Analyzing Neural Network-Based Generative Diffusion Models through Convex Optimization

Fangzhao Zhang ZFZHAO@STANFORD.EDU
Mert Pilanci PILANCI@STANFORD.EDU

Abstract

Diffusion models are gaining widespread use in cutting-edge image, video, and audio generation. Score-based diffusion models stand out among these methods, necessitating the estimation of score function of the input data distribution. In this study, we present a theoretical framework to analyze two-layer neural network-based diffusion models by reframing score matching and denoising score matching as convex optimization. We prove that training shallow neural networks for score prediction can be done by solving a single convex program and characterize the exact predicted score function. We also establish convergence results for neural network-based diffusion models with finite data. Our results provide a precise characterization of what neural network-based diffusion models learn in non-asymptotic settings.

1. Introduction

Diffusion models [16] were proposed to tackle the problem of sampling from an unknown distribution and are later shown to be able to generate high quality images [8]. Song et al. [17] recognize diffusion model as an example of score-based models which iteratively exploit Langevin dynamics to produce data from an unknown distribution. This approach only requires the estimation of the score function of the data distribution. Specifically, the simplest form of Langevin Monte Carlo procedure involves first sampling x^0 from an initial distribution, then repeating the following steps

$$x^t \leftarrow x^{t-1} + \frac{\epsilon}{2}\nabla_x \log p(x^{t-1}) + \sqrt{\epsilon} z^t,$$

where z^t is an independently generated i.i.d. Gaussian noise and ϵ is a small constant. Here, $\nabla_x \log p(x)$ is known as the score function of the distribution $p(x)$ we desire to sample from. It can be shown that under certain conditions [2], we obtain iterates distributed according to the target distribution $p(x)$ as ϵ tends to zero and number of iterations tends to infinity.

In practice, a deep neural network model s_θ is trained to minimize variants of the score matching objective $\mathbb{E}[\|s_\theta(x) - \nabla_x \log p_{\text{data}}(x)\|_2^2]$ and is used for score function estimation. The score matching objective can be shown to be equivalent up to a constant to

$$\mathbb{E}_{p_{\text{data}}(x)}\left[\text{tr}\left(\nabla_x s_\theta(x)\right) + \frac{1}{2}\|s_\theta(x)\|_2^2\right], \tag{1}$$

which is more practical since $\nabla_x \log p_{\text{data}}(x)$ is usually not directly available. Existing literature on the theory of diffusion models typically establishes convergence of diffusion process when the learned score function approximates the score of unknown data distribution well [2, 10], thus falls short in understanding the role of NN approximation error. However, in [13, 20, 21], the authors show NN-based score-based generative models given finite training data usually generalize well due to approximation errors introduced by limited model capacity and also optimization errors, recognizing the critical role NN approximation error plays in effectiveness of current large diffusion models.

Our contribution. Our work focuses on analyzing what neural network-based score model learns in finite regime. Specifically, we show that the score matching objective fitted with two-layer neural network can be reparametrized as a quadratic convex program and solved directly to global optimality and the predicted score function will be piecewise linear with kinks only at training data points. We also investigate cases where the convex program can be solved analytically and establish explicit Langevin sampling convergence result in this regime.

A notation section is involved in Appendix A for clarity.

2. Score Matching

In this section, we derive convex program for score matching fitting problem with two-layer neural network and establish convergence results for neural network-based Langevin sampling procedure. For sake of clarity, we present results for NN without skip connection here in the main content and leave results with more general architecture in Appendix D.3.

2.1. Score Matching Problem and Neural Network Architectures

Let s_θ denote a neural network parameterized by parameter θ, with n data samples, the training loss we consider is

$$\min_{\theta} \sum_{i=1}^{n} \text{tr}\left(\nabla_{x_i} s_\theta(x_i)\right) + \frac{1}{2}\|s_\theta(x_i)\|_2^2 + \frac{\beta}{2}\|\theta'\|_2^2, \tag{2}$$

where $\theta' \subseteq \theta$ denotes the parameters to be regularized. We note that a non-zero weight decay is indeed core for the optimal value to stay finite, see Appendix C for explanation, which rationalizes the additional weight decay term involved here. Let m denote number of hidden neurons. Consider two-layer neural network of general form as below

$$s_\theta(x) = W^{(2)}\sigma\left(W^{(1)}x + b^{(1)}\right) + Vx + b^{(2)}, \tag{3}$$

with activation function σ, parameter $\theta = \{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, V\}$ and $\theta' = \{W^{(1)}, W^{(2)}\}$ where $x \in \mathbb{R}^d$ is the input data, $W^{(1)} \in \mathbb{R}^{m\times d}$ is the first-layer weight, $b^{(1)} \in \mathbb{R}^m$ is the first-layer bias, $W^{(2)} \in \mathbb{R}^{d\times m}$ is the second-layer weight, $b^{(2)} \in \mathbb{R}^d$ is the second-layer bias and $V \in \mathbb{R}^{d\times d}$ is the skip connection coefficient.

2.2. Convex Programs

We describe here the derived convex program for univariate data only since our score prediction characterization and convergence result in below (Section 2.3) focus on univariate data. We defer the derived convex program for multivariate date to Appendix D.4 due to page limit.

Theorem 1 *When σ is ReLU or absolute value activation and $V = 0$, denote the optimal score matching objective value (2) with s_θ specified in (3) as p^*, when $m \geq$ len(y) and $\beta \geq 1$*[1]*,*

$$p^* = \min_{y} \quad \frac{1}{2}\|Ay\|_2^2 + b^T y + \beta\|y\|_1\,, \tag{4}$$

where the entries of A are determined by the pairwise distances between data points, and the entries of b correspond to the derivative of σ evaluated at entries of A (see Appendix D.2 for the formulas).

1. Note when $\beta < 1$, the optimal value to problem (2) may be unbounded, see Appendix C for explanation.

Proof See Appendix D.2. ■

Once an optimal solution $y^\star$ to the convex program (4) has been derived, we can reconstruct an optimal NN parameter set $\{W^{(1)\star}, W^{(2)\star}, b^{(1)\star}, b^{(2)\star}\}$ that achieves minimal training loss simply from data points $\{x_1, \cdots, x_n\}$ and $y^\star$. See Appendix D.2 for the reconstruction procedure. Given all this, with $y^\star$ known, for any test data $\hat{x}$, the predicted score is given by

$$\hat{y}(\hat{x}) = \sum_{i=1}^{n} (|y_i^\star| + |y_{i+n}^\star|)|\hat{x} - x_i| + b_0^\star,$$

where $b_0^\star$ is some constant computed from $y^\star$. Remarkably, the optimal score is a piecewise linear function with breakpoints only at a subset of data points. When training data points are highly separated, the optimal score approximately corresponds to the score function of a mixture of Gaussians with centroids at $\{\hat{x} : \hat{y}(\hat{x}) = 0\}$. The breakpoints delineate the ranges of each Gaussian component.

2.3. Score Prediction and Convergence Result

We now delve into the convex program (4) and show that with distinct data points and large weight decay, (4) can be solved analytically and the integration of predicted score function is always concave for ReLU activation, which aligns with theoretic assumptions for Langevin sampling procedures. We then establish convergence result for Langevin dynamics in this regime.

Score Prediction. Consider the case when σ is ReLU and $V = 0$, let $\mu = \sum_{i=1}^n x_i/n$ denote the sample mean and $v = \sum_{i=1}^n (x_i - \mu)^2/n$ denote the sample variance. When $\beta > \|b\|_\infty, y^\star = 0$ is optimal and the neural network will always predict zero score no matter what input it takes. When $\beta_1 < \beta \leq \|b\|_\infty$ for some threshold β_1[2], for any input data point $\hat{x}$, the predicted score $\hat{y}$ is

$$\begin{cases} \hat{y} = \frac{\beta - n}{nv}(\hat{x} - \mu), & x_1 \leq \hat{x} \leq x_n \\ \hat{y} = -(\frac{n-\beta}{2nv} + t)\hat{x} + (\frac{\beta - n}{2nv} + t)x_1 + \frac{n-\beta}{nv}\mu, & \hat{x} < x_1 \\ \hat{y} = (\frac{\beta - n}{2nv} + t)\hat{x} - (\frac{n-\beta}{2nv} + t)x_n + \frac{n-\beta}{nv}\mu, & \hat{x} > x_n \end{cases} \tag{5}$$

for some $|t| \leq \frac{n-\beta}{2nv}$. Left plot in Figure 1 provides a visualization of (5) and its integration. Note

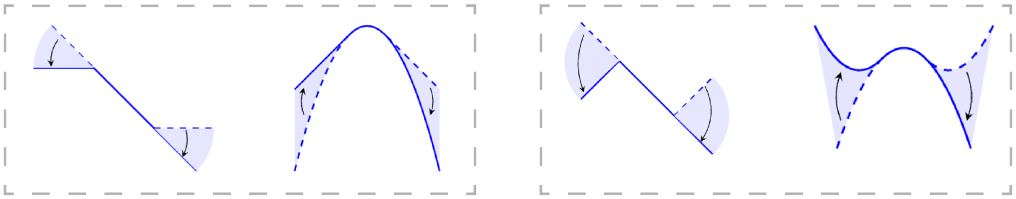

Figure 1: Predicted score function and its integration for univariate data with two-layer neural network with ReLU activation (left) and absolut value activation (right). See Section 2.3 for details.

within sampled data range, the predicted score function aligns with score function of Gaussian distribution parameterized by sample mean μ and sample variance v; outside sampled data range, the predicted score function is a linear interpolation. The integration of score function is always

2. See Appendix E.1.1 for value of β_1.

concave in this case, and therefore Langevin dynamics sampling with predicted score function has well-established convergence guarantees [3, 5, 6]. Contrarily, when σ is absolute value activation, the score prediction outside sampled data range is still a linear interpolation but with a different slope from what is predicted by the ReLU neural network, and the corresponding probability density is log-concave only when $t = 0$. Right plot in Figure 1 depicts the score prediction and its integration for absolute value σ.

Algorithm 1 Score Matching

Input: training data $x_1, \ldots, x_n \in \mathbb{R}^d$
minimize

$$\sum_{i=1}^{n} \frac{1}{2} s_\theta^2(x_i) + \nabla_\theta s_\theta(x_i) + \frac{\beta}{2} \|\theta'\|_2^2$$

Algorithm 2 Langevin Monte Carlo

Initialize: $x^0 \sim \mu_0(x)$
for $t = 1, 2, ..., T$ **do**
  $z^t \sim \mathcal{N}(0, 1)$
  $x^t \leftarrow x^{t-1} + \eta s_\theta(x^{t-1}) + \sqrt{2\eta} z^t$
end for

Convergence Result. Here we state our convergence result for Langevin sampling with NN-based score predictor. Strong convergence guarantees for Langevin Monte Carlo method are often contingent upon the log-concavity of the target distribution. Consider two-layer ReLU network without skip connection. We have derived that the NN-predicted score function for any input distribution is always concave given $\beta_1 < \beta \leq \|b\|_\infty$, thus we can exploit existing convergence results for log-concave sampling to derive the convergence of Langevin dynamics with a neural network-based score function, which we state formally as the below theorem.

Theorem 2 *When s_θ used in Algoritm 2 is of two-layer ReLU (without skip connection) trained to optimal with Algorithm 1 and $\beta_1 < \beta < n$, let π denote the target distribution (see Appendix E.2 for formula). In Algorithm 2, for any $\epsilon \in [0, 1]$, if we take step size $\eta \asymp \frac{\epsilon^2 nv}{n-\beta}$, then for $\overline{\mu} = T^{-1} \sum_{t=1}^{T} x^t$, it holds that $\sqrt{KL(\overline{\mu} \| \pi)} \leq \epsilon$ after $O\left(\frac{(n-\beta) W_2^2(\mu_0, \pi)}{nv\epsilon^4}\right)$ iterations, where W_2 denotes 2-Wasserstein distance.*

Proof See Appendix E.2. ■

To the best of our knowledge, prior to our study, there has been no characterization of the sample distribution generated by Algorithm 2 when the score model is trained using Algorithm 1.

3. Denoising Score Matching

We now reframe denoising score matching fitting problem with two-layer neural network as a convex program which can be solved to global optimality stably. We empirically verify the validity of our theoretic findings in Section 4. (We review background for denoising score matching in Appendix F).

