OPTIMAL NEURAL NETWORK APPROXIMATION OF WASSERSTEIN GRADIENT DIRECTION OF KL DIVERGENCE VIA CONVEX OPTIMIZATION*

YIFEI WANG[†], PENG CHEN[‡], MERT PILANCI[†], AND WUCHEN LI[§]

Abstract. The calculation of the direction of the Wasserstein gradient is vital for addressing problems related to posterior sampling and scientific computing. To approximate the Wasserstein gradient using finite samples, it is necessary to solve a variation problem. Our study focuses on the variation problem within the framework of two-layer networks with squared-ReLU activations. We present a semi-definite programming (SDP) relaxation as a solution, which can be viewed as an approximation of the Wasserstein gradient for a broader range of functions, including two-layer networks. By solving the convex SDP, we achieve the best approximation of the Wasserstein gradient direction in this function class. We also provide conditions to ensure the relaxation is tight. Additionally, we propose methods for practical implementation, such as subsampling and dimension reduction. The effectiveness and efficiency of our proposed method are demonstrated through numerical experiments, including Bayesian inference with PDE constraints and parameter estimation in COVID-19 modeling.

Key words. Bayesian inference, Convex Optimization, Neural Network, Semi-positive Definite Program.

MSC codes. 62F15, 41A30, 65K10

1. Introduction.

Bayesian inference is a crucial method for determining model parameters based on observational data. It is widely used in fields such as inverse problems, scientific computing, information science, and machine learning [46]. The core issue in Bayesian inference is obtaining samples from a posterior distribution, which describes the distribution of parameters based on both data and prior information.

The Wasserstein gradient flow, as first introduced in references such as [41, 2, 28], has been proven to be an efficient method for obtaining samples from a posterior distribution. This has led to growing interest in recent years. For example, the Wasserstein gradient flow of the Kullback-Leibler (KL) divergence is related to overdamped Langevin dynamics. Discretizing the overdamped Langevin dynamics results in the classical Langevin Monte Carlo Markov Chain (MCMC) algorithm. Therefore, the computation of the Wasserstein gradient flow offers a unique perspective on sampling algorithms. Additionally, the direction of the Wasserstein gradient also offers a deterministic method for updating a particle system as demonstrated in [10]. A number of efficient sampling algorithms have been developed by utilizing approxima-

*Submitted to the editors DATE.

Funding: W. Li is supported in part by AFOSR MURI FP 9550-18-1-502; in part by AFOSR YIP award No. FA9550-23-1-0087; and in part by the National Science Foundation (NSF) RTG: 2038080. P. Chen is partially supported by NSF DMS 2245674. Yifei Wang and Mert Pilanci was supported in part by the National Science Foundation (NSF) under Grant DMS-2134248; in part by the NSF CAREER Award under Grant CCF-2236829; in part by the U.S. Army Research Office Early Career Award under Grant W911NF-21-1-0242; and in part by the Office of Naval Research under Grant N00014-24-1-2164.

[†]Department of Electrical Engineering, Stanford University, Stanford, CA (wangyf18@stanford.edu, pilanci@stanford.edu).

[‡]School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, Atlanta, GA (pchen402@gatech.edu).

[§]Department of Mathematics, University of South Carolina, Columbia, SC (wuchen@mailbox.sc.edu).

tion or generalization of the Wasserstein gradient direction. Such examples include the Wasserstein gradient descent (WGD) with kernel density estimation (KDE) [35], Stein variational gradient descent (SVGD) [36], and neural variational gradient descent [15].

Neural networks have demonstrated impressive abilities in learning complex functions from data, as well as in Bayesian inverse problems [44, 40, 30, 32]. According to the universal approximation theorem of neural networks [23, 38], any complex function can be learned by a two-layer neural network with non-linear activations and a sufficient number of neurons. Furthermore, functions represented by neural networks provide a natural approximation to the Wasserstein gradient direction.

However, due to the nonlinear and nonconvex nature of neural networks, optimization algorithms such as stochastic gradient descent may not always find the global optimal solutions for the training problem. Recently, based on a line of research [42, 45, 4], the regularized training problem of two-layer neural networks with ReLU/polynomial activation and a convex loss function can be formulated as a convex program. By solving this convex program, it is possible to construct the entire set of global optima for the nonconvex training problem [52]. Theoretical analysis [51] has also shown that global optima of the training problem correspond to simpler models with better generalization properties. Numerical experiments have also shown that neural networks found by solving the convex program can achieve higher train accuracy and test accuracy compared to neural networks trained by SGD with the same number of parameters.

In this paper, we investigate a variational problem whose optimal solution corresponds to the Wasserstein gradient direction. Our focus is on the family of two-layer neural networks with squared ReLU activation. We formulate the regularized variational problem in terms of samples, and instead of directly training the neural network to minimize the loss, we analyze the convex dual problem of the training problem and study its semi-definite program (SDP) relaxation by analyzing the geometry of dual constraints. The resulting SDP can be efficiently solved by convex optimization solvers such as CVXPY [16]. We also analyze the choice of the regularization parameter and present a practical implementation using subsampling and dimension reduction to improve computational efficiency. Numerical experiments for PDE-constrained inference problems and Covid-19 parameter estimation problems demonstrate the effectiveness and efficiency of our method.

1.1. Related works. The time and spatial discretizations of Wasserstein gradient flows are extensively studied in literature [27, 28, 9, 10, 6, 37, 22]. Recently, neural networks have been applied in solving or approximating Wasserstein gradient flows [39, 34, 33, 1, 8, 24, 20]. For sampling algorithms, [15] learns the transportation function by solving an unregularized variational problem in the family of vector-output deep neural networks. Compared to these studies, we focus on a convex SDP relaxation of the variational problem induced by the Wasserstein gradient direction. Meanwhile, [21] form the Wasserstein gradient direction as the minimizer of the Bregman score and they apply deep neural networks to solve the induced variational problem. In short, we study the same variational variational problem but we focus on the two-layer neural networks, provide convex SDP relaxations and give sufficient conditions when the relaxation is exact.

In comparison to previous works on the convex optimization formulations of neural networks using SDP [4, 5], they focus on the polynomial activation and give the exact convex optimization formulation (instead of convex relaxation). In comparison, we focus on the neural networks with the squared ReLU activation, which has not been considered before. Our method can also apply to the analysis of supervised learning problems using neural networks with squared ReLU activation. Moreover, previous works on the convex optimization formulation of neural networks mainly focus on the supervised learning problem of two-layer neural networks using convex loss functions (e.g., squared loss, logistic loss). Our work utilizes a similar convex analytic framework to solve the variational problem of approximating the Wasserstein gradient direction, which is different from supervised learning. The convex optimization approach is based on the idea of infinite-width neural networks modeled as probability measures. The dual problem itself is equivalent to the convex dual problem when the neural network in the primal problem has infinitely many neurons. However, the convex optimization approach tackles networks of arbitrary width that are able to learn useful representations, while the infinite width is often limited to kernel methods.

2. Background. In this section, we briefly review the Wasserstein gradient descent and present its variational formulation. In particular, we focus on the Wasserstein gradient descent direction of KL divergence functional. Later on, we design a neural network convex optimization problem to approximate the Wasserstein gradient in samples.

2.1. Wasserstein gradient descent. Consider an optimization problem in the probability space:

$$\inf_{\rho\in\mathcal{P}} \mathrm{D}_{\mathrm{KL}}(\rho\|\pi) = \int \rho(x)(\log\rho(x) - \log\pi(x))dx, \tag{2.1}$$

Here the integral is taken over $\mathbb{R}^d$ and the objective functional $\mathrm{D}_{\mathrm{KL}}(\rho\|\pi)$ is the KL divergence from ρ to π. The variable is the density function ρ in the space $\mathcal{P} = \{\rho \in C^\infty(\mathbb{R}^d) | \int \rho dx = 1,\ \rho > 0\}$. The function $\pi \in C^\infty(\mathbb{R}^d)$ is a known probability density function of the posterior distribution. By solving the optimization problem (2.1) , we can generate samples from the posterior distribution.

A known fact [47, Chapter 8.3.1] is that the Wasserstein gradient descent flow for the optimization problem (2.1) satisfies

$$\begin{aligned}\partial_t\rho_t =& \nabla\cdot\left(\rho_t\nabla\frac{\delta}{\delta\rho_t}\mathrm{D}_{\mathrm{KL}}(\rho_t\|\pi)\right)\\ =& \nabla\cdot(\rho_t(\nabla\log\rho_t - \nabla\log\pi))\\ \overset{(a)}{=}& \Delta\rho_t - \nabla\cdot(\rho_t\nabla\log\pi),\end{aligned}$$

where $\rho_t(x) = \rho(x,t)$, $\frac{\delta}{\delta\rho_t}$ is the L^2 first variation operator w.r.t. ρ_t, $\nabla\cdot F$ denotes the divergence of a vector valued function $F:\mathbb{R}^d\to\mathbb{R}^d$ and Δ is the Laplace operator. In step (a) we use the fact that $\rho_t\nabla\log\rho_t = \nabla\rho_t$. This equation is also known as the gradient drift Fokker-Planck equation. It corresponds to the following updates in terms of samples :

$$dx_t = -(\nabla\log\rho_t(x_t) - \nabla\log\pi(x_t))dt. \tag{2.2}$$

Clearly, when $\rho_t = \pi$, the above dynamics reaches the equilibrium, which implies that the samples x_t are generated by the posterior distribution.

To solve the Wasserstein gradient flow (2.2), we consider a forward Eulerian discretization in time. In the l-th iteration, suppose that $\{x_l^n\}$ are samples drawn from

ρ_l. The update rule of Wasserstein gradient descent (WGD) on the particle system $\{x_l^n\}$ follows

$$x_{l+1}^n = x_l^n - \alpha_l \nabla \Phi_l(x_l^n), \tag{2.3}$$

where $\Phi_l : \mathbb{R}^d \to \mathbb{R}$ is a function which approximates $\log \rho_l - \log \pi$ and $\alpha_l > 0$ is the step size.

