, we need a larger outsourcing sample. In addition, n also needs to increase with the dimension of θ. Lastly, the more variable the data, i.e., the larger the value of τ, the larger n needs to be. As for β, it needs to satisfy

$$\beta = \tilde{\Omega}\left(r^{-2} + d/\alpha + (m\alpha)^{-1}\log(1/\rho)\right).$$

This indicates that a smaller value of r or α can lead to a larger required value of β.

Larger values of n or β in general lead to a larger computational cost for the external computing facility. For example, when using ULA defined in Algorithm 1 to draw samples from $\pi_\beta(\theta)$, n affects the cost per update of the underlying Markov chain, i.e., the cost of evaluating $\nabla\widehat{F}_n(\theta)$ is linear in n; β affects how fast the Markov chain converges to the target stationary distribution. Let $K(\beta, \delta_\beta)$ denote the number of iterations required to achieve $\|\hat{\pi}_{\beta,K} - \pi_\beta\|_{TV} \leq \delta_\beta$, where $\hat{\pi}_{\beta,K}$ is the marginal distribution of ULA at iteration K. Then, the computational cost for the external computing facility is of the order $dnK(\beta, \delta_\beta)$. We next analyze $K(\beta, \delta_\beta)$ in the context of ULA. The convergence of

ULA has been studied under various problem settings [12, 29, 45]. For general non-convex $\widehat{F}_n(\theta)$, Theorem 1 in [45] and Pinsker's inequality imply the following bound for $K(\beta, \delta_\beta)$:

Lemma 3.8. *Suppose π_β satisfies log Sobolev inequality with constant γ_β and $\nabla \widehat{F}_n(\theta)$ is L-Lipschitz. Then for ULA with step size $h \le \gamma_\beta \delta_\beta^2/(8d\beta L^2)$, we have*

$$K(\beta, \delta_\beta) = O\left(\frac{dL^2\beta^2|\log \delta_\beta|}{\gamma_\beta^2 \delta_\beta^2}\right).$$

Finding the constant γ_β can be challenging in general [35]. Meanwhile, the work [31] shows that as $\beta \to \infty$, $\gamma_\beta = O(1/\beta)$. This suggests that the computational cost for the external computing facility scales as $nd^2\beta^3$ in this case.

In actual implementations, many of the parameters required in our calculation of n and β may not be known (e.g., α, r, μ). In addition to the directional insights provided by our theoretical analysis, we also conduct some sensitivity analysis on n and β numerically to provide further guidance on how to choose them in practice in Section 4.1. Overall, we find that both n and β have "diminishing returns" as their value increases. We can achieve a reasonably good performance with an n that depends linearly on the dimension d and $\beta = O(1)$.

Given our discussion above, we next provide a summary of the computational costs of our proposed data outsourcing framework. Recall that $c(\mathcal{T})$ denotes the (worst case) computational cost for an optimization algorithm $\mathcal{T}$ to find a stationary point. Similar to the sampling cost, $c(\mathcal{T})$, in general, depends on the choice of algorithm, the dimension d, the size of the dataset N, and the value of the initial point. But in general, the dependence on these problem-specific parameters is linear or at most polynomial. For example, if gradient descent is applied, it is well known that the algorithm can find an ϵ-accurate stationary point with $c(\mathcal{T}) = O(dN/\epsilon)$ [33]. Since $F(\theta)$ is non-convex, one may need to try many different initial points to find the global minimum. Our proposed framework can help reduce the number of initial points required by smartly choosing these initial points, i.e., doing more advanced exploration. This imposes extra computational costs for the external computing facility, but saves in-house computational costs for the data organization. In particular, consider using ULA for SIPS and ULA combined with gradient descent for OIPS-SAO. Then

- In SIPS, the external computing facility will generate m samples with a total cost of order $nd^2\beta^3 m$. The data organization will spend $mc(\mathcal{T})$ in the exploitation/optimization stage.
- In OIPS-SAO, the external computing facility will generate m samples with a total cost of order $nd^2\beta^3 m + dmn/\epsilon$. The data organization will spend $c(\mathcal{T})$ in the optimization stage.

We note that m is in general a small number that does not depend on the problem-specific parameters. The above computational costs are to be compared to an in-house cost of $Mc(\mathcal{T})$, where $M = \Omega(r^{-d}\log(1/\rho))$, when using random initialization. We also note that the external computing facilities (e.g., cloud computing services) are in general computationally more well-equipped than the data organization.

We conclude this section by providing two further remarks

Remark 1. *Using data outsourcing and sampling-based method to find better initialization not only increases our chance of starting in the "right" neighborhood of the global minimum, but may also get us closer to the global minimum or other stationary points to start with when applying $\mathcal{T}$. In this case, it also helps reduce $c(\mathcal{T})$. When applying gradient descent with a properly chosen step size to a convex function, $c(\mathcal{T}) = O(dN\|\theta_0 - \theta\|/\epsilon)$, and when applying it to a strongly convex function, $c(\mathcal{T}) = O(dN\log(\|\theta_0 - \theta\|/\epsilon))$ [33]. In most examples tested in our numerical experiments, the latter*

benefit is relatively small. However, for the deep neural network example in Section 4.4, this benefit can be substantial.

Remark 2. *Note that n and β are determined by our desired confidence level $1-\rho$. In particular, it can be interpreted as the minimum sample size and inverse temperature required to achieve the desired accuracy. Let U_1 denote the payoff of finding the optimal solution and U_0 denote the payoff of not being able to find the optimal solution. We also denote h as the unit computational cost charged by the external computing facility. Then, we can determine the optimal outsourcing sample size $n(\delta, \rho^*, d)$ by solving the following utility maximization problem for ρ^*:*

$$\max_{\rho}(1-\rho)U_1 + \rho U_0 + hn(\delta,\rho,d)d^2\beta(\rho,d)^3 m$$

where $\beta(\rho,d) = C_0(\log(1/\rho)+d)/\alpha$.

3.4 Extension to ϵ-Global Minimum

One major constraint in our previous analysis is Assumption 3–the global minimizer is unique with an optimality gap of $\alpha > 0$. In practice, there can be multiple local minima that have function values very close to the global minimum. In this setting, it can be too ambitious to find the global minimum and it may be more reasonable to find an approximately optimal solution. In particular, given a user-specified accuracy level ϵ, we are interested in finding a local minimum whose objective value is within ϵ-distance from the optimal objective value, i.e., θ_i^* such that $F(\theta_i^*) \le F(\theta_0^*) + \epsilon$. We call such a local minimum an ϵ-global minimum of $F(\theta)$. In this subsection, we conduct performance analysis for our algorithms to find an ϵ-global minimum.

Let

$$\mathcal{J}_\epsilon^* = \left\{i : F(\theta_i^*) \le F(\theta_0^*) + \epsilon\right\}$$

be the index set of the ϵ-global minima. We also introduce the "attraction region" of the ϵ-global minima:

Definition 3.9 (Attraction region of ϵ-global minimums). Given an optimization algorithm $\mathcal{T}$, we define the attraction region of ϵ-global minimums of $F(\theta)$ as

$$\mathbb{B}_\epsilon^* = \left\{\theta \in \Theta : F\big(\mathcal{T}(\theta)\big) \le F(\theta_0^*) + \epsilon\right\}.$$

By definition, the optimization algorithm $\mathcal{T}$ converges to an ϵ-global minimum if and only if it starts with an initial point in $\mathbb{B}_\epsilon^*$. However, same as before, $\mathbb{B}_\epsilon^*$ is hard to characterize directly. Thus, we consider the following subset as a substitution:

$$\mathcal{B}_{\epsilon,r_\epsilon} := \bigcup_{i\in\mathcal{J}_\epsilon^*} \mathcal{B}_{r_\epsilon}(\theta_i^*) \subseteq \mathbb{B}_\epsilon^*.$$

To be concise, we only present the analysis for the annealing-based optimization approach (Algorithm 3-annealing). The results for the other algorithms are similar. Let $\mathcal{F}_{\epsilon,2}$ be the random event that the output of Algorithm 3-annealing fails to find an ϵ-global minimum of $F(\theta)$.