We consider the same neural network architecture described in Section 2.1 except that here we only consider case for $V = 0$. For sake of page limit, we present the results for multivariate data in Theorem 3 below, and leave a more specific case analysis for univariate date in Appendix F. Let $L \in \mathbb{R}^{n \times d}$ denote the label matrix, i.e., $L_i = \delta_i / \epsilon$, and $\mathcal{D} = \{D_i\}_{i=1}^{P}$ be the arrangement activation patterns for ReLU activation as defined in Section D.4, we have the following result for ReLU σ. See Appendix G.2 for also convex program for absolute value activation.

Theorem 3 *When σ is ReLU, $b^{(1)}, b^{(2)}, V$ and β all zero, denote the optimal denoising score matching objective value with s_θ specified in (3) as p^*, when $m \geq 2Pd$, under Assumption 10,*

$$p^* = \min_{W_i} \frac{1}{2} \left\| \sum_{i=1}^{P} D_i X W_i - L \right\|_F^2 . \tag{6}$$

Proof See Appendix G.2. ■

The derived convex program (6) is a simple least square fitting and any convex program solver can be used to solve it efficiently.

4. Numerical Results

4.1. Score Matching Simulations

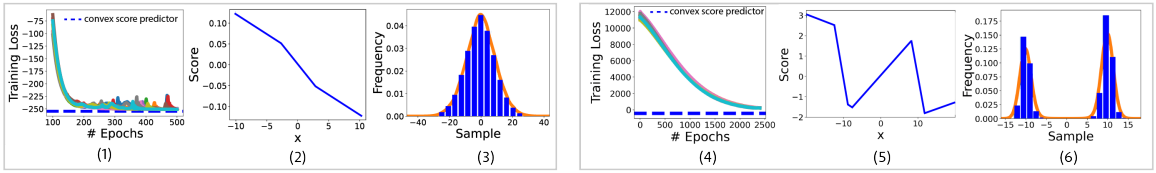

Figure 2: Simulation results for score matching tasks with two-layer ReLU neural network.

Plot (1) in Figure 2 compares objective values of non-convex training with Adam optimizer and our convex program loss solved via CVXPY [4]. The dashed blue line denotes our convex program objective value which solves the training problem globally and stably. Plot (2) is for score prediction, which verifies our analytical characterization in Section 2.3 and aligns with Figure 1 (we set $\beta = \|b\|_\infty - 1$). Plot (3) shows sampling histogram via (non-annealed) Langevin dynamics which recognizes the underline Gaussian as desired. The right figure in Figure 2 repeats the same experiments for two-component Gaussian mixture distribution with a slightly small β value since we know from Section 2.3 that $\beta = \|b\|_\infty - 1$ cannot capture Gaussian mixture distribution. Our convex program identifies the underline distribution accurately. See Appendix H.1 for more experimental details.

4.2. Denoising Score Matching Simulations

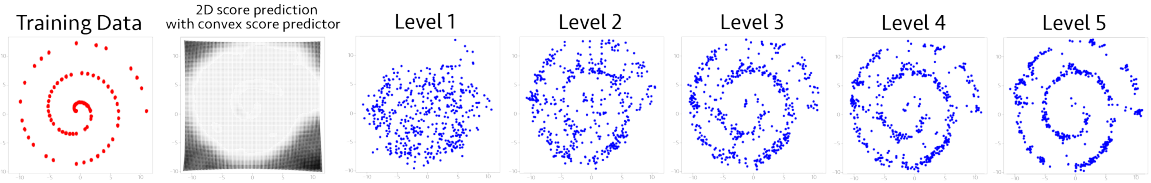

Figure 3: 2D simulation results for denoising score matching tasks with our convex score predictor. The right plots show denoising procedure with different noise levels in annealed Langevin sampling.

For denoising score matching fitting problem, we verify our derived program (6) for $2d$ spiral data and present sampling results with annealed Langevin process integrated with our convex score predictor. As shown in Figure 3, our convex program for denoising score matching works well in capturing training data distribution. See Appendix H.2 for more experimental details.

We defer more empirical observations to Appendix I.

References

[1] MOSEK ApS. *The MOSEK optimization toolbox*, 2019. URL YYY.

[2] Sinho Chewi. Log-concave sampling, 2023. URL https://chewisinho.github.io/main.pdf.

[3] Arnak S. Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities, 2016.

[4] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[5] Alain Durmus and Éric Moulines. Sampling from a strongly log-concave distribution with the unadjusted langevin algorithm. *arXiv: Statistics Theory*, 2016. URL https://api.semanticscholar.org/CorpusID:124591590.

[6] Alain Durmus and Eric Moulines. Non-asymptotic convergence analysis for the unadjusted langevin algorithm, 2016.

[7] Yinbin Han, Meisam Razaviyayn, and Renyuan Xu. Neural network-based score estimation in diffusion models: Optimization and generalization, 2024.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[9] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

[10] Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models, 2023.

[11] Lina Lin, Mathias Drton, and Ali Shojaie. Estimation of high-dimensional graphical models using regularized score matching, 2016.

[12] Aaron Mishkin, Arda Sahiner, and Mert Pilanci. Fast convex optimization for two-layer relu networks: Equivalent model classes and cone decompositions, 2022.

[13] Jakiw Pidstrigach. Score-based generative models detect manifolds, 2022.

[14] Mert Pilanci. From complexity to clarity: Analytical expressions of deep neural network weights via clifford's geometric algebra and convexity, 2024.

[15] Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks, 2020.

[16] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[17] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[18] Richard P Stanley et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13(389-496):24, 2004.

[19] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[20] Mingyang Yi, Jiacheng Sun, and Zhenguo Li. On the generalization of diffusion model, 2023.

[21] TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K. Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=shciCbSk9h.

[22] Chen Zeno, Greg Ongie, Yaniv Blumenfeld, Nir Weinberger, and Daniel Soudry. How do minimum-norm shallow denoisers look in function space?, 2023.

Appendix A. Notations

We first introduce some notations we will use in later sections. We use $\mathrm{sign}(x)$ to denote the sign function taking value 1 when $x \in [0, \infty)$ and -1 otherwise, and $\mathbb{1}$ to denote the 0-1 valued indicator function taking value 1 when the argument is a true Boolean statement. For any vector x, $\mathrm{sign}(x)$ and $\mathbb{1}\{x \geq 0\}$ applies elementwise. We denote the pseudoinverse of matrix A as $A^\dagger$. We denote subgradient of a convex function $f : \mathbb{R}^d \to \mathbb{R}$ at $x \in \mathbb{R}^d$ as $\partial f(x) \subseteq \mathbb{R}^d$. For any vector x, $\mathrm{len}(x)$ denote the dimension of x. Standard asymptotic notation is used, i.e., for any sequence $s_n \in \mathbb{R}$ and any given $c_n \in \mathbb{R}^+$, we use $s_n = \mathcal{O}(c_n)$ and $s_n = \Omega(c_n)$ to represent $s_n < \kappa c_n$ and $s_n > \kappa c_n$ respectively for some $\kappa > 0$. We use $s_n = \Theta(c_n)$ (also $a \asymp b$) when both $s_n = \mathcal{O}(c_n)$ and $s_n = \Omega(c_n)$.

Appendix B. More on Prior Work

Here we make a note on difference between our work and work [22], which tackles very similar problems as ours though there are several key differences. In [22], the authors study shallow neural network trained for score denoisers and characterize the exact neural network output. The authors show contractive property for NN-based denoiser and prove NN-based denoiser is advantageous against eMMSE denoiser. In our work, we study the exact score matching objective (1) which has not been considered in the other work and establish convergence result for NN-based score predictor which will be much harder to prove for NN-based denoiser due to involvement of noise. Moreover, for denoising score matching objective, we derive convex programs for arbitrary weight decay for multivariate data while the characterization in [22] is for vanishing weight decay. For multivariate data, the authors of [22] only consider modified objective with data belongs to special subspaces while our convex program holds in general. Finally, our analysis is based on convex optimization theory and no convexification is considered in [22]. Our work can be viewed as complementary to [22] in the sense that we study similar problems with different objectives and constraints from different angles.

Another work [7] establishes approximation error bound between true score function $\nabla \log p(X)$ and the GD minimizer to score-matching objective, which also serve as an approximation error bound between our convex program and true score in some cases while our method bypasses the potential local optimum problem caused by GD and derives the exact score function being predicted. The final result in [7] is presented as error bound on predicted score and true score asymptotically in number of hidden neurons m and number of data samples N with expectation over noises added to data and random initialization. Our work doesn't have such a bound while we can solve the training problem globally and thus escape local minimum with a convex program (which only holds with Assumption 3.8 in the other work) and derive the score function for finite data and neurons, i.e., we know exactly what's the predicted score function analytically (see Section 2.3 in our work) in certain regime while the other work only has an error bound on this.

Appendix C. Explanation for Unbounded Objective Value

Here we illustrate via a simple example that weight decay is necessary for the optimal objective value to stay finite. Follow notation in Section 2, consider for example only one data point and one hidden neuron, then objective function for the neural network with ReLU activation and no skip

connection would be

$$\frac{1}{2}((xw+b)_+\alpha+b_0)^2+w\alpha\mathbb{1}(xw+b\geq 0)+\frac{\beta}{2}(w^2+\alpha^2).$$

WLOG consider $x=1$, then when weight decay parameter $0\leq\beta<1$, set $b=-w+\sqrt{1-\beta}$ and $b_0=0$ above, we get

$$\frac{1-\beta}{2}\alpha^2+w\alpha+\frac{\beta}{2}(w^2+\alpha^2).$$

Then we can set $\alpha=-w$ and the above expression becomes $(\beta-1)w^2/2$. Thus the objective goes to minus infinity when w goes to infinity.

Appendix D. Proof in Section 2.2

D.1. Technical Lemmas

Lemma 4 *The below constraint set is strictly feasible only when $\beta>1$.*

$$\begin{cases}|z^T(x-1x_i)_+-1^T\mathbb{1}\{x-1x_i\geq 0\}|\leq\beta\\ |z^T(x-1x_i)_+-1^T\mathbb{1}\{x-1x_i>0\}|\leq\beta\\ |z^T(-x+1x_i)_++1^T\mathbb{1}\{-x+1x_i\geq 0\}|\leq\beta\\ |z^T(-x+1x_i)_++1^T\mathbb{1}\{-x+1x_i>0\}|\leq\beta\\ z^T1=0\end{cases}\qquad\forall i=1,\cdots,n$$

Proof *Consider without loss of generality that $x_1<x_2<\cdots<x_n$. Let $k=-\sum_{j=1}^m z_j(x_j-x_i)+m$ for some $1\leq m\leq n$, the first four constraints with $i=m$ are then $|z^Tx-(n+1)+k|,|z^Tx-(n+1)+k+1|,|k|,|k-1|$. When $i=n$, the first constraint is $\beta\geq 1$. Thus $\beta>1$ is necessary for the constraint set to be strictly feasible. Since we can always find $z^\star$ satisfying*

$$\begin{cases}x^Tz^\star=n\\ 1^Tz^\star=0\\ (x-1x_i)_+^Tz^\star-1^T\mathbb{1}\{x-1x_i\geq 0\}=0\end{cases}\quad\forall i=2,\cdots,n-1$$

Note such $z^\star$ satisfies all constraints in the original constraint set when $\beta>1$. Therefore when $\beta>1$, the original constraint is strictly feasible. ■

Lemma 5 *The below constraint set is strictly feasible only when $\beta>1$.*

$$\begin{cases}|z^T|x-1x_i|-1^T\text{sign}(x-1x_i)|\leq\beta\\ |z^T|-x+1x_i|+1^T\text{sign}(-x+1x_i)|\leq\beta\\ z^T1=0\end{cases}\quad\forall i=1,\cdots,n$$

Proof Consider without loss of generality that $x_1<x_2<\cdots<x_n$. Then taking $i=1$ and n in the first constraint gives $|z^Tx-n|\leq\beta$ and $|z^Tx-n+2|\leq\beta$. It's necessary to have $\beta>1$ and

$z^T x = n-1$ to have both constraints strictly satisfiable. Since we can always find $z^\star$ satisfying the below linear system

$$\begin{cases} x^T z^\star = n-1 \\ 1^T z^\star = 0 \\ |x - 1x_i|^T z^\star - 1^T \text{sign}(x - 1x_i) = -1 \quad \forall i = 2, \cdots, n-1 \end{cases}$$

Note such $z^\star$ also satisfies

$$|| - x + 1x_i|^T z^\star + 1^T \text{sign}(-x + 1x_i)| \leq 1$$

Therefore when $\beta > 1$, the original constraint set is strictly feasible. ■

Lemma 6 *The below constraint set is strictly feasible only when $\beta > 2$.*

$$\begin{cases} |z^T|x - 1x_i| - 1^T \text{sign}(x - 1x_i)| \leq \beta \\ |z^T| - x + 1x_i| + 1^T \text{sign}(-x + 1x_i)| \leq \beta \\ z^T 1 = 0 \\ z^T x = n \end{cases} \quad \forall i = 1, \cdots, n$$