2.2. Variational formulation of WGD. Given the particles $\{x_n\}_{n=1}^N$, we design the following variational problem to choose a suitable function Φ approximating the function $\log \rho - \log \pi$. Consider

$$\inf_{\Phi \in C^1(\mathbb{R}^d)} \frac{1}{2} \int \|\nabla \Phi(x) - (\nabla \log \rho(x) - \nabla \log \pi(x))\|_2^2 \rho(x) dx. \tag{2.4}$$

The objective function evaluates the least-square discrepancy between $\nabla \log \rho - \nabla \log \pi$ and $\nabla \Phi$ weighted by the density ρ. The optimal solution follows $\Phi = \log \rho - \log \pi$, up to a constant shift. Let $\mathcal{H} \subseteq C^1(\mathbb{R}^d)$ be a finite-dimensional function space. The following proposition gives a formulation of (2.4) in $\mathcal{H}$.

PROPOSITION 2.1. *Let $\mathcal{H} \subseteq C^1(\mathbb{R}^d)$ be a function space. The variational problem* (2.4) *in the domain $\mathcal{H}$ can be reformulated to*

$$\begin{aligned} &\inf_{\Phi \in \mathcal{H}} \frac{1}{2} \int \|\nabla \Phi(x)\|_2^2 \rho dx + \int \Delta \Phi(x) \rho(x) dx \\ &\qquad + \int \langle \nabla \log \pi(x), \nabla \Phi(x) \rangle \rho(x) dx. \end{aligned} \tag{2.5}$$

Proof. We first note that

$$\begin{aligned} &\frac{1}{2} \int \|\nabla \Phi - \nabla \log \rho + \nabla \log \pi\|_2^2 \rho dx \\ =&\frac{1}{2} \int \|\nabla \Phi\|_2^2 \rho dx + \int \langle \nabla \log \pi - \nabla \log \rho, \nabla \Phi \rangle \rho dx \\ &+ \frac{1}{2} \int \|\nabla \log \rho - \nabla \log \pi\|_2^2 \rho dx. \end{aligned} \tag{2.6}$$

We notice that the term $\frac{1}{2} \int \|\nabla \log \rho - \nabla \log \pi\|_2^2 \rho dx$ does not depend on Φ. Utilizing the integration by parts, we can compute that

$$\begin{aligned} \int \langle \nabla \log \rho, \nabla \Phi \rangle \rho dx =& \int \left\langle \frac{\nabla \rho}{\rho}, \nabla \Phi \right\rangle \rho dx \\ =& \int \langle \nabla \rho, \nabla \Phi \rangle dx \\ =& -\int \Delta \Phi \rho dx. \end{aligned} \tag{2.7}$$

Therefore, the variational problem (2.4) is equivalent to

$$\inf_{\Phi \in C^1(\mathbb{R}^d)} \frac{1}{2} \int \|\nabla \Phi\|_2^2 \rho dx + \int \langle \nabla \log \pi, \nabla \Phi \rangle \rho dx + \int \Delta \Phi \rho dx. \tag{2.8}$$

By restricting the domain to $\mathcal{H}$, we complete the proof. ∎

Remark 2.2. A similar variational problem has been studied in [15]. If we replace $\nabla\Phi$ for $\Phi \in \mathcal{H}$ by a vector field Ψ in a certain function family, then, the quantity in (2.5) is the negative regularized Stein discrepancy defined in [15] between ρ and π based on Ψ. This problem is also similar to the variational problem for the score matching estimator in [25] by parameterizing Φ in a given probabilistic model. In comparison, our method can be viewed as a special case of score matching by using a two-layer neural network.

Therefore, by replacing the density ρ by finite samples $\{x_n\}_{n=1}^N \sim \rho$, the problem (2.5) in terms of finite samples forms

$$\begin{aligned}\inf_{\Phi\in\mathcal{H}} \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2}\|\nabla\Phi(x_n)\|_2^2 + \Delta\Phi(x_n)\right)\\ + \frac{1}{N}\sum_{n=1}^{N}\langle\nabla\log\pi(x_n), \nabla\Phi(x_n)\rangle\,.\end{aligned} \tag{2.9}$$

3. Optimal neural network approximation of Wasserstein gradient. In this section, we focus on functional space $\mathcal{H}$ of functions represented by two-layer neural networks. We derive the primal and dual problems of the regularized Wasserstein variational problems. By analyzing the dual constraints, a convex SDP relaxation of the dual problem is obtained. We also present a practical implementation estimation of $\nabla\log\rho - \nabla\log\pi$ and discuss the choice of the regularization parameter.

Let ψ be an activation function. Consider the case where $\mathcal{H}$ is a class of two-layer neural network with the activation function $\psi(x)$:

$$\mathcal{H} = \left\{\Phi_{\boldsymbol{\theta}} \in C^1(\mathbb{R}^d) | \Phi_{\boldsymbol{\theta}}(x) = \alpha^T\psi(W^Tx)\right\}, \tag{3.1}$$

where $\boldsymbol{\theta} = (W, \alpha)$ is the parameter in the neural network with $W \in \mathbb{R}^{d\times m}$ and $\alpha \in \mathbb{R}^m$.

Remark 3.1. We can extend this model to handle by adding an entry of 1 in $x_1, \dots, x_n$, .

For two-layer neural networks, we can compute the gradient and Laplacian of $\Phi \in \mathcal{H}$ as follows:

$$\nabla\Phi_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{m}\alpha_i w_i \psi'(w_i^Tx) = W(\psi'(W^Tx)\circ\alpha), \tag{3.2}$$

$$\Delta\Phi_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{m}\alpha_i\|w_i\|_2^2\psi''(w_i^Tx). \tag{3.3}$$

Here $\circ$ represents the element-wise multiplication. By adding a regularization term to the variational problem (2.9), we obtain

$$\begin{aligned}\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{n=1}^{N}\left\|\sum_{i=1}^{m}\alpha_i w_i\psi'(w_i^Tx_n)\right\|_2^2\\ + \frac{1}{N}\sum_{n=1}^{N}\left\langle\sum_{i=1}^{m}\alpha_i w_i\psi'(w_i^Tx_n), \nabla\log\pi(x_n)\right\rangle\\ + \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{m}\alpha_i\|w_i\|_2^2\psi''(w_i^Tx_n) + \frac{\beta}{2}R(\boldsymbol{\theta}),\end{aligned} \tag{3.4}$$

where $\beta > 0$ is the regularization parameter. We focus on the squared ReLU activation $\psi(z) = (z)_+^2 = (\max\{z, 0\})^2$. Note that a non-vanishing second derivative is required for the Laplacian term in (3.3), which makes the ReLU activation inadequate. For this activation function, we consider the regularization function $R(\boldsymbol{\theta}) = \sum_{i=1}^m (\|w_i\|_2^3 + |\alpha_i|^3)$.

Remark 3.2. We note that $\nabla\Phi_{\boldsymbol{\theta}}(x)$ and $\Delta\Phi_{\boldsymbol{\theta}}(x)$ are all piece-wise degree-3 polynomials of the parameters $\boldsymbol{\theta}$. Hence, we consider a specific cubic regularization term above, analogous to [4]. By choosing this regularization term, we can derive a simplified dual problem.

By utilizing the arithmetic and geometric mean (AM-GM) inequality, we can rescale the first and second-layer parameters and formulate the regularized variational problem (3.4) as follows.

PROPOSITION 3.3 (Primal problem). *The regularized variational problem* (3.4) *can be reformulated to*

$$
\begin{aligned}
&\min_{W,\alpha} \frac{1}{2}\sum_{n=1}^N \left\|\sum_{i=1}^m \alpha_i w_i \psi'(w_i^T x_n)\right\|^2 \\
&\quad + \sum_{n=1}^N \sum_{i=1}^m \alpha_i \|w_i\|_2^2 \psi''(w_i^T x_n) \\
&\quad + \sum_{n=1}^N \left\langle \sum_{i=1}^m \alpha_i w_i \psi'(w_i^T x_n), \nabla \log \pi(x_n) \right\rangle + \tilde{\beta}\|\alpha\|_1, \\
&s.t.\ \|w_i\|_2 \le 1, i \in [m],
\end{aligned}
\tag{3.5}
$$

where $\tilde{\beta} = 3 \cdot 2^{-5/3} N\beta$ *and we denote* $[m] = \{1, \ldots, m\}$.

Proof. Suppose that $\hat{w}_i = \beta_i^{-1} w_i$ and $\hat{\alpha}_i = \beta_i^2 \alpha_i$, where $\beta_i > 0$ is a scale parameter for $i \in [m]$. Let $\boldsymbol{\theta}' = \{(\hat{w}_i, \hat{\alpha}_i)\}_{i=1}^m$. We note that

$$
\hat{\alpha}_i \hat{w}_i \psi'(\hat{w}_i^T x_n) = \beta_i \alpha_i w_i \psi'\left(\beta_i^{-1} w_i^T x_n\right) = \alpha_i w_i \psi'(w_i^T x_n),
\tag{3.6}
$$

and

$$
\hat{\alpha}_i \|\hat{w}_i\|_2^2 \psi''(\hat{w}_i^T x_n) = \alpha_i \|w_i\|_2^2 \psi''(\hat{w}_i^T x_n) = \alpha_i \|w_i\|_2^2 \psi''(w_i^T x_n).
\tag{3.7}
$$

This implies that $\Phi_{\boldsymbol{\theta}}(x) = \Phi_{\boldsymbol{\theta}'}(x)$ and $\nabla \cdot \Phi_{\boldsymbol{\theta}}(x) = \nabla \cdot \Phi_{\boldsymbol{\theta}'}(x)$. For the regularization term $R(\boldsymbol{\theta})$, we note that

$$
\begin{aligned}
\|\hat{w}_i\|_2^3 + \|\hat{\alpha}_i\|_2^3 =& \beta_i^6 |\alpha_i|^3 + \beta_i^{-3}\|w_i\|_2^3 \\
=& \beta_i^6 |\alpha_i|^3 + \frac{1}{2}\beta_i^{-3}\|w_i\|_2^3 + \frac{1}{2}\beta_i^{-3}\|w_i\|_2^3 \\
=& 3 \cdot 2^{-2/3} \|w_i\|_2^2 |\alpha_i|.
\end{aligned}
\tag{3.8}
$$

The optimal scaling parameter is given by $\alpha_i = 2^{-1/9} \frac{\|w_i\|_2^{1/3}}{|\alpha_i|_1^{1/3}}$. As the scaling operation does not change $\|w_i\|_2^2 |\alpha_i|$, we can simply let $\|w_i\|_2 = 1$. Thus, the regularization term $\frac{\beta}{2} R(\boldsymbol{\theta})$ becomes $\frac{\tilde{\beta}}{N} \sum_{i=1}^m \|w_i\|_1$. This completes the proof. ∎

In short, the optimal value of (3.4) and (3.5) are the same. We can obtain the optimal solution of (3.5) by rescaling the optimal solution of (3.4) and vice versa.