Theorem 3.10. *Suppose Assumptions 1, 4 and 5 hold. For any user-specified accuracy $\epsilon > 0$, pick r_ϵ such that $\sup_{\theta\in\Theta}\|\nabla^2 F(\theta)\|_{op}\cdot r_\epsilon^2 \le \epsilon$. In addition, assume the approximation accuracy δ satisfies $\delta < \mu \wedge r_\epsilon \wedge \epsilon/4$. For an arbitrary confidence level $\rho \in (0,1)$, if the sample size $n \ge n(\delta,\rho,d) = O(d\log(1/\rho)/\delta^2)$ and the inverse temperature $\beta = \Omega(r_\epsilon^{-2})$, then*

$$\mathbb{P}(\mathcal{F}_{3\epsilon,2}) \le \rho + \Big(\exp\big(-\beta\epsilon/2 + d\log\beta + C_d\big) + \delta_\beta\Big)^m.$$

Note that Theorem 3.10 establishes a similar performance guarantee as Theorem 3.7. However, the convergence rate in Theorem 3.10 is determined by a user-specified accuracy level ϵ instead of the optimality gap α.

4 NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments to demonstrate the performance of our proposed framework. We choose random start as a benchmark for comparison, which resembles the state-of-art approach when no outsourcing is available. Our first two examples are relatively simple problems, where the goal is to demonstrate the main ideas behind our framework. The last two examples are more sophisticated applications.

4.1 One-dimensional Nonconvex Function

In this section, we use a one-dimensional nonconvex function to demonstrate the robustness of our algorithms with respect to two key hyper-parameters: the outsourcing sample size n and the inverse temperature β. Specifically, we consider minimizing a polynomial function of the form

$$F(\theta) = a\theta^4 + b\theta^2 + c\theta,$$

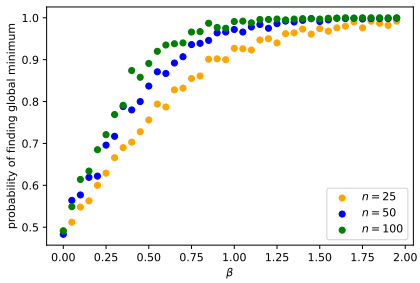
and assume the coefficients (a, b, c) are unknown but we can draw samples from $\mathcal{N}((a, b, c)^\top, \tau^2 I_3)$, where $\mathcal{N}(a, b)$ denotes a Gaussian distribution with mean vector a and covariance matrix b. Let the i-th samples be denote by $(\hat{a}_i, \hat{b}_i, \hat{c}_i)$. With an outsourcing dataset of size n, we have the following empirical objective:

$$\widehat{F}_n(\theta) = \left(\frac{1}{n}\sum_{i=1}^{n}\hat{a}_i\right)\theta^4 + \left(\frac{1}{n}\sum_{i=1}^{n}\hat{b}_i\right)\theta^2 + \left(\frac{1}{n}\sum_{i=1}^{n}\hat{c}_i\right)\theta.$$

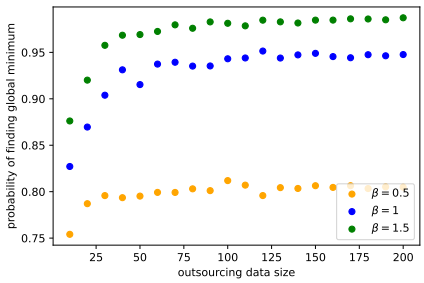
In this experiment, we first set $(a, b, c) = (1, -5, -1)$, in which case $F(\theta)$ has two local minima, with one being the global minimum. Since the target distribution $\pi_\beta(\theta) \propto \exp\{-\beta\widehat{F}_n(\theta)\}$ is one-dimensional, we apply the acceptance-rejection method with the proposal distribution been uniform$[-2, 2]$ [4] to generate independent samples from π_β exactly. For illustration, we focus on the sampling procedure (SIPS) with $m = 1$ and study the effect of n and β. In what follows, we refer to the probability of falling into the attraction basin of the global minimum as the success probability, i.e., it is the probability that starting from the given initial point, the in-house optimization finds the global minimum. 1000 replications are used in each scenario to estimate the success probability.

Figures 2 (a) and (c) show that the success probability increases as β increases. However, there is a diminishing return. In this example, $\beta = 1$ already leads to very good performance (>90% success probability) for different values of n and τ. In addition, given a target success probability, the (minimum) required β decreases with the outsource sample size n, and increases with the sample noise τ. Figure 2 (b) shows that the success probability increases as n increases, but there is again a diminishing return. In this example, $n = 50$ already leads to very good performance as long as β is large enough, i.e., $\beta \geq 1$. When β is too small, i.e., $\beta = 0.5$, the success probability converges to around 80% as n increases, and increasing n beyond 50 has almost no impact on improving the success probability. Lastly, Figure 2 (d) tests very large values of τ and shows that when the data are very noisy, we need a substantially larger outsourcing sample size to achieve good performance.

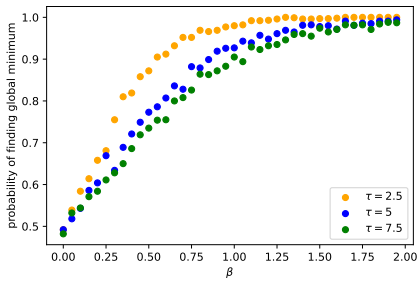
We next study the effect of the geometry of the objective function. In particular, we study the robustness of β and n when we vary the optimality gap α or the radius r of $\mathcal{B}_r(\theta_0^*)$ in Assumptions 2 and 3. We observe from Figures 3 (a) – (c) that as the optimality gap α decreases, larger n and β are required to achieve a good success probability. Figures 3 (d) – (f) show that r does not affect the performance of n or β much. This could be because, in this example, the size of the attraction region $\mathbb{B}_0^*$ does not change as r decreases.



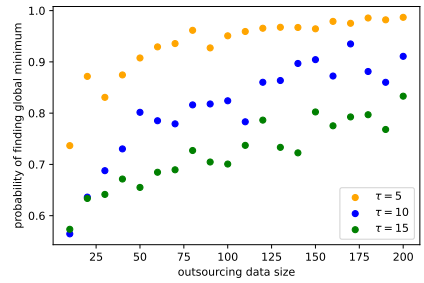
(a) Sensitivity of β ($\tau = 5$)



(b) Sensitivity of n ($\tau = 5$)



(c) Sensitivity of β ($n = 100$)



(d) Sensitivity of n ($\beta = 1.5$)

Fig. 2. Sensitivity analysis for n and β for different values of τ

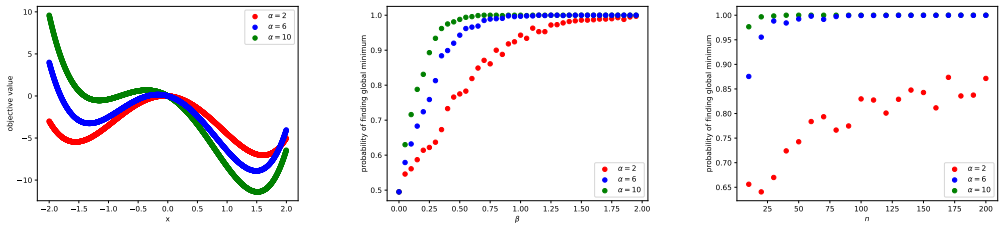
4.2 Gaussian Mixture Density

We study the problem of finding the mode with the largest probability density of a Gaussian mixture model using kernel density estimation. In particular, consider the objective function

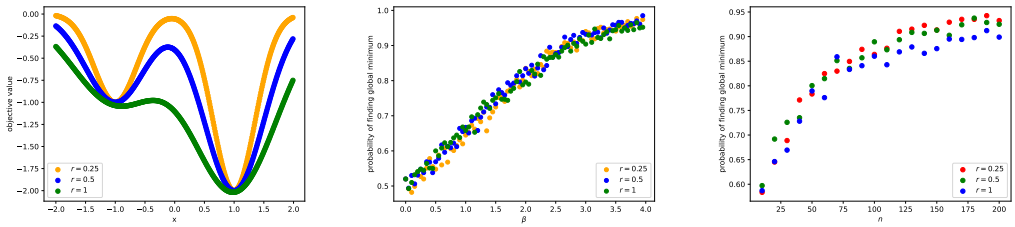
$$F(\theta) = \mathbb{E}_X\left[(2\pi\tau)^{-d/2} \cdot \exp\left\{-\frac{\|\theta - X\|^2}{2\tau^2}\right\}\right].$$

We assume X follows a Gaussian mixture distribution, that is, $X \sim \mathcal{N}(c_i, \tau^2 I_d)$ with probability p_i for $1 \le i \le J$, where the mixing weights p_i satisfy $0 < p_i < 1$ and $\sum_{i=1}^{J} p_i = 1$. When c_i's are well-separated, $F(\theta)$ has multiple local minima located near c_i's. In this case, the selection of the initial point is critical in optimizing $F(\theta)$.

