THE INDUCTIVE BIAS OF MINIMUM-NORM SHALLOW DIFFUSION MODELS THAT PERFECTLY FIT THE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

While diffusion models can generate high-quality images through the probability flow process, the theoretical understanding of this process is incomplete. A key open question is determining when the probability flow converges to the training samples used for denoiser training and when it converges to more general points on the data manifold. To address this, we analyze the probability flow of shallow ReLU neural network denoisers which interpolate the training data and have a minimal ℓ^2 norm of the weights. For intuition, we also examine a simpler dynamics which we call the score flow, and demonstrate that, in the case of orthogonal datasets, the score flow and probability flow follow similar trajectories. Both flows converge to a training point or a sum of training points. However, due to early stopping induced by the scheduler, the probability flow can also converge to a general point on the data manifold. This result aligns with empirical observations that diffusion models tend to memorize individual training examples and reproduce them during testing. Moreover, diffusion models can combine memorized foreground and background objects, indicating they can learn a "semantic sum" of training points. We generalize these results from the orthogonal dataset case to scenarios where the clean data points lie on an obtuse simplex. Simulations further confirm that the probability flow converges to one of the following: a training point, a sum of training points, or a point on the data manifold.

1 INTRODUCTION

In diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b), new images are sampled from the data distribution through an iterative process. Beginning with a random initialization, the model gradually denoises the image until a final image emerges. At their core, diffusion models learn the data distribution by estimating the score function of a Gaussian-blurred version of the data distribution. The connection between the score function and the denoiser, often called Tweedie's identity (Robbins, 1956; Miyasawa et al., 1961; Stein, 1981), holds only under optimal Bayes estimation. Moreover, for the estimated score to be a true gradient field, the denoiser must have a symmetric positive semidefinite Jacobian matrix (Chao et al., 2023; Manor & Michaeli, 2024). However, in practice, neural network denoisers are used, and their Jacobian matrix is generally non-symmetric, raising open questions about the convergence of the sampling process in score-based diffusion algorithms.

Diffusion models typically use a stochastic sampling process, which can be described by a stochastic differential equation (SDE) (Song et al., 2021b). Alternatively, a deterministic version of the sampling process can also be used, formulated as an ordinary differential equation (ODE) (Song et al., 2021a), called the probability flow ODE. We aim to theoretically analyze the probability flow, in order to illuminate this complex sampling process. However, practical diffusion architectures are typically deep and not fully connected, making it difficult to obtain theoretical guarantees without making additional strong assumptions (e.g., assuming a linearized regime like the neural tangent kernel (Jacot et al., 2018)). Therefore, in this paper we focus on diffusion models based on shallow ReLU neural network denoisers. These are both simple enough to allow for a theoretical investigation and rich enough to offer valuable insights.

To gain intuition into the dynamics of the probability flow ODE, we also explore a simpler ODE that corresponds to flowing in the direction of the score of the noisy data distribution, for a fixed noise-level. We call this the *score-flow* ODE. The score flow aims to sample from one of the modes of the noise-perturbed data distribution. We explore both the probability flow and the score flow ODEs for denoisers with minimal representation cost that perfectly fit the training data. Our analysis reveals that, for small noise levels, the trajectories of both flows is the same for a given initialization. However, the scheduler induces "early stopping", which determines whether the probability flow converges to training samples or to other points on the data manifold. This analysis provides insights into the stability and convergence properties of these processes.

Our contributions We investigate the probability and the score flow of shallow ReLU neural network denoisers in the context of interpolating noisy samples with minimal cost, specifically in the "low-noise regime", where noisy samples are well clustered.

- **Theoretical**: We prove that when the clean training points are orthogonal to one another, the probability flow and score flow follow a similar trajectory for a given initialization point. However, while the score flow converges only to a training point or to a sum of training points, the probability flow can also converge to a point on the boundary of the hyperbox whose vertices are all partial sums of the training points. This happens due to "early stopping" induced by the scheduler. We generalize this result to the case where the training points are the vertices of an obtuse simplex.
- **Experimental**: We train shallow denoisers that interpolate the training data with minimal representation cost on orthogonal datasets. We start by empirically demonstrating that the score flow ODE corresponding to a single such denoiser typically converges either to a sum of training points, which we call *virtual training points*, or to a general point on the boundary of the hyperbox (it converges to a training point only in rare occasions). We then show that the probability flow ODE, which uses a sequence of denoisers for varying noise levels, also converges to virtual points and to the boundary of the hyperbox, albeit at a somewhat lower frequency compared to the training points.

2 Setup and review of previous results

We study the denoising problem, where we observe a vector $\boldsymbol{y} \in \mathbb{R}^d$ that is a noisy observation of $\boldsymbol{x} \in \mathbb{R}^d$, i.e. $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$, such that $\mathbf{x}$ and $\boldsymbol{\epsilon}$ are statistically independent and $\boldsymbol{\epsilon}$ is Gaussian noise with zero mean and covariance matrix $\sigma^2 \boldsymbol{I}$. The MSE loss of any denoiser $\boldsymbol{h}(\boldsymbol{y})$ is

$$\mathcal{L}_{\mathrm{MSE}}(\boldsymbol{h}) = \mathbb{E}_{\mathbf{x},\mathbf{y}} \left\| \boldsymbol{h}(\mathbf{y}) - \mathbf{x} \right\|^2 , \tag{1}$$

where the expectation is over the joint probability distribution of $\mathbf{x}$ and $\mathbf{y}$. The minimizer of the MSE loss is the MMSE estimator

$$\boldsymbol{h}_{\mathrm{MMSE}}(\boldsymbol{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\mathbf{x} | \mathbf{y} = \boldsymbol{y} \right] . \tag{2}$$

In practice, since the true data distribution is unknown, we use empirical risk minimization with regularization. Consider a dataset consisting of M noisy samples for each of the N clean data points $\boldsymbol{x}_n$ such that $\boldsymbol{y}_{n,m} = \boldsymbol{x}_n + \boldsymbol{\epsilon}_{n,m}$, $n = 1, \ldots, N$, $m = 1, \ldots, M$. Then, one typically aims to minimize the loss

$$\mathcal{L}(\theta) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left\| \boldsymbol{h}_\theta(\boldsymbol{y}_{n,m}) - \boldsymbol{x}_n \right\|^2 + \lambda C(\theta) , \tag{3}$$

where θ are the parameters of the denoiser model $\boldsymbol{h}_\theta$ and $C(\theta)$ is a regularization term. Similarly to (Ongie et al., 2020; Zeno et al., 2023), we focus on a shallow ReLU network with a skip connection as the parametric model of interest, given by

$$\boldsymbol{h}_\theta(\boldsymbol{y}) = \sum_{k=1}^{K} \boldsymbol{a}_k [\boldsymbol{w}_k^\top \boldsymbol{y} + b_k]_+ + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{c} , \tag{4}$$

where $\theta = (\{\theta_k\}_{k=1}^K; \boldsymbol{c}, \boldsymbol{V})$ with $\theta_k = (b_k, \boldsymbol{a}_k, \boldsymbol{w}_k) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$ and $\boldsymbol{c} \in \mathbb{R}^d, \boldsymbol{V} \in \mathbb{R}^{d \times d}$ and the regularization term is a ℓ^2 penalty on the weights, but not on the biases and skip connections, i.e.,

$$C(\theta) = \frac{1}{2} \sum_{k=1}^{K} \left(\|\boldsymbol{a}_k\|^2 + \|\boldsymbol{w}_k\|^2 \right) . \tag{5}$$

Zeno et al. (2023) showed that in the "low-noise regime", i.e. when the clusters of noisy samples around each clean data point are well-separated[1], there are multiple solutions minimizing the empirical MSE (first term in equation 3). Each of these solutions has a different generalization capability. They studied the solution at which the ℓ_2 regularization of equation 5 is minimized.

Definition 1. *Let $\boldsymbol{h}_\theta : \mathbb{R}^d \to \mathbb{R}^d$ denote a shallow ReLU network of the form of equation 4. For any function $\boldsymbol{h} : \mathbb{R}^d \to \mathbb{R}^d$ realizable as a shallow ReLU network, we define its* ***representation cost*** *as*

$$R(\boldsymbol{h}) = \inf_{\theta:\, \boldsymbol{h}=\boldsymbol{h}_\theta} C(\theta) = \inf_{\theta:\, \boldsymbol{h}=\boldsymbol{h}_\theta} \sum_{k=1}^{K} \|\boldsymbol{a}_k\| \;\text{ s.t. }\; \|\boldsymbol{w}_k\| = 1,\ \forall k\,, \tag{6}$$

and a ***minimizer*** *of this cost, i.e., a 'min-cost' solution, as*

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h}} R(\boldsymbol{h}) \;\text{ s.t. }\; \boldsymbol{h}(\boldsymbol{y}_{n,m}) = \boldsymbol{x}_n \;\; \forall n, m\,. \tag{7}$$

In the multivariate case, finding an exact min-cost solution for finitely many noise realizations is generally intractable. Therefore, Zeno et al. (2023) simplified equation 7 by assuming that $\boldsymbol{h}(\boldsymbol{y}) = \boldsymbol{x}_n$ for all $\boldsymbol{y}$ in an open ball centered at $\boldsymbol{x}_n$. Specifically, letting $B(\boldsymbol{x}_n, \rho)$ denote the ball of radius ρ centered at $\boldsymbol{x}_n$, we simplify notations by writing this constraint as $\boldsymbol{h}(B(\boldsymbol{x}_n, \rho)) = \{\boldsymbol{x}_n\}$. Consider minimizing the representation cost under this constraint, that is, solving

$$\boldsymbol{h}^*_\rho(\boldsymbol{y}) \in \operatorname*{argmin}_{\boldsymbol{h}} R(\boldsymbol{h}) \;\text{ s.t. }\; \boldsymbol{h}(B(\boldsymbol{x}_n, \rho)) = \{\boldsymbol{x}_n\},\;\; \forall n. \tag{8}$$

Even this surrogate problem is still challenging to solve explicitly in the general case. Nonetheless, it can be solved for two specific configurations of training data points, which serve as prototypes for more general configurations. The first case is when all the data points form an obtuse simplex, i.e., the generalization of an obtuse triangle to higher dimensions, and the second case is when the data points form an equilateral triangle (see Appendix B).

3 The probability flow and the score flow

Once we have an explicit solution for the neural network denoiser, we estimate the score function by leveraging the connection between the MMSE denoiser and the score function (Robbins, 1956; Miyasawa et al., 1961; Stein, 1981),

$$\boldsymbol{h}_{\text{MMSE}}(\boldsymbol{y}) = \boldsymbol{y} + \sigma^2 \nabla \log p(\boldsymbol{y})\,, \tag{9}$$

where $p(\boldsymbol{y})$ is the probability density function of the noisy observation. From this relation, we can estimate the score function $\nabla \log p(\boldsymbol{y})$ as

$$\boldsymbol{s}(\boldsymbol{y}) = \frac{\boldsymbol{h}^*_\rho(\boldsymbol{y}) - \boldsymbol{y}}{\sigma^2}\,, \tag{10}$$

where $\boldsymbol{h}^*_\rho(\boldsymbol{y})$ is the minimum norm denoiser. In diffusion models, a stochastic process is typically used to sample new images. However, to generate unseen images from the data distribution, Song et al. (2021a) introduced a deterministic sampling process—the probability flow ODE (ordinary differential equation) (Song et al., 2021b; Karras et al., 2022).

We assume in this paper the variance exploding (VE) case, for which the probability flow ODE is given by

$$\forall t \in [0, T] : \frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = -\frac{1}{2}\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} \nabla \log p(\boldsymbol{y}_t, \sigma_t)\,, \tag{11}$$

where the score is estimated using the neural network denoiser $\nabla \log p(\boldsymbol{y}_t, \sigma_t) \approx \boldsymbol{s}(\boldsymbol{y}_t, \sigma_t)$, and $\sigma_t = \sqrt{t}$ is the scheduler. The minus sign in the probability flow ODE arises due to the reverse time variable: we initialize at $\boldsymbol{y}_T$, and finish at $\boldsymbol{y}_0$, a sample from the data distribution. In Appendix A we show that by using time re-scaling arguments the probability flow ODE is equivalent to the following ODE

$$\frac{\mathrm{d}\boldsymbol{y}_r}{\mathrm{d}r} = \boldsymbol{h}^*_{\rho_{g_r^{-1}}}(\boldsymbol{y}_r) - \boldsymbol{y}_r, \tag{12}$$

[1]The noise level in the low-noise regime, though small, is not negligible and has been noted as practically "useful" (Zeno et al., 2023), e.g. for diffusion sampling (Raya & Ambrogioni, 2023).

where $g_r = -\log \sigma_r$, assuming the radius of the noise balls satisfies $\rho_t = \alpha\sigma_t$ for some $\alpha > 0$.

Additionally, we will also analyze the score flow, which is a simplified case of equation 12 where ρ does not depend on t. Analyzing the score flow can be helpful in understanding the dynamics of the probability flow. The score flow represents the sampling process from one of the modes of the (multi-modal) distribution of $\boldsymbol{y}$. The score flow is initialized at $\boldsymbol{y}_0$ and for $t > 0$ follows

$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = \nabla \log p\left(\boldsymbol{y}\right) . \tag{13}$$

Using the estimated score function and time re-scaling $r = \frac{1}{\sigma^2}t$ we obtain the score flow

$$\frac{\mathrm{d}\boldsymbol{y}_r}{\mathrm{d}r} = \boldsymbol{h}^*_\rho(\boldsymbol{y}_r) - \boldsymbol{y}_r . \tag{14}$$

Notably, in contrast to the probability flow ODE, the min-cost denoiser here is independent of t.

4 THE PROBABILITY AND SCORE FLOW OF MIN-COST DENOISERS

In this section, we consider training sets that model different types of data manifolds, and state for each type the possible convergence points of the score and probability flows of min-cost solutions. As the score flow is a specific instance of probability flow (after time re-scaling) in which the variance profile is fixed, the difference between the convergence points of these two flows thus illuminates the effect of the variance reduction scheduling σ_t (and thus the ρ_t schedule) on the generated sample.

Specifically, we will consider datasets in which Zeno et al. (2023) found the min-cost solution $\boldsymbol{h}^*_\rho$ analytically: (1) orthogonal points, (2) points that form an obtuse angle with one of the points, and (3) a specific case of 3 training points forming an equilateral triangle.