For simplicity, we write $Y \in \mathbb{R}^{N\times d}$ whose n-row is $\nabla \log \pi(x_n)$ for $n \in [N]$. We introduce the slack variable $z_n = \sum_{i=1}^m \alpha_i w_i \psi'(x_n^T w_i)$ for $n \in [N]$ and denote $Z = \begin{bmatrix} z_1 & \dots & z_N \end{bmatrix}^T \in \mathbb{R}^{N\times d}$. Then, we can simplify the problem (3.5) to

$$\begin{aligned} \min_{W,\alpha,Z} \; & \frac{1}{2}\|Z\|_F^2 + \sum_{n=1}^N \sum_{i=1}^m \alpha_i \|w_i\|_2^2 \psi''(w_i^T x_n) \\ & + \operatorname{tr}(Y^T Z) + \tilde{\beta}\|\alpha\|_1, \\ \text{s.t. } & z_n = \sum_{i=1}^m \alpha_i w_i \psi'(x_n^T w_i), n \in [N], \\ & \|w_i\|_2 \leq 1, i \in [m]. \end{aligned} \tag{3.9}$$

To derive the convex relaxation of the neural network training problem, the dual problem plays an important role. By applying the Lagrangian duality, we can derive the dual problem of (3.9) as follows.

Proposition 3.4 (Dual problem). *The dual problem of the regularized variational problem* (3.9) *is*

$$\begin{aligned} & -\frac{1}{2}\|\Lambda + Y\|_F^2, \\ s.t. \; & \max_{w:\|w\|_2\leq 1} \left| \sum_{n=1}^N \|w\|_2^2 \psi''(x_n^T w) - \lambda_n^T w \psi'(x_n^T w) \right| \leq \tilde{\beta}, \end{aligned} \tag{3.10}$$

which provides a lower-bound on (3.9)*. .*

Proof. Consider the Lagrangian function

$$\begin{aligned} L(Z, W, \alpha, \Lambda) =& \frac{1}{2}\|Z\|_F^2 + \sum_{n=1}^N \sum_{i=1}^m \alpha_i \|w_i\|_2^2 \psi''(w_i^T x_n) + \operatorname{tr}(Y^T Z) + \tilde{\beta}\|\alpha\|_1 \\ & + \sum_{n=1}^N \lambda_n^T \left(z_n - \sum_{i=1}^m \alpha_i w_i \psi'(x_n^T w_i) \right) \\ =& \tilde{\beta}\|\alpha\|_1 + \sum_{i=1}^m \alpha_i \sum_{n=1}^N \left(\|w_i\|_2^2 \psi''(w_i^T x_n) - \lambda_n^T w_i \psi'(x_n^T w_i) \right) \\ & + \frac{1}{2}\|Z\|_F^2 + \operatorname{tr}((Y+\Lambda)^T Z). \end{aligned} \tag{3.11}$$

For fixed W, the constraints on Z and α are linear and the strong duality holds. Thus, we can exchange the order of $\min_{Z,\alpha}$ and $\max_\Lambda$. Thus, we can compute that

$$\begin{aligned} & \min_{W\in\mathcal{W},Z,\alpha} \max_\Lambda L(Z, W, \alpha, \Lambda) \\ =& \min_{W\in\mathcal{W}} \max_\Lambda \min_{\alpha,Z} L(Z, W, \alpha, \Lambda) \\ =& \min_{W\in\mathcal{W}} \max_\Lambda \min_{\alpha,Z} \tilde{\beta}\|\alpha\|_1 + \sum_{i=1}^m \alpha_i \sum_{n=1}^N \left(\|w_i\|_2^2 \psi''(w_i^T x_n) - \lambda_n^T w_i \psi'(x_n^T w_i) \right) + \frac{1}{2}\|Z\|_F^2 + \operatorname{tr}((Y+\Lambda)^T Z) \\ =& \min_{W\in\mathcal{W}} \max_\Lambda -\frac{1}{2}\|\Lambda + Y\|_F^2 + \sum_{i=1}^m \mathbb{I}\left(\max_{w_i:\|w_i\|_2\leq 1} \left| \sum_{n=1}^N \|w_i\|_2^2 \psi''(w_i^T x_n) - \lambda_n^T w_i \psi'(x_n^T w_i) \right| \leq \tilde{\beta} \right). \end{aligned} \tag{3.12}$$

By exchanging the order of min and max, we can derive the dual problem:

(3.13)

$$
\begin{aligned}
&\max_{\Lambda} \min_{W\in\mathcal{W}} -\frac{1}{2}\|\Lambda+Y\|_F^2+\sum_{i=1}^m \mathbb{I}\left(\max_{w_i:\|w_i\|_2\leq 1}\left|\sum_{n=1}^N \|w_i\|_2^2\psi''(w_i^Tx_n)-\lambda_n^Tw_i\psi'(x_n^Tw_i)\right|\leq\tilde\beta\right)\\
&=\max_{\Lambda} -\frac{1}{2}\|\Lambda+Y\|_F^2 \text{ s.t. } \max_{w_i:\|w_i\|_2\leq 1}\left|\sum_{n=1}^N \|w_i\|_2^2\psi''(w_i^Tx_n)-\lambda_n^Tw_i\psi'(x_n^Tw_i)\right|\leq\tilde\beta, i\in[m]\\
&=\max_{\Lambda} -\frac{1}{2}\|\Lambda+Y\|_F^2 \text{ s.t. } \max_{w:\|w\|_2\leq 1}\left|\sum_{n=1}^N \|w\|_2^2\psi''(w^Tx_n)-\lambda_n^Tw\psi'(x_n^Tw)\right|\leq\tilde\beta, i\in[m]
\end{aligned}
$$

This completes the proof. ∎

We note that the dual problem can be infeasible if the regularization parameter $\tilde\beta$ is below a certain threshold. In other words, if the regularization term is missing or the regularization parameter is not large enough, the optimal value of the dual problem is $-\infty$ and the primal problem is not lower bounded.

3.1. Analysis of dual constraints and the relaxed dual problem. Now, we analyze the constraint in the dual problem. We note that it is closely related to the regularization parameter, which we will discuss later. For simplicity, we take $\psi''(0)=0$ as the subgradient of $\psi'(z)$ at $z=0$, i.e., taking the left derivative of $\psi'(z)$ at $z=0$. Let $X=[x_1,\dots,x_N]^T\in\mathbb{R}^{N\times d}$. Denote the set of all possible hyper-plane arrangements corresponding to the rows of X as

$$
\mathcal{S}=\{\mathbf{diag}(\mathbb{I}(Xw\geq 0))|w\in\mathbb{R}^d, w\neq 0\}. \tag{3.14}
$$

Here $\mathbb{I}(s)=1$ if the statement s is correct and $\mathbb{I}(s)=0$ otherwise. Let $p=|\mathcal{S}|$ be the cardinality of $\mathcal{S}$, and write $\mathcal{S}=\{D_1,\dots,D_p\}$. According to [12], we have the upper bound $p\leq 2r\left(\frac{e(N-1)}{r}\right)^r$, where $r=\text{rank}(X)$. Based on the analysis of the dual constraints, we can derive a convex SDP as a relaxed dual problem.

Proposition 3.5 (Relaxed dual problem). *The relaxed dual problem is the following SDP:*

$$
\begin{aligned}
&\max_{\Lambda,\{r^{(j,-)},r^{(j,+)}\}_{j=1}^p} -\frac{1}{2}\|\Lambda+Y\|_F^2,\\
&s.t.\ \tilde A_j(\Lambda)+\tilde B_j+\sum_{n=0}^N r_n^{(j,-)}H_n^{(j)}+\tilde\beta e_{d+1}e_{d+1}^T\succeq 0\\
&\quad -\tilde A_j(\Lambda)-\tilde B_j+\sum_{n=0}^N r_n^{(j,+)}H_n^{(j)}+\tilde\beta e_{d+1}e_{d+1}^T\succeq 0\\
&\quad r^{(j,+)}\geq 0, r^{(j,-)}\geq 0, j\in[p],
\end{aligned} \tag{3.15}
$$

where we denote $[p]=\{1,\dots,p\}$. For $j\in[p]$, we denote $A_j(\Lambda)=-\Lambda^TD_jX-X^TD_j\Lambda$, $B_j=2\operatorname{tr}(D_j)I_d$, $\tilde A_j(\Lambda)=\begin{bmatrix}A_j(\Lambda)&0\\0&0\end{bmatrix}$, $\tilde B_j=\begin{bmatrix}B_j&0\\0&0\end{bmatrix}$, $H_0^{(j)}=\begin{bmatrix}I_d&0\\0&-1\end{bmatrix}$ and $H_n^{(j)}=\begin{bmatrix}0&(1-2(D_j)_{nn})x_n\\(1-2(D_j)_{nn})x_n^T&0\end{bmatrix}$, $n\in[N]$ The vector $e_{d+1}\in\mathbb{R}^{d+1}$ satisfies that $(e_{d+1})_i=0$ for $i\in[d]$ and $(e_{d+1})_{d+1}=1$.