Proof *Consider without loss of generality that $x_1 < x_2 < \cdots < x_n$. Then taking $i = n$ in the first constraint gives $| - n + (n-2)| \leq \beta$, which indicates that $\beta > 2$ is necessary for the constraint set to be strictly feasible. Since we can always find $z^\star$ satisfying*

$$\begin{cases} x^T z^\star = n \\ 1^T z^\star = 0 \\ |x - 1x_i|^T z^\star - 1^T \text{sign}(x - 1x_i) = 0 \quad \forall i = 2, \cdots, n-1 \end{cases}$$

Note such $z^\star$ also satisfies

$$|| - x + 1x_i|^T z^\star + 1^T \text{sign}(-x + 1x_i)| \leq 2$$

Therefore when $\beta > 2$, the original constraint set is strictly feasible. ■

D.2. Proof of Theorem 1

D.2.1. PROOF OF THEOREM 1 FOR RELU ACTIVATION

Proof Consider data $x \in \mathbb{R}^n$. Let m denote number of hidden neurons, then we have first layer weight $w \in \mathbb{R}^m$, first layer bias $b \in \mathbb{R}^m$, second layer weight $\alpha \in \mathbb{R}^m$ and second layer bias $b_0 \in \mathbb{R}$. The score matching objective is reduced to

$$p^\star = \min_{w,\alpha,b} \frac{1}{2} \left\| \sum_{j=1}^m (xw_j + 1b_j)_+ \alpha_j + 1b_0 \right\|_2^2 + 1^T \left(\sum_{j=1}^m w_j \alpha_j \mathbb{1}\{xw_j + 1b_j \geq 0\} \right) + \frac{1}{2}\beta \sum_{j=1}^m (w_j^2 + \alpha_j^2).$$

According to Lemma 2 in [15], after rescaling, the above problem is equivalent to

$$\min_{\substack{w,\alpha,b\\|w_j|=1}} \frac{1}{2}\left\|\sum_{j=1}^{m}(xw_j+1b_j)_+\alpha_j+1b_0\right\|_2^2+1^T\left(\sum_{j=1}^{m}w_j\alpha_j\mathbb{1}\{xw_j+1b_j\geq 0\}\right)+\beta\sum_{j=1}^{m}|\alpha_j|,$$

which can be written as

$$\begin{aligned}\min_{\substack{w,\alpha,b,r_1,r_2\\|w_j|=1}} &\frac{1}{2}\|r_1\|_2^2+1^Tr_2+\beta\sum_{j=1}^{m}|\alpha_j|\\ \text{s.t. } r_1&=\sum_{j=1}^{m}(xw_j+1b_j)_+\alpha_j+1b_0\\ r_2&=\sum_{j=1}^{m}w_j\alpha_j\mathbb{1}\{xw_j+1b_j\geq 0\}.\end{aligned}$$

The dual problem writes

$$\begin{aligned}d^\star=\max_{z_1,z_2}\min_{\substack{w,\alpha,b,r_1,r_2\\|w_j|=1}} &\frac{1}{2}\|r_1\|_2^2+1^Tr_2+\beta\sum_{j=1}^{m}|\alpha_j|+z_1^T\left(r_1-\sum_{j=1}^{m}(xw_j+1b_j)_+\alpha_j-1b_0\right)\\ &+z_2^T\left(r_2-\sum_{j=1}^{m}w_j\alpha_j\mathbb{1}\{xw_j+1b_j\geq 0\}\right),\end{aligned}$$

which gives a lower bound of $p^\star$. Minimizing over r_1, r_2, α_j above gives

$$\begin{aligned}\max_z\min_{b_0}&-\frac{1}{2}\|z\|_2^2-b_0z^T1\\ &\text{s.t. } |z^T(xw_j+1b_j)_+-w_j1^T\mathbb{1}\{xw_j+1b_j\geq 0\}|\leq\beta,\quad\forall|W_j|=1,\forall b_j.\end{aligned}$$

For the constraints to hold, we must have $z^T1=0$ and b_j takes values over x_j's. The above is equivalent to

$$\begin{aligned}\max_z\quad&-\frac{1}{2}\|z\|_2^2\\ \text{s.t.}\ &\begin{cases}|z^T(x-1x_i)_+-1^T\mathbb{1}\{x-1x_i\geq 0\}|\leq\beta\\ |z^T(x-1x_i)_+-1^T\mathbb{1}\{x-1x_i>0\}|\leq\beta\\ |z^T(-x+1x_i)_++1^T\mathbb{1}\{-x+1x_i\geq 0\}|\leq\beta\\ |z^T(-x+1x_i)_++1^T\mathbb{1}\{-x+1x_i>0\}|\leq\beta\\ z^T1=0\end{cases}\qquad\forall i=1,\dots,n,\end{aligned}\tag{7}$$

According to Lemma 4, when $\beta \geq 1$, the constraints in (7) are feasible for affine constraints, thus Slater's condition holds and the dual problem writes

$$
\begin{aligned}
d^\star = \min_{\substack{z_0,\ldots,z_7,z_8 \\ \text{s.t.} z_0,\ldots,z_7 \geq 0}} \max_z & -\frac{1}{2}\|z\|_2^2 + \sum_{i=1}^n z_{0i}\left(z^T(x-1x_i)_+ - 1^T\mathbb{1}\{x-1x_i \geq 0\} + \beta\right) \\
& + \sum_{i=1}^n z_{1i}\left(-z^T(x-1x_i)_+ + 1^T\mathbb{1}\{x-1x_i \geq 0\} + \beta\right) + \sum_{i=1}^n z_{2i}\left(z^T(x-1x_i)_+ - 1^T\mathbb{1}\{x-1x_i > 0\} + \beta\right) \\
& + \sum_{i=1}^n z_{3i}\left(-z^T(x-1x_i)_+ + 1^T\mathbb{1}\{x-1x_i > 0\} + \beta\right) + \sum_{i=1}^n z_{4i}\left(z^T(-x+1x_i)_+ + 1^T\mathbb{1}\{-x+1x_i \geq 0\} + \beta\right) \\
& + \sum_{i=1}^n z_{5i}\left(-z^T(-x+1x_i)_+ - 1^T\mathbb{1}\{-x+1x_i \geq 0\} + \beta\right) + \sum_{i=1}^n z_{6i}\left(z^T(-x+1x_i)_+ + 1^T\mathbb{1}\{-x+1x_i > 0\} + \beta\right) \\
& + \sum_{i=1}^n z_{7i}\left(-z^T(-x+1x_i)_+ - 1^T\mathbb{1}\{-x+1x_i > 0\} + \beta\right) + z_8 z^T 1,
\end{aligned}
$$

which is equivalent to

$$
\min_{\substack{z_0,\ldots,z_7,z_8 \\ \text{s.t.} z_0,\ldots,z_7 \geq 0}} \max_z -\frac{1}{2}\|z\|_2^2 + e^T z + f,
$$

where

$$
\begin{aligned}
e = \sum_{i=1}^n z_{0i}(x-1x_i)_+ - \sum_{i=1}^n z_{1i}(x-1x_i)_+ + \sum_{i=1}^n z_{2i}(x-1x_i)_+ - \sum_{i=1}^n z_{3i}(x-1x_i)_+ + \sum_{i=1}^n z_{4i}(-x+1x_i)_+ \\
- \sum_{i=1}^n z_{5i}(-x+1x_i)_+ + \sum_{i=1}^n z_{6i}(-x+1x_i)_+ - \sum_{i=1}^n z_{7i}(-x+1x_i)_+ + 1z_8,
\end{aligned}
$$

and

$$
\begin{aligned}
f = & -\sum_{i=1}^n z_{0i}1^T\mathbb{1}\{x-1x_i \geq 0\} + \sum_{i=1}^n z_{1i}1^T\mathbb{1}\{x-1x_i \geq 0\} - \sum_{i=1}^n z_{2i}1^T\mathbb{1}\{x-1x_i > 0\} \\
& + \sum_{i=1}^n z_{3i}1^T\mathbb{1}\{x-1x_i > 0\} + \sum_{i=1}^n z_{4i}1^T\mathbb{1}\{-x+1x_i \geq 0\} - \sum_{i=1}^n z_{5i}1^T\mathbb{1}\{-x+1x_i \geq 0\} \\
& + \sum_{i=1}^n z_{6i}1^T\mathbb{1}\{-x+1x_i > 0\} - \sum_{i=1}^n z_{7i}1^T\mathbb{1}\{-x+1x_i > 0\} + \beta(\sum_{i=0}^7 \|z_i\|_1).
\end{aligned}
$$

Maximizing over z gives

$$
\min_{\substack{z_0,\ldots,z_7,z_8 \\ \text{s.t.} z_0,\ldots,z_7 \geq 0}} \frac{1}{2}\|e\|_2^2 + f,
$$

Simplifying to get

$$
\begin{aligned}
\min_{y_0,y_1,y_2,y_3,y_4} \frac{1}{2}\|A_1(y_0+y_1) + A_2(y_2+y_3) + 1y_4\|_2^2 + 1^T C_1 y_0 - 1^T C_3 y_2 \\
+ 1^T C_2 y_1 - 1^T C_4 y_3 + \beta(\|y_0\|_1 + \|y_1\|_1 + \|y_2\|_1 + \|y_3\|_1).
\end{aligned}
$$

Minimizing over y_4 gives the convex program (4) in Theorem 1 with $A = [\bar{A}_1, \bar{A}_1, \bar{A}_2, \bar{A}_2] \in \mathbb{R}^{n\times 4n}, b = [1^T C_1, 1^T C_2, -1^T C_3, -1^T C_4]^T \in \mathbb{R}^{4n}$ where $\bar{A}_1 = \left(I - \frac{1}{n}11^T\right) A_1, \bar{A}_2 = \left(I - \frac{1}{n}11^T\right) A_2$ with $[A_1]_{ij} = (x_i - x_j)_+$ and $[A_2]_{ij} = (-x_i + x_j)_+, [C_1]_{ij} = \mathbb{1}\{x_i - x_j \geq 0\}, [C_2]_{ij} = \mathbb{1}\{x_i - x_j > 0\}, [C_3]_{ij} = \mathbb{1}\{-x_i + x_j \geq 0\}, [C_4]_{ij} = \mathbb{1}\{-x_i + x_j > 0\}$. Once we obtain optimal solution $y^\star$ to problem (4), we can take

$$\begin{cases} w_j^\star = \sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = -\sqrt{|y_j^\star|} x_j \text{ for } j = 1, \ldots, n, \\ w_j^\star = \sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = -\sqrt{|y_j^\star|}(x_{j-n} + \epsilon) \text{ for } j = n+1, \ldots, 2n, \\ w_j^\star = -\sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = \sqrt{|y_j^\star|} x_{j-2n} \text{ for } j = 2n+1, \ldots, 3n, \\ w_j^\star = -\sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = \sqrt{|y_j^\star|}(x_{j-3n} - \epsilon) \text{ for } j = 3n+1, \ldots, 4n, \\ b_0^\star = -\frac{1}{n} 1^T ([A_1, A_1, A_2, A_2] y^\star), \end{cases} \tag{8}$$

then score matching objective has the same value as optimal value of convex program (4) as $\epsilon \to 0$, which indicates $p^\star = d^\star$ and the above parameter set is optimal. ■

D.2.2. PROOF OF THEOREM 1 FOR ABSOLUTE VALUE ACTIVATION

Proof Consider data $x \in \mathbb{R}^n$. Let m denote number of hidden neurons, then we have first layer weight $w \in \mathbb{R}^m$, first layer bias $b \in \mathbb{R}^m$, second layer weight $\alpha \in \mathbb{R}^m$ and second layer bias $b_0 \in \mathbb{R}$. Then the score matching objective is reduced to

$$p^\star = \min_{w,\alpha,b} \frac{1}{2} \left\| \sum_{j=1}^m |xw_j + 1b_j| \alpha_j + 1b_0 \right\|_2^2 + 1^T \left(\sum_{j=1}^m w_j \alpha_j \text{sign}(xw_j + 1b_j) \right) + \frac{1}{2} \beta \sum_{j=1}^m (w_j^2 + \alpha_j^2).$$