We begin with the following simple, yet general, observation on the dynamics of score flow. For this dynamics, the stability condition for a stationary point $\boldsymbol{y}$ is that any eigenvalue of the Jacobian matrix of the score function with respect to the input $\boldsymbol{y}$, i.e., $\lambda\left(\boldsymbol{J}\left(\boldsymbol{y}\right)\right)$ satisfies

$$\mathrm{Re}\{\lambda\left(\boldsymbol{J}\left(\boldsymbol{y}\right)\right)\} < 0 \,. \tag{15}$$

We next show that in any model that perfectly fits an open ball of radius $\rho > 0$ around the training points (and thus also interpolates the training set), the clean data points are stable stationary points of the score flow. This implies that, when initialized near these points, the process can converge to the clean data points.

Proposition 1. *Let $\rho > 0$ be arbitrary. Let $\mathbf{h}\left(\boldsymbol{y}\right)$ be a denoiser that satisfies $\mathbf{h}(B(\boldsymbol{x}_n, \rho)) = \{\boldsymbol{x}_n\}$ for all $n \in [N]$ (and thus interpolates the training data). Then, any training point $\boldsymbol{y} \in \{\boldsymbol{x}_n\}_{n=1}^N$ is a stable stationary point of equation 13 where we estimate the score using $s\left(\boldsymbol{y}\right) = \frac{\boldsymbol{h}(\boldsymbol{y})-\boldsymbol{y}}{\sigma^2}$.*

Proof. For all $\boldsymbol{y} \in \{\boldsymbol{x}_n\}_{n=1}^N$ we get that $s\left(\boldsymbol{y}\right) = 0$ since the denoiser interpolates the training data. In addition, for all $\boldsymbol{y} \in \mathrm{int}\left(B(\boldsymbol{x}_n, \rho)\right)$ the Jacobian matrix is

$$\boldsymbol{J}\left(\boldsymbol{y}\right) = -\frac{1}{\sigma^2}\boldsymbol{I} \,, \tag{16}$$

therefore the stability condition of equation 15 holds. □

This result implies that, when the score function is differentiable and the training points are the only stationary points, the score flow will converge to the training points with probability 1.

4.1 ORTHOGONAL DATASETS

For simplicity, we begin with the case of a dataset composed of orthogonal points. Specifically, suppose that we have N training points $\{\boldsymbol{x}_n\}_{n=0}^{N-1}$ where $\boldsymbol{x}_0 = \mathbf{0}$ and the remaining training points are orthogonal, i.e., $\boldsymbol{x}_i^\top \boldsymbol{x}_j = 0$ for all $i, j > 0$ with $i \neq j$. [2] This approximates the behavior of data in many generic distributions (e.g., standard normal), which becomes more orthogonal in higher

[2]The result holds for the general case where $\boldsymbol{x}_0$ is non-zero, provided that $\left(\boldsymbol{x}_i - \boldsymbol{x}_0\right)^\top \left(\boldsymbol{x}_j - \boldsymbol{x}_0\right) = 0$.

dimensions. Let $\boldsymbol{u}_n = \boldsymbol{x}_n/\|\boldsymbol{x}_n\|$ for all $n = 1, ..., N-1$. A minimizer of equation 8, $\boldsymbol{h}^*_\rho$, is given by (Zeno et al., 2023, proof of Theorem 3)

$$\boldsymbol{h}^*_\rho(\boldsymbol{y}) = \sum_{n=1}^{N-1} \frac{\|\boldsymbol{x}_n\|}{\|\boldsymbol{x}_n\| - 2\rho} \left([\boldsymbol{u}_n^\top \boldsymbol{y} - \rho]_+ - [\boldsymbol{u}_n^\top \boldsymbol{y} - (\|\boldsymbol{x}_n\| - \rho)]_+ \right) \boldsymbol{u}_n. \tag{17}$$

We prove (Appendix B.1) the set of stationary points is the set of all possible sums of training points.

Theorem 1. *Suppose that the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ are orthogonal. Then, the set of the stable stationary points of equation 13 is $\mathcal{A} = \{\sum_{n\in\mathcal{I}} \boldsymbol{x}_n \mid \mathcal{I} \subseteq [N-1]\}$.*

This implies that the stationary points are the vertices of a hyperbox. Next, we prove (in Appendix B.2) that the score flow converges to the vertex of the hyperbox closest to the initialization $\boldsymbol{y}_0$. Also, for some $\boldsymbol{y}_0$, score flow first converges to the hyperbox boundary, then to a specific vertex.

Theorem 2. *Suppose that the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ are orthogonal. Consider the score flow where we estimate the score using $\boldsymbol{s}(\boldsymbol{y}) = \frac{\boldsymbol{h}^*_\rho(\boldsymbol{y}) - \boldsymbol{y}}{\sigma^2}$ and an initialization point $\boldsymbol{y}_0$. If $\forall i \in [N-1] : \boldsymbol{u}_i^\top \boldsymbol{y}_0 \neq \frac{\|\boldsymbol{x}_i\|}{2}$, then*

- *We converge to the closest vertex of the hyperbox to the initialization $\boldsymbol{y}_0$.*
- *If the closest point to $\boldsymbol{y}_0$ on the hyperbox is a point on its boundary which is not a vertex, then $\forall \epsilon < \min_i |\boldsymbol{u}_i^\top \boldsymbol{y}_0|$ there exists $\rho_0(\epsilon)$ and $T_0(\epsilon, \rho), T_1(\rho)$ such that for all $\rho < \rho_0(\epsilon)$ and all $T \in [T_0(\epsilon, \rho), T_1(\rho)]$, the point $\boldsymbol{y}_T$ is not a stable stationary point and at most at distance ϵ from the boundary of the hyperbox.*

Next, we consider the probability flow. For tractable analysis, we approximate the score estimator for small noise levels (i.e., for all $\min_{n\in[N-1]} \frac{\rho_t}{\|\boldsymbol{x}_n\|} \ll 1$) via Taylor's approximation to obtain

$$\boldsymbol{s}(\boldsymbol{y}, t) = \frac{1}{\sigma_t^2} \left(\sum_{n=1}^{N-1} \boldsymbol{u}_n \phi(\boldsymbol{u}_n^\top \boldsymbol{y}) - \left(\boldsymbol{I} - \sum_{n=1}^{N-1} \boldsymbol{u}_n \boldsymbol{u}_n^\top \right) \boldsymbol{y} \right) \tag{18}$$

where

$$\phi(z) = \begin{cases} -z & z < \rho_t \\ \rho_t \left(\frac{2}{\|\boldsymbol{x}_n\|} z - 1 \right) & \rho_t < z < \|\boldsymbol{x}_n\| - \rho_t \\ \|\boldsymbol{x}_n\| - z & z > \|\boldsymbol{x}_n\| - \rho_t \end{cases} . \tag{19}$$

With this approximation, one can show the probability flow and the score flow have a similar trajectory (for small ρ), if they have the same initialization point. However, the ρ_t scheduler in probability flow induces "early stopping". This can lead to the probability flow to converge to a non-vertex boundary point (in contrast to score flow), or to influence the speed of convergence to a stationary point. We show this in the following result for the probability flow (proved in Appendix B.3)

Theorem 3. *Suppose that the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ are orthogonal. Consider the probability flow where $\sigma_t = \sqrt{t}$, we estimate the score using equation 18, and $\boldsymbol{y}_T$ is the initialization point. If $\forall i \in [N-1] : \boldsymbol{u}_i^\top \boldsymbol{y}_T \neq \frac{\|\boldsymbol{x}_i\|}{2}$, then*

- *If the closest point to $\boldsymbol{y}_T$ on the hyperbox is a vertex, then we converge to this vertex.*
- *If the closest point to $\boldsymbol{y}_T$ on the hyperbox is not a vertex, then $\exists \tau(\boldsymbol{y}_T, \rho_T)$ such that we converge to the closest vertex to the initialization point $\boldsymbol{y}_T$ if $T > \tau(\boldsymbol{y}_T, \rho_T)$, and we converge to a point on the boundary of the hyperbox if $T < \tau(\boldsymbol{y}_T, \rho_T)$.*

Theorem 3 shows that the probability flow converges to a vertex of the hyperbox or a point on the boundary of the hyperbox. We consider this hyperbox boundary as an implicit data manifold—the diffusion model samples from this hyperbox boundary even though we did not assume an explicit sampling model that generated the training data, such as a distribution supported on the manifold. However, in some cases probability flow ODE can converge to specific points in this manifold: the training points, or sums of training points ("virtual points").

This result aligns well with empirical findings that diffusion models can memorize individual training examples and generate them during sampling (Carlini et al., 2023). In addition, an empirical result shows that Stable Diffusion (Rombach et al., 2022) can reproduce training data by piecing together foreground and background objects that it has memorized (Somepalli et al., 2023). This behavior resembles our result that the probability flow can also converge to sums of training points. In Stable Diffusion we observe a "semantic sum" of training points; however, our analysis focuses on the probability flow of a simple 1-hidden-layer model, while in deep neural networks summations in deeper layers can translate into more intricate semantic combinations.

4.2 Obtuse-angle datasets

We continue with the case of a non-orthogonal dataset. Specifically, suppose the convex hull of the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ is a $(N-1)$-simplex such that $\boldsymbol{x}_0$ forms an obtuse angle with all other vertices; we assume WLOG that $\boldsymbol{x}_0 = 0$. We refer to this as an obtuse simplex. Let $\boldsymbol{u}_n = \boldsymbol{x}_n / \|\boldsymbol{x}_n\|$ for all $n = 1, ..., N-1$. In this case, the minimizer $\boldsymbol{h}_\rho^*$ is still given by equation 17.

In Figure 1, we illustrate the normalized score flow for the case of an obtuse 2-simplex (see Figure 6 in Appendix E for the unnormalized score flow). The normalized score function is the score function multiplied by the log of the norm of the score and divided by the norm of the score. As shown, the training points are stationary points. Next, we prove (in Appendix B.4) that, in the general case of N training points, the set of stable stationary points is a subset of the set of all partial sums of the training points. Additionally, we demonstrate that when the angles between data points are nearly orthogonal, a stable stationary point corresponding to the sum of the points exists.

Theorem 4. *Suppose the convex hull of the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ is an obtuse simplex. Then, the set $\mathcal{A}$ of the stable stationary points of equation 13 satisfies $\{\boldsymbol{x}_n\}_{n=0}^{N-1} \subseteq \mathcal{A} \subseteq \{\sum_{n\in\mathcal{I}} \boldsymbol{x}_n \mid \mathcal{I} \subseteq \{0, 1, \cdots, N-1\}\}$. In addition, the point $\sum_{n\in\mathcal{I}} \boldsymbol{x}_n$, where $\mathcal{I} \subseteq \{0, 1, \cdots, N-1\}$ and $|\mathcal{I}| \geq 2$ if $0 \notin \mathcal{I}$ and $|\mathcal{I}| \geq 3$ if $0 \in \mathcal{I}$, is a stable stationary point if $\min_{k\in\mathcal{I}} \left\{\sum_{i\in\mathcal{I}\setminus\{k\}} \boldsymbol{u}_k^\top \boldsymbol{u}_i \|\boldsymbol{x}_i\|\right\} > -\rho$.*

The condition $\min_{k\in\mathcal{I}} \sum_{i\in\mathcal{I}\setminus\{k\}} \boldsymbol{u}_k^\top \boldsymbol{u}_i \|\boldsymbol{x}_i\| > -\rho$ holds for almost orthogonal dataset (and $\rho > 0$).

Next, we prove (in Appendix B.5) that in the general case with N training points, for small noise levels (i.e., small ρ) and an initialization point close to the chords connecting the origin to each training point ($\boldsymbol{x}_n$), the score flow first converges to a point along a chord connecting the origin and another training point, and then to an edge of the chord ($\boldsymbol{0}$ or $\boldsymbol{x}_n$, depending on initialization).

Theorem 5. *Suppose the convex hull of the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ is an obtuse simplex. Given an initial point $\boldsymbol{y}_0$ such that $\rho < \boldsymbol{u}_i^\top \boldsymbol{y}_0 < \|\boldsymbol{x}_i\| - \rho$ and $\boldsymbol{u}_j^\top \boldsymbol{y}_0 < \rho$ for all $j \neq i$, consider the score flow where we estimate the score using $\boldsymbol{s}(\boldsymbol{y}) = \frac{\boldsymbol{h}_\rho^*(\boldsymbol{y}) - \boldsymbol{y}}{\sigma^2}$. Then we converge to the closest edge of the chord. In addition, for all $\epsilon \in (0, \boldsymbol{u}_i^\top \boldsymbol{y}_0)$ there exists $\rho_0(\epsilon)$ and $T_0(\epsilon, \rho), T_1(\rho)$ such that for all $\rho < \rho_0(\epsilon)$ the point $\boldsymbol{y}_T$ is not a stable stationary point and at most at distance ϵ from the line between $\boldsymbol{x}_1$ and $\boldsymbol{x}_i$ for $T_0(\epsilon, \rho) < T < T_1(\rho)$.*

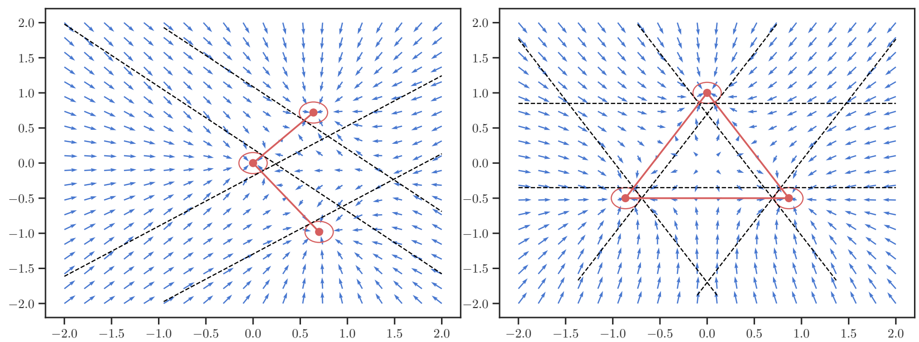
We next turn to the probability flow. To this end, we assume that the initial point $\boldsymbol{y}_T$ is such that $\rho_T < \boldsymbol{u}_i^\top \boldsymbol{y}_T < \|\boldsymbol{x}_i\| - \rho_T$ and $\boldsymbol{u}_j^\top \boldsymbol{y}_T < \rho$ for all $j \neq i$. We again use Taylor's approximation in the small-noise level regime (specifically, for all $i \in [N-1]$ $\frac{\rho_t}{\|\boldsymbol{x}_n\|} \ll 1$), to obtain the following score estimation at a point $\boldsymbol{y}$ such that $\rho_t < \boldsymbol{u}_i^\top \boldsymbol{y} < \|\boldsymbol{x}_i\| - \rho_t$ and $\boldsymbol{u}_j^\top \boldsymbol{y} < \rho_t$ for all $j \neq i$ is

$$\boldsymbol{s}(\boldsymbol{y}, t) = \frac{1}{\sigma_t^2}\left(\left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|}\rho_t\right)\boldsymbol{u}_i\boldsymbol{u}_i^\top - \boldsymbol{I}\right)\boldsymbol{y} - \rho_t \boldsymbol{u}_i\right). \tag{20}$$

We now have the following result regarding probability flow (proved in Appendix B.6)

Theorem 6. *Suppose the convex hull of the training points $\{\boldsymbol{x}_0, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N-1}\} \subset \mathbb{R}^d$ is an obtuse simplex. Given an initial point $\boldsymbol{y}_T$ such that $\rho_T < \boldsymbol{u}_i^\top \boldsymbol{y}_T < \|\boldsymbol{x}_i\| - \rho_T$ and $\boldsymbol{u}_j^\top \boldsymbol{y}_T < \rho_T$ for all $j \neq i$. Consider the probability flow where $\sigma_t = \sqrt{t}$ and we estimate the score using equation 20. Then, $\exists \tau(\boldsymbol{y}_T, \rho_T))$ such that we converge to a point on the line connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_i$ if $T < \tau(\boldsymbol{y}_T, \rho_T)$ and if $T \geq \tau(\boldsymbol{y}_T, \rho_T)$ we converge to the closest point in the set $\{\boldsymbol{x}_0, \boldsymbol{x}_i\}$ to $\boldsymbol{y}_T$.*

Theorem 6 shows that the probability flow converges to a point on the chord or to one of the edges of the chord. In this scenario, we consider the chords as the implicit data manifold.