We start with an example with $d = 5$ and $J = 10$, and implement SIPS, OIPS-annealing, and OIPS-SAO with $n = 100$, $\beta = 2$, and $m = 50$. To draw samples from $\pi_\beta(\theta)$, we use ULA with stepsize $h = 0.1$ and run 1000 iterations in total. For OIPS-SAO, we further apply gradient descent (GD) for 100 iterations to obtain $\widehat{\mathcal{T}}(\theta_i)$, $i = 1, \ldots, m$. We also consider two benchmarks: random start with a single initial point (to be compared to OIPS) and random start with $m = 50$ initial points (to be compared to SIPS). For the in-house optimization, i.e., to optimize $F(\theta)$, given an initial point, we apply GD, where the gradient is estimated using the batch mean with batch size 1000. In this optimization phase, we run 20 iterations of GD and take the objective value at the last iteration as the convergent function value. Figure 4 shows the distribution of convergent function values under different algorithms based on 100 replications of each algorithm. We observe that SIPS and



(a) Objective function with different α (b) Sensitivity of β ($n = 100$) (c) Sensitivity of n ($\beta = 2$)



(d) Functions with different r (e) Robustness of β (f) Robustness of n

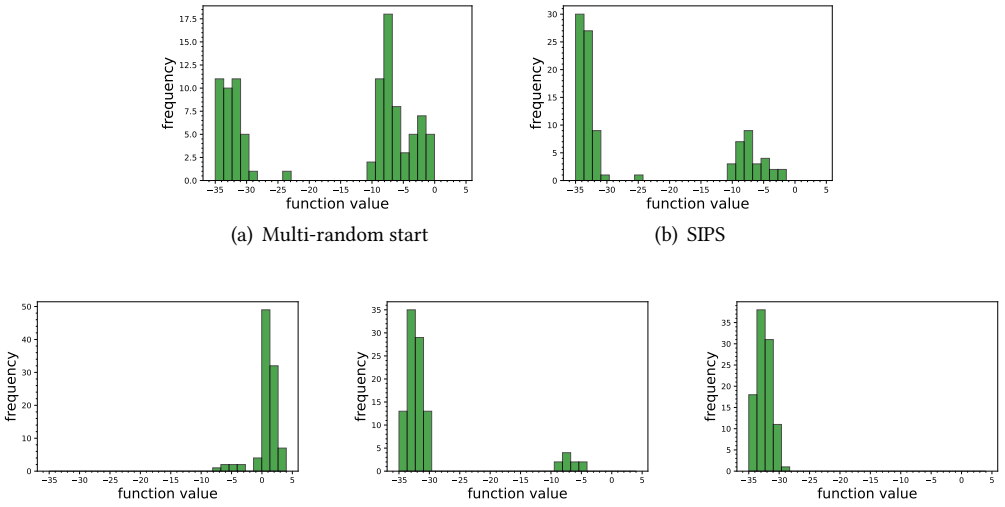
Fig. 3. Sensitivity analysis of n and β for different values of α and r

OIPS outperform their corresponding random start benchmarks significantly. In particular, when comparing random start from multiple initial points with SIPS, the average convergent objective values are -16.7 and -25.3 respectively (p-value = 0.000). When comparing random start with a single initial point to OIPS-annealing, the average convergent objective values are 0.71 and -29.8 respectively (p-value = 0.000). OIPS-SAO is achieving even better performance with an average convergent objective value of -32.5.

We further test the performance of our algorithm for different problem dimensions. In particular, we fix the number of modes $J = 3$ and vary d from 10 to 35. Motivated by our theoretical analysis, we set the outsourcing sample size at $n = 4d$, i.e., linear in d. We fix $\beta = 2$. For simplicity of demonstration, we only OIPS-annealing algorithm with a single random start (both require only a single round of in-house optimization). Figure 5 shows the distribution of convergent function values under different algorithms based on 100 replications of each algorithm. We observe that in all scenarios tested, the OIPS-annealing algorithm finds the global minimum with a high probability, suggesting $n = 4d$ is a good outsourcing sample size choice in this case. In addition, OIPS-annealing outperforms the random start benchmark significantly.

4.3 Generalized multinomial logit model

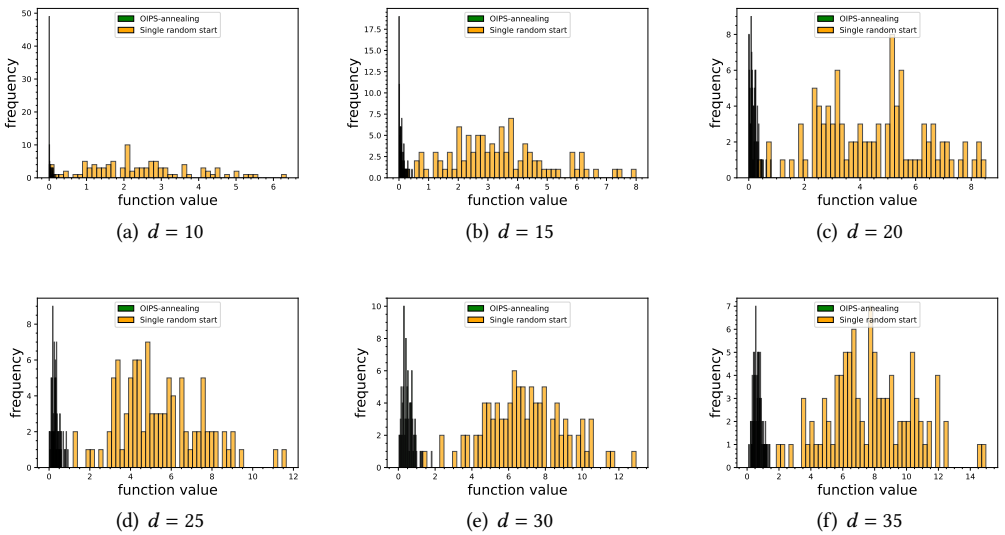
We study an application of our algorithms for maximum likelihood estimation of the generalized multinomial logit (GMNL) model. The multinomial logit model (MNL) is a classic model to study consumer choices. GMNL is an extension of MNL, which accommodates the scaling heterogeneity in utility coefficients through an individual-specific scaling factor [14]. Such a generalization makes the negative log-likelihood function nonconvex. In practice, GD or BFGS with random starts is employed for the estimation [43].



(a) Multi-random start (b) SIPS

(c) Single random start (d) OIPS-annealing (e) OIPS-SAO

Fig. 4. Histogram of convergent function values of mixture Gaussian density ($d = 5, J = 10$).



(a) $d = 10$ (b) $d = 15$ (c) $d = 20$

(d) $d = 25$ (e) $d = 30$ (f) $d = 35$

Fig. 5. Histogram of convergent function values of mixture Gaussian density under OIPS-annealing versus random start ($J = 3$, d varies from 10 to 35).