According to Lemma 2 in [15], after rescaling, the above problem is equivalent to

$$\min_{\substack{w,\alpha,b \\ |w_j|=1}} \frac{1}{2} \left\| \sum_{j=1}^m |xw_j + 1b_j| \alpha_j + 1b_0 \right\|_2^2 + 1^T \left(\sum_{j=1}^m w_j \alpha_j \text{sign}(xw_j + 1b_j) \right) + \beta \sum_{j=1}^m |\alpha_j|,$$

which can be written as

$$\begin{aligned} \min_{\substack{w,\alpha,b,r_1,r_2 \\ |w_j|=1}} \quad & \frac{1}{2}\|r_1\|_2^2 + 1^T r_2 + \beta \sum_{j=1}^m |\alpha_j| \\ \text{s.t.} \quad & r_1 = \sum_{j=1}^m |xw_j + 1b_j| \alpha_j + 1b_0 \\ & r_2 = \sum_{j=1}^m w_j \alpha_j \text{sign}(xw_j + 1b_j). \end{aligned} \tag{9}$$

The dual problem of (9) writes

$$d^\star = \max_{z_1,z_2} \min_{\substack{w,\alpha,b,r_1,r_2 \\ |w_j|=1}} \frac{1}{2}\|r_1\|_2^2 + 1^T r_2 + \beta \sum_{j=1}^m |\alpha_j| + z_1^T \left(r_1 - \sum_{j=1}^m |xw_j + 1b_j|\alpha_j - 1b_0 \right) + z_2^T \left(r_2 - \sum_{j=1}^m w_j \alpha_j \text{sign}(xw_j + 1b_j) \right),$$

which is a lower bound of optimal value to the original problem, i.e., $p^\star \geq d^\star$. Minimizing over r_1 and r_2 gives

$$\max_z \min_{\substack{w,\alpha,b \\ |w_j|=1}} -\frac{1}{2}\|z\|_2^2 + \beta \sum_{j=1}^m |\alpha_j| - z^T \left(\sum_{j=1}^m |xw_j + 1b_j|\alpha_j + 1b_0 \right) + 1^T \sum_{j=1}^m w_j \alpha_j \text{sign}(xw_j + 1b_j).$$

Minimizing over α_j gives

$$\begin{aligned} &\max_z \min_{b_0} -\frac{1}{2}\|z\|_2^2 - b_0 z^T 1 \\ &\quad \text{s.t. } |z^T|xw_j + 1b_j| - w_j 1^T \text{sign}(xw_j + 1b_j)| \leq \beta, \quad \forall |w_j| = 1, \forall b_j, \end{aligned}$$

which is equivalent to

$$\begin{aligned} &\max_z \min_{b_0} -\frac{1}{2}\|z\|_2^2 - b_0 z^T 1 \\ &\quad \text{s.t.} \begin{cases} |z^T|x + 1b_j| - 1^T \text{sign}(x + 1b_j)| \leq \beta \\ |z^T| - x + 1b_j| + 1^T \text{sign}(-x + 1b_j)| \leq \beta \end{cases} \quad \forall b_j. \end{aligned}$$

For the constraints to hold, we must have $z^T 1 = 0$ and b_j takes values over x_j's. Furthermore, since sign is discontinuous at input 0, we add another function sign^* which takes value -1 at input 0 to cater for the constraints. The above is equivalent to

$$\begin{aligned} &\max_z \quad -\frac{1}{2}\|z\|_2^2 \\ &\text{s.t.} \begin{cases} |z^T|x - 1x_i| - 1^T \text{sign}(x - 1x_i)| \leq \beta \\ |z^T|x - 1x_i| - 1^T \text{sign}^*(x - 1x_i)| \leq \beta \\ |z^T| - x + 1x_i| + 1^T \text{sign}(-x + 1x_i)| \leq \beta \\ |z^T| - x + 1x_i| + 1^T \text{sign}^*(-x + 1x_i)| \leq \beta \\ z^T 1 = 0 \end{cases} \quad \forall i = 1, \ldots, n. \end{aligned} \tag{10}$$

Since the second constraint overlaps with the third, and the fourth constraint overlaps with the first, (10) is equivalent to

$$\begin{aligned} &\max_z -\frac{1}{2}\|z\|_2^2 \\ &\text{s.t.} \begin{cases} |z^T|x - 1x_i| - 1^T \text{sign}(x - 1x_i)| \leq \beta \\ |z^T| - x + 1x_i| + 1^T \text{sign}(-x + 1x_i)| \leq \beta \\ z^T 1 = 0 \end{cases} \quad \forall i = 1, \ldots, n. \end{aligned} \tag{11}$$

According to Lemma 5, when $\beta \geq 1$, the constraints in (11) are feasible for affine constraints, thus Slater's condition holds and the dual problem writes

$$\begin{aligned} d^\star = \min_{\substack{z_0,z_1,z_2,z_3,z_4 \\ \text{s.t.} z_0,z_1,z_2,z_3 \geq 0}} \max_z & -\frac{1}{2}\|z\|_2^2 + \sum_{i=1}^n z_{0i}\left(z^T|x-1x_i| - 1^T\text{sign}(x-1x_i) + \beta\right) \\ & + \sum_{i=1}^n z_{1i}\left(-z^T|x-1x_i| + 1^T\text{sign}(x-1x_i) + \beta\right) \\ & + \sum_{i=1}^n z_{2i}\left(z^T|-x+1x_i| + 1^T\text{sign}(-x+1x_i) + \beta\right) \\ & + \sum_{i=1}^n z_{3i}\left(-z^T|-x+1x_i| - 1^T\text{sign}(-x+1x_i) + \beta\right) \\ & + z_4 z^T 1, \end{aligned}$$

which is equivalent to

$$\min_{\substack{z_0,z_1,z_2,z_3,z_4 \\ \text{s.t.} z_0,z_1,z_2,z_3 \geq 0}} \max_z -\frac{1}{2}\|z\|_2^2 + e^T z + f,$$

where

$$e = \sum_{i=1}^n z_{0i}|x-1x_i| - \sum_{i=1}^n z_{1i}|x-1x_i| + \sum_{i=1}^n z_{2i}|-x+1x_i| - \sum_{i=1}^n z_{3i}|-x+1x_i| + 1z_4$$

and

$$\begin{aligned} f = & -\sum_{i=1}^n z_{0i}1^T\text{sign}(x-1x_i) + \sum_{i=1}^n z_{1i}1^T\text{sign}(x-1x_i) + \sum_{i=1}^n z_{2i}1^T\text{sign}(-x+1x_i) \\ & - \sum_{i=1}^n z_{3i}1^T\text{sign}(-x+1x_i) + \beta(\|z_0\|_1 + \|z_1\|_1 + \|z_2\|_1 + \|z_3\|_1). \end{aligned}$$

Maximizing over z gives

$$\min_{\substack{z_0,z_1,z_2,z_3,z_4 \\ \text{s.t.} z_0,z_1,z_2,z_3 \geq 0}} \frac{1}{2}\|e\|_2^2 + f.$$

Simplifying to get

$$\min_{y_1,y_2,z} \frac{1}{2}\|A_1(y_1+y_2) + 1z\|_2^2 + 1^T C_1 y_1 - 1^T C_2 y_2 + \beta(\|y_1\|_1 + \|y_2\|_1).$$

Minimizing over z gives the convex program (4) in Theorem 1 where $A = [\bar{A}_1, \bar{A}_1] \in \mathbb{R}^{n\times 2n}, b = [1^T C_1, -1^T C_2]^T \in \mathbb{R}^{2n}$ with $\bar{A}_1 = \left(I - \frac{1}{n}11^T\right)A_1$, $[A_1]_{ij} = |x_i - x_j|$, $[C_1]_{ij} = \text{sign}(x_i - x_j)$ and $[C_2]_{ij} = \text{sign}(-x_i + x_j)$. Once we obtain optimal solution $y^\star$ to problem (4), we can take

$$\begin{cases} w_j^\star = \sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = -\sqrt{|y_j^\star|}x_j \text{ for } j = 1,\dots,n, \\ w_j^\star = -\sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = \sqrt{|y_j^\star|}x_{j-n} \text{ for } j = n+1,\dots,2n, \\ b_0^\star = -\frac{1}{n}1^T([A_1, A_1]y^\star), \end{cases}$$

then score matching objective has the same value as optimal value of convex program (4), which indicates $p^\star = d^\star$ and the above parameter set is optimal. ■

D.3. Convex Programs for More Model Architectures

D.3.1. RELU ACTIVATION WITH SKIP CONNECTION

Theorem 7 *When σ is ReLU and $V \neq 0$, denote the optimal score matching objective value (2) with s_θ specified in (3) as $p^\star$, when $m \geq len(y)$ and $\beta \geq 1$,*

$$p^* = \min_y \quad \frac{1}{2}\|Ay\|_2^2 + b^T y + c + 2\beta\|y\|_1 \,, \tag{12}$$

A, b, c and reconstruction rule for θ is specficied in the proof below.

Proof Here we reduce score matching objective including ReLU activation to score matching objective including absolute value activation and exploits results in Theorem 8. Let $\{w^r, b^r, \alpha^r, v^r\}$ denotes parameter set corresponding to ReLU activation, consider another parameter set $\{w^a, b^a, \alpha^a, v^a\}$ satisfying

$$\begin{cases} \alpha^r = 2\alpha^a, \\ w^r = w^a, \\ b^r = b^a, \\ b_0^r = b_0^a - \frac{1}{2}\sum_{j=1}^m b_j^r \alpha_j^r, \\ v^r = v^a - \frac{1}{2}\sum_{j=1}^m w_j^r \alpha_j^r. \end{cases}$$

Then the score matching objective

$$\min_{w^r,\alpha^r,b^r,v^r} \frac{1}{2}\left\|\sum_{j=1}^m (xw_j^r + 1b_j^r)_+ \alpha_j^r + xv^r + 1b_0^r\right\|_2^2 + 1^T\left(\sum_{j=1}^m w_j^r \alpha_j^r \mathbb{1}\{xw_j^r + 1b_j^r \geq 0\}\right) + nv^r + \frac{\beta}{2}\sum_{j=1}^m (w_j^{r2} + \alpha_j^{r2})$$

is equivalent to

$$\min_{w^a,\alpha^a,b^a,v^a} \frac{1}{2}\left\|\sum_{j=1}^m |xw_j^a + 1b_j^a|\alpha_j^a + xv^a + 1b_0^a\right\|_2^2 + 1^T\left(\sum_{j=1}^m w_j^a \alpha_j^a \text{sign}(xw_j^a + 1b_j^a)\right) + nv^a + \frac{\beta}{2}\sum_{j=1}^m \left(w_j^{a2} + 4\alpha_j^{a2}\right).$$

According to Lemma 2 in [15], after rescaling, the above problem is equivalent to

$$\min_{\substack{w^a,\alpha^a,b^a,v^a \\ |w_j^a|=1}} \frac{1}{2}\left\|\sum_{j=1}^m |xw_j^a + 1b_j^a|\alpha_j^a + xv^a + 1b_0^a\right\|_2^2 + 1^T\left(\sum_{j=1}^m w_j^a \alpha_j^a \text{sign}(xw_j^a + 1b_j^a)\right) + nv^a + 2\beta\sum_{j=1}^m |\alpha_j^a|. \tag{13}$$

Following similar analysis as in Appendix 8 with a different rescaling factor we can derive the convex program (12) with $A = B^{\frac{1}{2}}A_1, b = A_1^T(-n\bar{x}/\|\bar{x}\|_2^2) + b_1, c = -n^2/(2\|\bar{x}\|_2^2)$ where $B = I - P_{\bar{x}}$ with $P_{\bar{x}} = \bar{x}\bar{x}^T/\|\bar{x}\|_2^2$, and A_1, b_1 are identical to A, b defined in Section D.2.2 respectively.