(a) (b)

Figure 1: **The normalized score function of obtuse and acute simplex**. The red dots are the training points $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$. The black lines are the ReLU boundaries. In Figure (a) we plot the score function of an obtuse triangle. In Figure (b) we plot an equilateral triangle.

4.3 An equilateral triangle dataset

Finally, for completeness, we consider the score flow in the case where the training points form the vertices of an equilateral triangle (as this is the last remaining dataset case for which the min-cost denoiser is analytically solvable (Zeno et al., 2023)). We prove (in Appendix B.7) that, given an initialization point near the edge of the triangle, the score flow first converges to the face of the triangle (the implicit data manifold here) and then to the vertex closest to the initialization point $\boldsymbol{y}_0$.

Proposition 2. *Suppose the convex hull of the training points $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \in \mathbb{R}^d$ is an equilateral triangle. Given an initial point $\boldsymbol{y}_0$ such that $i \in \{1, 2\}$ $-\frac{\|\boldsymbol{x}_i\|}{2} + \rho < \boldsymbol{u}_i^\top \boldsymbol{y}_0 < \|\boldsymbol{x}_i\| - \rho$ and $\boldsymbol{u}_3^\top \boldsymbol{y} < -\frac{\|\boldsymbol{x}_3\|}{2} + \rho$, consider the score flow where we estimate the score using $\boldsymbol{s}(\boldsymbol{y}) = \frac{\boldsymbol{h}_\rho^*(\boldsymbol{y}) - \boldsymbol{y}}{\sigma^2}$. Then we converge to the closest vertex to the $\boldsymbol{y}_0$. In addition, for all $0 < \epsilon < (\boldsymbol{u}_1 + \boldsymbol{u}_2)^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}\|}{2}$ there exists $\rho_0(\epsilon)$ and $T_0(\rho, \epsilon), T_1(\rho)$ such that for all $\rho < \rho_0(\epsilon)$ the point $\boldsymbol{y}_T$ is not a stable stationary point and at most ϵ distance from the line between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for $T_0(\rho, \epsilon) < T < T_1(\rho)$.*

Without loss of generality, we can permute the training points indices $\{1, 2, 3\}$ in the above result. The probability flow for this case can be also analyzed, similarly to what we did in previous cases.

5 Simulations

In this section, we demonstrate the findings of Theorems 1, 2 and 3 in shallow neural networks. In practical settings, the continuous probability flow ODE given by equation 11 is discretized to S timesteps, as

$$\boldsymbol{y}_{t-1} = \boldsymbol{y}_t + (\sigma_t^2 - \sigma_{t-1}^2) \frac{(\boldsymbol{h}_{\rho_t}^*(\boldsymbol{y}_t) - \boldsymbol{y}_t)}{2\sigma_t^2}, \quad t = T, \ldots, 1, \tag{21}$$

where $\boldsymbol{h}_{\rho_t}^*(\boldsymbol{y}_t)$ is modeled as a series of S denoisers (usually with weight sharing), which are applied consecutively to gradually denoise the signal. In this setting, the sampling should theoretically be initialized at $T = \infty$, however in practice it is initialized from a finite timestep T, which is chosen such that $\sigma_T \gg \|\boldsymbol{x}_i\|$ for all i. Similarly, the score-flow of equation 13 is discretized as

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \gamma \frac{(\boldsymbol{h}_{\rho_{t_0}}^*(\boldsymbol{y}_t) - \boldsymbol{y}_t)}{\sigma_{t_0}^2}, \quad t = 0, 1, \ldots, \tag{22}$$

where γ is some step size and here t_0 is a fixed timestep (so that all iterations are with the same denoiser). Note that here t increases along the iterations, and since we use a single denoiser, there is no constraint on the number of iterations we can perform.

It should be noted that while our theorems characterize only the low-noise regime, here we simulate a more practical sampling process, which starts the sampling from large noise. Namely, the initialization

($\boldsymbol{y}_T$ in equation 21 and $\boldsymbol{y}_0$ in equation 22) is drawn from a Gaussian with large σ. Thus, our theoretical analysis becomes relevant only once the dynamics enter the low-noise regime.

To demonstrate our results for the case of an orthogonal dataset, we use orthonormal training samples, set $\sigma_t = \sqrt{t}$, and choose $T = 100$ to ensure an effectively high noise at the beginning of the sampling process. We train a set of $S = 150$ denoisers, ensuring 50 equally-spaced noise levels in the "low-noise regime" and 100 equally-spaced noise levels outside it. We train our networks on data in dimension $d = 30$, with $M = 500$ noisy samples per training sample, taking the dimension of the hidden layer of the networks to be $K = 300$.

To be consistent with our theory, which assumes the denoiser achieves exact interpolation over the noisy training samples, we use a non-standard training protocol to enforce close-to-exact interpolation. Specifically, we pose the denoiser training as the equality constrained optimization problem

$$\min_{\theta} C(\theta) \;\; s.t. \;\; \boldsymbol{h}_\theta(\boldsymbol{y}_{n,m}) = \boldsymbol{x}_n, \;\; \forall n, m \tag{23}$$

which we optimize using the Augmented Lagrangian (AL) method (see, e.g., (Nocedal & Wright, 2006)). Specifically, we define

$$\mathcal{L}_{AL}(\theta, \boldsymbol{Q}, \mu) := C(\theta) + \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{\mu}{2} \|\boldsymbol{h}_\theta(\boldsymbol{y}_{n,m}) - \boldsymbol{x}_n\|^2 + \langle \boldsymbol{q}^{(n,m)}, \boldsymbol{h}_\theta(\boldsymbol{y}_{n,m}) - \boldsymbol{x}_n \rangle \tag{24}$$

where $\mu \in \mathbb{R}_{>0}$, $\boldsymbol{q}^{(n,m)} \in \mathbb{R}^d$ represents a vector of Lagrange multipliers, and $\boldsymbol{Q} \in \mathbb{R}^{d \times MN}$ is the matrix whose columns are $\boldsymbol{q}^{(n,m)}$ for all $m = 1, ..., M$, $n = 1, ..., N$. Then, starting from an initialization of $\mu_0 > 0$ and $\boldsymbol{Q}_0 = \boldsymbol{0}$, for $k = 0, 1, ..., \mathcal{K}$ we perform the iterative updates:

$$\theta_{k+1} = \arg\min_{\theta} \mathcal{L}_{AL}(\theta_k, \boldsymbol{Q}_k, \mu_k) \tag{25}$$

$$\boldsymbol{q}_{k+1}^{(n,m)} = \boldsymbol{q}_k^{(n,m)} + \mu_k(\boldsymbol{h}_\theta(\boldsymbol{y}_{n,m}) - \boldsymbol{x}_n), \;\; \forall n, m \tag{26}$$

$$\mu_{k+1} = \eta \mu_k, \tag{27}$$

where $\eta > 1$ is a fixed constant. The solution of equation 25 is approximated by following standard training using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 10^{-4} for 10^4 iterations. We additionally take $\eta = 3$ and $\mathcal{K} = 7$, and decrease the learning rate by 0.5 after each iterative update.

We start by demonstrating the existence of *virtual* training points, that is, stable stationary points that are sums of training points, as predicted by Theorem 1. We take a denoiser from the "low-noise regime" ($\sigma_t = 0.095$ in this example) and run 10 fixed-point iterations on all the predicted virtual points that consist of combinations of pairs, triplets and quadruplets of the training points. In Figure 2 we plot the percentage of these runs that converged within an L_∞ distance of 0.2 to the predicted virtual point. As can be seen, 98.6% of the predicted virtual points composed of pairs of training points are stable in practice, and the stability of virtual points decreases as higher-numbers of combinations are considered. Nevertheless, the absolute number of stable virtual points increases substantially as higher-numbers of combinations are considered. Specifically, in the same example a total of 429, 3390, and 6965 stationary points were found for the pairs, triplet and quadruplet combinations. The increase in the absolute numbers is due to the higher number of higher-order sums. The decrease in percentages is due to small deviations in the ReLU boundaries of the trained denoiser compared to the theoretical optimal denoiser. These deviations have a greater impact on stationary points that involve sums of more training points.

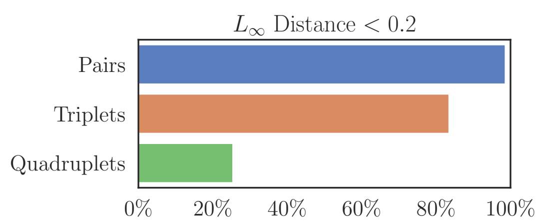
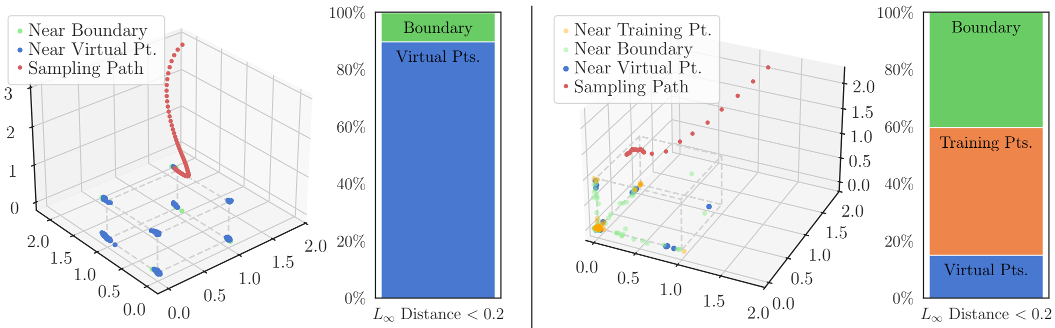


Figure 2: **Existence of stable virtual training points.** We run fixed-point iterations on a single denoiser, starting from all possible pair-wise, triplet-wise, and quadruplet-wise combinations of training samples. The plot shows the percentage of points that converged within an L_∞ distance of 0.2 to the original, virtual, input point.

Next, we explore the full dynamics of the diffusion process. We start with the score flow for a single denoiser from timestep t_0, which corresponds to noise level $\sigma_{t_0} = 0.095$. We randomly



(a) Score Flow. (b) Probability Flow.

Figure 3: **Projection to three dimensions and convergence types frequency of randomly sampled points.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$, for both sampling using the score flow (3a) and a regular diffusion process (3b). For each, we plot on the right the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points. On the left, we plot the sampling results projected to three dimensions, along with the path a single point took until convergence. In score flow, all points converged to either virtual points or to boundaries of the hyperbox, which is evident in the point clusters in the locations of the projected virtual points. For probability flow, the bias induced by the "large-noise regime" denoisers diffusion causes more samples to converge around the the training points and their adjacent boundaries. Nevertheless, a large percentage of samples still converge in the vicinity of virtual points. The paths the points take towards the hyperbox draws them first to the closest boundary, and then, if the steps sizes and amount permit, travel along the edges towards the closest stable stationary points.

sample 500 points from $\mathcal{N}(0, 100\boldsymbol{I})$, and apply 3000 score-flow iterations to each, with a step size of $\gamma = 5 \cdot 10^{-4}$. The right hand side of Figure 3a shows the percentage of points that converged within an L_∞ distance of 0.2 to either virtual points, training points, or a boundary of the hyperbox. On the left hand side of Figure 3a, we plot the projection of all samples on three dimensions. Out of 500 samples, almost all points converged to virtual points, which is expected in random initialization due to their larger number, compared to the training points. The rest of the points converged to the hyperbox's boundaries. The path the points take towards the hyperbox first draws them to the closest boundary, and then they drift along the boundary towards the closest stable stationary point.

Finally, we examine a full diffusion process with the probability ODE. Here we follow equation 21 using $S = 150$ trained denoisers, starting again from 500 randomly sampled points from $\mathcal{N}(0, \sigma_T \boldsymbol{I})$. Our results hold where the noise level is small compared to the norm of the training samples. Therefore, denoisers of large noise levels are not expected to have stable virtual points. In probability flow most noise levels are large compared to this norm, as the sampling process begins with a large variance (in the VE case). Specifically, in our example only the last 50 denoisers have small noise levels. Yet, as can be seen on the right hand side of Figure 3b, a large percentage of the samples produced are virtual points. In contrast to the score flow case, the start of the sampling process here attracts most samples towards the mean of the training points, as any optimal-MSE denoiser would, which creates a biased starting point to the the sampling process in the "low-noise regime". From this regime onwards, the points travel along the boundaries of the hyperbox towards their nearest stable points, which is usually a training point. This behaviour is demonstrated on the left side of Figure 3b, where the projected path of a random point is drawn starting from the 90$^{\text{th}}$ step.

Please refer to Appendix E for comparisons of additional thresholds, and to Appendix C and D for discussions on the effects of the training set size and the minimum norm constraint.