Suppose that there are J products. The utility that customer i chooses alternative j is $U_{ij} = x_j^\top \phi_i + \epsilon_{ij}$, where x_j is a p-dimensional vector of the attributes of product j, $\phi_i \in \mathbb{R}^p$ is the vector of utility coefficients, and ϵ_{ij} is an idiosyncratic error term that follows a standard Gumbel distribution. The customer chooses the product with the highest utility and the probability of choosing product

k is $P_{ik} = \exp(x_k^\top \phi_i)/\sum_{j=1}^J \exp(x_j^\top \phi_i)$. In GMNL, $\phi_i = \exp\{z_i^\top \psi + \nu_i\}\phi$, where z_i is a q-dimensional vector of customer characteristics, ψ is a q-dimensional heterogeneity coefficient, and ν_i is an independent random shock that follows the standard Gaussian distribution. Let the binary variable $y_{ij} \in \{0, 1\}$ denote whether customer i chooses product j. Then the likelihood for customer i is

$$L_i = \mathbb{E}\Big[\prod_{k=1}^J \Big(\frac{\exp(x_k^\top \phi_i)}{\sum_{j=1}^J \exp(x_j^\top \phi_i)}\Big)^{y_{ik}}\Big].$$

We use simulation, i.e., simulated data, to approximate the above expectation. The model parameter $\theta = (\phi, \psi)$ can be estimated by minimizing the empirical negative log-likelihood function

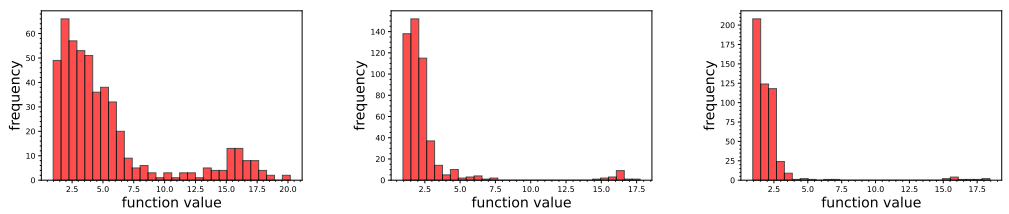
$$F(\theta) = -\frac{1}{N}\sum_{i=1}^N \log\Big(\frac{1}{R}\sum_{r=1}^R \prod_{k=1}^J \Big(\frac{\exp(x_k^\top \phi_i^{[r]})}{\sum_{j=1}^J \exp(x_j^\top \phi_i^{[r]})}\Big)^{y_{ik}}\Big), \tag{6}$$

where $\phi_i^{[r]} = \exp\{z_n^\top \psi + \nu_n^{[r]}\}\phi$, which is the r-th draw from the distribution of ϕ_i, and R is the total number of random draws. In our experiment, we use $R = 100$.

We consider an instance with dimension parameters $p = 10, q = 5$, and $J = 5$ products, i.e., $d = p + q = 15$. We set the true parameters $\phi^* = (1, \ldots, 1, -1, \ldots, -1)$ and $\psi^* = (1, \ldots, 1)$, and generate product attributes x_j and agent characteristics z_i from the standard Gaussian distribution. Then, we simulate the agents' choices using the GMNL model and obtain choice data y_{ij}. The generated data contain $N = 1000$ customers and takes the form $\{x_j, z_i, y_{ij}\}_{1\le i\le N, 1\le j\le J}$.

For the outsourcing exploration, we apply OIPS-annealing and OIPS-SAO with $n = 100$, $\beta = 2$, and $m = 50$. To sample from $\pi_\beta(\theta)$, we apply ULA with step size 0.1 and run $K = 1000$ iterations in total. For OIPS-SAO, we further apply GD with 500 iterations to find $\widehat{\mathcal{T}}(\theta_i)$, $i = 1, \ldots, m$. For the in-house optimization, we apply GD with stepsize 0.1 to optimize the negative log-likelihood (6). We run GD for 100 iterations, and take the objective value at the last iteration as the convergent function value. We choose random start with a single initial point as the benchmark when no outsourcing is available. Figure 6 shows the distribution of convergent objective values based on 500 repetitions of the algorithms. We observe that OIPS-SAO and OIPS-annealing outperform random start significantly. In particular, comparing random start to OIPS-annealing, the average convergent objective values are 5.28 and 2.55 respectively (p-value = 0.000). OIPS-SAO is performing even better with an average convergent value of 2.09.

We can also compare the computational costs for the external computing facility under different exploration strategies. For random start, we incur zero cost. For OIPS-annealing, the computation cost for the external computing facility is $dnK = 1.5 \times 10^6$. For OIPS-SAO, the computational cost for the external computing facility is $dnK + 500dmn = 3.9 \times 10^7$.



(a) Single random start (b) OIPS-annealing (c) OIPS-SAO

Fig. 6. Histogram of negative log-likelihood of GMNL model

4.4 Image identification with deep neural network

In this section, we consider a classic image classification problem using convolutional neural networks (CNN). When training CNN, we need to minimize a nonconvex loss function of a very high dimension. The most commonly used initialization in the literature is random start (Gaussian or uniform initialization) [19]. In this section, we compare our outsourcing idea with random start.

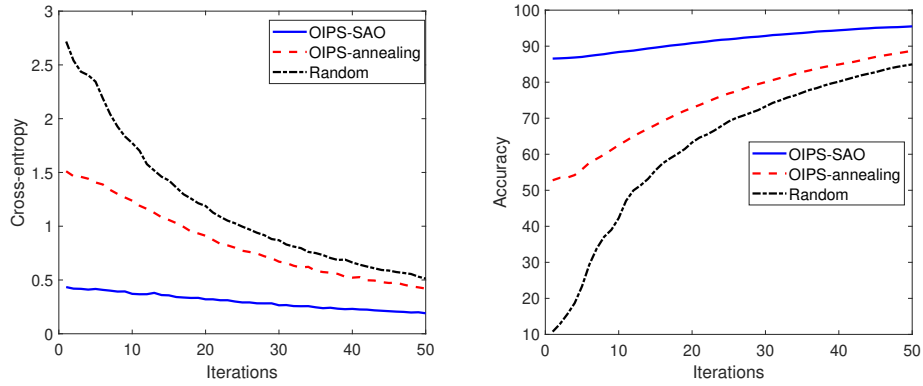
In particular, we train a CNN to classify the handwritten numbers 0-9 based on 28×28 digit images from the dataset – MNIST, which contains 10000 images. We consider a CNN with three convolutional layers, three batch normalization layers, and one fully connected layer. For the i-th convolutional layer, $i = 1, 2, 3$, the number of filters is $2^{(i+2)}$ with size 3×3. Max pooling is used together with the ReLU activation function. For the final fully connected layer, we use the Softmax activation function for classification. Overall, our CNN contains 21690 parameters. In other words, we need to solve an optimization problem of dimension 21690. Lastly, we split 80% data as the training set and 20% as the testing set. Cross-entropy is used as the loss function.

We implement OIPS-annealing and OIPS-SAO with $n = 1000$, $\beta = 2$, and $m = 50$, and compare their performance with the random start (with a single initialization) benchmark. To sample from $\pi_\beta(\theta)$, ULA with stochastic gradient is applied. For OIPS-SAO, we further run stochastic gradient descent with 15 iterations to optimize $\widehat{F}_n$. For the in-house optimization, we use stochastic gradient descent with momentum, and we run the algorithm for 50 iterations. All stochastic gradients are based on batch means with a batch size of 128. Here we use a stochastic gradient instead of the full gradient, because this is a state-of-art approach.

We implement the procedure 100 times and plot the averages performance metrics (the value of the objective function (cross-entropy) on the training set and the corresponding accuracy on the testing set) at different iterations in the in-house optimization stage in Figure 7. We observe that OIPS-SAO and OIPS-annealing achieve better training loss and testing accuracy over random start, with OIPS-SAO performing the best. The performance improvement of our procedures is the largest when the number of iterations at the in-house optimization stage is small, and it gets smaller as the number of iterations gets larger. The energy landscape of the loss function of Neural Networks is less well-understood, but it has been shown that there can be multiple local minima with near-optimal performance (see, e.g., [16]). If this is the case, the main advantage of using outsourcing for initialization is to get us closer to a good local minimum. That is why we see a large performance improvement over random start when the number of iterations is small. In this case, outsourcing is most beneficial when the in-house computing resource is very limited and the data organization can only run a relatively small number of iterations.

In terms of computational cost, we run this experiment on a computer where the CPU is an Intel® Xeon® Processor E-2286M (8-cores 2.40-GHz Turbo, 16 MB). The GPU is Nvidia Quadro RTX 4000 w/8GB. In the outsourcing stage, it takes on average 118 and 828 seconds to run OIPS-annealing and OIPS-SAO respectively. For the in-house optimization stage, it takes on average 30 seconds to run 50 iterations of stochastic gradient descent with momentum. Note that even though the computational cost at the exploration stage is relatively large, this cost is incurred at the external computing facility where computational resources can be much cheaper.

Lastly, to test the robustness of our method, we implement OIPS-annealing and OIPS-SAO with different values of n and β. In particular, we vary the outsourcing sample size n from 800 to 1200 and the inverse temperature β from 1 to 3. Tables 1 and 2 summarize the results. Overall, we observe similar performances for different values of n and β, with only a small amount of performance improvement as n, β increasing. This demonstrates that our method is quite robust to the choice of n and β in a reasonable range.