Here, $\bar{x}_j := x_j - \sum_i x_i/n$ denotes mean-subtracted data vector. The optimal solution set to (13) is given by

$$\begin{cases} w_j^{a\star} = \sqrt{2|y_j^\star|}, \alpha_j^{a\star} = \sqrt{|y_j^\star|/2}, b_j^{a\star} = -\sqrt{2|y_j^\star|}x_j \text{ for } j = 1, \dots, n, \\ W_j^{a\star} = -\sqrt{2|y_j^\star|}, \alpha_j^{a\star} = \sqrt{|y_j^\star|/2}, b_j^{a\star} = \sqrt{2|y_j^\star|}x_{j-n} \text{ for } j = n+1, \dots, 2n, \\ v^{a\star} = -(\bar{x}^T A_1 y^\star + n)/\|\bar{x}\|_2^2, \\ b_0^{a\star} = -\frac{1}{n} 1^T([A_1', A_1']y^\star + xv^{a\star}), \end{cases}$$

where A_1' is A_1 defined in Appendix D.2.2 and $y^\star$ is optimal solution to convex program (12). Then the optimal parameter set $\{w^r, b^r, \alpha^r, z^r\}$ is given by

$$\begin{cases} w_j^{r\star} = \sqrt{2|y_j^\star|}, \alpha_j^{r\star} = \sqrt{2|y_j^\star|}, b_j^{r\star} = -\sqrt{2|y_j^\star|}x_j \text{ for } j = 1, \dots, n, \\ w_j^{r\star} = -\sqrt{2|y_j^\star|}, \alpha_j^{r\star} = \sqrt{2|y_j^\star|}, b_j^{r\star} = \sqrt{2|y_j^\star|}x_{j-n} \text{ for } j = n+1, \dots, 2n, \\ v^{r\star} = -(\bar{x}^T A_1 y^\star + n)/\|\bar{x}\|_2^2 - \sum_{j=1}^m w_j^{r\star}\alpha_j^{r\star}/2, \\ b_0^{r\star} = -\frac{1}{n} 1^T([A_1', A_1']y^\star + x(-(\bar{x}^T A_1 y^\star + n)/\|\bar{x}\|_2^2)) - \sum_{j=1}^m b_j^{r\star}\alpha_j^{r\star}/2. \end{cases}$$

■

D.3.2. ABSOLUTE VALUE ACTIVATION WITH SKIP CONNECTION

Theorem 8 *When σ is absolute value activation and $V \neq 0$, denote the optimal score matching objective value (2) with s_θ specified in (3) as $p^\star$, when $m \geq len(y)$ and $\beta \geq 2$,*

$$p^* = \min_y \quad \frac{1}{2}\|Ay\|_2^2 + b^T y + \beta\|y\|_1 \,, \tag{14}$$

A, b and reconstruction rule for θ is specficied in the proof below.

Proof Consider data matrix $x \in \mathbb{R}^n$, then the score matching objective is reduced to

$$p^\star = \min_{w,\alpha,b,v} \frac{1}{2}\left\| \sum_{j=1}^m |xw_j + 1b_j|\alpha_j + xv + 1b_0 \right\|_2^2 + 1^T \left(\sum_{j=1}^m w_j\alpha_j \text{sign}(xw_j + 1b_j) \right) + nv + \frac{1}{2}\beta \sum_{j=1}^m (w_j^2 + \alpha_j^2).$$

Following similar analysis as in Appendix D.2.2, we can derive the dual problem as

$$d^\star = \max_{z_1, z_2} \min_{\substack{w,\alpha,b,v,r_1,r_2 \\ |w_j|=1}} \frac{1}{2}\|r_1\|_2^2 + 1^T r_2 + nv + \beta \sum_{j=1}^m |\alpha_j| + z_1^T \left(r_1 - \sum_{j=1}^m |xw_j + 1b_j|\alpha_j - xv - 1b_0 \right) + z_2^T \left(r_2 - \sum_{j=1}^m w_j\alpha_j \text{sign}(xw_j + 1b_j) \right).$$

which gives a lower bound of $p^\star$. Minimizing over r_1 and r_2 gives

$$\max_{z_1} \min_{\substack{w,\alpha,b,v \\ |w_j|=1}} -\frac{1}{2}\|z_1\|_2^2 + nv + \beta \sum_{j=1}^m |\alpha_j| - z_1^T \left(\sum_{j=1}^m |xw_j + 1b_j|\alpha_j + xv + 1b_0 \right) + 1^T \sum_{j=1}^m w_j\alpha_j \text{sign}(xw_j + 1b_j).$$

Minimizing over v gives

$$\max_{\substack{z_1 \\ z_1^T x = n}} \min_{\substack{w,\alpha,b \\ |w_j|=1}} -\frac{1}{2}\|z_1\|_2^2 + \beta \sum_{j=1}^m |\alpha_j| - z_1^T \left(\sum_{j=1}^m |xw_j + 1b_j|\alpha_j + 1b_0 \right) + 1^T \sum_{j=1}^m w_j \alpha_j \text{sign}(xw_j + 1b_j).$$

Minimizing over α_j gives

$$\begin{aligned} &\max_z \min_{b_0} -\frac{1}{2}\|z\|_2^2 - b_0 z^T 1 \\ &\quad \text{s.t.} \begin{cases} z^T x = n \\ |z^T|xw_j + 1b_j| - w_j 1^T \text{sign}(xw_j + 1b_j)| \le \beta, \quad \forall |w_j| = 1, \forall b_j. \end{cases} \end{aligned}$$

Following same logic as in Appendix D.2.2, the above problem is equivalent to

$$\begin{aligned} &\max_z \quad -\frac{1}{2}\|z\|_2^2 \\ &\text{s.t.} \begin{cases} |z^T|x - 1x_i| - 1^T \text{sign}(x - 1x_i)| \le \beta \\ |z^T| - x + 1x_i| + 1^T \text{sign}(-x + 1x_i)| \le \beta \\ z^T 1 = 0 \\ z^T x = n \end{cases} \qquad \forall i = 1, \dots, n. \end{aligned} \tag{15}$$

According to Lemma 6, when $\beta \ge 2$, the constraints in (15) are feasible for affine constraints, thus Slater's condition holds and the dual problem writes

$$\begin{aligned} d^\star = \min_{\substack{z_0,z_1,z_2,z_3,z_4,z_5 \\ \text{s.t.} z_0,z_1,z_2,z_3 \ge 0}} \max_z &-\frac{1}{2}\|z\|_2^2 + \sum_{i=1}^n z_{0i} \left(z^T|x - 1x_i| - 1^T \text{sign}(x - 1x_i) + \beta \right) \\ &+ \sum_{i=1}^n z_{1i} \left(-z^T|x - 1x_i| + 1^T \text{sign}(x - 1x_i) + \beta \right) \\ &+ \sum_{i=1}^n z_{2i} \left(z^T| - x + 1x_i| + 1^T \text{sign}(-x + 1x_i) + \beta \right) \\ &+ \sum_{i=1}^n z_{3i} \left(-z^T| - x + 1x_i| - 1^T \text{sign}(-x + 1x_i) + \beta \right) \\ &+ z_4(z^T x - n) + z_5 z^T 1, \end{aligned}$$

which is equivalent to

$$\min_{\substack{z_0,z_1,z_2,z_3,z_4,z_5 \\ \text{s.t.} z_0,z_1,z_2,z_3 \ge 0}} \max_z -\frac{1}{2}\|z\|_2^2 + e^T z + f,$$

where

$$e = \sum_{i=1}^n z_{0i}|x - 1x_i| - \sum_{i=1}^n z_{1i}|x - 1x_i| + \sum_{i=1}^n z_{2i}| - x + 1x_i| - \sum_{i=1}^n z_{3i}| - x + 1x_i| + xz_4 + 1z_5,$$

and

$$f = -\sum_{i=1}^{n} z_{0i} 1^T \text{sign}(x - 1x_i) + \sum_{i=1}^{n} z_{1i} 1^T \text{sign}(x - 1x_i) + \sum_{i=1}^{n} z_{2i} 1^T \text{sign}(-x + 1x_i)$$
$$- \sum_{i=1}^{n} z_{3i} 1^T \text{sign}(-x + 1x_i) - z_4 n + \beta(\|z_0\|_1 + \|z_1\|_1 + \|z_2\|_1 + \|z_3\|_1).$$

Maximizing over z gives

$$\min_{\substack{z_0,z_1,z_2,z_3,z_4,z_5 \\ \text{s.t.} z_0,z_1,z_2,z_3 \geq 0}} \frac{1}{2}\|e\|_2^2 + f.$$

Simplifying to get

$$\min_{y_0,y_1,y_2,y_3} \frac{1}{2} \left\| A_1'(y_0 + y_1) + xy_2 + 1y_3 \right\|_2^2 + 1^T C_1 y_0 - 1^T C_2 y_1 + n y_2 + \beta(\|y_1\|_1 + \|y_2\|_1), \tag{16}$$

where A_1', C_1, C_2 are as A_1, C_1, C_2 defined in Appendix D.2.2. Minimizing over y_3 gives $y_3 = -1^T(A_1'(y_0 + y_1) + xy_2)/n$ and (16) is reduced to

$$\min_{y_0,y_1,y_2} \frac{1}{2} \left\| \bar{A}_1'(y_0 + y_1) + \bar{x} y_2 \right\|_2^2 + 1^T C_1 y_0 - 1^T C_2 y_1 + n y_2 + \beta(\|y_1\|_1 + \|y_2\|_1),$$

where $\bar{A}_1'$ is as $\bar{A}_1$ defined in Appendix D.2.2. Minimizing over y_2 gives $y_2 = -\left(\bar{x}^T \bar{A}_1'(y_0 + y_1) + n\right)/\|\bar{x}\|_2^2$ and the above problem is equivalent to the convex program (14) in Theorem 8 with $A = B^{\frac{1}{2}} A_1, b = A_1^T(-n\bar{x}/\|\bar{x}\|_2^2) + b_1, c = -n^2/(2\|\bar{x}\|_2^2)$ where $B = I - P_{\bar{x}}$ with $P_{\bar{x}} = \bar{x}\bar{x}^T/\|\bar{x}\|_2^2$, and A_1, b_1 are identical to A, b defined in Section D.2.2 respectively. Once we obtain optimal solution $y^\star$ to problem (14), we can take

$$\begin{cases} w_j^\star = \sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = -\sqrt{|y_j^\star|} x_j \text{ for } j = 1, \ldots, n, \\ w_j^\star = -\sqrt{|y_j^\star|}, \alpha_j^\star = \sqrt{|y_j^\star|}, b_j^\star = \sqrt{|y_j^\star|} x_{j-n} \text{ for } j = n+1, \ldots, 2n, \\ v^\star = -(\bar{x}^T A_1 y^\star + n)/\|\bar{x}\|_2^2, \\ b_0^\star = -\frac{1}{n} 1^T([A_1', A_1']y^\star + xv^\star), \end{cases}$$

then score matching objective has the same value as optimal value of convex program (14), which indicates $p^\star = d^\star$ and the above parameter set is optimal.

■

D.4. Convex Program for Multivariate Data

To state the convex program for multivariate data, we first introduce the concept of arrangement matrices. When d is arbitrary, for data matrix $X \in \mathbb{R}^{n \times d}$ and any arbitrary vector $u \in \mathbb{R}^d$, We consider the set of diagonal matrices

$$\mathcal{D} := \{\text{diag}(\mathbb{1}\{Xu \geq 0\})\},$$

which takes value 1 or 0 along the diagonal that indicates the set of possible arrangement activation patterns for the ReLU activation. Indeed, we can enumerate the set of sign patterns as $\mathcal{D} = \{D_i\}_{i=1}^P$ where P is bounded by

$$P \leq 2r\left(\frac{e(n-1)}{r}\right)^r$$

for $r = \text{rank}(X)$ [15, 18]. Since the proof of Theorem 1 is closely tied to reconstruction of optimal neurons and does not trivially extend to multivariate data, we instead build on [12] and employ an alternative duality-free proof to derive our conclusion for multivariate data. The result holds for zero $b^{(1)}, b^{(2)}, V$ and β, i.e., when there is no bias term, skip connection, and weight decay added. We present here result for ReLU σ. See Appendix D.4 for conclusion for the case when σ is absolute value activation.