The optimal value of (3.15) *gives a lower bound on the dual problem* (3.10)*, and hence on the primal problem* (3.9).

Proof. Based on the hyper-plane arrangements $D_1, \dots, D_p$, the dual constraint is equivalent to that for all $j \in [p]$,

$$\left|2\operatorname{tr}(D_j)\|w\|_2^2 - 2w^T\Lambda^T D_j X w\right| \le \tilde{\beta} \tag{3.16}$$

holds for all $w \in \mathbb{R}^d$ satisfying $\|w\|_2 \le 1, (2D_j - I)Xw \ge 0$. This is equivalent to say that for all $j \in [p]$

$$\begin{aligned} \tilde{\beta} \ge & \min 2\operatorname{tr}(D_j)\|w\|_2^2 - 2w^T\Lambda^T D_j X w, \\ & \text{s.t. } \|w\|_2 \le 1, 2(D_j - I)Xw \ge 0, \\ -\tilde{\beta} \le & \max 2\operatorname{tr}(D_j)\|w\|_2^2 - 2w^T\Lambda^T D_j X w, \\ & \text{s.t. } \|w\|_2 \le 1, 2(D_j - I)Xw \ge 0. \end{aligned} \tag{3.17}$$

From a convex optimization perspective, the natural idea to interpret the constraint (3.17) is to transform the minimization problem into a maximization problem. We can rewrite the minimization problem in (3.17) as a trust region problem with inequality constraints:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \ & w^T \left(B_j + A_j(\Lambda)\right) w, \\ \text{s.t. } & \|w\|_2 \le 1, (2D_j - I)Xw \ge 0. \end{aligned} \tag{3.18}$$

As the problem (3.18) is a convex problem, by taking the dual of (3.18) w.r.t. w, we can transform (3.18) into a maximization problem. However, as (3.18) is a trust region problem with inequality constraints, the dual problem of (3.18) can be very complicated. According to [26], the optimal value of the problem (3.18) is bounded by the optimal value of the following SDP

$$\begin{aligned} \min_{Z \in \mathbb{S}^{d+1}} \ & \operatorname{tr}((\tilde{A}_j(\Lambda) + \tilde{B}_j)Z), \\ \text{s.t. } & \operatorname{tr}(H_n^{(j)} Z) \le 0, n = 0, \dots, N, \\ & Z_{d+1,d+1} = 1, Z \succeq 0. \end{aligned} \tag{3.19}$$

from below.

LEMMA 3.6. *The dual problem of SDP* (3.19) *takes the form*

$$\max -\gamma, \ \textit{s.t. } S = \tilde{A}_j(\Lambda) + \tilde{B}_j + \sum_{n=0}^{N} r_n H_n^{(j)} + \gamma e_{d+1} e_{d+1}^T, r \ge 0, S \succeq 0, \tag{3.20}$$

in variables $r = \begin{bmatrix} r_0 \\ \vdots \\ r_N \end{bmatrix} \in \mathbb{R}^{N+1}$ *and* $\gamma \in \mathbb{R}$.

Proof. Consider the Lagrangian

$$L(Z, r, \gamma) = \operatorname{tr}((\tilde{A}_j(\Lambda) + \tilde{B}_j)Z) + \sum_{n=0}^{N} r_n \operatorname{tr}(H_n^{(j)} Z) + \gamma(\operatorname{tr}(Z e_{d+1} e_{d+1}^T) - 1), \tag{3.21}$$

where $r \in \mathbb{R}_+^{N+1}$ and $\gamma \in \mathbb{R}$. By minimizing $L(Z, r, \gamma)$ w.r.t. $Z \in \mathbb{S}_+^{d+1}$, we derive the dual problem (3.20). ∎

The constraints on Λ in the dual problem (3.10) include that the optimal value of (3.19) is bounded from below by $-\tilde{\beta}$. According to Lemma 3.6, this constraint is equivalent to that there exist $r \in \mathbb{R}^{N+1}$ and γ such that

$$-\gamma \geq -\tilde{\beta}, S = \tilde{A}_j(\Lambda) + \tilde{B}_j + \sum_{n=0}^{N} r_n H_n^{(j)} + \gamma e_{d+1} e_{d+1}^T, r \geq 0, S \succeq 0. \tag{3.22}$$

As $e_{d+1} e_{d+1}^T$ is positive semi-definite, the above condition on Λ is also equivalent to that there exist $r \in \mathbb{R}^{N+1}$ such that

$$\tilde{A}_j(\Lambda) + \tilde{B}_j + \sum_{n=0}^{N} r_n H_n^{(j)} + \tilde{\beta} e_{d+1} e_{d+1}^T \succeq 0, r \geq 0. \tag{3.23}$$

Therefore, the following convex set of Λ

$$\left\{ \Lambda : \tilde{A}_j(\Lambda) + \tilde{B}_j + \sum_{n=0}^{N} r_n^{(j,-)} H_n^{(j)} + \tilde{\beta} e_{d+1} e_{d+1}^T \succeq 0,\ r^{(j,-)} \geq 0 \right\} \tag{3.24}$$

is a subset of the set of Λ satisfying the dual constraints

$$\left\{ \Lambda : \min_{\|w\|_2 \leq 1, (2D_j - I)w \geq 0} w^T \left(B_j + A_j(\Lambda) \right) w \geq -\tilde{\beta} \right\}. \tag{3.25}$$

On the other hand, the constraint on Λ

$$\max_{\|w\|_2 \leq 1, (2D_j - I)w \geq 0} w^T \left(B_j + A_j(\Lambda) \right) w \leq \tilde{\beta} \tag{3.26}$$

is equivalent to

$$\min_{\|w\|_2 \leq 1, (2D_j - I)w \geq 0} -w^T \left(B_j + A_j(\Lambda) \right) w \geq -\tilde{\beta}. \tag{3.27}$$

By applying the previous analysis on the above trust region problem, the following convex set of Λ

$$\left\{ \Lambda : -\tilde{A}_j(\Lambda) - \tilde{B}_j + \sum_{n=0}^{N} r_n^{(j,+)} H_n^{(j)} + \tilde{\beta} e_{d+1} e_{d+1}^T \succeq 0,\ r^{(j,+)} \geq 0 \right\} \tag{3.28}$$

is a subset of the set of Λ satisfying the dual constraints

$$\left\{ \Lambda : \max_{\|w\|_2 \leq 1, (2D_j - I)w \geq 0} w^T \left(B_j + A_j(\Lambda) \right) w \leq \tilde{\beta} \right\}. \tag{3.29}$$

Therefore, replacing the dual constraint by

$$\begin{aligned} &\tilde{A}_j(\Lambda) + \tilde{B}_j + \sum_{n=0}^{N} r_n^{(j,-)} H_n^{(j)} + \tilde{\beta} e_{d+1} e_{d+1}^T \succeq 0, j \in [p], \\ &-\tilde{A}_j(\Lambda) - \tilde{B}_j + \sum_{n=0}^{N} r_n^{(j,+)} H_n^{(j)} + \tilde{\beta} e_{d+1} e_{d+1}^T \succeq 0, j \in [p], \\ &r^{(j,-)} \geq 0, r^{(j,+)} \geq 0, j \in [p], \end{aligned} \tag{3.30}$$

we obtain the relaxed dual problem. As its feasible domain is a subset of the feasible domain of the dual problem, the optimal value of the relaxed dual problem gives a lower bound for the optimal value of the dual problem. ∎

Now we consider the case when the relaxation is inexact, i.e., the relaxed dual problem has a smaller optimal value compared to the dual problem. In this case, the relaxed bi-dual problem provides insights on approximating the primal problem via convex optimization, which is derived as follows. As an equivalent formulation of the convex dual problem (3.15), it can be viewed as a convex relaxation of the primal problem (3.9).

PROPOSITION 3.7 (Relaxed bi-dual problem). *The dual of the relaxed dual problem* (3.15) *is as follows*

$$
\begin{aligned}
\min_{Z,\{(S^{(j,+)},S^{(j,-)})\}_{j=1}^p} \quad & \frac{1}{2}\|Z+Y\|_F^2-\frac{1}{2}\|Y\|_F^2 \\
& +\sum_{j=1}^p \operatorname{tr}(\tilde{B}_j(S^{(j,+)}-S^{(j,-)})) \\
& +\tilde{\beta}\sum_{j=1}^p \operatorname{tr}\left((S^{(j,+)}+S^{(j,-)})e_{d+1}e_{d+1}^T\right), \\
\text{s.t. } \quad & Z=\sum_{j=1}^p \tilde{A}_j^*(S^{(j,-)}-S^{(j,+)}), \\
& \operatorname{tr}(S^{(j,-)}H_n^{(j)})\leq 0, \operatorname{tr}(S^{(j,+)}H_n^{(j)})\leq 0, \\
& n=0,\dots,N, j\in[p].
\end{aligned} \tag{3.31}
$$

Here A_j^* is the adjoint operator of the linear operator A_j.