6 Related work

Memorization and Generalization in Deep Generative Models Several recent works have sought to explain the transition from memorization to generalization in deep generative models, both from a theoretical and empirical perspective. One early line of work in this vein studied memorization in over-parametrized (non-denoising) autoencoders (Radhakrishnan et al., 2019; 2020). This work shows that over-parameterized autoencoders trained to low cost are locally contractive about each training sample, such that training images can be recovered by iteratively applying the autoencoder to noisy inputs. A theoretical explanation of this phenomenon using a neural tangent kernel analysis is given in (Jiang & Pehlevan, 2020). More recent work has also shown that state-of-the-art diffusion models exhibit a similar form of memorization, such that extraction of training samples is possible by identifying stable stationary points of the diffusion process (Carlini et al., 2023). Additionally, when trained on few images, several works have shown that the outputs of diffusion models are strongly biased towards the training set, and thus fail to generalize (Somepalli et al., 2023; Yoon et al., 2023; Kadkhodaie et al., 2024). A recent empirical study suggests that memorization and generalization in diffusion models are mutually exclusive phenomenon, and successful generation occurs only when memorization fails (Yoon et al., 2023; Zhang et al., 2023). Beyond these empirical studies, recent work has put forward theoretical explanations for generalization in score-based models. In (Pidstrigach, 2022), the authors show that score-based models can learn manifold structure in the data generating distribution. A complementary perspective is provided by Kadkhodaie et al. (2024), which argues that diffusion models implicitly encode geometry-adaptive harmonic representations.

Representation costs and neural network denoisers Several other works have investigated overparameterized autoencoding/denoising networks with minimal representation cost (i.e., minimial ℓ^2-norm of parameters). Function space characterizations of min-norm solutions of shallow fully connected neural networks are given in (Savarese et al., 2019; Ongie et al., 2020; Parhi & Nowak, 2021; Shenouda et al., 2023); extensions to deep networks and emergent bottleneck structure are considered in (Jacot, 2022; Jacot et al., 2022; Jacot, 2023; Wen & Jacot, 2024). The present work relies on the shallow min-norm solutions derived by Zeno et al. (2023) for specific configurations of data points, but goes beyond this work in studying the dynamics of its associated flows.

A recent study investigates properties of shallow min-norm solutions to a score matching objective (Zhang & Pilanci, 2024), building off of a line of work that studies min-norm solutions from a convex optimization perspective (Pilanci & Ergen, 2020; Ergen & Pilanci, 2020; Sahiner et al., 2021; Wang & Pilanci, 2021). In the case of univariate data, an explicit min-norm solution of the score-matching objective is derived, and convergence results are given for Langevin sampling with the neural network-learned score function. Additionally, in the multivariate case, general min-norm solutions to the score-matching loss are characterized as minimizers of a quadratic program. Our results differ from (Zhang & Pilanci, 2024) in that we study different optimization formulations (denoising loss versus score-matching loss) and inference procedures (probability- and score-flow versus Langevin dynamics). Our results focus on high-dimensional data belonging to a simplex, while Zhang & Pilanci (2024) give convergence guarantees only in the case of univariate data.

7 Discussion

Conclusions. We explored the probability flow ODE of shallow neural networks with minimal representation cost. We showed that for orthogonal dataset and obtuse-angle dataset the probability flow and the score flow follows the same trajectory given the same initialization point and small noise level. The scheduler in probability flow induces "early stopping", which results in converging to a boundary point instead of a specific vertex (as in score flow) or speed up convergence to a specific vertex. One possible extension of this work is to analyze the probability flow ODE in the case of variance-preserving processes. This is an important case since practical diffusion models more often use variance-preserving forward and backward processes.

Limitations. A key limitation of our analysis is the assumption (inherited from Zeno et al. (2023)) that the denoiser interpolates data across a full d-dimensional ball centered around each clean training sample, where d represents the input dimension. In real-world scenarios, the number of noisy samples is typically smaller than the input dimension d. A more accurate approach might involve assuming that the denoiser interpolates over an $(M-1)$-dimensional disc around each training sample, reflecting the norm concentration of Gaussian noise in high-dimensional spaces. Furthermore, for mathematical tractability, our analysis focuses on a single hidden layer model.

ETHICS STATEMENT

This paper presents a theoretical analysis of diffusion models under specific constraints, aiming to enhance the understanding of generative models. This may lead to greater transparency when using these models. Moreover, we anticipate that insights gained from these simpler cases will shed light on the memorization and generalization behaviors in large-scale diffusion models, which pose privacy concerns. Lastly, we note the neural network examined in this paper is a shallow one, whereas practical contemporary implementations almost always involve deep networks. Naturally, addressing deep networks from the outset would pose an impassable barrier. In general, our guiding principle for research works that aim to understand new or not-yet-understood phenomena is that we should first study it in the simplest model that shows it, so as not to get distracted by possible confounders, and to enable a detailed analytic understanding. For example, when exploring or teaching a statistical problem issues, we would typically start with linear regression, understand the phenomena in this simple case, and then move on to more complex models. Thus, we hope the example we set in this paper will help promote this guiding principle for research and teaching.

REPRODUCIBILITY STATEMENT

The paper fully discloses all the information needed for reproducing the results. We provide full and detailed proofs for all claims in the paper in Appendices A and B. The details of the experimental results are detailed in Section 5, including hyper-parameters and training configuration. Additionally, code will be published upon acceptance.

References

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023. 6, 10

Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, and Chun-Yi Lee. On investigating the conservative property of score-based generative models. In *International Conference on Machine Learning (ICML)*, 2023. 1

Tolga Ergen and Mert Pilanci. Convex geometry of two-layer ReLU networks: Implicit autoencoding and interpretable models. In *International Conference on Artificial Intelligence and Statistics*, pp. 4024–4033. PMLR, 2020. 10

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

Arthur Jacot. Implicit bias of large depth networks: a notion of rank for nonlinear functions. In *The Eleventh International Conference on Learning Representations*, 2022. 10

Arthur Jacot. Bottleneck structure in learned features: Low-dimension vs regularity tradeoff. *Advances in Neural Information Processing Systems*, 36:23607–23629, 2023. 10

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf. 1

Arthur Jacot, Eugene Golikov, Clément Hongler, and Franck Gabriel. Feature learning in l_2-regularized DNNs: Attraction/repulsion and sparsity. *Advances in Neural Information Processing Systems*, 35:6763–6774, 2022. 10

Yibo Jiang and Cengiz Pehlevan. Associative memory in iterated overparameterized sigmoid autoencoders. In *International conference on machine learning*, pp. 4828–4838. PMLR, 2020. 10

Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ANvmVS2Yr0. 10, 26

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022. 3

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 3, 2015. 8

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=adSGeugiuj. 1

Koichi Miyasawa et al. An empirical bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38(181-188):1–2, 1961. 1, 3

Jorge Nocedal and Stephen J Wright. Penalty and augmented lagrangian methods. *Numerical Optimization*, pp. 497–528, 2006. 8

Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1lNPxHKDH. 2, 10

Rahul Parhi and Robert D Nowak. Banach space representer theorems for neural networks and ridge splines. *Journal of Machine Learning Research*, 22(43):1–40, 2021. 10

Jakiw Pidstrigach. Score-based generative models detect manifolds. *Advances in Neural Information Processing Systems*, 35:35852–35865, 2022. 10

Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning*, pp. 7695–7705. PMLR, 2020. 10

A Radhakrishnan, KD Yang, M Belkin, and C Uhler. Memorization in overparameterized autoencoders. In *Deep Phenomena Workshop, International Conference on Machine Learning*, 2019. 10

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. Overparameterized neural networks implement associative memory. *Proceedings of the National Academy of Sciences*, 117 (44):27162–27170, 2020. 10

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=lxGFGMMSVl. 3

Herbert Robbins. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, 1954-1955*, volume 1, pp. 157–163. Berkeley and Los Angeles: University of California Press, 1956. 1, 3

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 6

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. In *International Conference on Learnining Representations (ICLR)*, 2021. 10

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory*, pp. 2667–2690. PMLR, 2019. 10

Joseph Shenouda, Rahul Parhi, Kangwook Lee, and Robert D Nowak. Vector-valued variation spaces and width bounds for DNNs: Insights on weight decay regularization. *arXiv preprint arXiv:2305.16534*, 2023. 10

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015. 1

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6048–6058, 2023. 6, 10, 26

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP. 1, 3

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS. 1, 3

Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, 1981. 1, 3

Yifei Wang and Mert Pilanci. The convex geometry of backpropagation: Neural network gradient flows converge to extreme points of the dual convex program. In *International Conference on Learning Representations*, 2021. 10

Yuxiao Wen and Arthur Jacot. Which frequencies do CNNs need? emergent bottleneck structure in feature learning. *arXiv preprint arXiv:2402.08010*, 2024. 10

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. 10

Chen Zeno, Greg Ongie, Yaniv Blumenfeld, Nir Weinberger, and Daniel Soudry. How do minimum-norm shallow denoisers look in function space? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=gdzxWGGxWE. 2, 3, 4, 5, 7, 10, 15

Fangzhao Zhang and Mert Pilanci. Analyzing neural network-based generative diffusion models through convex optimization. *arXiv preprint arXiv:2402.01965*, 2024. 10

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. 2023. 10

A PROOFS OF RESULTS IN SECTION 3

The probability flow ODE is given by

$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = -\frac{1}{2}\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}\nabla \log p\left(\boldsymbol{y}_t, \sigma_t\right) \tag{28}$$

$$= -\sigma_t \frac{\mathrm{d}\sigma_t}{\mathrm{d}t}\nabla \log p\left(\boldsymbol{y}_t, \sigma_t\right) . \tag{29}$$

First, we apply change of variable as follows

$$r = g\left(t\right) = -\log \sigma_t \tag{30}$$

$$\frac{\mathrm{d}r}{\mathrm{d}t} = -\frac{1}{\sigma_t}\frac{\mathrm{d}\sigma_t}{\mathrm{d}t} \tag{31}$$

$$\frac{\mathrm{d}t}{\mathrm{d}r} = \left(-\frac{1}{\sigma_t}\frac{\mathrm{d}\sigma_t}{\mathrm{d}t}\right)^{-1} . \tag{32}$$

Therefore,

$$\frac{\mathrm{d}y_t}{\mathrm{d}r} = \frac{\mathrm{d}y_t}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}r} = \left(-\sigma_t \frac{d\sigma_t}{dt}\nabla \log p\left(y_t, \sigma_t\right)\right)\left(-\frac{\sigma_t}{\frac{d\sigma_t}{dt}}\right) \tag{33}$$

$$= \sigma_t^2 \nabla \log p\left(y_t, \sigma_t\right) \tag{34}$$

Next, we estimate the score function using a neural network denoiser, and substitute $t = g^{-1}\left(r\right)$ to obtain

$$\frac{\mathrm{d}y_r}{\mathrm{d}r} = h^*_{\rho(g^{-1}(r))}(y_r) - y_r . \tag{35}$$

B PROOFS OF RESULTS IN SECTION 4

In this section we use the following Propositions from (Zeno et al., 2023).

Proposition 3. *Suppose that the convex hull of the training points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \subset \mathbb{R}^d$ is a $(N-1)$-simplex such that $\boldsymbol{x}_1$ forms an obtuse angle with all other vertices, i.e., $(\boldsymbol{x}_j - \boldsymbol{x}_1)^\top(\boldsymbol{x}_i - \boldsymbol{x}_1) < 0$ for all $i \neq j$ with $i, j > 1$. Then the minimizer $\boldsymbol{h}^*_\rho$ of equation 8 is unique and is given by*

$$\boldsymbol{h}^*_\rho(\boldsymbol{y}) = \boldsymbol{x}_1 + \sum_{n=2}^{N} \boldsymbol{u}_n \phi_n(\boldsymbol{u}_n^\top(\boldsymbol{y} - \boldsymbol{x}_1)) \tag{36}$$

where $\boldsymbol{u}_n = \frac{\boldsymbol{x}_n - \boldsymbol{x}_1}{\|\boldsymbol{x}_n - \boldsymbol{x}_1\|}$, $\phi_n(t) = s_n([t - a_n]_+ - [t - b_n]_+)$, with $a_n = \rho$, $b_n = \|\boldsymbol{x}_n - \boldsymbol{x}_1\| - \rho$, and $s_n = \|\boldsymbol{x}_n - \boldsymbol{x}_1\|/(b_n - a_n)$ for all $n = 2, ..., N$.

Proposition 4. *Suppose the convex hull of the training points $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \in \mathbb{R}^d$ is an equilateral triangle. Assume the norm-balls $B_n := B(\boldsymbol{x}_n, \rho)$ centered at each training point have radius $\rho < \|\boldsymbol{x}_n - \boldsymbol{x}_0\|/2$, $n = 1, 2, 3$, where $\boldsymbol{x}_0 = \frac{1}{3}(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3)$ is the centroid of the triangle. Then a minimizer $\boldsymbol{h}^*_\rho$ of equation 8 is given by*

$$\boldsymbol{h}^*_\rho(\boldsymbol{y}) = \boldsymbol{u}_1\phi_1(\boldsymbol{u}_1^\top(\boldsymbol{y} - \boldsymbol{x}_0)) + \boldsymbol{u}_2\phi_2(\boldsymbol{u}_2^\top(\boldsymbol{y} - \boldsymbol{x}_0)) + \boldsymbol{u}_3\phi_3(\boldsymbol{u}_3^\top(\boldsymbol{y} - \boldsymbol{x}_0)) + \boldsymbol{x}_0, \tag{37}$$

where $\phi_n(t) = s_n([t - a_n]_+ - [t - b_n]_+)$ with $\boldsymbol{u}_n = \frac{\boldsymbol{x}_n - \boldsymbol{x}_0}{\|\boldsymbol{x}_n - \boldsymbol{x}_0\|}$, $a_n = -\frac{1}{2}\|\boldsymbol{x}_n - \boldsymbol{x}_0\| + \rho$, $b_n = \|\boldsymbol{x}_n - \boldsymbol{x}_0\| - \rho$, and $s_n = \|\boldsymbol{x}_n - \boldsymbol{x}_0\|/(b_n - a_n)$.