Theorem 9 *When σ is ReLU, $b^{(1)}, b^{(2)}, V$ and β all zero, denote the optimal score matching objective value (2) with s_θ specified in (3) as p^*, when $m \geq 2Pd$, under Assumption 10,*

$$p^* = \min_{W_i} \frac{1}{2}\left\|\sum_{i=1}^P D_i X W_i\right\|_F^2 + \sum_{i=1}^P tr(D_i)tr(W_i). \tag{17}$$

Prior work [9, 11] observes that with linear activation, the optimal weight matrix of score fitting reduces to empirical precision matrix which models the correlation between data points and the authors exploit this fact in graphical model construction. Here we show that the optimal W_i's solved for (17) correspond to piecewise empirical covariance estimator and therefore the non-linear two-layer NN is a more expressive model compared to prior linear models. To see this, we first write $\tilde{V} = [\text{tr}(D_1)I, \text{tr}(D_2)I, \ldots, \text{tr}(D_P)I], W = [W_1, \ldots, W_P]^T, \tilde{X} = [D_1X, \ldots, D_PX]$, then the convex program (17) can be rewritten as

$$\min_W \frac{1}{2}\|\tilde{X}W\|_F^2 + \langle \tilde{V}, W\rangle. \tag{18}$$

When the optimal value is finite, e.g., $\tilde{V} \in \text{range}(\tilde{X}^T\tilde{X})$, an optimal solution to (18) is given by

$$\begin{aligned} W^\star &= -(\tilde{X}^T\tilde{X})^\dagger \tilde{V} \\ &= -\begin{bmatrix} \sum_{k\in S_{11}} X_k X_k^T & \sum_{k\in S_{12}} X_k X_k^T & \cdots \\ \sum_{k\in S_{21}} X_k X_k^T & \sum_{k\in S_{22}} X_k X_k^T & \cdots \\ & \cdots & \end{bmatrix}^\dagger \begin{bmatrix} \text{tr}(D_1)I \\ \text{tr}(D_2)I \\ \vdots \\ \text{tr}(D_P)I \end{bmatrix}, \end{aligned}$$

where $S_{ij} = \{k : X_k^T u_i \geq 0, X_k^T u_j \geq 0\}$ and u_i is the generator of $D_i = \text{diag}(\mathbb{1}\{Xu_i \geq 0\})$. The above expression for $W^\star$ can be seen as a (negative) piecewise empirical covariance estimator which partitions the space with hyperplane arrangements. When $P = 1$ and $D_1 = I$, $|W^\star| = (\tilde{X}^T\tilde{X})^\dagger$ reduces to the empirical precision matrix corresponding to linear activation model.

We now proceed to prove Theorem 9. Here we first depict the assumption required for Theorem 9 to hold in Assumption 10. Note if Assumption 10 is not true, original Theorem 9 still holds with equal sign replaced by greater than or equal to, which can be trivially seen from our proof of Theorem 9 below. Assumption 10 has already been characterized in Proposition 3.1 in [12], here

we restate it for sake of completeness. First we define for each activation pattern $D_i \in \mathcal{D}$, the set of vectors that induce D_i as

$$\mathcal{K}_i = \{u \in \mathbb{R}^d : (2D_i - I)Xu \succeq 0\}.$$

Assumption 10 *Let $\mathcal{D}_X$ denote the activation pattern set $\mathcal{D}$ induced by dataset X, assume for any $D_i \in \mathcal{D}_X$, $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$.*

According to Proposition 3.1 in [12], Assumption 10 is satisfied whenever data matrix X is full row-rank. Empirically, Assumption 10 holds with high probability according to experiments in [12].

D.4.1. FORMAL PROOF

Proof When $X \in \mathbb{R}^{n\times d}$ for some $d > 1$. Let m denote the number of hidden neurons, then the score matching objective can be reduced to

$$p^\star = \min_{u_j, v_j} \sum_{i=1}^{n} \left(\frac{1}{2} \left\| \sum_{j=1}^{m} (X_i u_j)_+ v_j^T \right\|_2^2 + \text{tr} \left(\nabla_{X_i} \left[\sum_{j=1}^{m} (X_i u_j)_+ v_j^T \right] \right) \right), \tag{19}$$

which can be rewritten as

$$\min_{u_j, v_j} \frac{1}{2} \left\| \sum_{j=1}^{m} (Xu_j)_+ v_j^T \right\|_F^2 + 1^T \left(\sum_{j=1}^{m} \mathbb{1}\{Xu_j \geq 0\} v_j^T u_j \right). \tag{20}$$

Let $D_j' = \text{diag}\left(\mathbb{1}\{Xu_j \geq 0\}\right)$, then problem (20) is equivalent to

$$\min_{u_j, v_j} \frac{1}{2} \left\| \sum_{j=1}^{m} D_j' X u_j v_j^T \right\|_F^2 + \sum_{j=1}^{m} \text{tr}(D_j') v_j^T u_j. \tag{21}$$

Thus

$$p^\star = \min_{\substack{W_j = u_j v_j^T \\ (2D_j' - I)Xu_j \geq 0}} \frac{1}{2} \left\| \sum_{j=1}^{m} D_j' X W_j \right\|_F^2 + \sum_{j=1}^{m} \text{tr}(D_j')\text{tr}(W_j) \tag{22}$$

$$\geq \min_{W_j} \frac{1}{2} \left\| \sum_{j=1}^{P} D_j X W_j \right\|_F^2 + \sum_{j=1}^{P} \text{tr}(D_j)\text{tr}(W_j), \tag{23}$$

where $D_1, \ldots, D_P$ enumerates all possible sign patterns of $\text{diag}\left(\mathbb{1}\{Xu \geq 0\}\right)$. To prove the reverse direction, let $\{W_j^\star\}$ be the optimal solution to the convex program (17), we provide a way to reconstruct optimal $\{u_j, v_j\}$ which achieves the lower bound value. We first factorize each $W_j^\star = \sum_{k=1}^{d} \tilde{u}_{jk} \tilde{v}_{jk}^T$. According to Theorem 3.3 in [12], for any $\{j, k\}$, under Assumption 10, we can write $\tilde{u}_{jk} = \tilde{u}'_{jk} - \tilde{u}''_{jk}$ such that $\tilde{u}'_{jk}, \tilde{u}''_{jk} \in \mathcal{K}_j$ with $\mathcal{K}_j = \{u \in \mathbb{R}^d : (2D_j - I)Xu \succeq 0\}$. Therefore, when $m \geq 2Pd$, we can set $\{u_j, v_j\}$ to enumerate through $\{\tilde{u}'_{jk}, \tilde{v}_{jk}\}$ and $\{\tilde{u}''_{jk}, -\tilde{v}_{jk}\}$ to achieve optimal value of (17). With absolute value activation, the conclusion holds by replacing D_j with $\text{diag}\left(\text{sign}(Xu_j)\right)$ and $D_1, \ldots, D_P$ enumerate all possible sign patterns of $\text{diag}\left(\text{sign}(Xu)\right)$. ■

Appendix E. Proof in Section 2.3

E.1. Proof of Score Prediction

E.1.1. SCORE PREDICTION FOR RELU WITHOUT SKIP CONNECTION

Proof The optimality condition for convex program (4) is

$$0 \in A^T Ay + b + \beta\theta_1, \tag{24}$$

where $\theta_1 \in \partial\|y\|_1$. To show $y^\star$ satisfies optimality condition (24), let a_i denote the ith column of A. Check the first entry,

$$a_1^T Ay + b_1 + \beta(-1) = nvy_1^\star - nvy_{3n}^\star + n - \beta = 0.$$

Check the $3n$th entry,

$$a_{3n}^T Ay + b_{3n} + \beta = -nvy_1^\star + nvy_{3n}^\star - n + \beta = 0.$$

For jth entry with $j \notin \{1, 3n\}$, note

$$\begin{aligned} & |a_j^T Ay + b_j| \\ = & |a_j^T (a_1 y_1^\star + a_{3n} y_{3n}^\star) + b_j| \\ = & |a_j^T (a_1 y_1^\star - a_1 y_{3n}^\star) + b_j| \\ = & \left| \frac{\beta - n}{nv} a_j^T a_1 + b_j \right|. \end{aligned}$$

Since $|b_j| \leq n - 1$, by continuity, $|a_j^T Ay + b_j| \leq \beta$ should hold as we decrease β a little further to threshold $\beta_1 = \max_{j \notin \{1,3n\}} |a_j^T Ay + b_j|$. Therefore, $y^\star$ is optimal.

■

E.1.2. SCORE PREDICTION FOR ABSOLUTE VALUE ACTIVATION WITHOUT SKIP CONNECTION

Proof Assume without loss of generality data points are ordered as $x_1 < \ldots < x_n$, then

$$b = [n, n-2, \cdots, -(n-2), n-2, n-4, \cdots, -n].$$

The optimality condition to the convex program (4) is given by

$$0 \in A^T Ay + b + \beta\theta_1, \tag{25}$$

where $\theta_1 \in \partial\|y\|_1$. To show $y^\star$ satisfies optimality condition (25), let a_i denote the ith column of A. We check the first entry

$$a_1^T Ay + b_1 + \beta(-1) = nvy_1 - nvy_n + n - \beta = 0.$$

We then check the last entry

$$a_n^T Ay + b_n + \beta = -nvy_1 + nvy_n - n + \beta = 0.$$

For jth entry with $1 < j < n$, note

$$\begin{aligned} & |a_j^T Ay + b_j| \\ = & |a_j^T (a_1 y_1 + a_n y_n) + b_j| \\ = & |a_j^T (a_1 y_1 - a_1 y_n) + b_j| \\ = & \left| \frac{\beta - n}{nv} a_j^T a_1 + b_j \right|. \end{aligned}$$

Since $|b_j| \leq n - 2$, by continuity, $|a_j^T Ay + b_j| \leq \beta$ should hold as we decrease β a little further to some threshold $\beta_2 = \max_{j \notin \{1,n\}} |a_j^T Ay + b_j|$. Therefore, $y^\star$ satisfies (25). ■

E.1.3. SCORE PREDICTION FOR RELU WITH SKIP CONNECTION

In the convex program (12), $y = 0$ is an optimal solution when $2\beta \geq \|b\|_\infty$. Therefore, following the reconstruction procedure described in Appendix D.3.1, the corresponding neural network parameter set is given by $\{W^{(1)} = 0, b^{(1)} = 0, W^{(2)} = 0, b^{(2)} = \mu/v, V = -1/v\}$ with μ and v denotes the sample mean and sample variance as described in Section 2.3. For any test data $\hat{x}$, the corresponding predicted score is given by

$$\hat{y} = V\hat{x} + b^{(2)} = -\frac{1}{v}(\hat{x} - \mu),$$

which gives the score function of Gaussian distribution with mean being sample mean and variance being sample variance.Therefore, adding skip connection would change the zero score prediction to a linear function parameterized by sample mean and variance in the large weight decay regime.

E.1.4. SCORE PREDICTION FOR ABSOLUTE VALUE ACTIVATION WITH SKIP CONNECTION

Consider convex program (14), when $\beta > \|b\|_\infty$, $y = 0$ is optimal. Following the reconstruction procedure described in Appendix D.3.2, the corresponding neural network parameter set is given by $\{W^{(1)} = 0, b^{(1)} = 0, W^{(2)} = 0, b^{(2)} = \mu/v, V = -1/v\}$ with μ and v denotes the sample mean and sample variance as described in Section 2.3. For any testing data $\hat{x}$, the corresponding predicted score is given by

$$\hat{y} = V\hat{x} + b^{(2)} = -\frac{1}{v}(\hat{x} - \mu),$$

which is the score function of Gaussian distribution with mean being sample mean and variance being sample variance, just as the case for σ being ReLU activation and $V \neq 0$ described in Appendix D.3.1.

E.2. Proof of Convergence Result

The target distribution π satisfies

$$\pi \propto \begin{cases} \exp(\frac{\beta-n}{2nv}x^2 - \frac{\mu(\beta-n)}{nv}x), & x_1 \leq x \leq x_n, \\ \exp((\frac{\beta-n}{4nv} - \frac{t}{2})x^2 + (\frac{\beta-n}{2nv} + t)x_1 x \\ \quad +(\frac{\mu(n-\beta)}{nv})x + (\frac{n-\beta}{4nv} - \frac{t}{2})x_1^2), & x < x_1, \\ \exp((\frac{\beta-n}{4nv} + \frac{t}{2})x^2 - (\frac{n-\beta}{2nv} + t)x_n x \\ \quad +(\frac{\mu(n-\beta)}{nv})x + (\frac{t}{2} + \frac{n-\beta}{4nv})x_n^2), & x > x_n, \end{cases}$$

for some $|t| \leq \frac{n-\beta}{2nv}$.

Proof When $\beta_1 < \beta \leq n$, the predicted score function is differentiable almost everywhere with least slope 0 and largest slope $(n-\beta)/(nv)$. Then since the integrated score function is weakly concave, Theorem 2 follows case 1 in Theorem 4.3.6 in [2]. ■

Appendix F. Denoising Score Matching Background

To tackle the difficulty in computation of trace of Jacobian required in score matching objective (1), denoising score matching has been proposed in [19]. It then becomes widely used in practical generative models, especially for its natural conjunction with annealed Langevin sampling procedure, which forms the current mainstream noising/denoising paradigm of large-scale diffusion models being used in popular applications.