(a) Training loss vs. iteration (b) Accuracy on testing set vs. iteration

Fig. 7. Evolution of training loss and accuracy on testing set

Table 1. Sensitivity analysis of the OIPS-annealing algorithm

outsourcing sample size n	800	800	800	1000	1000	1000	1200	1200	1200
inverse temperature β	1	2	3	1	2	3	1	2	3
training loss	0.42	0.42	0.41	0.42	0.43	0.40	0.43	0.43	0.41
accuracy on testing(%)	87.8	87.79	87.43	87.85	87.28	87.79	87.52	87.5	88.29

Table 2. Sensitivity analysis of the OIPS-SAO algorithm

outsourcing sample size n	800	800	800	1000	1000	1000	1200	1200	1200
inverse temperature β	1	2	3	1	2	3	1	2	3
training loss	0.20	0.19	0.18	0.17	0.17	0.15	0.14	0.13	0.12
accuracy on testing(%)	95.85	95.77	95.6	96.32	96.17	96.05	96.52	96.82	96.74

5 CONCLUSION, LIMITATIONS, FUTURE WORK

We have designed three algorithms using outsourced data to find good initial points. They are better than the popular random start approach. In both theoretical analysis and numerical tests, OIPS-SAO performs better than OIPS-annealing, but has higher computational costs in general. Our proposed approach is most beneficial when the underlying objective function/loss function is smooth but non-convex and the global minimum leads to a much smaller loss than other local minima, i.e., there is a strong motivation to find the global minimum. Our approach saved the number of initializations the data organization needs to try in-house to find the global minimum. We note that in some machine learning tasks, such as matrix completion and wide neural networks, local minima already have good statistical properties or prediction power [16, 34]. In these situations, our proposed approach can get us closer to a local minimum to start with, but the benefit would not be as substantial as in the former cases.

Our work has the following three limitations, which can be seen as possible future directions. 1) Our theoretical development provides important insights on how to use data outsourcing to achieve better initialization, including how to choose the outsourcing data size and how to design

sampling algorithms for the outsourced computing facility. However, the optimal choice of n and β depends on problem-specific characteristics that may not be fully known to us (e.g., strong convexity parameter), which limits our ability to fully optimize the choice of n and β in practice. It would be interesting to take a more algorithmic approach to fine-tune these parameters in future research. 2) We assume the outsourced data is drawn randomly from the population data. In practice, the outsourced data might be from a biased distribution or need additional privacy encryption. 3) Our analysis focuses on the large β scenario. In practice, we would prefer to use a moderate β, i.e., $\beta = O(1)$, due to the sampling cost. For the sampling task, there are two main challenges: multi-modality due to the non-convexity of $F(\theta)$ and high-dimensionality, both of which are active areas of research. Various advanced sampling methods have been developed in the literature to address these challenges in special cases [20, 41, 42], which can be utilized in the outsourcing exploration stage for more efficient sampling.

REFERENCES

[1] Emile Aarts and Jan Korst. 1989. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. John Wiley & Sons, Inc.

[2] Zeyuan Allen-Zhu. 2018. Natasha 2: Faster Non-convex Optimization Than SGD. In *Advances in Neural Information Processing Systems*.

[3] Jordan T Ash and Ryan P Adams. 2020. On Warm-Starting Neural Network Training. In *34th Conference on Neural Information Processing Systems*.

[4] Søren Asmussen and Peter W Glynn. 2007. *Stochastic simulation: algorithms and analysis*. Vol. 57. Springer.

[5] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *Siam Review* 60, 2 (2018), 223–311.

[6] Xi Chen, Simon S Du, and Xin T Tong. 2020. On Stationary-Point Hitting Time and Ergodicity of Stochastic Gradient Langevin Dynamics. *Journal of Machine Learning Research* 21, 68 (2020), 1–41.

[7] Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma. 2019. Gradient descent with random initialization: Fast global convergence for nonconvex phase retrieval. *Mathematical Programming* 176, 1 (2019), 5–37.

[8] Sabrina De Capitani Di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. 2007. A data outsourcing architecture combining cryptography and access control. In *Proceedings of the 2007 ACM workshop on Computer security architecture*. 63–69.

[9] Jing Dong and Xin T Tong. 2020. Spectral Gap of Replica Exchange Langevin Diffusion on Mixture Distributions. *arXiv preprint arXiv:2006.16193* (2020).

[10] Jing Dong and Xin T Tong. 2021. Replica exchange for non-convex optimization. *Journal of Machine Learning Research* 22, 173 (2021), 1–59.

[11] Alain Durmus and Eric Moulines. 2017. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability* 27, 3 (2017), 1551–1587.

[12] A. Durmus and E. Moulines. 2017. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *Annals of Applied Probability* 27, 3 (2017), 1551–1587.

[13] Raaz Dwivedi, Yuansi Chen, Martin J Wainwright, and Bin Yu. 2018. Log-concave sampling: Metropolis-Hastings algorithms are fast!. In *Conference on Learning Theory*. PMLR, 793–797.

[14] Denzil G Fiebig, Michael P Keane, Jordan Louviere, and Nada Wasi. 2010. The generalized multinomial logit model: accounting for scale and coefficient heterogeneity. *Marketing Science* 29, 3 (2010), 393–421.

[15] Sara Foresti. 2010. *Preserving privacy in data outsourcing*. Vol. 99. Springer Science & Business Media.

[16] Rong Ge, Chi Jin, and Yi Zheng. 2017. No Spurious Local Minima in Nonconvex Low Rank Problems: A Unified Geometric Analysis. In *Proceedings of the International Conference on Machine Learning*.

[17] Rong Ge, Holden Lee, and Andrej Risteski. 2018. Simulated tempering Langevin Monte Carlo II: An improved proof using soft Markov chain decomposition. *arXiv preprint arXiv:1812.00793* (2018).

[18] Saeed Ghadimi and Guanghui Lan. 2016. Accelerated Gradient Methods for Nonconvex Nonlinear and Stochastic Programming. *Mathematical Programming* 156, 1-2 (2016), 59–99.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[20] M. Hairer, A.M. Stuart, and S.J. Vollmer. 2014. Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions. *Ann. Appl. Probab.* 24, 6 (2014), 2455–2490.

[21] Boris Hanin and David Rolnick. 2018. How to start training: The effect of initialization and architecture. In *32nd Conference on Neural Information Processing Systems*.

[22] Rie Johnson and Tong Zhang. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems* 26 (2013), 315–323.
[23] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2019. SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning. (2019).
[24] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.
[25] Holden Lee, Andrej Risteski, and Rong Ge. 2018. Beyond log-concavity: Provable guarantees for sampling multi-modal distributions using simulated tempering langevin monte carlo. In *NeurIPS*.
[26] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.
[27] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189* (2019).
[28] Yue M Lu and Gen Li. 2017. Spectral initialization for nonconvex estimation: high-dimensional limit and phase transitions. In *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 3015–3019.
[29] Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I Jordan. 2019. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences* 116, 42 (2019), 20881–20885.
[30] Song Mei, Yu Bai, Andrea Montanari, et al. 2018. The landscape of empirical risk for nonconvex losses. *Annals of Statistics* 46, 6A (2018), 2747–2774.
[31] Georg Menz and André Schlichting. 2014. Poincaré and logarithmic Sobolev inequalities by decomposition of the energy landscape. *The Annals of Probability* 42, 5 (2014), 1809–1884.
[32] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics* 21, 6 (1953), 1087–1092.
[33] Yurii Nesterov. 2003. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media.
[34] Dohyung Park, Anastasios Kyrillidis, Constantine Carmanis, and Sujay Sanghavi. 2017. Non-square Matrix Sensing without Spurious Local Minima Via the Burer-Monteiro Approach. In *Artificial Intelligence and Statistics*.
[35] Loucas Pillaud-Vivien, Francis Bach, Tony Lelièvre, Alessandro Rudi, and Gabriel Stoltz. 2020. Statistical estimation of the poincaré constant and application to sampling multimodal distributions. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2753–2763.
[36] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. 2017. Non-convex Learning via Stochastic Gradient Langevin Dynamics: A Nonasymptotic Analysis. In *Proceedings of the Conference on Learning Theory*.
[37] Gareth O Roberts, Richard L Tweedie, et al. 1996. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* 2, 4 (1996), 341–363.
[38] Pierangela Samarati and Sabrina De Capitani Di Vimercati. 2010. Data protection in outsourcing scenarios: Issues and directions. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. 1–14.
[39] Mark Schmidt, Nicolas Le Roux, and Francis Bach. 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162, 1-2 (2017), 83–112.
[40] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. 2021. *Lectures on stochastic programming: modeling and theory*. SIAM.
[41] Nicholas G Tawn, Gareth O Roberts, and Jeffrey S Rosenthal. 2020. Weight-preserving simulated tempering. *Statistics and Computing* 30, 1 (2020), 27–41.
[42] Xin T. Tong, Mathias Morzfeld, and Youssef M. Marzouk. 2020. MALA-within-Gibbs samplers for high-dimensional distributions with sparse conditional structure. *SIAM Journal on Scientific Computing* 42, 3 (2020), A1765–A1788.
[43] Kenneth E Train. 2009. *Discrete choice methods with simulation*. Cambridge university press.
[44] Aad W. Vaart and Jon A. Wellner. 2013. *Weak convergence and empirical processes: with applications to statistics*. Springer Science & Business Media.
[45] Santosh Vempala and Andre Wibisono. 2019. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *Advances in neural information processing systems* 32 (2019).
[46] Xiao Wang, Shiqian Ma, Donald Goldfarb, and Wei Liu. 2017. Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization* 27, 2 (2017), 927–956.
[47] Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer, 681–688.
[48] Graham R Wood, David LJ Alexander, and David W Bulger. 2002. Approximation of the distribution of convergence times for stochastic global optimisation. *Journal of Global Optimization* 22, 1 (2002), 271–284.
[49] Dawn B Woodard, Scott C Schmidler, Mark Huber, et al. 2009. Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *The Annals of Applied Probability* 19, 2 (2009), 617–640.