B.1 PROOF OF THEOREM 1

Proof. In the case of orthogonal dataset where for all $i \neq j$ $\boldsymbol{x}_i^\top \boldsymbol{x}_j = 0$ and $\boldsymbol{x}_0 = 0$, the score function is

$$\boldsymbol{s}\left(\boldsymbol{y}\right) = \frac{\boldsymbol{h}^*_\rho(\boldsymbol{y}) - \boldsymbol{y}}{\sigma^2} \tag{38}$$

$$= \frac{\sum_{i=1}^{N-1} \boldsymbol{e}_n \frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho}\left([y_i - \rho]_+ - [y_i - (\|\boldsymbol{x}_i\| - \rho)]_+\right) - \boldsymbol{y}}{\sigma^2} . \tag{39}$$

The Jacobian matrix is

$$J_{ij}\left(\boldsymbol{y}\right)=\frac{\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\|-2\rho}\Delta_i\left(\boldsymbol{y}\right)\delta_{i,j}-\delta_{i,j}}{\sigma^2}\,, \tag{40}$$

where $\Delta_n\left(\boldsymbol{y}\right)$ indicates if only one of the ReLU functions is activated. In matrix form, we obtain

$$\boldsymbol{J}\left(\boldsymbol{y}\right)=\frac{1}{\sigma^2}\left(\operatorname{diag}\left(\frac{\|\boldsymbol{x}_1\|}{\|\boldsymbol{x}_1\|-2\rho}\Delta_1\left(\boldsymbol{y}\right),\cdots,\frac{\|\boldsymbol{x}_{N-1}\|}{\|\boldsymbol{x}_{N-1}\|-2\rho}\Delta_{N-1}\left(\boldsymbol{y}\right)\right)-\boldsymbol{I}\right)\,, \tag{41}$$

where $\Delta_n\left(\boldsymbol{y}\right)\in\{0,1\}$. In this case, the stability condition is

$$\operatorname{Re}\{\lambda\left(\boldsymbol{J}\left(\boldsymbol{y}\right)\right)\}=\lambda\left(\boldsymbol{J}\left(\boldsymbol{y}\right)\right)<0\,. \tag{42}$$

Note that for $\Delta_i\left(\boldsymbol{y}\right)=1$

$$\lambda\left(\boldsymbol{J}\left(\boldsymbol{y}\right)\right)=\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\|-2\rho}\Delta_i\left(\boldsymbol{y}\right)-1>0\,. \tag{43}$$

Therefore, a stationary point is stable if and only if for all $i\in[N-1]$ $\Delta_i\left(\boldsymbol{y}\right)=0$. We define the set $\mathcal{A}=\{\sum_{n\in\mathcal{I}}\boldsymbol{x}_n|\mathcal{I}\in\mathcal{P}([N-1])\}$. Note that the set of points where the score is zero and $\Delta_i\left(\boldsymbol{y}\right)=0$ for all $i\in[N-1]$ is $\mathcal{A}$. $\square$

B.2 Proof of Theorem 2

Proof. We assume WLOG that for all $i\in[N-1]$ $\boldsymbol{u}_i=\boldsymbol{e}_i$. We can analyze the ODE equation 14 along each orthogonal direction separately. In each direction, we divide the ODE into the following cases:

If $y_i\leq\rho$ or $i>N-1$, the score function is

$$s_i\left(y_i\right)=-\frac{y_i}{\sigma^2}\,. \tag{44}$$

Therefore, according to Lemma 1,

$$(\boldsymbol{y}_t)_i=(\boldsymbol{y}_0)_i e^{-\frac{t}{\sigma^2}} \tag{45}$$

and we converge to zero.

If $y_i\geq\|\boldsymbol{x}_i\|-\rho$, the score function is

$$s_i\left(y_i\right)=\frac{\|\boldsymbol{x}_i\|-y_i}{\sigma^2}\,. \tag{46}$$

Therefore, according to Lemma 1,

$$(\boldsymbol{y}_t)_i=(\boldsymbol{y}_0)_i e^{-\frac{t}{\sigma^2}}+\|\boldsymbol{x}_i\|\left(1-e^{-\frac{t}{\sigma^2}}\right) \tag{47}$$

$$=\left(y_0-\|\boldsymbol{x}_i\|\right)e^{-\frac{t}{\sigma^2}}+\|\boldsymbol{x}_i\| \tag{48}$$

and we converge to $\|\boldsymbol{x}_i\|$.

Finally, if $\rho<y_i<\|\boldsymbol{x}_i\|-\rho$, the score function is

$$s_i\left(y_i\right)=\frac{1}{\sigma^2}\left(\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\|-2\rho}-1\right)y_i-\frac{\|\boldsymbol{x}_i\|\rho}{\|\boldsymbol{x}_i\|-2\rho}\right)\,. \tag{49}$$

Therefore, according to Lemma 1,

$$(\boldsymbol{y}_t)_i=(\boldsymbol{y}_0)_i e^{\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\|-2\rho}-1\right)\frac{t}{\sigma^2}}+\frac{\|\boldsymbol{x}_i\|}{2}\left(1-e^{\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\|-2\rho}-1\right)\frac{t}{\sigma^2}}\right) \tag{50}$$

$$=\left((\boldsymbol{y}_0)_i-\frac{\|\boldsymbol{x}_i\|}{2}\right)e^{\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\|-2\rho}-1\right)\frac{t}{\sigma^2}}+\frac{\|\boldsymbol{x}_i\|}{2}\,. \tag{51}$$

Here, if $(\boldsymbol{y}_0)_i = \frac{\|\boldsymbol{x}_i\|}{2}$ we converge to $\frac{\|\boldsymbol{x}_i\|}{2}$; if $(\boldsymbol{y}_0)_i > \frac{\|\boldsymbol{x}_i\|}{2}$ then we converge to $\|\boldsymbol{x}_i\|$; if $(\boldsymbol{y}_0)_i < \frac{\|\boldsymbol{x}_i\|}{2}$ then we converge to zero.

There are multiple initializations in which the closest point on the hyperbox is a point on the boundary which is not a vertex. We first consider the case where there exist a non empty set $\mathcal{I} \subset [N-1]$ such that for all $i \in \mathcal{I}$ $\rho < (\boldsymbol{y}_0)_i < \|\boldsymbol{x}_i\| - \rho$, and for all $j \in [N] \setminus \mathcal{I}$ $(\boldsymbol{y}_0)_j < \rho$ or $(\boldsymbol{y}_0)_j > \|\boldsymbol{x}_i\| - \rho$. We define $\Delta T_i(\rho)$ time to reach the edge of the partition, i.e. $\|\boldsymbol{x}_i\| - \rho$ (when $(\boldsymbol{y}_0)_i > \|\boldsymbol{x}_i\| - \rho$) starting from the initialization point, and $\Delta \tilde{T}_j(\rho, \epsilon)$ time to reach ϵ distance from zero or $\|\boldsymbol{x}_i\|$ starting from the initialization point:

$$\Delta T_i(\rho) = \sigma^2 \frac{\|\boldsymbol{x}_i\| - 2\rho}{2\rho} \log\left(\frac{\frac{\|\boldsymbol{x}_i\|}{2} - \rho}{(\boldsymbol{y}_0)_i - \frac{\|\boldsymbol{x}_i\|}{2}}\right) \tag{52}$$

$$\Delta \tilde{T}_j(\rho, \epsilon) = \sigma^2 \log\left(\frac{(\boldsymbol{y}_0)_i}{\epsilon}\right). \tag{53}$$

Since $\rho = \alpha\sigma$ we get that

$$\Delta T_i(\rho) = \rho \frac{\|\boldsymbol{x}_i\| - 2\rho}{2\alpha^2} \log\left(\frac{\frac{\|\boldsymbol{x}_i\|}{2} - \rho}{(\boldsymbol{y}_0)_i - \frac{\|\boldsymbol{x}_i\|}{2}}\right) \tag{54}$$

$$\Delta \tilde{T}_j(\rho, \epsilon) = \left(\frac{\rho}{\alpha}\right)^2 \log\left(\frac{(\boldsymbol{y}_0)_i}{\epsilon}\right). \tag{55}$$

Note that $\exists \rho_0(\epsilon) > 0$ such that $\forall \rho < \rho_0(\epsilon,)$

$$T_0 = \max_j \Delta \tilde{T}_j(\rho, \epsilon) < T < T_1 = \min_i \Delta T_i(\rho), \tag{56}$$

since $\exists \rho_0(\epsilon)$ such that

$$\left(\frac{\rho_0}{\alpha}\right)^2 \log\left(\frac{(\boldsymbol{y}_0)_i}{\epsilon}\right) < \rho_0 \frac{\|\boldsymbol{x}_i\| - 2\rho_0}{2\alpha^2} \log\left(\frac{\frac{\|\boldsymbol{x}_i\|}{2} - \rho_0}{(\boldsymbol{y}_0)_i - \frac{\|\boldsymbol{x}_i\|}{2}}\right) \tag{57}$$

$$\log\left(\frac{(\boldsymbol{y}_0)_i}{\epsilon}\right) < \frac{\|\boldsymbol{x}_i\| - 2\rho_0}{2\rho_0} \log\left(\frac{\frac{\|\boldsymbol{x}_i\|}{2} - \rho_0}{(\boldsymbol{y}_0)_i - \frac{\|\boldsymbol{x}_i\|}{2}}\right). \tag{58}$$

We can similarly derive the time interval during which $\boldsymbol{y}_T$ is at most ϵ distance from the boundary of the hyperbox and is not at a stationary point for additional initializations. Specifically, for all $i \in [N-1]$ $\rho < (\boldsymbol{y}_0)_i < \|\boldsymbol{x}_i\| - \rho$ is such an initialization point. □

B.3 Proof of Theorem 3

First, we prove the following lemma.

Lemma 1. *consider the following affine ODE*

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = a y_t + b \tag{59}$$

with initial point y_T, where $a \neq 0$. The solution is

$$y = e^{a(t-T)}\left(y_T - \frac{b}{a}\left(e^{-a(t-T)} - 1\right)\right). \tag{60}$$

Proof. We verify directly that this is indeed the solution, since

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = a e^{a(t-T)}\left(y_T - \frac{b}{a}\left(e^{-at} - 1\right)\right) + e^{a(t-T)} b e^{-a(t-T)} \tag{61}$$

$$= a e^{a(t-T)}\left(y_T - \frac{b}{a}\left(e^{-(t-T)t} - 1\right)\right) + b = a y_t + b \tag{62}$$

$$y_T = \left(y_T - \frac{b}{a}(1-1)\right) = y_T. \tag{63}$$

□

Next, we prove the main Theorem.

Proof. We assume WLOG that for all $i \in [N-1]$ $\boldsymbol{u}_i = \boldsymbol{e}_i$. We can analyze the score flow along each orthogonal direction separately. In each direction, we divide the ODE to the following cases:

If $i \notin [N-1]$, then equation 12 is

$$\frac{\mathrm{d}y_r}{\mathrm{d}r} = -y\,. \tag{64}$$

Note that the initial point is at $r_0 = -\log\sqrt{T}$. Using Lemma 1, we obtain

$$\left(\boldsymbol{y}_r\right)_i = \left(\boldsymbol{y}_T\right)_i e^{-1\left(r+\log\sqrt{T}\right)}\,. \tag{65}$$

Since $r = -\log\sqrt{t}$, we further obtain

$$\left(\boldsymbol{y}_t\right)_i = \left(\boldsymbol{y}_T\right)_i e^{\left(\log\sqrt{t}-\log\sqrt{T}\right)} = \left(\boldsymbol{y}_T\right)_i e^{\left(\log\sqrt{\frac{t}{T}}\right)} = \left(\boldsymbol{y}_T\right)_i \sqrt{\frac{t}{T}}\,. \tag{66}$$

Therefore, we obtain $\left(\boldsymbol{y}_0\right)_i = 0$.

We now consider now the case where $i \in [N-1]$.

In the case where $y_i < \rho_t$, equation 12 is

$$\frac{\mathrm{d}y_r}{\mathrm{d}r} = -y\,. \tag{67}$$

So, similarly to the previous case, we obtain $\left(\boldsymbol{y}_0\right)_i = 0$.

In the case where $y_i > \|x_i\| - \rho_t$, equation 12 is

$$\frac{\mathrm{d}y_r}{\mathrm{d}r} = \|\boldsymbol{x}_i\| - y\,. \tag{68}$$

Note that the initial point is at $r_0 = -\log\sqrt{T}$. Using Lemma 1 we obtain

$$\begin{aligned}
\left(\boldsymbol{y}_r\right)_i &= e^{-1\left(r+\log\sqrt{T}\right)}\left(\left(\boldsymbol{y}_T\right)_i + \|\boldsymbol{x}_i\|\left(e^{-1\left(r+\log\sqrt{T}\right)} - 1\right)\right) & (69)\\
&= \|\boldsymbol{x}_i\| + \left(\left(\boldsymbol{y}_T\right)_i - \|\boldsymbol{x}_i\|\right) e^{-1\left(r+\log\sqrt{T}\right)}\,. & (70)
\end{aligned}$$

Since $r = -\log\sqrt{t}$, we further obtain

$$\begin{aligned}
\left(\boldsymbol{y}_t\right)_i &= \|\boldsymbol{x}_i\| + \left(\left(\boldsymbol{y}_T\right)_i - \|\boldsymbol{x}_i\|\right) e^{\left(\log\sqrt{t}-\log\sqrt{T}\right)} = & (71)\\
&= \|\boldsymbol{x}_i\| + \left(\left(\boldsymbol{y}_T\right)_i - \|\boldsymbol{x}_i\|\right)\sqrt{\frac{t}{T}}\,. & (72)
\end{aligned}$$

Therefore, we obtain $\left(\boldsymbol{y}_0\right)_i = \|\boldsymbol{x}_i\|$.