To briefly review, denoising score matching first perturbs data points with a predefined noise distribution and then estimates the score of the perturbed data distribution. When the noise distribution is chosen to be standard Gaussian, for some noise level $\epsilon > 0$, the objective is equivalent to

$$\min_{\theta} \ \mathbb{E}_{x\sim p_{\text{data}}(x)}\mathbb{E}_{\delta\sim\mathcal{N}(0,I)} \left\| s_\theta(x+\epsilon\delta) - \frac{\delta}{\epsilon} \right\|_2^2,$$

with the empirical version given by

$$\mathbf{DSM}(s_\theta) = \sum_{i=1}^{n} \frac{1}{2} \left\| s_\theta(x_i+\epsilon\delta_i) - \frac{\delta_i}{\epsilon} \right\|_2^2, \tag{26}$$

where $\{x_i\}_{i=1}^n$ are samples from $p_{\text{data}}(x)$ and $\{\delta_i\}_{i=1}^n$ are samples from standard Gaussian. The final training loss we consider is the above score matching objective together with weight decay term, which writes

$$\min_{\theta} \ \mathbf{DSM}(s_\theta(x)) + \frac{\beta}{2}\|\theta'\|_2^2, \tag{27}$$

where $\theta' \subseteq \theta$ denotes the parameters to be regularized. Unlike for score matching objective where weight decay is important for optimal objective value to stay finite, here for denoising objective, weight decay is unnecessary and can be removed. In our derived convex program, we allow β to be arbitrarily close to zero so the result is general. Note (27) circumvents the computation of trace of Jacobian and is thus more applicable for training tasks in large data regime.

Univariate Data. Consider training data $x_1, \ldots, x_n \in \mathbb{R}$. Denoising score matching fitting with objective (27) is equivalent to solving a lasso problem in the sense that both problems have same optimal value and an optimal NN parameter set which achieves minimal loss can be derived from the solution to the corresponding lasso program. The difference between convex program of denoising score matching fitting and that of score matching fitting is that no linear term is included in this scenario. We detail this finding in the following theorem,

Theorem 11 *When σ is ReLU or absolute value activation and $V = 0$, denote the optimal denoising score matching objective value (27) with s_θ specified in (3) as p^*, when $m \geq$ len(y) and $\beta > 0$,*

$$p^* = \min_{y} \quad \frac{1}{2}\|Ay+b\|_2^2 + \beta\|y\|_1, \tag{28}$$

where the entries of A are determined by the pairwise distances between data points.

Proof See Appendix G.1. ■

For demonstration, consider σ being ReLU, then coefficient matrix $A \in \mathbb{R}^{n\times 2n}$ is concatenation of two $n \times n$ matrices, i.e., $A = [\bar{A}_1, \bar{A}_2]$ where $\bar{A}_1 = (I - 11^T/n)A_1$ is the column-mean-subtracted version of A_1 and $[A_1]_{ij} = (x_i - x_j)_+$ measures pairwise distance between data points. Similarly, we have $\bar{A}_2 = (I - 11^T/n)A_2$ with $[A_2]_{ij} = (-x_i + x_j)_+$. Label vector b is the mean-subtracted version of original training label $l = [\sigma_1/\epsilon, \sigma_2/\epsilon, \cdots, \sigma_n/\epsilon]^T$. Once an optimal $y^\star$ to (28) has been derived, we can construct an optimal NN parameter set that achieves minimal training loss out of $y^\star$ and data points. See Appendix G.1 for details. Under this construction, with value of $y^\star$ known, given any test data $\hat{x}$, NN-predicted score is

$$\hat{y}(\hat{x}) = -\sum_{i=1}^{n} y_i^\star (x - x_i)_+ - \sum_{i=1}^{n} y_{n+i}^\star (-x + x_i)_+ + b_0^\star,$$

with $b_0^\star = \frac{1}{n}1^T([A_1, A_2]y^\star + l)$. We then proceed to present our multivariate data result, which holds for $b^{(1)}, b^{(2)}$ and β being all zero due to a change of our proof paradigm.

Appendix G. Proof in Section 3

G.1. Proof of Theorem 11

G.1.1. PROOF OF THEOREM 11 FOR RELU ACTIVATION

Consider data matrix $x \in \mathbb{R}^n$. Let m denote number of hidden neurons, then we have first layer weight $w \in \mathbb{R}^m$, first layer bias $b \in \mathbb{R}^m$, second layer weight $\alpha \in \mathbb{R}^m$ and second layer bias $b_0 \in \mathbb{R}$. Let l denotes the label vector, i.e, $l = [\delta_1/\epsilon, \delta_2/\epsilon, \ldots, \delta_n/\epsilon]^T$. The score matching objective is reduced to

$$p^\star = \min_{w,\alpha,b} \frac{1}{2} \left\| \sum_{j=1}^{m} (xw_j + 1b_j)_+ \alpha_j + 1b_0 - l \right\|_2^2 + \frac{1}{2}\beta \sum_{j=1}^{m} (w_j^2 + \alpha_j^2).$$

According to Lemma 2 in [15], after rescaling, the above problem is equivalent to

$$\min_{\substack{w,\alpha,b \\ |w_j|=1}} \frac{1}{2} \left\| \sum_{j=1}^{m} (xw_j + 1b_j)_+ \alpha_j + 1b_0 - l \right\|_2^2 + \beta \sum_{j=1}^{m} |\alpha_j|,$$

which can be rewritten as

$$\begin{aligned} \min_{\substack{w,\alpha,b,r \\ |w_j|=1}} \ & \frac{1}{2}\|r\|_2^2 + \beta \sum_{j=1}^{m} |\alpha_j| \\ \text{s.t. } & r = \sum_{j=1}^{m} (xw_j + 1b_j)_+ \alpha_j + 1b_0 - l. \end{aligned} \tag{29}$$

The dual of problem (29) writes

$$d^\star = \max_z -\frac{1}{2}\|z\|_2^2 + z^T l$$
$$\text{s.t.} \begin{cases} |z^T(x - 1x_i)_+| \le \beta \\ |z^T(-x + 1x_i)_+| \le \beta \\ z^T 1 = 0 \end{cases} \qquad \forall i = 1, \dots, n.$$

Note the constraint set is strictly feasible since $z = 0$ always satisfies the constraints, Slater's condition holds and we get the dual problem as

$$d^\star = \min_{\substack{z_0, z_1, z_2, z_3, z_4 \\ \text{s.t.} z_0, z_1, z_2, z_3 \ge 0}} \frac{1}{2}\|e\|_2^2 + f,$$

where $e = \sum_{i=1}^n z_{0i}(x - 1x_i)_+ - \sum_{i=1}^n z_{1i}(x - 1x_i)_+ + \sum_{i=1}^n z_{2i}(-x + 1x_i)_+ - \sum_{i=1}^n z_{3i}(-x + 1x_i)_+ + 1z_4 + l$ and $f = \beta(\|z_0\|_1 + \|z_1\|_1 + \|z_2\|_1 + \|z_3\|_1)$. Simplify to get

$$\min_y \frac{1}{2}\|Ay + \bar{l}\|_2^2 + \beta\|y\|_1,$$

with $A = [\bar{A}_1, \bar{A}_2] \in \mathbb{R}^{n \times 2n}$ where $\bar{A}_1 = \left(I - \frac{1}{n}11^T\right) A_1, \bar{A}_2 = \left(I - \frac{1}{n}11^T\right) A_2$ with $[A_1]_{ij} = (x_i - x_j)_+$ and $[A_2]_{ij} = (-x_i + x_j)_+$. $\bar{l}_j = l_j - \sum_i l_i/n$ is the mean-subtracted label vector. Once we obtain optimal solution $y^\star$ to problem (28), we can take

$$\begin{cases} w_j^\star = \sqrt{|y_j^\star|}, \alpha_j^\star = -\sqrt{|y_j^\star|}, b_j^\star = -\sqrt{|y_j^\star|}x_j \text{ for } j = 1, \dots, n, \\ w_j^\star = -\sqrt{|y_j^\star|}, \alpha_j^\star = -\sqrt{|y_j^\star|}, b_j^\star = \sqrt{|y_j^\star|}x_{j-n} \text{ for } j = n+1, \dots, 2n, \\ b_0^\star = \frac{1}{n}1^T([A_1, A_2]y^\star + l), \end{cases}$$

then denoising score matching objective has the same value as optimal value of convex program (28), which indicates $p^\star = d^\star$ and the above parameter set is optimal.

G.1.2. PROOF OF THEOREM 11 FOR ABSOLUTE VALUE ACTIVATION

Proof Consider data matrix $x \in \mathbb{R}^n$. Let m denote number of hidden neurons, then we have first layer weight $w \in \mathbb{R}^m$, first layer bias $b \in \mathbb{R}^m$, second layer weight $\alpha \in \mathbb{R}^m$ and second layer bias $b_0 \in \mathbb{R}$. Let l denotes the label vector, i.e, $l = [\delta_1/\epsilon, \delta_2/\epsilon, \dots, \delta_n/\epsilon]^T$. The score matching objective is reduced to

$$p^\star = \min_{w,\alpha,b} \frac{1}{2}\left\|\sum_{j=1}^m (xw_j + 1b_j)_+\alpha_j + 1b_0 - l\right\|_2^2 + \frac{1}{2}\beta\sum_{j=1}^m (w_j^2 + \alpha_j^2).$$

According to Lemma 2 in [15], after rescaling, the above problem is equivalent to

$$\min_{\substack{w,\alpha,b \\ |w_j|=1}} \frac{1}{2}\left\|\sum_{j=1}^m (xw_j + 1b_j)_+\alpha_j + 1b_0 - l\right\|_2^2 + \beta\sum_{j=1}^m |\alpha_j|,$$

which can be rewritten as

$$\begin{aligned} \min_{\substack{w,\alpha,b,r \\ |w_j|=1}} \ & \frac{1}{2}\|r\|_2^2 + \beta \sum_{j=1}^{m} |\alpha_j| \\ \text{s.t. } & r = \sum_{j=1}^{m} (xw_j + 1b_j)_+ \alpha_j + 1b_0 - l. \end{aligned} \tag{30}$$

The dual of problem (30) writes

$$\begin{aligned} d^\star = \max_z \ & -\frac{1}{2}\|z\|_2^2 + z^T l \\ \text{s.t. } & \begin{cases} |z^T(x - 1x_i)_+| \le \beta \\ |z^T(-x + 1x_i)_+| \le \beta \\ z^T 1 = 0 \end{cases} \qquad \forall i = 1, \dots, n. \end{aligned}$$

Note the constraint set is strictly feasible since $z = 0$ always satisfies the constraints, Slater's condition holds and we get the dual problem as

$$d^\star = \min_{\substack{z_0,z_1,z_2,z_3,z_4 \\ \text{s.t.} z_0,z_1,z_2,z_3 \ge 0}} \frac{1}{2}\|e\|_2^2 + f,$$

where $e = \sum_{i=1}^n z_{0i}(x - 1x_i)_+ - \sum_{i=1}^n z_{1i}(x - 1x_i)_+ + \sum_{i=1}^n z_{2i}(-x + 1x_i)_+ - \sum_{i=1}^n z_{3i}(-x + 1x_i)_+ + 1z_4 + l$ and $f = \beta(\|z_0\|_1 + \|z_1\|_1 + \|z_2\|_1 + \|z_3\|_1)$. Simplify to get

$$\min_y \frac{1}{2}\|Ay + \bar{l}\|_2^2 + \beta\|y\|_1.$$

where $A = \left(I - \frac{1}{n}11^T\right) A_3 \in \mathbb{R}^{n\times n}$ with $[A_3]_{ij} = |x_i - x_j|$, $\bar{l}$ is the same as defined in Appendix G.1.1. Once we obtain optimal solution $y^\star$ to problem (28), we can take

$$\begin{cases} w_j^\star = \sqrt{|y_j^\star|}, \alpha_j^\star = -\sqrt{|y_j^\star|}, b_j^\star = -\sqrt{|y_j^\star|}x_j \text{ for } j = 1, \dots, n, \\ w_j^\star = -\sqrt{|y_j^\star|}, \alpha_j^\star = -\sqrt{|y_j^\star|}, b_j^\star = \sqrt{|y_j^\star|}x_{j-n} \text{ for } j = n+1, \dots, 2n, \\ b_0^\star = \frac{1}{n}1^T([A_1, A_2]y^\star + l), \end{cases}$$

then denoising score matching objective has the same value as optimal value of convex program (28), which indicates $p^\star = d^\star$ and the above parameter set is optimal. ■