Proof. Consider the Lagrangian function

$$
\begin{aligned}
&L(\Lambda,\mathbf{r},\mathbf{S}) \\
&=-\frac{1}{2}\|\Lambda+Y\|_2^2-\sum_{j=1}^p \operatorname{tr}\left(S^{(j,-)}\left(\tilde{A}_j(\Lambda)+\tilde{B}_j+\sum_{n=0}^N r_n^{(j,-)}H_n^{(j)}+\frac{\tilde{\beta}}{2}e_{d+1}e_{d+1}^T\right)\right) \\
&\quad -\sum_{j=1}^p \operatorname{tr}\left(S^{(j,+)}\left(-\tilde{A}_j(\Lambda)-\tilde{B}_j+\sum_{n=0}^N r_n^{(j,+)}H_n^{(j)}+\frac{\tilde{\beta}}{2}e_{d+1}e_{d+1}^T\right)\right),
\end{aligned} \tag{3.32}
$$

where we write

$$
\begin{aligned}
\mathbf{r}&=\left(r^{(1,-)},\dots,r^{(p,-)},r^{(1,+)},\dots,r^{(p,+)}\right)\in\left(\mathbb{R}^{N+1}\right)^{2p}, \\
\mathbf{S}&=\left(S^{(1,-)},\dots,S^{(p,-)},S^{(1,+)},\dots,S^{(p,+)}\right)\in\left(\mathbb{S}_+^{d+1}\right)^{2p}.
\end{aligned} \tag{3.33}
$$

Here we write $\mathbb{S}_+^{d+1}=\{S\in\mathbb{S}^{d+1}|S\succeq 0\}$. By maximizing w.r.t. Λ and $\mathbf{r}$, we derive the bi-dual problem (3.31). ∎

As (3.15) is a convex problem and the Slater's condition is satisfied, the optimal values of (3.15) and (3.31) are same. The bi-dual problem (3.31) is closely related to the primal problem (3.9). Indeed, any feasible solutions of the primal problem (3.5) can be mapped to feasible solutions of (3.31). We note that the mapping from the primal solution to the bi-dual solution cannot go both ways unless these two problems are equivalent.

THEOREM 3.8. *Suppose that* (Z, W, α) *is feasible to the primal problem* (3.9). *Then, there exist matrices* $\{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p$ *constructed from* (W, α) *such that*

$(Z, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$ *is feasible to the relaxed bi-dual problem* (3.31). *Moreover, the objective value of the relaxed bi-dual problem* (3.31) *at* $(Z, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$ *is the same as objective value of the primal problem* (3.9) *at* (Z, W, α).

Proof. Suppose that (Z, W, α) is a feasible solution to (3.5). Let $D_{j_1}, \dots, D_{j_k}$ be the enumeration of $\{\mathbf{diag}(\mathbb{I}(Xw_i \geq 0)) | i \in [m]\}$. For $i \in [k]$, we let

$$S^{(j_i,+)} = \sum_{l:\alpha_l \geq 0, \mathbf{diag}(\mathbb{I}(Xw_l \geq 0)) = D_{j_i}} \alpha_l \begin{bmatrix} w_l w_l^T & w_l \\ w_l^T & 1 \end{bmatrix}, S^{(j_i,-)} = 0, \tag{3.34}$$

and

$$S^{(j_i,+)} = 0, S^{(j_i,-)} = -\sum_{l:\alpha_l < 0, \mathbf{diag}(\mathbb{I}(Xw_l \geq 0)) = D_{j_i}} \alpha_l \begin{bmatrix} w_l w_l^T & w_l \\ w_l^T & 1 \end{bmatrix}. \tag{3.35}$$

For $j \notin \{j_1, \dots, j_k\}$, we simply set $S^{(j,+)} = 0, S^{(j,-)} = 0$. As $\|w_i\|_2 \leq 1$ and $D_{j_i} = \mathbb{I}(Xw_i \geq 0)$, we can verify that $\mathrm{tr}(S^{(j,-)} H_n^{(j)}) \leq 0, \mathrm{tr}(S^{(j,+)} H_n^{(j)}) \leq 0$ are satisfied for $j = j_1, \dots, j_m$ and $n = 0, 1, \dots, N$. This is because for $n = 0$, as $H_0^{(j_i)} = \begin{bmatrix} I_d & 0 \\ 0 & -1 \end{bmatrix}$, it follows that

$$\begin{aligned} \mathrm{tr}(S^{(j_i,+)} H_0^{(j_i)}) &= \sum_{l:\alpha_l \geq 0, \mathbf{diag}(\mathbb{I}(Xw_l \geq 0)) = D_{j_i}} \alpha_l (\|w_l\|^2 - 1) \leq 0, \\ \mathrm{tr}(S^{(j_i,-)} H_0^{(j_i)}) &= -\sum_{l:\alpha_l < 0, \mathbf{diag}(\mathbb{I}(Xw_l \geq 0)) = D_{j_i}} \alpha_l (\|w_l\|^2 - 1) \leq 0. \end{aligned} \tag{3.36}$$

For $n = 1, \dots, N$, we have

$$\begin{aligned} \mathrm{tr}(S^{(j_i,+)} H_0^{(j_i)}) &= \sum_{l:\alpha_l \geq 0, \mathbf{diag}(\mathbb{I}(Xw_l \geq 0)) = D_{j_i}} 2\alpha_l (1 - 2(D_{j_i})_{nn}) x_n^T w_l \leq 0, \\ \mathrm{tr}(S^{(j_i,-)} H_0^{(j_i)}) &= -\sum_{l:\alpha_l < 0, \mathbf{diag}(\mathbb{I}(Xw_l \geq 0)) = D_{j_i}} \alpha_l (1 - 2(D_{j_i})_{nn}) x_n^T w_l \leq 0. \end{aligned} \tag{3.37}$$

Based on the above transformation, we can rewrite the bidual problem in the form of the primal problem (3.9). For $S \in \mathbb{S}^{d+1}$, we note that

$$\begin{aligned} &\mathrm{tr}(S \tilde{A}_j(\Lambda)) \\ =& -\mathrm{tr}((\Lambda^T D_j X + X^T D_j \Lambda) S_{1:d,1:d}) \\ =& -2\,\mathrm{tr}(\Lambda^T D_j X S_{1:d,1:d}), \end{aligned}$$

where $S_{1:d,1:d}$ denotes the $d \times d$ block of S consisting the first d rows and columns. This implies that $\tilde{A}_j^*(S) = -2 D_j X S_{1:d,1:d}$. Hence, we have

$$\begin{aligned} \tilde{A}_{j_i}(S^{(j_i,+)} - S^{(j_i,-)}) &= -\sum_{l:\mathbf{diag}(\mathbb{I}(Xw_l \geq 0)} 2\alpha_l D_{j_i} X w_l w_l^T \\ &= -\sum_{l:\mathbf{diag}(\mathbb{I}(Xw_l \geq 0)} 2\alpha_l (Xw_l)_+ w_l^T. \end{aligned}$$

Therefore, we have

$$\sum_{j=1}^{p} \tilde{A}_j^*(S^{(j,-)} - S^{(j,+)}) = 2\sum_{i=1}^{m} \alpha_i (Xw_i)_+ w_i^T.$$

As n-th row of Z satisfies that $z_n = 2\sum_{i=1}^m \alpha_i w_i (x_n^T w_i)_+$, this implies that

$$Z = 2\sum_{i=1}^{m} \alpha_i (Xw_i)_+ w_i^T = \sum_{j=1}^{p} \tilde{A}_j^*(S^{(j,-)} - S^{(j,+)}).$$

Hence $(Z, \{(S^{(j,-)}, (S^{(j,-)}\}_{j=1}^p)$ is feasible to the relaxed bi-dual problem (3.31).

We can also compute that

$$\sum_{j=1}^{p} \operatorname{tr}(\tilde{B}_j(S^{(j,+)} - S^{(j,-)})) = 2\sum_{i=1}^{m} \alpha_i \sum_{n=1}^{N} \mathbb{I}(x_n^T w_i \geq 0)\|w_i\|_2^2,$$

and

$$\sum_{j=1}^{p} \operatorname{tr}\left((S^{(j,+)} + S^{(j,-)}) e_{d+1} e_{d+1}^T\right) = \sum_{i=1}^{m} |\alpha_i|.$$

Thus, the primal problem (3.9) with (Z, W, α) and the relaxed bi-dual problem (3.31) with $(Z, \{(S^{(j,-)}, (S^{(j,-)}\}_{j=1}^p)$ have the same objective value. ∎

Let $J(Z, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$ denote the objective value of the relaxed bi-dual problem (3.31) at a feasible solution $(Z, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$. Let (Z^*, W^*, α^*) denote a globally optimal solution of the primal problem (3.9). By Theorem 3.8, there exist matrices $\{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p$ such that $(Z^*, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$ is a feasible solution of the relaxed bi-dual problem (3.31) and $J(Z^*, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$ is the same as the objective value of (3.9) at its global minimum (Z^*, W^*, α^*). On the other hand, let $(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p)$ denote an optimal solution of the relaxed bi-dual problem (3.31). From the optimality of $(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p)$, we have

$$J(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p) \leq J(Z^*, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p).$$

Note that at (Z^*, W^*, α^*) we obtain the optimal approximation of $\nabla \log \rho - \nabla \log \pi$ at $x_1, \dots, x_N$ in the family of two-layer squared-ReLU networks (3.1). Smaller or equal objective value of the relaxed bi-dual problem (3.31) can be achieved at the pair $(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p)$ than at $(Z^*, \{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p)$. Therefore, we can view $\tilde{Z}^*$ gives an optimal approximation of $\nabla \log \rho - \nabla \log \pi$ evaluated on $x_1, \dots, x_N$ in a broader function family including the two-layer squared ReLU neural networks.

From the derivation of the relaxed bi-dual problem, we have the relation $\tilde{Z}^* = -\Lambda^* - Y$, where $(\Lambda^*, \{r^{(j,+)}, r^{(j,-)})$ is optimal to the relaxed dual problem (3.15) and $(\tilde{Z}^*, \{\tilde{S}^{(j,+)}, \tilde{S}^{(j,-)}\}_{j=1}^p)$ is optimal to the relaxed bi-dual problem (3.31). Therefore, by solving Λ^* from the relaxed dual problem (3.15), we can use $-\Lambda^* - Y$ as the approximation of $\nabla \log \rho - \nabla \log \pi$ evaluated on $x_1, \dots, x_N$.

Remark 3.9. We note that solving the proposed convex optimization problem 3.15 renders the approximation of the Wasserstein gradient direction. Compared to the two-layer ReLU networks, it induces a broader class of functions represented by $\{S^{(j,+)}, S^{(j,-)}\}_{j=1}^p$. This contains more variables than the neural network function.