In the case where $\rho_t < y_i < \|\boldsymbol{x}_i\| - \rho_t$, equation 12 is

$$\frac{\mathrm{d}y_r}{\mathrm{d}r} = \rho_{g_r^{-1}}\left(\frac{2}{\|\boldsymbol{x}_i\|}y - 1\right)\,. \tag{73}$$

Note that

$$\begin{aligned}
\rho_t &= \alpha\sigma_t = \alpha\sqrt{t} & (74)\\
g_r^{-1} &= e^{-2r}\,. & (75)
\end{aligned}$$

Therefore,

$$\rho_r = \alpha e^{-r} \tag{76}$$

so we obtain the following ODE:

$$\frac{\mathrm{d}y_r}{\mathrm{d}r} = \alpha e^{-r}\left(\frac{2}{\|\boldsymbol{x}_i\|}y - 1\right)\,. \tag{77}$$

Next, we apply additional time re-scaling

$$k = -\alpha e^{-r} \tag{78}$$
$$\frac{\mathrm{d}k}{\mathrm{d}r} = \alpha e^{-r} = \rho_r \tag{79}$$
$$\frac{\mathrm{d}r}{\mathrm{d}k} = \alpha^{-1} e^{r} = \rho_r^{-1} . \tag{80}$$

So, we get the following ODE:

$$\frac{\mathrm{d}y_r}{\mathrm{d}k} = \frac{\mathrm{d}y_r}{\mathrm{d}r}\frac{\mathrm{d}r}{\mathrm{d}k} = \alpha e^{-r}\left(\frac{2}{\|\boldsymbol{x}_i\|}y - 1\right)\alpha^{-1}e^{r} = \frac{2}{\|\boldsymbol{x}_i\|}y - 1 \tag{81}$$
$$\frac{\mathrm{d}y_k}{\mathrm{d}k} = \frac{2}{\|\boldsymbol{x}_i\|}y - 1 . \tag{82}$$

Note that the initial point is at $k_0 = -\alpha\sqrt{T}$. Using Lemma 1 we obtain

$$(\boldsymbol{y}_k)_i = e^{\frac{2}{\|\boldsymbol{x}_i\|}(k+\alpha\sqrt{T})}\left((\boldsymbol{y}_T)_i + \frac{\|\boldsymbol{x}_i\|}{2}\left(e^{-\frac{2}{\|\boldsymbol{x}_i\|}(k+\alpha\sqrt{T})} - 1\right)\right) \tag{83}$$
$$= \frac{\|x_i\|}{2} + \left((\boldsymbol{y}_T)_i - \frac{\|x_i\|}{2}\right)e^{\frac{2}{\|x_i\|}(k+\alpha\sqrt{T})} . \tag{84}$$

Since $k = -\alpha e^{-r}$ and $r = -\log\sqrt{t}$, we obtain

$$(\boldsymbol{y}_r)_i = \frac{\|x_i\|}{2} + \left((\boldsymbol{y}_T)_i - \frac{\|x_i\|}{2}\right)e^{\frac{2}{\|x_i\|}(-\alpha e^{-r}+\alpha\sqrt{T})} \tag{85}$$
$$(\boldsymbol{y}_t)_i = \frac{\|x_i\|}{2} + \left((\boldsymbol{y}_T)_i - \frac{\|x_i\|}{2}\right)e^{\frac{2}{\|x_i\|}(-\alpha\sqrt{t}+\alpha\sqrt{T})} . \tag{86}$$

So, we obtain $(\boldsymbol{y}_0)_i = \frac{\|x_i\|}{2} + \left((\boldsymbol{y}_T)_i - \frac{\|x_i\|}{2}\right)e^{\frac{2\alpha\sqrt{T}}{\|x_i\|}}$. Given an initialization point $\boldsymbol{y}_T$, let $\mathcal{I} \subseteq [N-1]$ be a non empty set such that $\rho < (\boldsymbol{y}_T)_i < \|\boldsymbol{x}_i\| - \rho$ for all $i \in \mathcal{I}$ and either $(\boldsymbol{y}_T)_i < \rho$ or $(\boldsymbol{y}_T)_i > \|\boldsymbol{x}_i\| - \rho$ for all $j \in [N-1] \setminus \mathcal{I}$. Then, if

$$T > \max_{i\in\mathcal{I}}\left(\frac{\|x_i\|}{2\alpha}\right)^2 \log^2\left(\frac{\frac{\|x_i\|}{2}}{(\boldsymbol{y}_T)_i - \frac{\|x_i\|}{2}}\right) , \tag{87}$$

we converge to the closest point in the set $\mathcal{A} = \{\sum_{n\in\mathcal{I}} \boldsymbol{x}_n \mid \mathcal{I} \subseteq [N-1]\}$ to the initialization point $\boldsymbol{y}_T$, where $\{\boldsymbol{x}_n\}_{n=0}^{N-1}$ is the training set. We instead converge to the closest boundary of the hyperbox to the initialization point $\boldsymbol{y}_T$ if

$$T < \max_{i\in\mathcal{I}}\left(\frac{\|x_i\|}{2\alpha}\right)^2 \log^2\left(\frac{\frac{\|x_i\|}{2}}{(\boldsymbol{y}_T)_i - \frac{\|x_i\|}{2}}\right) . \tag{88}$$

□

B.4 Proof of Theorem 4

Proof. In the case where the convex hull of the training points is an $(N-1)$-simplex, such that $\boldsymbol{x}_0$ forms an obtuse angle with all other vertices and $\boldsymbol{x}_0 = 0$, the score function is

$$\boldsymbol{s}(\boldsymbol{y}) = \frac{\boldsymbol{h}_\rho^*(\boldsymbol{y}) - \boldsymbol{y}}{\sigma^2} \tag{89}$$
$$= \frac{\sum_{n=1}^{N-1}\frac{\|\boldsymbol{x}_n\|}{\|\boldsymbol{x}_n\|-2\rho}\boldsymbol{u}_n\left([\boldsymbol{u}_n^\top\boldsymbol{y} - \rho]_+ - [\boldsymbol{u}_n^\top\boldsymbol{y} - (\|\boldsymbol{x}_n\| - \rho)]_+\right) - \boldsymbol{y}}{\sigma^2} . \tag{90}$$

The Jacobian matrix is

$$J_{ij}(\boldsymbol{y}) = \frac{\sum_{n=1}^{N-1}\frac{\|\boldsymbol{x}_n\|}{\|\boldsymbol{x}_n\|-2\rho}(u_n)_i(u_n)_j\Delta_n(\boldsymbol{y}) - \delta_{i,j}}{\sigma^2} , \tag{91}$$

where $\Delta_n(\boldsymbol{y})$ indicates if only one of the ReLU functions is activated. In matrix form we obtain

$$\boldsymbol{J}(\boldsymbol{y}) = \frac{1}{\sigma^2}\left(\boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{I}\right), \tag{92}$$

where

$$\boldsymbol{U} = \left(\Delta_1(\boldsymbol{y})\sqrt{\gamma_1}\boldsymbol{u}_1, \cdots, \Delta_{N-1}(\boldsymbol{y})\sqrt{\gamma_{N-1}}\boldsymbol{u}_{N-1}\right) \tag{93}$$

$$\gamma_n = \frac{\|\boldsymbol{x}_n\|}{\|\boldsymbol{x}_n\| - 2\rho} \tag{94}$$

$$\Delta_n(\boldsymbol{y}) \in \{0, 1\}. \tag{95}$$

Note that the Jacobian matrix is a real and symmetric matrix therefore it has real eigenvalues. In this case, the stability condition is

$$\mathrm{Re}\{\lambda(\boldsymbol{J}(\boldsymbol{y}))\} = \lambda(\boldsymbol{J}(\boldsymbol{y})) < 0. \tag{96}$$

For any $\boldsymbol{a} \in \mathbb{R}^d$

$$\boldsymbol{a}^\top \boldsymbol{J}(\boldsymbol{y})\boldsymbol{a} \leq \lambda_{\max}(\boldsymbol{J}(\boldsymbol{y}))\boldsymbol{a}^\top\boldsymbol{a}. \tag{97}$$

This holds in particular for $\boldsymbol{a} \in \mathcal{S}^{d-1}$, therefore

$$\lambda_{\max}(\boldsymbol{J}) \geq \boldsymbol{a}^\top \frac{1}{\sigma^2}\left(\boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{I}\right)\boldsymbol{a} \tag{98}$$

$$= \frac{1}{\sigma^2}\left(\left\|\boldsymbol{a}^\top\boldsymbol{U}\right\|_2^2 - 1\right). \tag{99}$$

If we choose $\boldsymbol{a} = \boldsymbol{u}_n$ such that $\Delta_n(\boldsymbol{y}) \neq 0$, then $\left\|\boldsymbol{a}^\top\boldsymbol{U}\right\|_2^2 > 1$, since $\gamma_n > 1$. Therefore, a stationary point is stable if and only if for all $n \in \{1, \cdots, N-1\}$ $\Delta_i(\boldsymbol{y}) = 0$. Note that if $\boldsymbol{y}$ is such that $\Delta_n(\boldsymbol{y}) = 0$ for all $n \in \{1, \cdots, N-1\}$, then there exists $\mathcal{I} \in \mathcal{P}(0, 1, \cdots, N-1)$ such that

$$f^*(\boldsymbol{y}) = \sum_{n\in\mathcal{I}} \boldsymbol{x}_n. \tag{100}$$

Therefore, $\boldsymbol{y}^* = \sum_{n\in\mathcal{I}} \boldsymbol{x}_n$ is a stationary point if and only if for all $i \in \{1, \cdots, N-1\}$ $\Delta_i(\boldsymbol{y}^*) = 0$. Note that the set of stable stationary points is not empty, since for all $i \in [N]$ the point $\boldsymbol{y}^* = \boldsymbol{x}_i$ is a stable stationary point because $\boldsymbol{f}^*(\boldsymbol{y}^*) = \boldsymbol{x}_i$, and thus $\Delta_n(\boldsymbol{y}^*) = 0$ for all $n \in \{1, \cdots, N-1\}$.

The condition for the point $\sum_{n\in\mathcal{I}} \boldsymbol{x}_n$ where $\mathcal{I} \subseteq [N]$ and $|\mathcal{I}| \geq 2$ if $0 \notin \mathcal{I}$ and $|\mathcal{I}| \geq 3$ if $0 \in \mathcal{I}$ to be a stable stationary point, is that for all $\forall k \in \mathcal{I}$

$$\sum_{i\in\mathcal{I}} \boldsymbol{u}_k^\top \boldsymbol{x}_i > \|\boldsymbol{x}_k\| - \rho, \tag{101}$$

which is equivalent to that for all $\forall k \in \mathcal{I}$

$$\sum_{i\in\mathcal{I}\setminus\{k\}} \boldsymbol{u}_k^\top \boldsymbol{x}_i > -\rho. \tag{102}$$

This set of inequality is equivalent to the condition

$$\min_{k\in\mathcal{I}} \left\{ \sum_{i\in\mathcal{I}\setminus\{k\}} \boldsymbol{u}_k^\top \boldsymbol{u}_i \|\boldsymbol{x}_i\| \right\} > -\rho. \tag{103}$$

□

B.5 Proof of Theorem 5

First, we prove the following lemma.

Lemma 2. *Consider the following system of affine ODE*

$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{y}_t + \boldsymbol{b}\,, \tag{104}$$

with the initial condition $\boldsymbol{y}_0$, where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is a non singular matrix. The solution is

$$\boldsymbol{y}_t = e^{\boldsymbol{A}t}\left(\boldsymbol{y}_0 - \boldsymbol{A}^{-1}\left(e^{-\boldsymbol{A}t} - \boldsymbol{I}\right)\boldsymbol{b}\right)\,. \tag{105}$$

In the case where $\boldsymbol{A}$ is also symmetric, the solution can be written as

$$\boldsymbol{y}_t = \sum_{i=1}^{d} \boldsymbol{v}_i\left(\boldsymbol{v}_i^\top \boldsymbol{y}_0\right)e^{\lambda_i t} - \sum_{i=1}^{d} \boldsymbol{v}_i\left(\boldsymbol{v}_i^\top \boldsymbol{b}\right)\lambda_i^{-1}\left(1 - e^{\lambda_i t}\right)\,, \tag{106}$$

where $\sum_{i=1}^{d}\lambda_i \boldsymbol{v}_i\boldsymbol{v}_i^\top$ is the eigenvalue decomposition of the matrix $\boldsymbol{A}$.

Proof. We verify directly that this is indeed the solution, since

$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = \boldsymbol{A}e^{\boldsymbol{A}t}\left(\boldsymbol{y}_0 - \boldsymbol{A}^{-1}\left(e^{-\boldsymbol{A}t} - \boldsymbol{I}\right)\boldsymbol{b}\right) + e^{\boldsymbol{A}t}e^{-\boldsymbol{A}t}\boldsymbol{b} = \boldsymbol{A}\boldsymbol{y}_t + \boldsymbol{b} \tag{107}$$

$$\boldsymbol{y}_0 = I\left(\boldsymbol{y}_0 - \boldsymbol{A}^{-1}\left(\mathbf{I} - \mathbf{I}\right)\boldsymbol{b}\right) = \boldsymbol{y}_0\,. \tag{108}$$

In the case where $\boldsymbol{A}$ is also symmetric,

$$e^{\boldsymbol{A}t} = \sum_{k=0}^{\infty}\frac{1}{k!}\left(\boldsymbol{A}t\right)^k = \boldsymbol{V}\left(\sum_{k=0}^{\infty}\frac{1}{k!}t^k\boldsymbol{\Lambda}^k\right)\boldsymbol{V}^\top \tag{109}$$

$$= \boldsymbol{V}\operatorname{diag}\left(e^{\lambda_1 t}, \cdots, e^{\lambda_d t}\right)\boldsymbol{V}^\top = \sum_{i=1}^{d} e^{\lambda_i t}\boldsymbol{v}_i\boldsymbol{v}_i^\top \tag{110}$$

$$e^{-\boldsymbol{A}t} = \sum_{i=1}^{d} e^{-\lambda_i t}\boldsymbol{v}_i\boldsymbol{v}_i^\top\,. \tag{111}$$

Therefore,

$$\boldsymbol{y}_t = e^{\boldsymbol{A}t}\left(\boldsymbol{y}_0 - \boldsymbol{A}^{-1}\left(e^{-\boldsymbol{A}t} - \boldsymbol{I}\right)\boldsymbol{b}\right) \tag{112}$$

$$= \sum_{i=1}^{d}\boldsymbol{v}_i\boldsymbol{v}_i^\top e^{\lambda_i t}\left(\boldsymbol{y}_0 - \sum_{k=1}^{d}\boldsymbol{v}_k\boldsymbol{v}_k^\top\lambda_i^{-1}\sum_{j=1}^{d}\boldsymbol{v}_j\boldsymbol{v}_j^\top\left(e^{-\lambda_j t} - 1\right)\boldsymbol{b}\right) \tag{113}$$

$$= \sum_{i=1}^{d}\boldsymbol{v}_i\boldsymbol{v}_i^\top e^{\lambda_i t}\left(\boldsymbol{y}_0 - \sum_{k=1}^{d}\boldsymbol{v}_k\lambda_k^{-1}\boldsymbol{v}_k^\top\left(e^{-\lambda_k t} - 1\right)\boldsymbol{b}\right) \tag{114}$$

$$= \sum_{i=1}^{d}\boldsymbol{v}_i\left(\boldsymbol{v}_i^\top\boldsymbol{y}_0\right)e^{\lambda_i t} - \sum_{i=1}^{d}\boldsymbol{v}_i\left(\boldsymbol{v}_i^\top\boldsymbol{b}\right)\lambda_i^{-1}\left(1 - e^{\lambda_i t}\right)\,. \tag{115}$$

□

Next, we prove Theorem 5.