G.2. Proof of Theorem 3

Proof When $X \in \mathbb{R}^{n\times d}$ for some $d > 1$, when $\beta = 0$, the score matching objective can be reduced to

$$p^\star = \min_{u_j, v_j} \sum_{i=1}^{n} \frac{1}{2} \left\| \sum_{j=1}^{m} (X_i u_j)_+ v_j^T - L_i \right\|_2^2,$$

which can be rewritten as

$$\min_{u_j,v_j} \frac{1}{2} \left\| \sum_{j=1}^{m} (Xu_j)_+ v_j^T - Y \right\|_F^2. \tag{31}$$

Let $D_j' = \text{diag}\left(\mathbb{1}\{Xu_j \geq 0\}\right)$, then problem (31) is equivalent to

$$\min_{u_j,v_j} \frac{1}{2} \left\| \sum_{j=1}^{m} D_j' X u_j v_j^T - Y \right\|_F^2.$$

Therefore,

$$\begin{aligned} p^\star = & \min_{\substack{W_j = u_j v_j^T \\ (2D_j' - I)Xu_j \geq 0}} \frac{1}{2} \left\| \sum_{j=1}^{m} D_j' X W_j - Y \right\|_F^2 \\ \geq & \min_{W_j} \frac{1}{2} \left\| \sum_{j=1}^{P} D_j X W_j - Y \right\|_F^2, \end{aligned}$$

where $D_1, \ldots, D_P$ enumerate all possible sign patterns of $\text{diag}\left(\mathbb{1}\{Xu \geq 0\}\right)$. Under Assumption 10, the construction of optimal parameter set follows Appendix D.4.1. With absolute value activation, the same conclusion holds by replacing D_j' to be $\text{diag}\left(\text{sign}(Xu_j)\right)$ and $D_1, \ldots, D_P$ enumerate all possible sign patterns of $\text{diag}\left(\text{sign}(Xu)\right)$. ■

Appendix H. Details for Numerical Experiments in Section 4

H.1. Score Matching Fitting

For Gaussian data experiment, the training dataset contains $n = 500$ data points sampled from standard Gaussian. For non-convex neural network training, we run 10 trials with different random parameter initiations and solve with Adam optimizer with step size $1e-2$. We train for 500 epochs. We run Langevin dynamics sampling (Algorithm 2) with convex score predictor with 10^5 data points and $T = 500$ iterations, we take μ_0 to be uniform distribution from -10 to 10 and $\epsilon = 1$.

For Gaussian mixture experiment, the training dataset contains two Gaussian component each containing 500 data points, with centers at -10 and 10 and both have standard variance. We take $\beta = 20$.

H.2. Denoising Score Matching Fitting

For spiral data simulation, we first generate 100 data points forming a spiral as shown in the left most plot in Figure 3. We then add five levels of Gaussian noise with mean zero and standard deviation $[0.5, 0.1, 0.05, 0.03, 0.01]$. Thus the training data set contains 500 noisy data points. We fit five $2d$ convex score predictors corresponding to each noise level. We solve the convex program with CVXPY [4] with MOSEK solver [1]. The score plot corresponding to fitting our convex program with noise level 0.03. For annealed Langevin sampling, we sample 500 data points in total, starting

from uniform distribution on $[-10, 10]$ interval. We set $\epsilon = 1$ in each single Langevin process in Algorithm 2. We present the sample scatter plots sequentially after sample with 0.5 noise level score predictor for 5 steps (Level 1), 0.1 noise level score predictor for 5 steps (Level 2), 0.05 noise level score predictor for 5 steps (Level 3), 0.03 noise level score predictor for 5 steps (Level 4), and finally 0.01 noise level score predictor for 15 steps (Level 5).

H.2.1. CONVEX REFORMULATIONS OF DENOISING SCORE MATCHING FOR THE SPIRAL DATA GENERATION

Since convex program (6) requires to iterate over all activation pattern D_i's, here for easier implementation, we instead follow a variant of (6) which has been derived in Theorem 14 of [14], we replicate here for completeness. The formal theorem and our implementation follow Theorem 13 below. We first state the definition of Maximum Chamber Diameter, which is used in statement of Theorem 13 for theoretic soundness.

Definition 12 *We define the Maximum Chamber Diameter, denoted as*

$$\mathscr{D}(X) := \max_{\substack{w,v\in\mathbb{R}^d, \|w\|_2=\|v\|_2=1 \\ sign(Xw)=sign(Xv)}} \|w - v\|_2.$$

Consider the denoising score matching objective for a two-layer ReLU model given by

$$p_v^* \triangleq \min_{W^{(1)},W^{(2)},b} \left\| \sum_{j=1}^{m} \sigma(XW_j^{(1)})W_j^{(2)} - L \right\|_F^2 + \lambda \sum_{j=1}^{m} \|W_j^{(1)}\|_p^2 + \|W_j^{(2)}\|_p^2. \tag{32}$$

Here, the label matrix $L \in \mathbb{R}^{n\times d}$ contains the d-dimensional noise labels as in equation (6), and $W^{(1)} \in \mathbb{R}^{d\times m}$, $W^{(2)} \in \mathbb{R}^{m\times d}$. We introduce the following equivalent convex program.

$$\hat{p}_v \triangleq \min_{Z\in\mathbb{R}^{c\times d}} \|KZ - L\|_F^2 + \lambda \sum_{j=1} \|Z_j\|_2, \tag{33}$$

where Z_j is the j-th column of the matrix Z and $c = \binom{n}{d-1}$.

Theorem 13 *Define the matrix K as follows*

$$K_{ij} = \frac{\left(x_i \wedge x_{j_1} \wedge \cdots \wedge x_{j_{d-1}}\right)_+}{\|x_{j_1} \wedge \ \ldots \ \wedge x_{j_{d-1}}\|_p},$$

where the multi-index $j = (j_1, ..., j_{d-1})$ is over all combinations of $d-1$ rows. It holds that

- *when $p = 1$, the convex problem (33) is equivalent to the non-convex problem (32), i.e., $p_v^* = \hat{p}_v$.*
- *when $p = 2$, the convex problem (33) is a $\frac{1}{1-\epsilon}$ approximation of the non-convex problem (32), i.e., $p_v^* \leq \hat{p}_v \leq \frac{1}{1-\epsilon} p_v^*$, where $\epsilon \in (0, 1)$ is an upper-bound on the maximum chamber diameter $\mathscr{D}(X)$.*

An neural network achieving the above approximation bound can be constructed as follows:

$$f(x) = \sum_j Z_j^* \frac{\left(x_i \wedge x_{j_1} \wedge \cdots \wedge x_{j_{d-1}}\right)_+}{\|x_{j_1} \wedge \ \ldots \ \wedge x_{j_{d-1}}\|_p}, \tag{34}$$

where Z^* is an optimal solution to (33).

See Theorem 14 in [14] for proof. In our simulation in Section 4.2, we implement convex program (33), solve with CVXPY [4], and get score prediction from equation (34).

Appendix I. Additional Simulation Results

In this section, we give more simulation results besides those discussed in main text in Section 4. In Section I.1 we show simulation results for score matching tasks with more neural network types and data distributions. In section I.2 we show more simulation results for denoising score matching tasks.

I.1. Supplemental Simulation for Score Matching Fitting

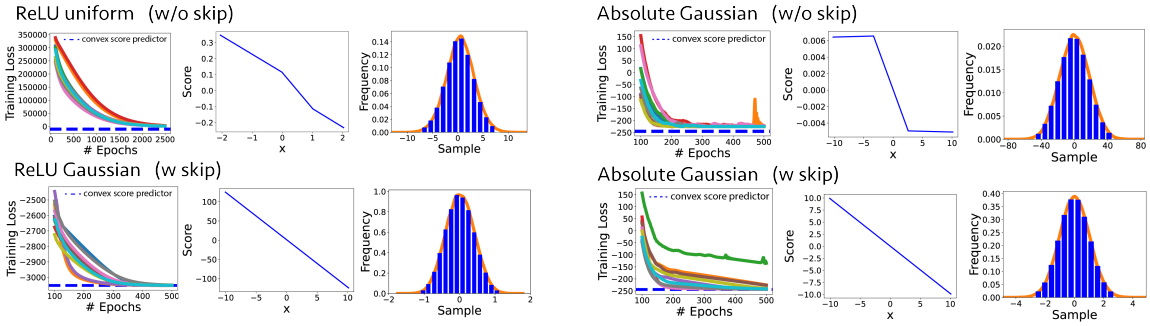

Figure 4: Simulation results for score matching tasks with two-layer neural network. The left subplots for all four categories show training loss where the dashed blue lines indicate loss of convex score predictor. The middle plots show score prediction by convex score predictor. The right plots show sampling histograms via plain Langevin process with convex score predictor. See Appendix I.1 for details.

Here we verify our findings in Section 2.3 with more experiments. The upper left plot in Figure 4 shows results for two-layer ReLU network without skip connection and with training data of uniform distribution on range $[0, 1]$. Here we still set $\beta = \|b\|_\infty - 1$ as in Section 4.1. Our theoretic analysis in Section 2.3 reveals that for this β value, the predicted score corresponding to Gaussian distribution characterized with sample mean and sample variance, which is corroborated by our simulation results here, i.e., the mid-subplot shows score function contained in left plot in Figure 1. The upper right plot follows same experimental setup except that here we experiment with absolute value activation instead of ReLU and the training data is standard normal. The predicted score is aligned with our theoretic derivation in right plot in Figure 1.

The bottom left and bottom right plots are for networks with skip connection. We set $\beta = \|b\|_\infty + 1$. Our theory in Section 2.3 concludes that for this β value, NNs without skip connection would predict zero scores while NNs with skip connection predict linear score corresponding to Gaussian distribution chractrized with sample mean and sample variance, which is supported by our simulation results.

For non-convex training, we run 10 trials with different random parameter initiations. It can be observed that our convex program always solves the training problem globally. Note for absolute value activation NNs (top right and bottom right plots), non-convex training sometimes sticks with local optimality, reflected by the gap of convergence value between non-convex training and our convex fitting.

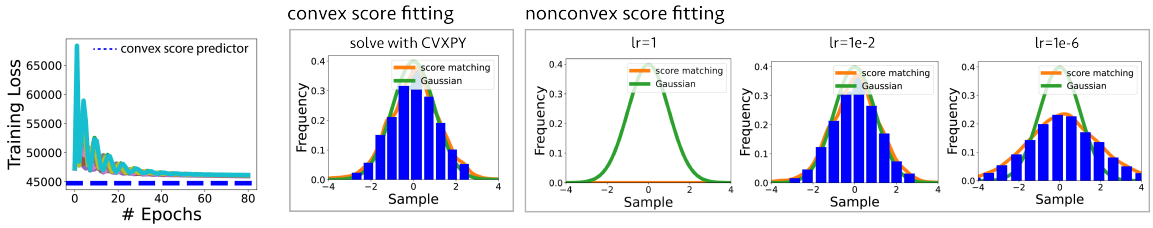

Figure 5: Simulation results for denoising score matching tasks with two-layer ReLU neural network. The left plot shows training loss where the dashed blue line indicates loss of convex score predictor (28). The second plot shows sampling histogram via annealed Langevin process with convex score predictor. The third, fourth, and fifth plots show sampling histograms via annealed Langevin process with non-convex score predictors trained with learning rates $1, 1e-2, 1e-6$ respectively. The ground truth distribution is standard Gaussian, which is recovered by our model.

I.2. Supplemental Simulation for Denoising Score Matching Fitting

For experiment in Figure 5, training data is standard Gaussian and $\beta = 0.5$ is adopted. we take ten noise levels with standard deviation $[\sigma_1, \ldots, \sigma_L]$ being the uniform grid from 1 to 0.01. For each noise level, we sample 10 steps. Initial sample points follow uniform distribution in range $[-1, 1]$. The non-convex training uses Adam optimizer and takes 200 epochs. Left most plot in Figure 5 shows the training loss of 10 non-convex fittings with stepsize $lr = 1e-2$ and our convex fitting. It can be observed that our convex fitting achieves lower training loss than all non-convex fittings. The second plot in Figure 5 shows annealed Langevin sampling histogram using our convex score predictor, which captures the underline Gaussian distribution. The right three plots show annealed Langevin sampling histograms with non-convex fitted score predictor trained with different learning rates. With $lr = 1$, training loss diverges and thus the predict score diverges from the true score of training data. Thus the sample histogram diverges from Gaussian. With $lr = 1e-2$, non-convex fitted NN recognizes the desired distribution while with $lr = 1e-6$, the NN is not trained enough thus the sampling results resemble Gaussian to some extent but not accurately. These results show that non-convex fitted score predictor is sometimes unstable due to training hyperparameter

setting while convex fitted score predictor is usually much more reliable and thus gains empirical advantage.