3.2. Choice of the regularization parameter. As the constraints in the relaxed dual problem (3.15) depend on the regularization parameter $\tilde{\beta}$, it is possible that for small $\tilde{\beta}$, the relaxed dual problem (3.15) is infeasible. Consider the following SDP

$$\begin{aligned} \min\ \tilde{\beta},\ \text{s.t.}\ &\tilde{A}_j(\Lambda)+\tilde{B}_j+\sum_{n=0}^{N} r_n^{(j,-)}H_n^{(j)}+\tilde{\beta}e_{d+1}e_{d+1}^T\succeq 0,\\ &-\tilde{A}_j(\Lambda)-\tilde{B}_j+\sum_{n=0}^{N} r_n^{(j,+)}H_n^{(j)}+\tilde{\beta}e_{d+1}e_{d+1}^T\succeq 0,\\ &r^{(j,-)}\geq 0, r^{(j,+)}\geq 0, j\in[p]. \end{aligned} \tag{3.38}$$

Here the variables are $\tilde{\beta},\Lambda$ and $\{r^{(j,+)},r^{(j,-)}\}_{j=1}^p$. Let $\tilde{\beta}_1$ be the optimal value of the above problem. Then, only for $\tilde{\beta}\geq\tilde{\beta}_1$, there exists $\Lambda\in\mathbb{R}^{N\times d}$ satisfying the constraints in (3.15). In other words, the relaxed dual problem (3.15) is feasible. We also note that $\tilde{\beta}_1$ only depends on the samples X and it does not depend on the value of $\nabla\log\pi$ evaluated on $x_1,\dots,x_N$. On the other hand, consider the following SDP

$$\begin{aligned} \min\ \tilde{\beta},\ \text{s.t.}\ &\tilde{A}_j(Y)+\tilde{B}_j+\sum_{n=0}^{N} r_n^{(j,-)}H_n^{(j)}+\tilde{\beta}e_{d+1}e_{d+1}^T\succeq 0,\\ &-\tilde{A}_j(Y)-\tilde{B}_j+\sum_{n=0}^{N} r_n^{(j,+)}H_n^{(j)}+\tilde{\beta}e_{d+1}e_{d+1}^T\succeq 0,\\ &r^{(j,-)}\geq 0, r^{(j,+)}\geq 0, j\in[p], \end{aligned} \tag{3.39}$$

where the variables are $\tilde{\beta}$ and $\{r^{(j,+)},r^{(j,-)}\}_{j=1}^p$. Let $\tilde{\beta}_2$ be the optimal value of the above problem. For $\tilde{\beta}\geq\tilde{\beta}_2$, as $\mathbf{Y}$ is feasible for the constraints in (3.15), the optimal value of the relaxed dual problem (3.15) is 0. In short, only when $\tilde{\beta}\in[\tilde{\beta}_1,\tilde{\beta}_2]$, the variational problem (3.15) is non-trivial. To ensure that solving the relaxed dual problem (3.15) gives a good approximation of the Wasserstein gradient direction, we shall avoid choosing $\tilde{\beta}$ either too small or too large.

3.3. Practical implementation. Although the number p of all possible hyperplane arrangements is upper bounded by $2r((N-1)e/r)^r$ with $r=\text{rank}(X)$, it is computationally costly to enumerate all possible p matrices $D_1,\dots,D_p$ to represent the constraints in the relaxed dual problem (3.5). In practice, we first randomly sample M i.i.d. random vectors $u_1,\dots,u_M\sim\mathcal{N}(0,I_d)$ and generate a subset $\hat{\mathcal{S}}=\{\mathbf{diag}(\mathbb{I}(Xu_j\geq 0)|j\in[M]\}$. of $\mathcal{S}$. Then, we optimize the randomly sub-sampled version of the relaxed dual problem based on the subset $\hat{\mathcal{S}}$ and obtain the solution Λ. Here $-\Lambda-Y$ is used as the direction to update the particle system X. If the regularization parameter is too large, then we will have $-\Lambda-Y=0$, which makes the particle system unchanged. Therefore, to ensure that $\tilde{\beta}$ is not too large, we decay $\tilde{\beta}$ by a factor $\gamma_1\in(0,1)$. This also appears in [19]. On the other hand, if $\tilde{\beta}$ is too small resulting the relaxed dual problem (3.5) infeasible, we increase $\tilde{\beta}$ by multiplying γ_2^{-1}, where $\gamma_2\in(0,1)$. The overall algorithm is summarized in Algorithm 3.1.

Applying the standard interior point method [7] leads to the computational time

$$O((\max\{N,d^2\}\hat{p})^6). \tag{3.40}$$

For high-dimensional problems, i.e., d is large, the computational cost of solving (3.15) can be large. In this case, we apply the dimension-reduction techniques [55, 11, 48] to

Algorithm 3.1 Convex neural Wasserstein descent

Require: initial positions $\{x_0^n\}_{n=1}^N$, step size α_l, initial regularization parameter $\tilde{\beta}_0$, $\gamma_1, \gamma_2 \in (0,1)$.

```
while not converge do
    Form X_l and Y_l based on {x_l^n}_{n=1}^N and {∇ log π(x_l^n)}_{n=1}^N.
    Solve Λ_l from the relaxed dual problem (3.15) with β̃ = β̃_l.
    if the relaxed dual problem with β̃ = β̃_l is infeasible then
        Set X_{l+1} = X_l for n ∈ [N] and set β̃_{l+1} = γ_2^{-1} β̃_l.
    else
        Update X_{l+1} = X_l + α_l(Λ_l + Y_l) for n ∈ [N] and set β̃_{l+1} = γ_1 β̃_l.
    end if
end while
```

reduce the parameter dimension d to a data-informed intrinsic dimension $\hat{d}$, which is often very low, i.e., $\hat{d} \ll d$, thus significantly reducing the computational time (3.40).

4. Numerical experiments. In this section, we present numerical results to compare WGD approximated by neural networks (WGD-NN) and WGD approximated using convex optimization formulation of neural networks (WGD-cvxNN). The performance of compared methods is assessed by the sample goodness-of-fit of the posterior. For WGD-NN, in each iteration, it updates the particle system using (2.3) with a function Φ represented by a two-layer squared ReLU neural network. The parameters of the neural network are obtained by directly solving the nonconvex optimization problem (3.4). For high-dimensional problems, we apply the dimension reduction technique and compare the projected versions (pWGD-NN and pWGD-cvxNN).

We note that although the cost for solving the relaxed dual problem (3.15) using standard convex optimization solvers in WGD-cvxNN can be higher compared to that by a direct neural network training in WGD-NN, this cost difference is negligible in the entire optimization dominated by the likelihood evaluation when the model (e.g., PDE) is expensive to solve. In such cases, WGD-cvxNN and WGD-NN have similar computational complexity but WGD-cvxNN achieves better performance. We use the standard convex optimization solver CVXPY [16] with MOSEK[3] inner solver. Applying randomized SDP solvers [54], randomized second-order methods [43, 31] or advanced SDP solvers [56, 53, 49] for the large-scale problem can improve the computation time. Moreover, the induced SDPs have specific structures of many similar constraints. Solving the SDP (3.15) can be accelerated by designing a specialized convex optimization solver, which is left for future work.

4.1. A two-dimensional example. We test and compare the performance of WGD-cvxNN and WGD-NN on a bimodal two-dimensional double-banana posterior distribution introduced in [14]. We first generate 300 posterior samples by a Stein variational Newton (SVN) method [14] as the reference, as shown in Figure 1. We evaluate the performance of WGD-NN and WGD-cvxNN by calculating the maximum mean discrepancy (MMD) between their samples in each iteration and the reference samples. In the comparison, we use $N = 50$ samples and run for 100 iterations with step sizes $\alpha_l = 10^{-3}$. For WGD-cvxNN, we set $\beta = 1$, $\gamma_1 = 0.95$ and $\gamma_2 = 0.95^{10}$. For WGD-NN, we use $m = 200$ neurons and optimize the regularized training problem (3.4) using all samples with the Adam optimizer [29] with learning rate 10^{-3} for 200 sub-iterations. We also set the regularization parameter $\beta = 1$ and decrease it by a factor of 0.95 in each iteration. We find that this setup of parameters is more suitable.

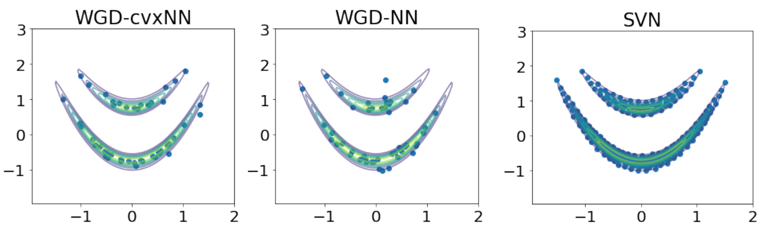

Fig. 1: Two-dimensional example. Posterior density and sample distributions by WGD-cvxNN and WGD-NN at the final step of 100 iterations, compared to the reference SVN samples (right).

The posterior density and the sample distributions by WGD-cvxNN and WGD-NN at the final step of 100 iterations are shown in Figure 1. It can be observed that WGD-cvxNN provides more representative samples than WGD-NN for the posterior density. In Figure 2, we plot the MMD of the samples by WGD-cvxNN and WGD-NN compared to the reference SVN samples at each iteration. We observe that the samples by WGD-cvxNN achieve much smaller MMD than those of WGD-NN compared to the reference SVN samples, which is consistent with the results shown in Figure 1.