Proof. We assume WLOG that $\boldsymbol{x}_0 = 0$. Given the initial point $\boldsymbol{y}_0$ such that $\boldsymbol{y}_0$ such that $\rho < \boldsymbol{u}_i^\top\boldsymbol{y}_0 < \|\boldsymbol{x}_i\| - \rho$ and $\boldsymbol{u}_j^\top\boldsymbol{y}_0 < \rho$ for all $j \neq i$, the score is given by

$$\boldsymbol{s}\left(\boldsymbol{y}\right) = \frac{1}{\sigma^2}\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho}\boldsymbol{u}_i\left(\boldsymbol{u}_i^\top\boldsymbol{y} - \rho\right) - \boldsymbol{y}\right) \tag{116}$$

$$= \frac{1}{\sigma^2}\left(\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho}\boldsymbol{u}_i\boldsymbol{u}_i^\top - \boldsymbol{I}\right)\boldsymbol{y} - \frac{\|\boldsymbol{x}_i\|\,\rho}{\|\boldsymbol{x}_i\| - 2\rho}\boldsymbol{u}_i\right)\,. \tag{117}$$

According to Lemma 2, the score flow in the partition $\rho < \boldsymbol{u}_i^\top \boldsymbol{y} < \|\boldsymbol{x}_i\| - \rho$ and $\boldsymbol{u}_j^\top \boldsymbol{y} < \rho$ for all $j \neq i$ is

$$\boldsymbol{y}_t = \sum_{k=1}^{d} \boldsymbol{v}_k \left(\boldsymbol{v}_k^\top \boldsymbol{y}_0\right) e^{\lambda_k \frac{t}{\sigma^2}} - \sum_{k=1}^{d} \boldsymbol{v}_k \left(\boldsymbol{v}_k^\top \boldsymbol{b}\right) \lambda_k^{-1} \left(1 - e^{\lambda_k \frac{t}{\sigma^2}}\right), \tag{118}$$

where the matrix $\boldsymbol{A} = \left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho} \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I}\right)$. The eigenvalue decomposition of $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^\top \tag{119}$$

$$\mathbf{V} = \begin{pmatrix} \boldsymbol{u}_i & \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_{d-1} \end{pmatrix} \tag{120}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\frac{2\rho}{\|\boldsymbol{x}_i\| - 2\rho}, -1, \cdots, -1\right), \tag{121}$$

where $\boldsymbol{w}_j \in \boldsymbol{u}_i^\perp$. Since,

$$\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho} \boldsymbol{u}_i \mathbf{u}_i^\top - \boldsymbol{I}\right) \boldsymbol{u}_i = \left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho} - 1\right) \boldsymbol{u}_i \tag{122}$$

$$= \frac{2\rho}{\|\boldsymbol{x}_i\| - 2\rho} \boldsymbol{u}_i \tag{123}$$

$$\left(\frac{\|\boldsymbol{x}_i\|}{\|\boldsymbol{x}_i\| - 2\rho} \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I}\right) \boldsymbol{w}_j = -\boldsymbol{w}_j, \tag{124}$$

and $\boldsymbol{b} = -\frac{\|\boldsymbol{x}_i\|\rho}{\|\boldsymbol{x}_i\| - 2\rho} \boldsymbol{u}_i$. So, we get

$$\boldsymbol{y}_t = \boldsymbol{u}_i \left(\left(\boldsymbol{u}_i^\top \boldsymbol{y}_0\right) e^{\frac{2\rho}{\|\boldsymbol{x}_i\| - 2\rho} \frac{t}{\sigma^2}} + \frac{\|\boldsymbol{x}_i\|}{2} \left(1 - e^{\frac{2\rho}{\|\boldsymbol{x}_i\| - 2\rho} \frac{t}{\sigma^2}}\right)\right) + \sum_{k=2}^{d} \boldsymbol{v}_k \left(\boldsymbol{v}_k^\top \boldsymbol{y}_0\right) e^{-\frac{t}{\sigma^2}}. \tag{125}$$

Note that we can analyze the score flow along each orthogonal direction separately. Next, we divide it into the following cases:

If $\boldsymbol{u}_i^\top \boldsymbol{y}_0 = \frac{\|\boldsymbol{x}_i\|}{2}$, then

$$\boldsymbol{y}_t = \boldsymbol{u}_i \frac{\|\boldsymbol{x}_i\|}{2} + \sum_{k=2}^{d} \boldsymbol{v}_k \left(\boldsymbol{v}_k^\top \boldsymbol{y}_0\right) e^{-\frac{t}{\sigma^2}}. \tag{126}$$

Therefore, we converge to the point $\boldsymbol{y}_\infty = \boldsymbol{u}_i \frac{\|\boldsymbol{x}_i\|}{2}$.

If $\boldsymbol{u}_i^\top \boldsymbol{y}_0 > \frac{\|\boldsymbol{x}_i\|}{2}$, then we converge to $\boldsymbol{y}_\infty = \boldsymbol{u}_i \|\boldsymbol{x}_i\|$, and if $\boldsymbol{u}_i^\top \boldsymbol{y}_0 < \frac{\|\boldsymbol{x}_i\|}{2}$ then we converge to $\boldsymbol{y}_\infty = \boldsymbol{x}_1 = 0$ (since then the score function is $\frac{\|\boldsymbol{x}_i\| - \boldsymbol{y}}{\sigma^2}$ or $-\frac{\boldsymbol{y}}{\sigma^2}$).

We assume WLOG that $\boldsymbol{u}_i^\top \boldsymbol{y}_0 > \frac{\|\boldsymbol{x}_i\|}{2}$. We define $\Delta T_{u_i}(\rho)$ time to reach the edge of the partition, i.e. $\|\boldsymbol{x}_i\| - \rho$ starting from the initialization point, and $\Delta T_{v_k}(\rho, \epsilon)$ time to reach ϵ distance from zero (the data manifold) starting from the initialization point.

$$\Delta T_{u_i}(\rho) = \sigma^2 \frac{\|\boldsymbol{x}_i\| - 2\rho}{2\rho} \log\left(\frac{\frac{\|\boldsymbol{x}_i\|}{2} - \rho}{\boldsymbol{u}_i^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}_i\|}{2}}\right) \tag{127}$$

$$\Delta T_{v_k}(\rho, \epsilon) = \sigma^2 \log\left(\frac{\boldsymbol{v}_k^\top \boldsymbol{y}_0}{\epsilon}\right). \tag{128}$$

Since $\rho = \alpha\sigma$, we get that

$$\Delta T_{u_i}(\rho) = \rho \frac{\|\boldsymbol{x}_i\| - 2\rho}{2\alpha^2} \log\left(\frac{\frac{\|\boldsymbol{x}_i\|}{2} - \rho}{\boldsymbol{u}_i^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}_i\|}{2}}\right) \tag{129}$$

$$\Delta T_{v_k}(\rho, \epsilon) = \left(\frac{\rho}{\alpha}\right)^2 \log\left(\frac{\boldsymbol{v}_k^\top \boldsymbol{y}_0}{\epsilon}\right). \tag{130}$$

Similarly to B.2, we get that $\exists \rho_0(\epsilon) > 0$ such that $\forall \rho < \rho_0(\epsilon,)$

$$T_0 = \max_k \Delta T_{v_k}(\epsilon) < T < \Delta T_{u_i}(\rho). \tag{131}$$

□

B.6 Proof of Theorem 6

Proof. The estimated score function at the initialization is

$$\sigma_t^2 \boldsymbol{s}(\boldsymbol{y}, t) = \left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \right) \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I} \right) \boldsymbol{y} - \rho_t \boldsymbol{u}_i \,. \tag{132}$$

Next, we project the estimated score along $\boldsymbol{u}_i$ and the orthogonal direction, so we get

$$\boldsymbol{u}_i \boldsymbol{u}_i^\top \sigma_t^2 \boldsymbol{s}(\boldsymbol{y}, t) = \left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \right) \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{u}_i \boldsymbol{u}_i^\top \right) \boldsymbol{y} - \rho_t \boldsymbol{u}_i \tag{133}$$

$$= \boldsymbol{u}_i \rho_t \left(\frac{2}{\|\boldsymbol{x}_i\|} \boldsymbol{u}_i^\top \boldsymbol{y} - 1 \right) \tag{134}$$

$$\left(\boldsymbol{I} - \boldsymbol{u}_i \boldsymbol{u}_i^\top \right) \sigma_t^2 \boldsymbol{s}(\boldsymbol{y}, t) = \left(\boldsymbol{I} - \boldsymbol{u}_i \boldsymbol{u}_i^\top \right) \left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \right) \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I} \right) \boldsymbol{y} - \rho_t \left(\boldsymbol{I} - \boldsymbol{u}_i \boldsymbol{u}_i^\top \right) \boldsymbol{u}_i \tag{135}$$

$$= \left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \right) \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I} \right) \boldsymbol{y} - \left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \right) \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{u}_i \boldsymbol{u}_i^\top \right) \boldsymbol{y} \tag{136}$$

$$= \left(\left(1 + \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \right) \boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I} \right) \boldsymbol{y} - \frac{2}{\|\boldsymbol{x}_i\|} \rho_t \boldsymbol{u}_i \boldsymbol{u}_i^\top \tag{137}$$

$$= \left(\boldsymbol{u}_i \boldsymbol{u}_i^\top - \boldsymbol{I} \right) \boldsymbol{y} \,. \tag{138}$$

Therefore, the projected score onto $\boldsymbol{u}_i$ is $\frac{\rho_t \left(\frac{2}{\|\boldsymbol{x}_i\|} \boldsymbol{u}_i^\top \boldsymbol{y} - 1 \right)}{\sigma_t^2}$, and the projected score function onto $\boldsymbol{w}_j \in \boldsymbol{u}_i^\perp$ is $-\frac{\boldsymbol{w}_j^\top y}{\sigma_t^2}$, so we get the same estimated score as in Theorem 3 (we can analyze the score flow along each orthogonal direction separately). Therefore, along $\boldsymbol{w}_j$ we get

$$\boldsymbol{w}_j^\top \boldsymbol{y}_t = \boldsymbol{w}_j^\top \boldsymbol{y}_T e^{\left(\log \sqrt{t} - \log \sqrt{T} \right)} = \boldsymbol{w}_j^\top \boldsymbol{y}_T e^{\left(\log \sqrt{\frac{t}{T}} \right)} = \left(\boldsymbol{y}_T \right)_i \sqrt{\frac{t}{T}} \,. \tag{139}$$

So, we obtain $\boldsymbol{w}_j^\top \boldsymbol{y}_0 = 0$. Along $\boldsymbol{u}_i$ we get

$$\boldsymbol{u}_i^\top \boldsymbol{y}_t = \frac{\|x_i\|}{2} + \left(\boldsymbol{u}_i^\top \boldsymbol{y}_T - \frac{\|x_i\|}{2} \right) e^{\frac{2}{\|x_i\|} \left(-\alpha \sqrt{t} + \alpha \sqrt{T} \right)} \,, \tag{140}$$

so we obtain $\boldsymbol{w}_j^\top \boldsymbol{y}_0 = \frac{\|x_i\|}{2} + \left(\boldsymbol{u}_i^\top \boldsymbol{y}_T - \frac{\|x_i\|}{2} \right) e^{\frac{2\alpha\sqrt{T}}{\|x_i\|}}$. Then, if

$$T \geq \left(\frac{\|x_i\|}{2\alpha} \right)^2 \log^2 \left(\frac{\frac{\|x_i\|}{2}}{\left(\boldsymbol{y}_T \right)_i - \frac{\|x_i\|}{2}} \right) \,, \tag{141}$$

we converge to the closest point in the set $\{\boldsymbol{x}_0, \boldsymbol{x}_i\}$ to the initialization point $\boldsymbol{y}_T$ since the estimated score is equal to $-\frac{\boldsymbol{y}}{\sigma_t^2}$ or $\frac{\|\boldsymbol{x}_i\| - \boldsymbol{y}}{\sigma_t^2}$ and we converge to 0 or $\|\boldsymbol{x}_i\|$ (as in Theorem 3), and if

$$T < \left(\frac{\|x_i\|}{2\alpha} \right)^2 \log^2 \left(\frac{\frac{\|x_i\|}{2}}{\left(\boldsymbol{y}_T \right)_i - \frac{\|x_i\|}{2}} \right) \,, \tag{142}$$

we converge to a point on the line connecting $\boldsymbol{x}_0$ and $\boldsymbol{x}_i$. □

B.7 Poof of Proposition 2

Proof. We assume WLOG that $\boldsymbol{x}_0 = 0$. Note that since the convex hull of the training points is an equilateral triangle, then $\|x_i\| = \|x\|$. Given the initial point $\boldsymbol{y}_0$ such that $i \in \{1, 2\} - \frac{\|\boldsymbol{x}\|}{2} + \rho <$

$\boldsymbol{u}_i^\top \boldsymbol{y} < \|\boldsymbol{x}\| - \rho$ and $\boldsymbol{u}_3^\top \boldsymbol{y} < -\frac{\|\boldsymbol{x}\|}{2} + \rho$, the score is given by

$$\boldsymbol{s}(\boldsymbol{y}) = \frac{1}{\sigma^2}\left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\sum_{i=1}^{2}\left(\boldsymbol{u}_i\boldsymbol{u}_i^\top \boldsymbol{y} + \frac{1}{2}\boldsymbol{x}_i - \boldsymbol{u}_i\rho\right) - \mathbf{y}\right) \tag{143}$$

$$= \frac{1}{\sigma^2}\left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\boldsymbol{u}_1\boldsymbol{u}_1^\top + \boldsymbol{u}_2\boldsymbol{u}_2^\top\right) - \boldsymbol{I}\right)\boldsymbol{y} \tag{144}$$

$$+ \frac{1}{\sigma^2}\left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\frac{1}{2}\|\boldsymbol{x}\| - \rho\right)\boldsymbol{u}_1 + \frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\frac{1}{2}\|\boldsymbol{x}\| - \rho\right)\boldsymbol{u}_2\right). \tag{145}$$