[50] Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. 2018. Global Convergence of Langevin Dynamics Based Algorithms for Nonconvex Optimization. In *Advances in Neural Information Processing Systems*.
[51] Zelda B Zabinsky. 2003. *Stochastic adaptive search for global optimization*. Vol. 72. Springer Science & Business Media.
[52] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. 2019. Fixup initialization: Residual learning without normalization. In *Seventh International Conference on Learning Representations*.
[53] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. 2021. FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data. *IEEE Transactions on Signal Processing* 69 (2021), 6055–6070.

A TECHNICAL VERIFICATIONS

A.1 Approximation accuracy of $\widehat{F}_n(\theta)$ and data complexity

Proof of Lemma 3.2. To prove the results about gradient and Hessian convergence, we can apply Theorem 1 in [30] directly. Specifically, under Assumptions 1 and 4, when $n > Cd\log(d)$, with probability at least $1-\rho$, we have

$$\begin{aligned}\sup_{\theta\in\Theta}\|\nabla F(\theta)-\nabla\widehat{F}_n(\theta)\| &\le \tau\sqrt{\frac{Cd\log(n)}{n}},\\ \sup_{\theta\in\Theta}\|\nabla^2 F(\theta)-\nabla^2\widehat{F}_n(\theta)\|_{\mathrm{op}} &\le \tau^2\sqrt{\frac{Cd\log(n)}{n}}.\end{aligned} \tag{7}$$

For the stationary points convergence, based on Theorem 2 in [30], under Assumptions 1 and 4, when $n \ge 4Cd\log(n)\cdot((\tau^2/\sigma^2)\vee(\tau^4/\eta^2))$, the empirical loss function $\widehat{F}_n(\theta)$ is $(\sigma/2,\eta/2)$-strongly Morse and possesses $K+1$ stationary points with probability at least $1-\rho$. Furthermore, there is a one-to-one correspondence between $(\theta_0^*,\dots,\theta_K^*)$, the stationary points of $F(\theta)$, and $(\hat\theta_0^*,\dots,\hat\theta_K^*)$, the stationary points of $\widehat{F}_n(\theta)$. Moreover, when $n \ge 4Cd\log(n)/\eta_*^2$,

$$\max_{0\le i\le K}\|\theta_i^*-\hat\theta_i^*\| \le \frac{2\tau}{\eta}\sqrt{\frac{Cd\log(n)}{n}}. \tag{8}$$

It remains to establish the uniform convergence result for $\widehat{F}_n$. Although it is not directly available in [30], the proof follows a similar idea. For self-completeness, we provide the details here.

First of all, given the parameter space Θ, let $\Theta_\varepsilon := \{\theta_1,\dots,\theta_J\}$ be a ε-covering net. In other words, for arbitrary $\theta\in\Theta$, there exists certain $\theta_{j(\theta)}\in\Theta_\varepsilon$ such that $\|\theta-\theta_{j(\theta)}\|\le\varepsilon$. Thus, for any $\theta\in\Theta$, we have

$$\big|\widehat{F}_n(\theta)-F(\theta)\big| \le \big|\widehat{F}_n(\theta)-\widehat{F}_n(\theta_{j(\theta)})\big| + \big|\widehat{F}_n(\theta_{j(\theta)})-F(\theta_{j(\theta)})\big| + \big|F(\theta)-F(\theta_{j(\theta)})\big|. \tag{9}$$

For any $t>0$, we denote by

$$A_t = \Big\{\sup_{\theta\in\Theta}\big|\widehat{F}_n(\theta)-\widehat{F}_n(\theta_{j(\theta)})\big|\ge t/3\Big\},\ B_t = \Big\{\sup_{\theta_j\in\Theta_\varepsilon}\big|\widehat{F}_n(\theta_j)-F(\theta_j)\big|\ge t/3\Big\},$$
$$\text{and } C_t = \Big\{\sup_{\theta\in\Theta}\big|F(\theta)-F(\theta_{j(\theta)})\big|\ge t/3\Big\}.$$

Then we have

$$\mathbb{P}\Big(\sup_{\theta\in\Theta}\big|\widehat{F}_n(\theta)-F(\theta)\big|\ge t\Big) \le \mathbb{P}(A_t)+\mathbb{P}(B_t)+\mathbb{P}(C_t).$$

In the next, we upper bound the three parts in above inequality respectively. For the last part, we have

$$\big|F(\theta)-F(\theta_{j(\theta)})\big| \le \sup_{\theta\in\Theta}\|\nabla F(\theta)\|\cdot\|\theta-\theta_{j(\theta)}\| \le L^*\cdot\varepsilon.$$

Hence, when $t \geq 3\varepsilon L^*$, the deterministic event C_t would never happen and $\mathbb{P}(C_t) = 0$. For the second part, under Assumption 4, by applying the union bound and the sub-Gaussian concentration inequality, we have

$$\begin{aligned}\mathbb{P}(B_t) &\leq \left|\Theta_\varepsilon\right| \cdot P\Big(\left|\widehat{F}_n(\theta_j) - F(\theta_j)\right| \geq t/3\Big) \\ &\leq |\Theta_\varepsilon| \cdot \exp\big\{-nt^2/(18\tau^2)\big\} \leq (2/\varepsilon)^d \cdot \exp\big\{-nt^2/(18\tau^2)\big\}.\end{aligned}$$

Thus, when

$$t > 5\tau \cdot \sqrt{\frac{\log(2/\rho) + d\log(2/\varepsilon)}{n}},$$

we have $\mathbb{P}(B_t) \leq \rho/2$. For the first part, by Markov inequality, we have

$$\mathbb{P}(A_t) \leq \frac{3\mathbb{E}\big[\sup_{\theta\in\Theta} |\widehat{F}_n(\theta) - \widehat{F}_n(\theta_{j(\theta)})|\big]}{t} \leq \frac{3\varepsilon \cdot \mathbb{E}\big[\sup_{\theta\in\Theta} \|\nabla\widehat{F}_n(\theta)\|\big]}{t}.$$

By Assumption 4, we have

$$\mathbb{E}\Big[\sup_{\theta\in\Theta} \|\nabla\widehat{F}_n(\theta)\|\Big] \leq \mathbb{E}\Big[\sup_{\theta\in\Theta} \|\nabla\widehat{F}_n(\theta) - \nabla\widehat{F}_n(\theta^*)\|\Big] + \mathbb{E}\big[\|\nabla\widehat{F}_n(\theta^*)\|\big] \leq 2J^* + H,$$

which implies that

$$\mathbb{P}(A_t) \leq 3\varepsilon(2J^* + H)/t.$$

Taking $t \geq 6\varepsilon(2J^* + H)/\rho$, we have $\mathbb{P}(A_t) \leq \rho/2$.