4.2. PDE-constrained linear Bayesian inference. In this experiment, we consider a linear Bayesian inference problem constrained by a partial differential equation (PDE) model for contaminant diffusion in environmental engineering in the domain $D = (0, 1)$,

$$-\kappa\Delta u + \nu u = \xi \quad \text{in } D,$$

where ξ is a contaminant source field parameter in domain D, u is the contaminant concentration which we can observe at some locations, κ and ν are diffusion and reaction coefficients. For simplicity, we set $\kappa, \nu = 1$, $u(0) = u(1) = 0$, and consider 15 pointwise observations of u with 1% noise, equidistantly distributed in D. We consider a Gaussian prior distribution $\xi \sim \mathcal{N}(0, C)$ with covariance given by a differential operator $C = (-\delta\Delta + \gamma I)^{-\alpha}$ with $\delta, \gamma, \alpha > 0$ representing the correlation length and variance, which is commonly used in geoscience. We set $\delta = 0.1, \gamma = 1, \alpha = 1$. In this linear setting, the posterior is Gaussian with the mean and covariance given analytically, which are used as a reference to assess the sample goodness. We solve this forward model by a finite element method with piece-wise linear elements on a uniform mesh of size 2^k, $k \geq 1$. We project this high-dimensional parameter to the data-informed low dimensions as in [48] to alleviate the curse of dimensionality when applying WGD-cvxNN and WGD-NN, which we call pWGD-cvxNN and pWGD-NN, respectively. For $k = 4$ we have 17 dimensions for the discrete parameter and 4 dimensions after projection.

We run pWGD-cvxNN and pWGD-NN using 16 samples for 200 iterations with $\alpha_l = 10^{-3}$, $\beta = 5$, $\gamma_1 = 0.95$, and $\gamma_2 = 0.95^{10}$ for both methods. We use $m = 200$ neurons for pWGD-NN and train it by the Adam optimizer for 200 sub-iterations as in the first example. From Figure 3, we observe that pWGD-cvxNN achieves better root mean squared error (RMSE) than pWGD-NN for both the sample mean and the sample variance compared to the reference.

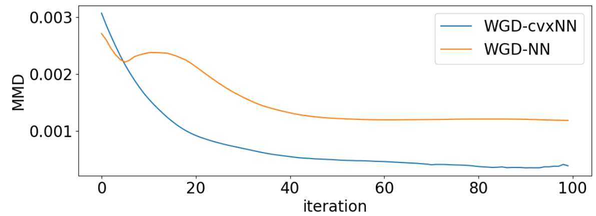

Fig. 2: Two-dimensional example. Maximum mean discrepancy (MMD) of WGD-cvxNN and WGD-NN samples compared to the reference SVN samples.

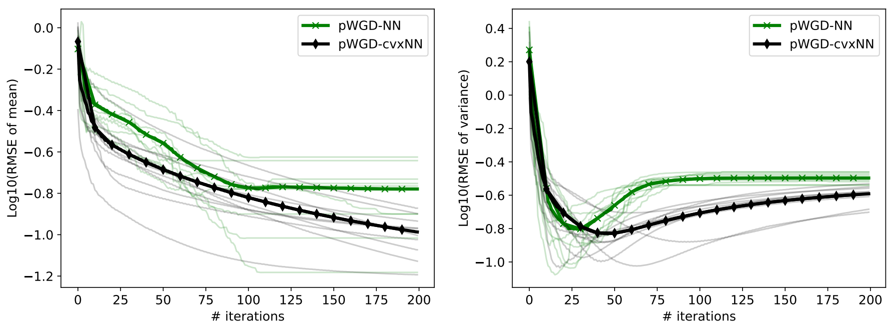

Fig. 3: PDE-constrained linear Bayesian inference. Ten trials and the RMSE of the sample mean (top) and sample variance (bottom) by pWGD-NN and pWGD-cvxNN at different iterations.

4.3. PDE-constrained nonlinear Bayesian inference. In this experiment, we consider a nonlinear Bayesian inference problem constrained by the following partial differential equation (PDE) [11] with application to subsurface (Darcy) flow in a physical domain $D = (0, 1)^2$,

$$\begin{aligned} \mathbf{v} + e^{\xi}\nabla u &= 0 \quad \text{in } D, \\ \nabla \cdot \mathbf{v} &= h \quad \text{in } D, \end{aligned} \tag{4.1}$$

where u is pressure, $\mathbf{v}$ is velocity, h is force, e^{ξ} is a random (permeability) field equipped with a Gaussian prior $\xi \sim \mathcal{N}(\xi_0, C)$ with covariance operator $C = (-\delta\Delta + \gamma I)^{-\alpha}$ where we set $\delta = 0.1, \gamma = 1, \alpha = 2$ and $\xi_0 = 0$. This problem is widely used in many areas, for instance, in estimating permeability in groundwater flow, thermal conductivity in material science, or electrical impedance in medical imaging, We impose Dirichlet boundary conditions $u = 1$ on the top boundary and $u = 0$ on the bottom boundary, and homogeneous Neumann boundary conditions on the left and right boundaries for u. We use a finite element method with piecewise linear elements for the discretization of the problem, resulting in 81 dimensions for the discrete parameter. The data is generated as pointwise observation of the pressure field at 49 points equidistantly distributed in $(0, 1)^2$, corrupted with additive 5% Gaussian noise. We use a DILI-MCMC algorithm [13] with 10000 effective samples to compute the sample mean and sample variance, which are used as the reference values to assess the goodness of the samples.

We run pWGD-cvxNN and pWGD-NN with 64 samples for ten trials with step size $\alpha_l = 10^{-3}$, where we set $\beta = 10$, $\gamma_1 = 0.95$, and $\gamma_2 = 0.95^{10}$ for both methods. The RMSE of the sample mean and sample variance are shown in Figure 4 for the

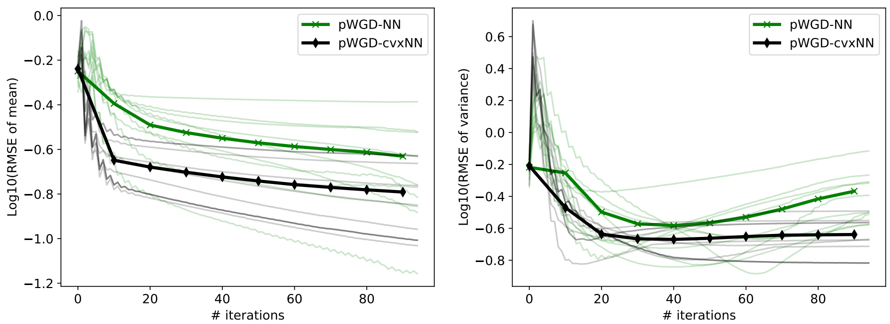

Fig. 4: PDE-constrained non-linear Bayesian inference. Ten trials and the RMSE of the sample mean (top) and sample variance (bottom) by pWGD-NN and pWGD-cvxNN at different iterations.

two methods at each of the iterations. We can observe that pWGD-cvxNN achieves smaller errors for both the sample mean and the sample variance compared to pWGD-NN at each iteration. Moreover, pWGD-cvxNN provides a much smaller variation of the sample mean and sample variance for the ten trials compared to pWGD-NN. Furthermore, by an effective reduction of the parameter dimension from 81 to data-informed 20 in our pWGD-cvxNN, as used and analyzed in [55, 11, 48], the time for solving the SDP is significantly reduced from about 800 seconds to about 0.7 seconds in average, making our pWGD-cvxNN computationally efficient.

4.4. Bayesian inference for COVID-19. In this experiment, we use Bayesian inference to learn the dynamics of the transmission and severity of COVID-19 from the recorded data for New York state. We use the model, parameter, and data as in [11]. More specifically, we use a compartmental model for the modeling of the transmission and outcome of COVID-19. We take the number of hospitalized cases as the observation data to infer a social distancing parameter, a time-dependent stochastic process that is equipped with a Tanh–Gaussian prior to model the transmission reduction effect of social distancing, which becomes 96 dimensions after discretization.

We use the projected Stein variational gradient descent (pSVGD) method [11] as the reference to evaluate the goodness of samples. We run pWGD-cvxNN and pWGD-NN using 64 samples for 100 iterations with step size $\alpha_l = 10^{-3}$, where we set $\beta = 10$, $\gamma_1 = 0.95$, and $\gamma_2 = 0.95^{10}$ for both methods as in the last example. From Figure 5 we can observe that pWGD-cvxNN produces more consistent results than pWGD-NN compared to the reference pSVGD results, for both the sample mean and 90% credible interval, both in the inference of the social distancing parameter and in the prediction of the hospitalized cases.

5. Conclusion. In the context of Bayesian inference, we approximate the Wasserstein gradient direction by the gradient of functions in the family of two-layer neural networks. We propose a convex SDP relaxation of the dual of the variational primal problem, which can be solved efficiently using convex optimization methods instead of directly training the neural network as a nonconvex optimization problem. In particular, we established that the gradient obtained by the new formulation and convex optimization is at least as good as the one approximated by functions in the family of two-layer neural networks, which is demonstrated by various numerical experiments. By stacking the two-layer neural networks in each step together, our

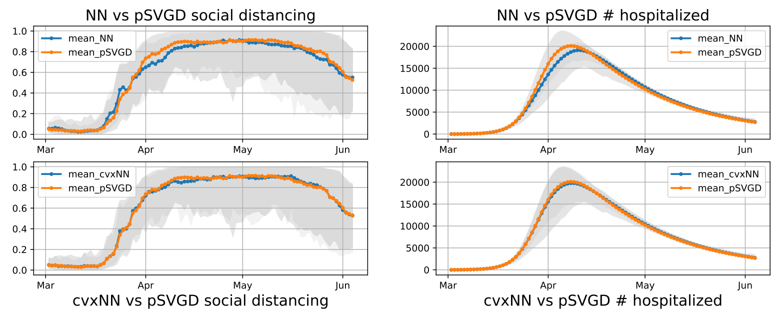

Fig. 5: Bayesian inference for COVID-19. Comparison of pWGD-cvxNN and pWGD-NN to the reference by pSVGD for Bayesian inference of the social distancing parameter (left) from the data of the hospitalized cases (right) with sample mean and 90% credible interval.

proposed method formulates a deep neural network to learn the transportation map from the prior to the posterior. In future studies, specialized optimization solvers for the structured SDPs, including the relaxed dual problem, can lead to significant accelerations of our proposed method. We also expect to apply deep neural networks for the approximation of Wasserstein gradient flows based on recent works on convex optimization formulations of deep neural networks [50, 17, 18]. Detailed study of the conditions where the SDP relaxation is tight is of great interest as it provides more insight from the convex optimization perspective to understand how neural networks fit the data and which kind of datasets is easier to learn. We also expect to bound the number of hyperplane arrangements needed for an approximate solution and give a useful guarantee that bounds the distance between solutions to perturbations of the convex problem in future research.

REFERENCES

[1] D. Alvarez-Melis, Y. Schiff, and Y. Mroueh, *Optimizing functionals on the space of probabilities with input convex neural networks*, arXiv preprint arXiv:2106.00774, (2021).

[2] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*, Springer Science & Business Media, 2005.

[3] M. ApS, *Mosek optimization suite*, 2019.

[4] B. Bartan and M. Pilanci, *Neural spectrahedra and semidefinite lifts: Global convex optimization of polynomial activation neural networks in fully polynomial-time*, arXiv preprint arXiv:2101.02429, (2021).

[5] B. Bartan and M. Pilanci, *Training quantized neural networks to global optimality via semidefinite programming*, in International Conference on Machine Learning, PMLR, 2021, pp. 694–704.

[6] C. Bonet, N. Courty, F. Septier, and L. Drumetz, *Sliced-wasserstein gradient flows*, arXiv preprint arXiv:2110.10972, (2021).

[7] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[8] C. Bunne, L. Meng-Papaxanthos, A. Krause, and M. Cuturi, *Jkonet: Proximal optimal transport modeling of population dynamics*, arXiv preprint arXiv:2106.06345, (2021).

[9] J. A. Carrillo, K. Craig, L. Wang, and C. Wei, *Primal dual methods for wasserstein gradient flows*, Foundations of Computational Mathematics, (2021), pp. 1–55.

[10] J. A. Carrillo, D. Matthes, and M.-T. Wolfram, *Lagrangian schemes for wasserstein gradient flows*, Handbook of Numerical Analysis, 22 (2021), pp. 271–311.

[11] P. Chen and O. Ghattas, *Projected stein variational gradient descent*, Advances in Neural Information Processing Systems, 33 (2020), pp. 1947–1958.

[12] T. M. Cover, *Geometrical and statistical properties of systems of linear inequalities with*

applications in pattern recognition, IEEE transactions on electronic computers, (1965), pp. 326–334.

[13] T. Cui, K. J. Law, and Y. M. Marzouk, *Dimension-independent likelihood-informed mcmc*, Journal of Computational Physics, 304 (2016), pp. 109–137.

[14] G. Detommaso, T. Cui, A. Spantini, Y. Marzouk, and R. Scheichl, *A stein variational newton method*, arXiv preprint arXiv:1806.03085, (2018).

[15] L. L. di Langosco, V. Fortuin, and H. Strathmann, *Neural variational gradient descent*, arXiv preprint arXiv:2107.10731, (2021).

[16] S. Diamond and S. Boyd, *CVXPY: A Python-embedded modeling language for convex optimization*, Journal of Machine Learning Research, 17 (2016), pp. 1–5.

[17] T. Ergen and M. Pilanci, *Global optimality beyond two layers: Training deep relu networks via convex programs*, in International Conference on Machine Learning, PMLR, 2021, pp. 2993–3003.

[18] T. Ergen and M. Pilanci, *Path regularization: A convexity and sparsity inducing regularization for parallel relu networks*, arXiv preprint arXiv:2110.09548, (2021).

[19] T. Ergen, A. Sahiner, B. Ozturkler, J. Pauly, M. Mardani, and M. Pilanci, *Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization*, arXiv preprint arXiv:2103.01499, (2021).

[20] J. Fan, A. Taghvaei, and Y. Chen, *Variational wasserstein gradient flow*, arXiv preprint arXiv:2112.02424, (2021).

[21] X. Feng, Y. Gao, J. Huang, Y. Jiao, and X. Liu, *Relative entropy gradient sampler for unnormalized distributions*, arXiv preprint arXiv:2110.02787, (2021).

[22] C. Frogner and T. Poggio, *Approximate inference with wasserstein gradient flows*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2581–2590.

[23] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural networks, 2 (1989), pp. 359–366.

[24] H. J. Hwang, C. Kim, M. S. Park, and H. Son, *The deep minimizing movement scheme*, arXiv preprint arXiv:2109.14851, (2021).

[25] A. Hyvärinen and P. Dayan, *Estimation of non-normalized statistical models by score matching.*, Journal of Machine Learning Research, 6 (2005).

[26] V. Jeyakumar and G. Li, *Trust-region problems with linear inequality constraints: exact sdp relaxation, global optimality and robust optimization*, Mathematical Programming, 147 (2014), pp. 171–206.

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

[28] O. Junge, D. Matthes, and H. Osberger, *A fully discrete variational scheme for solving nonlinear fokker–planck equations in multiple space dimensions*, SIAM Journal on Numerical Analysis, 55 (2017), pp. 419–443.

[29] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[30] J. Kruse, G. Detommaso, R. Scheichl, and U. Köthe, *Hint: Hierarchical invertible neural transport for density estimation and bayesian inference*, arXiv preprint arXiv:1905.10687, (2019).

[31] J. Lacotte, Y. Wang, and M. Pilanci, *Adaptive newton sketch: Linear-time optimization with quadratic convergence and effective hessian dimensionality*, in International Conference on Machine Learning, PMLR, 2021, pp. 5926–5936.

[32] S. Lan, S. Li, and B. Shahbaba, *Scaling up bayesian uncertainty quantification for inverse problems using deep neural networks*, arXiv preprint arXiv:2101.03906, (2021).

[33] A. T. Lin, S. W. Fung, W. Li, L. Nurbekyan, and S. J. Osher, *Alternating the population and control neural networks to solve high-dimensional stochastic mean-field games*, Proceedings of the National Academy of Sciences, 118 (2021).

[34] A. T. Lin, W. Li, S. Osher, and G. Montúfar, *Wasserstein proximal of gans*, arXiv preprint arXiv:2102.06862, (2021).

[35] C. Liu, J. Zhuo, P. Cheng, R. Zhang, and J. Zhu, *Understanding and accelerating particle-based variational inference*, in International Conference on Machine Learning, PMLR, 2019, pp. 4082–4092.

[36] Q. Liu and D. Wang, *Stein variational gradient descent: A general purpose bayesian inference algorithm*, in Advances in neural information processing systems, 2016, pp. 2378–2386.

[37] A. Liutkus, U. Simsekli, S. Majewski, A. Durmus, and F.-R. Stöter, *Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions*, in International Conference on Machine Learning, PMLR, 2019, pp. 4104–4113.

[38] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, *The expressive power of neural networks:*

A view from the width, in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6232–6240.

[39] P. Mokrov, A. Korotin, L. Li, A. Genevay, J. Solomon, and E. Burnaev, *Large-scale wasserstein gradient flows*, arXiv preprint arXiv:2106.00736, (2021).

[40] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, *Ot-flow: Fast and accurate continuous normalizing flows via optimal transport*, arXiv preprint arXiv:2006.00104, (2020).

[41] F. Otto, *The geometry of dissipative evolution equations: the porous medium equation*, Communications in Partial Differential Equations, 26 (2001), pp. 101–174.

[42] M. Pilanci and T. Ergen, *Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks*, in International Conference on Machine Learning, PMLR, 2020, pp. 7695–7705.

[43] M. Pilanci and M. J. Wainwright, *Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence*, SIAM Journal on Optimization, 27 (2017), pp. 205–245.

[44] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in International conference on machine learning, PMLR, 2015, pp. 1530–1538.

[45] A. Sahiner, T. Ergen, J. Pauly, and M. Pilanci, *Vector-output relu neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms*, arXiv preprint arXiv:2012.13329, (2020).

[46] A. M. Stuart, *Inverse problems: a Bayesian perspective*, Acta numerica, 19 (2010), pp. 451–559.

[47] C. Villani, *Topics in optimal transportation*, American Mathematical Soc., 2003.

[48] Y. Wang, P. Chen, and W. Li, *Projected wasserstein gradient descent for high-dimensional bayesian inference*, arXiv preprint arXiv:2102.06350, (2021).

[49] Y. Wang, K. Deng, H. Liu, and Z. Wen, *A decomposition augmented lagrangian method for low-rank semidefinite programming*, arXiv preprint arXiv:2109.11707, (2021).

[50] Y. Wang, T. Ergen, and M. Pilanci, *Parallel deep neural networks have zero duality gap*, arXiv preprint arXiv:2110.06482, (2021).

[51] Y. Wang, Y. Hua, E. Candés, and M. Pilanci, *Overparameterized relu neural networks learn the simplest models: Neural isometry and exact recovery*, arXiv preprint arXiv:2209.15265, (2022).

[52] Y. Wang, J. Lacotte, and M. Pilanci, *The hidden convex optimization landscape of two-layer relu neural networks: an exact characterization of the optimal solutions*, arXiv preprint arXiv:2006.05900, (2020).

[53] L. Yang, D. Sun, and K.-C. Toh, *Sdpnal +: a majorized semismooth newton-cg augmented lagrangian method for semidefinite programming with nonnegative constraints*, Mathematical Programming Computation, 7 (2015), pp. 331–366.

[54] A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher, *Scalable semidefinite programming*, SIAM Journal on Mathematics of Data Science, 3 (2021), pp. 171–200.

[55] O. Zahm, T. Cui, K. Law, A. Spantini, and Y. Marzouk, *Certified dimension reduction in nonlinear bayesian inverse problems*, arXiv preprint arXiv:1807.03712, (2018).

[56] X.-Y. Zhao, D. Sun, and K.-C. Toh, *A newton-cg augmented lagrangian method for semidefinite programming*, SIAM Journal on Optimization, 20 (2010), pp. 1737–1765.