Finally, by taking

$$\varepsilon^* = \rho\tau/(6dn(2J^* + H)),\ t^* = 5\tau\sqrt{(\log(2/\rho) + d\log(2/\varepsilon))/n},$$

and utilizing the fact that $H \leq \tau^2 d^{c_h}, J_* \leq \tau^3 d^{c_h}$, when $n \geq Cd\log(d)$, we have

$$\mathbb{P}\left(\sup_{\theta\in\Theta}\left|\widehat{F}_n(\theta) - F(\theta)\right| \geq \tau\sqrt{\frac{Cd\log(n)}{n}}\right) \leq \rho.$$

Now, given an approximation accuracy δ, when

$$n \geq \max\Big\{Cd\tau^2\log n/\delta^2, 4Cd\log n((\tau^2/\sigma^2) \wedge (\tau^4/\eta^2)), 4Cd\log n/\eta_*^2, Cd\log d\Big\},$$

we have

$$\begin{aligned}&\sup_{\theta\in\Theta} |F(\theta) - \widehat{F}_n(\theta)| \leq \delta,\ \sup_{\theta\in\Theta} \|\nabla F(\theta) - \nabla\widehat{F}_n(\theta)\| \leq \delta, \\ &\sup_{\theta\in\Theta} \|\nabla^2 F(\theta) - \nabla^2\widehat{F}_n(\theta)\|_{\mathrm{op}} \leq \delta,\ \text{ and } \max_{0\leq i\leq K} \|\theta_i^* - \hat{\theta}_i^*\| \leq \delta.\end{aligned}$$

with probability at least $1 - \rho$. □

A.2 Performance analysis of the sampling approach

Proof of Proposition 3.3. First note that when $\widehat{F}_n(\theta)$ is a δ-approximation, for any $\theta \notin \mathcal{B}_r(\theta_0^*)$, by Assumption 3 we have

$$\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*) \geq \big(F(\theta) - \delta\big) - \big(F(\theta_0^*) + \delta\big) \geq \alpha - 2\delta > 0.$$

Hence, by the definition of π_β, we have

$$\begin{aligned}\mathbb{P}\Big(\tilde{\theta}_\beta \in \mathcal{B}_r(\theta_0^*)\Big) &= \frac{\int_{\mathcal{B}_r(\theta_0^*)} \exp(-\beta \widehat{F}_n(\theta))d\theta}{\int_{\mathcal{B}_r(\theta_0^*)} \exp(-\beta \widehat{F}_n(\theta))d\theta + \int_{\Theta/\mathcal{B}_r(\theta_0^*)} \exp(-\beta \widehat{F}_n(\theta))d\theta} \\ &= \frac{\int_{\mathcal{B}_r(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta}{\int_{\mathcal{B}_r(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta + \int_{\Theta/\mathcal{B}_r(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta} \\ &\geq \frac{\int_{\mathcal{B}_r(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta}{\int_{\mathcal{B}_r(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta + \exp(-\beta(\alpha - 2\delta)) \cdot \mathrm{Vol}(\Theta/\mathcal{B}_r(\theta_0^*))},\end{aligned}$$

where $\mathrm{Vol}(\Theta/\mathcal{B}_r(\theta_0^*))$ denotes the volume of set $\Theta/\mathcal{B}_r(\theta_0^*)$.

On the other hand, based on the regularity condition of Hessian and the definition of δ-approximation, we have $\|\nabla^2 \widehat{F}_n(\theta)\|_{\mathrm{op}} \leq H + L^* + \delta$. As a result, for any $\theta \in \mathcal{B}_r(\theta_0^*)$,

$$\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*) \leq 2(H + L^* + \delta) \cdot \big(\|\theta - \theta_0^*\|^2\big).$$

Hence,

$$\begin{aligned}&\int_{\mathcal{B}_r(\theta_0^*)} \exp\big(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)]\big)d\theta \geq \int_{\mathcal{B}_r(\theta_0^*)} \exp\big(-2\beta(H + L^* + \delta)\|\theta - \theta_0^*\|^2\big)d\theta \\ =&\Big(\pi\beta^{-1}/(H + L^* + \delta) \cdot \big(\Psi(2r\sqrt{\beta(H + L^* + \delta)})) - 1/2\big)\Big)^d,\end{aligned}$$

where $\Psi(\cdot)$ denotes the CDF of standard normal distribution. Note that for $\beta \geq (H + L^* + \delta)^{-1} r^{-2}$, $\Psi\left(2r\sqrt{\beta(H + L^* + \delta)}\right) - 1/2 \geq \Psi(1) - 1/2$. Then, for $C_d = d\log d + d\log(H + L^* + \delta) + 3d$,

$$1 - \mathbb{P}\Big(\tilde{\theta}_\beta \in \mathcal{B}_r(\theta_0^*)\Big) \leq \exp\big(-\beta\big(\alpha - 2\delta\big) + d\log\beta + C_d\big).$$

Finally, by setting $\delta = \alpha/4$, we obtain the result. □

Proof of Lemma 3.4. Let $\theta_1, \ldots, \theta_m$ be samples from $\widehat{\mathcal{M}}$. Let $\mathcal{G}_l$ denote the σ-algebra generated by $\{\theta_1, \ldots, \theta_l\}$. For any measurable set B

$$\begin{aligned}\mathbb{P}(\theta_1 \notin B, \ldots, \theta_m \notin B) &= \mathbb{E}\Big[\prod_{i=1}^{m} 1_{(\theta_i \notin B)}\Big] \\ &= \mathbb{E}\Big[\prod_{i=1}^{m-1} 1_{(\theta_i \notin B)} \cdot \mathbb{P}(\theta_m \notin B | \mathcal{G}_{m-1})\Big] \\ &\leq \mathbb{E}\Big[\prod_{i=1}^{m-1} 1_{(\theta_i \notin B)} \cdot (\pi_\beta(B^c) + \delta_\beta)\Big] \text{ by Assumption 5} \\ &= (\pi_\beta(B^c) + \delta_\beta) \cdot \mathbb{P}(\theta_1 \notin B, \ldots, \theta_{m-1} \notin B),\end{aligned}$$

where B^c is the complement of B. By induction, we have

$$\mathbb{P}(\theta_1 \notin B, \ldots, \theta_m \notin B) \leq (\pi_\beta(B^c) + \delta_\beta)^m.$$

□

Proof of Theorem 3.5. We use $\mathcal{I}_n(\delta)$ to denote the random event that $\widehat{F}_n(\theta)$ is a δ-approximation of $F(\theta)$. First, based on Proposition 3.3, $\mathbb{P}(\mathcal{I}_n^c(\delta)) \le \rho$ for $n \ge n(\delta, \rho, d)$. Then,

$$\mathbb{P}(\mathcal{F}_0) - \rho \le \mathbb{P}\big(\mathcal{F}_0 \cap I_n(\delta)\big) \le \mathbb{P}\big(\mathcal{F}_0 | I_n(\delta)\big).$$

By the definition of δ-approximation, conditional on $\mathcal{I}(\delta)$, if at least one of $(\theta_1, \cdots, \theta_L)$ falls into $\mathcal{B}_r(\theta_0^*)$, $\mathcal{F}_0$ would not happen. Hence, by Lemma 3.4, we have

$$\mathbb{P}(\mathcal{F}_0 | \mathcal{I}_n(\delta)) \le \big(\pi_\beta(\mathcal{B}_r^c(\theta_0^*)) + \delta_\beta\big)^m$$

Based on Lemma 3.3, for $\beta = \Omega(r^{-2})$,

$$\pi_\beta(\mathcal{B}_r^c(\theta_0^*)) \le \exp(-\beta\alpha/2 + d\log\beta + C_d).$$

for some constant C. Then,

$$\mathbb{P}(\mathcal{F}_0) \le \rho + \big(\exp(-\beta\alpha/2 + d\log\beta + C_d) + \delta_\beta\big)^m.$$

for some constant $C > 0$, and we finish the proof. □

A.3 Performance analysis of the optimization approaches

In this section, we establish the performance guarantee of the optimization approach, i.e., Algorithm 3. We first analyze the sample selection rule with the SAO approach, which set $\theta_*^0 = \widehat{\mathcal{T}}(\theta_{i^*}^0)$ where $i^* = \text{argmin}_{1\le i\le L}\big\{\widehat{F}_n(\widehat{\mathcal{T}}(\theta_i^0))\big\}$.

Proof of Theorem 3.6. Under Assumption 2, $F(\theta)$ is μ-strongly convex in $\mathcal{B}_r(\theta_0^*)$. When $\delta < \mu$ and $\widehat{F}_n(\theta)$ is a δ-approximation, we have

$$\sup_{\theta\in\Theta} \|\nabla^2 F(\theta) - \nabla^2 \widehat{F}_n(\theta)\|_{\text{op}} \le \delta \text{ and } \|\hat{\theta}_0^* - \theta_0^*\| \le \delta.$$

This implies that $\widehat{F}_n(\theta)$ is $(\mu-\delta)$-strongly convex in $\mathcal{B}_r(\theta_0^*)$ and $\hat{\theta}_0^*$ is the unique minimum of $\widehat{F}_n(\theta)$ in $\mathcal{B}_r(\theta_0^*)$. Hence, starting from any $\theta \in \mathcal{B}_r(\theta_0^*)$, the optimization algorithm $\widehat{\mathcal{T}}$ can converge to $\hat{\theta}_0^*$, which implies that

$$\mathbb{P}(\widehat{\mathcal{T}}(\tilde{\theta}_\beta) \ne \hat{\theta}_0^*) \le \mathbb{P}(\tilde{\theta}_\beta \notin \mathcal{B}_r(\theta_0^*)).$$

Then Proposition 3.3 provides an upper bound for $\mathbb{P}(\widehat{\mathcal{T}}(\tilde{\theta}_\beta) \ne \hat{\theta}_0^*)$. Finally, note that if at least one of $\theta_1, \theta_2, \ldots, \theta_m$ falls into $\mathcal{B}_r(\theta_0^*)$, $\hat{\theta}_0^*$ can be found by $\widehat{\mathcal{T}}$ and will be selected as the initial point to optimize $F(\theta)$. Thus, $\mathbb{P}(\mathcal{F}_1) \le \mathbb{P}(\mathcal{F}_0)$. By Theorem 3.5, we obtain the upper bound for $\mathbb{P}(\mathcal{F}_1)$. □

Proof of Theorem 3.7. We first show that if at least one point of $(\theta_1, \ldots, \theta_m)$ is in $\mathcal{B}_{r_0}(\theta_0^*)$, the annealing approach would select a point that falls into $\mathcal{B}_{r_0}(\theta_0^*)$. When $\widehat{F}_n(\theta)$ is a δ-approximation of $F(\theta)$, if $\theta_i \in \mathcal{B}_{r_0}(\theta_0^*)$, for any $\theta_j \notin \mathcal{B}_{r_0}(\theta_0^*)$, we have

$$\begin{aligned}
\widehat{F}_n(\theta_i) \le F(\theta_i) + \delta &\le F(\theta_0^*) + \delta + \frac{1}{2} r_0^2 \cdot \sup_{\theta\in\Theta} \|\nabla^2 F(\theta)\|_{\text{op}} \\
&< F(\theta_0^*) + \delta + \frac{\alpha}{2} \\
&\le F(\theta_j) - \frac{\alpha}{2} + \delta \text{ by Assumption 3} \\
&< \widehat{F}_n(\theta_j) \text{ as } \delta \le \alpha/4.
\end{aligned}$$

Hence, if the algorithm selects some $\theta_j \ne \theta_i$, we must have $\widehat{F}_n(\theta_j) \le \widehat{F}_n(\theta_i)$, which implies that θ_j is in $\mathcal{B}_{r_0}(\theta_0^*)$ as well. The remaining proof follows the same line of arguments as Theorem 3.5. □

A.4 Convergence analysis of ULA

Proof of Lemma 3.8. By Pinsker inequality, to bound the total variation distance by δ_β, it suffices to ensure that $\mathrm{KL}(\hat{\pi}_{\beta,K}||\pi_\beta) \le 2\delta_\beta^2$, where KL denotes the Kullback–Leiber divergence.

Next, Theorem 1 in [45] indicates the $2\delta_\beta^2$-bound for the KL divergence can be achieved by setting

$$h = \frac{\gamma_\beta \delta_\beta^2}{8(L\beta)^2 d} \text{ and } K = \Theta\left(\frac{1}{\gamma_\beta h}|\log \delta_\beta|\right) = \Theta\left(\frac{dL^2\beta^2|\log\delta_\beta|}{\gamma_\beta^2\delta_\beta^2}\right).$$

Note that $L\beta$ is the Lipschitz constant for $\nabla \log \pi_\beta$. □

A.5 Performance analysis for extension to ϵ-Global Minimum

Proof of Theorem 3.10. For Algorithm 3-annealing, note that if there is a sample $\theta_i \in \mathcal{B}_{r_\epsilon}(\theta_i^*) \subseteq \mathcal{B}_{\epsilon,r_\epsilon}$, then we have

$$\widehat{F}_n(\theta_i) \le F(\theta_i) + \delta \le F(\theta_i^*) + \sup_{\theta\in\Theta} \|\nabla^2 F(\theta)\|_{\mathrm{op}} \cdot r_\epsilon^2 + \delta \le F(\theta_0^*) + 2\epsilon + \delta.$$

Next, if the algorithm pick θ_j for $j \ne i$, then

$$F(\theta_j) \le \widehat{F}_n(\theta_j) + \delta \le \widehat{F}_n(\theta_i) + \delta \le F(\theta_0^*) + 2\epsilon + 2\delta.$$

For $\delta \le \epsilon/2$, by definition θ_j is a 3ϵ-global minimum.

In what follows, we estimate the probability that a sample drawn from π_β fails to fall into $\mathcal{B}_{\epsilon,r_\epsilon}$. Note that for $\theta \notin \mathcal{B}_{\epsilon,r_\epsilon}$,

$$\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*) \ge \epsilon - 2\delta.$$

Then,

$$\begin{aligned}
\mathbb{P}\big(\tilde{\theta}_\beta \in \mathcal{B}_{\epsilon,r_\epsilon}\big) &= \frac{\int_{\mathcal{B}_{\epsilon,r_\epsilon}} \exp(-\beta\widehat{F}_n(\theta))d\theta}{\int_{\mathcal{B}_{\epsilon,r_\epsilon}} \exp(-\beta\widehat{F}_n(\theta))d\theta + \int_{\Theta/\mathcal{B}_{\epsilon,r_\epsilon}} \exp(-\beta\widehat{F}_n(\theta))d\theta} \\
&\ge \frac{\int_{\mathcal{B}_{\epsilon,r_\epsilon}} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta}{\int_{\mathcal{B}_{\epsilon,r_\epsilon}} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta + \exp(-\beta(\epsilon-2\delta))\cdot \mathrm{Vol}(\Theta/\mathcal{B}_{\epsilon,r_\epsilon})} \\
&\ge \frac{\int_{\mathcal{B}_{r_\epsilon}(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta}{\int_{\mathcal{B}_{r_\epsilon}(\theta_0^*)} \exp(-\beta[\widehat{F}_n(\theta) - \widehat{F}_n(\theta_0^*)])d\theta + \exp(-\beta(\epsilon-2\delta))\cdot \mathrm{Vol}(\Theta/\mathcal{B}_{\epsilon,r_\epsilon})}.
\end{aligned}$$

Similar to the proof of Proposition 3.3, for $\beta = \Omega(r_\epsilon^{-2})$, we have

$$\mathbb{P}\Big(\tilde{\theta}_\beta \notin \mathcal{B}_{\epsilon,r_\epsilon}\Big) = \exp\big\{-\beta\big(\epsilon - 2\delta\big) + d\log\beta + C_d\big\}.$$

For $\delta \le \epsilon/4$, we have

$$\mathbb{P}\Big(\tilde{\theta}_\beta \notin \mathcal{B}_{\epsilon,r_\epsilon}\Big) = \exp\big(-\beta\epsilon/2 + d\log\beta + C_d\big).$$

Lastly, we use $\mathcal{I}_n(\delta)$ to denote the random event that $\widehat{F}_n(\theta)$ is a δ-approximation of $F(\theta)$. Similar to the proof of Theorem 3.5, we have

$$\mathbb{P}(\mathcal{F}_{3\epsilon,2}) - \rho \le \mathbb{P}\big(\mathcal{F}_{3\epsilon,2} \cap I_n(\delta)\big) \le \mathbb{P}\big(\mathcal{F}_{3\epsilon,2}|I_n(\delta)\big).$$

and

$$\mathbb{P}(\mathcal{F}_{3\epsilon,2}|\mathcal{I}_n(\delta)) \le \Big(\exp\big(-\beta\epsilon/2 + d\log\beta + C_d\big) + \delta_\beta\Big)^m.$$

□