According to Lemma 2, the score flow in the partition $i \in \{1,2\}$ $-\frac{\|\boldsymbol{x}\|}{2} + \rho < \boldsymbol{u}_i^\top \boldsymbol{y} < \|\boldsymbol{x}\| - \rho$ and $\boldsymbol{u}_3^\top \boldsymbol{y} < -\frac{\|\boldsymbol{x}\|}{2} + \rho$ is

$$\boldsymbol{y}_t = \sum_{k=1}^{2}\boldsymbol{v}_k\left(\boldsymbol{v}_k^\top \boldsymbol{y}_0\right)e^{\lambda_k\frac{t}{\sigma^2}} - \sum_{k=1}^{2}\boldsymbol{v}_k\left(\boldsymbol{v}_k^\top \boldsymbol{b}\right)\lambda_k^{-1}\left(1 - e^{\lambda_k\frac{t}{\sigma^2}}\right), \tag{146}$$

where the matrix $\boldsymbol{A} = \left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\boldsymbol{u}_1\boldsymbol{u}_1^\top + \boldsymbol{u}_2\boldsymbol{u}_2^\top\right) - \boldsymbol{I}\right)$. The eigenvalue decomposition of $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top \tag{147}$$

$$\mathbf{V} = \begin{pmatrix}\frac{\boldsymbol{u}_1 - \boldsymbol{u}_2}{\sqrt{2\left(1 - \boldsymbol{u}_1^\top \boldsymbol{u}_2\right)}} & \frac{\boldsymbol{u}_1 + \boldsymbol{u}_2}{\sqrt{2\left(1 + \boldsymbol{u}_1^\top \boldsymbol{u}_2\right)}}\end{pmatrix} \tag{148}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(1 - \boldsymbol{u}_1^\top \boldsymbol{u}_2\right) - 1, \frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(1 + \boldsymbol{u}_1^\top \boldsymbol{u}_2\right) - 1\right), \tag{149}$$

since,

$$\left(\frac{\|\mathbf{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\boldsymbol{u}_1\boldsymbol{u}_1^\top + \boldsymbol{u}_2\boldsymbol{u}_2^\top\right) - \boldsymbol{I}\right)(\boldsymbol{u}_1 - \boldsymbol{u}_2) = \frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\boldsymbol{u}_2^\top \boldsymbol{u}_1 - \boldsymbol{u}_1\boldsymbol{u}_1^\top \boldsymbol{u}_2 - \boldsymbol{u}_2\right) - (\boldsymbol{u}_1 - \boldsymbol{u}_2) \tag{150}$$

$$= \left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(1 - \boldsymbol{u}_2^\top \boldsymbol{u}_1\right) - 1\right)(\boldsymbol{u}_1 - \boldsymbol{u}_2) \tag{151}$$

$$\left(\frac{\|\mathbf{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\boldsymbol{u}_1\boldsymbol{u}_1^\top + \boldsymbol{u}_2\boldsymbol{u}_2^\top\right) - \boldsymbol{I}\right)(\boldsymbol{u}_1 + \boldsymbol{u}_2) = \frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\boldsymbol{u}_2^\top \boldsymbol{u}_1 + \boldsymbol{u}_1\boldsymbol{u}_1^\top \boldsymbol{u}_2 + \boldsymbol{u}_2\right) - (\boldsymbol{u}_1 + \boldsymbol{u}_2) \tag{152}$$

$$= \left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(1 + \boldsymbol{u}_2^\top \boldsymbol{u}_1\right) - 1\right)(\boldsymbol{u}_1 + \boldsymbol{u}_2), \tag{153}$$

and $\boldsymbol{b} = \frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\frac{1}{2}\|\boldsymbol{x}\| - \rho\right)\boldsymbol{u}_1 + \frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\left(\frac{1}{2}\|\boldsymbol{x}\| - \rho\right)\boldsymbol{u}_2$. We assume WLOG that,

$$\boldsymbol{u}_1 = \begin{pmatrix}0\\1\end{pmatrix}, \quad \boldsymbol{u}_2 = \begin{pmatrix}\frac{\sqrt{3}}{2}\\-\frac{1}{2}\end{pmatrix}, \quad \boldsymbol{u}_3 = \begin{pmatrix}-\frac{\sqrt{3}}{2}\\-\frac{1}{2}\end{pmatrix}, \tag{154}$$

and we get

$$\boldsymbol{v}_1 = \frac{1}{\sqrt{3}}(\boldsymbol{u}_1 - \boldsymbol{u}_2) \tag{155}$$

$$\boldsymbol{v}_2 = \boldsymbol{u}_1 + \boldsymbol{u}_2 = -\boldsymbol{u}3 \tag{156}$$

$$\lambda_1 = \frac{\frac{3}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho} - 1 > 0 \tag{157}$$

$$\lambda_2 = -\left(1 - \frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\| - 2\rho}\right) < 0\,. \tag{158}$$

$$\boldsymbol{y}_t = \frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)\left(\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0\right) e^{\left(\frac{\frac{3}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1\right)\frac{\frac{t}{\sigma^2}}{\sigma^2}} \tag{159}$$

$$+ \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\left(\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0\right) e^{-\left(1-\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\right)\frac{t}{\sigma^2}} \tag{160}$$

$$- \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\frac{1}{2}\|\boldsymbol{x}\|-\rho\right)\left(\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1\right)^{-1}\left(1-e^{-\left(1-\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\right)\frac{t}{\sigma^2}}\right). \tag{161}$$

Note that,

$$\left(\frac{\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\left(\frac{1}{2}\|\boldsymbol{x}\|-\rho\right)\right)\left(\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1\right)^{-1} = \frac{\|\boldsymbol{x}\|\left(\frac{1}{2}\|\boldsymbol{x}\|-\rho\right)}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\frac{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}{-\|\boldsymbol{x}\|+2\rho} \tag{162}$$

$$= \frac{\|\boldsymbol{x}\|\left(\frac{1}{2}\|\boldsymbol{x}\|-\rho\right)}{-\|\boldsymbol{x}\|+2\rho} = -\frac{\|\boldsymbol{x}\|}{2}. \tag{163}$$

Therefore,

$$\boldsymbol{y}_t = \frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)\left(\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0\right) e^{\left(\frac{\frac{3}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1\right)\frac{t}{\sigma^2}} \tag{164}$$

$$+ \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\left(\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0\right) e^{-\left(1-\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\right)\frac{t}{\sigma^2}} \tag{165}$$

$$- \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\left(-\frac{\|\boldsymbol{x}\|}{2}\right)\left(1-e^{-\left(1-\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\right)\frac{t}{\sigma^2}}\right) \tag{166}$$

$$= \frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)\left(\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0\right) e^{\left(\frac{\frac{3}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1\right)\frac{t}{\sigma^2}} \tag{167}$$

$$+ \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\left(\left(\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}\|}{2}\right) e^{-\left(1-\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\right)\frac{t}{\sigma^2}} + \frac{\|\boldsymbol{x}\|}{2}\right). \tag{168}$$

Note that we can analyze the score flow along each orthogonal direction separately. Next, we divide it into the following cases:

If $\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 = 0$, then

$$\boldsymbol{y}_t = \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\left(\left(\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}\|}{2}\right) e^{-\left(1-\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}\right)\frac{t}{\sigma^2}} + \frac{\|\boldsymbol{x}\|}{2}\right), \tag{169}$$

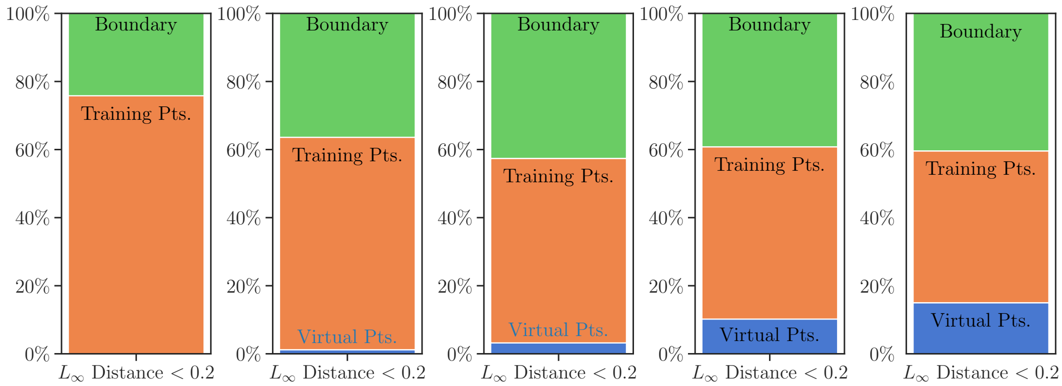
and we converge to the point $\boldsymbol{y}_\infty = \left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)\frac{\|\boldsymbol{x}\|}{2}$.

If $\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 > 0$, then we converge to $\boldsymbol{y}_\infty = \boldsymbol{x}_1$, and if $\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 < 0$, then we converge to $\boldsymbol{y}_\infty = \boldsymbol{x}_2$.

We assume WLOG that $\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 > 0$. We define $\Delta T_d\left(\rho, \epsilon\right)$ as the time to reach ϵ distance from the data manifold (the line connecting the training points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) starting from initialization point $\boldsymbol{y}_0$, and $\Delta T_e\left(\rho\right)$ the time to reach the edge of the partition starting from initialization point $\boldsymbol{y}_0$. We assume WLOG that $\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 > \frac{\|\boldsymbol{x}\|}{2}$ and $\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}\|}{2} > \epsilon$

$$\Delta T_d\left(\rho, \epsilon\right) = \frac{\sigma^2}{\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1}\log\left(\frac{\epsilon}{\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}\|}{2}}\right) \tag{170}$$

$$\Delta T_e\left(\rho\right) = \frac{\sigma^2}{\frac{\frac{3}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho}-1}\log\left(\frac{\frac{1}{2}\|\boldsymbol{x}\|-\rho}{\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0}\right). \tag{171}$$



(a) $N = 10$. (b) $N = 15$. (c) $N = 20$. (d) $N = 25$. (e) $N = 30$.

Figure 4: **Convergence types frequency of randomly sampled points in diffusion sampling for different N.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$ for diffusion sampling, using different training set sizes, N. We plot the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points. The generalization increases with N, drawing a larger percentage of samples to converge in the vicinity of virtual points and the boundaries of the hyperbox.

Since $\rho = \alpha\sigma$, we get that

$$\Delta T_d\left(\rho, \epsilon\right) = \frac{\rho^2}{\alpha^2\left(\frac{\frac{1}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho} - 1\right)} \log\left(\frac{\epsilon}{\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0 - \frac{\|\boldsymbol{x}\|}{2}}\right) \tag{172}$$

$$\Delta T_e\left(\rho\right) = \frac{\rho^2}{\alpha^2\left(\frac{\frac{3}{2}\|\boldsymbol{x}\|}{\frac{3}{2}\|\boldsymbol{x}\|-2\rho} - 1\right)} \log\left(\frac{\frac{1}{2}\|\boldsymbol{x}\| - \rho}{\frac{1}{\sqrt{3}}\left(\boldsymbol{u}_1 - \boldsymbol{u}_2\right)^\top \boldsymbol{y}_0}\right). \tag{173}$$

Similar to B.2 we get that $\exists \rho_0\left(\epsilon\right) > 0$ such that $\forall \rho < \rho_0\left(\epsilon,\right)$

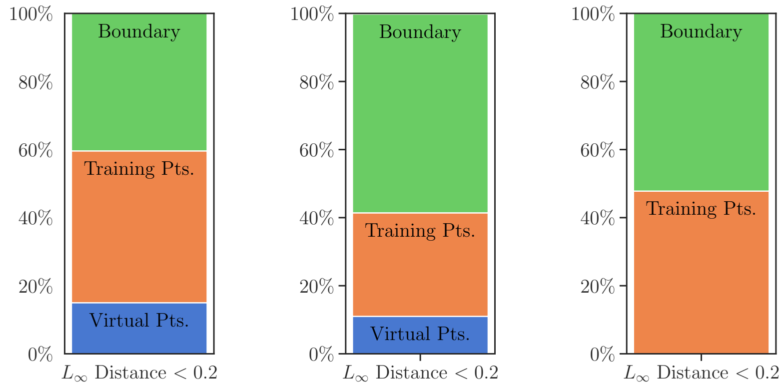
$$T_0 = \Delta T_d\left(\rho, \epsilon\right) < T < T_1 = \Delta T_e\left(\rho\right) . \tag{174}$$

□

C The Effect of the Number of Training Samples

The effect of the training set size has been explored in several past works (Somepalli et al., 2023; Kadkhodaie et al., 2024), as explored in detail in Section 6. Here we continue the analysis from Section 5 to investigate the effect of changing N, the training set size, on the full dynamics of the diffusion process with the probability ODE. Specifically, we repeat the experiment from Section 5 while reducing N. All the hyperparameters are kept the same, except for M which we increase to 2000 for $N = 10$ only, to prevent over-fitting in the large-noise regime. Figure 4 shows the percentage of points that converged within an L_∞ distance of 0.2 to either virtual points, training points, or a boundary of the hyperbox, for the different N values. The generalization increases with N, drawing a larger percentage of samples to converge in the vicinity of virtual points, or to boundaries of the hyperbox. This aligns with the results of Kadkhodaie et al. (2024).

When considering the effect of oversampling duplications, previous works observed that diffusion models tend to overfit more to duplicate training points than to other training points (Somepalli et al., 2023). However, here we study the regime in which the model perfectly fits all the training points. In practice, if duplicate training points would cause the neural network to fit them better, at the expense of the other training points. Then, we expect our analysis to effectively hold, but only for the training points that are well-fitted and their associated virtual points. Therefore, this mirrors the case of decreasing N, and will cause more convergence to the duplicated training points and increase memorization.



(a) AL method, $\lambda = 1$. (b) Weight decay, $\lambda = 0.25$. (c) No weight decay.

Figure 5: **Convergence types frequency of randomly sampled points in diffusion sampling for training with AL method, weight decay, and without weight decay.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$ for diffusion sampling, using different training configurations. We plot the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points. The minimum norm constraint is necessary for inducing the bias towards virtual training points and the boundaries of the hyperbox. Additionally, standard training protocol using weight decay regularization simulates well the minimum norm denoiser, which is achieved by the use of the AL method.

D The Minimum Norm Assumption

Theorems 2, 3, 4, 5 and 6 all hold in the case of a minimum norm denoiser, in which the denoiser achieves exact interpolation over the noisy training samples. To enforce a consistent denoiser, we used a non-standard training protocol in Section 5. Specifically, we optimize an equality constrained optimization problem using the Augmented Lagrangian method. Here we verify the the robustness of our results and the necessity of the minimum norm assumption by repeating the experiment from Section 5 when using standard training, with and without the use of weight decay. Specifically, all the hyper parameters and Adam optimizer are kept the same, and only the loss function changes to directly optimize equation 3. Training with weight decay should result in a denoiser that is similar to the min-norm solution. Figure 5 shows the percentage of points that converged within an L_∞ distance of 0.2 to either virtual points, training points, or a boundary of the hyperbox, for the different training configurations. The use of weight decay in a standard training protocol induces a similar bias to that achieved by the using Augmented Lagrangian method.

E Additional simulations

Figure 1 shows the normalized score flow for the case of an obtuse 2-simplex. The normalization was done for visualization purposes only, since the norm of the score decreases as it approaches the ReLU boundaries. In Figure 6 we illustrate the unnormalized score flow. Figure 7 shows the trajectory of score flow of the exact score function, and the green line is trajectory of the score flow of the approximated score function as can be seen the trajectories are practically identical.

We next repeat the statistical analysis done in Section 5 for different thresholds. Figure 8 demonstrates the existence of virtual points, in an analogous way to Figure 2, for the L_2 metric. Figures 9 and 10 offer additional insights to the right side of Figure 3a. Specifically, in Figure 9 we compare the results of the convergence types frequency of randomly sampled points with score flow when using different thresholds of the L_∞ distance. In Figure 10 we instead use the L_2 metric. Similarly, Figures 11 and 12 depict additional comparisons to the right side of Figure 3b, for both the L_∞ and L_2 distance metrics.

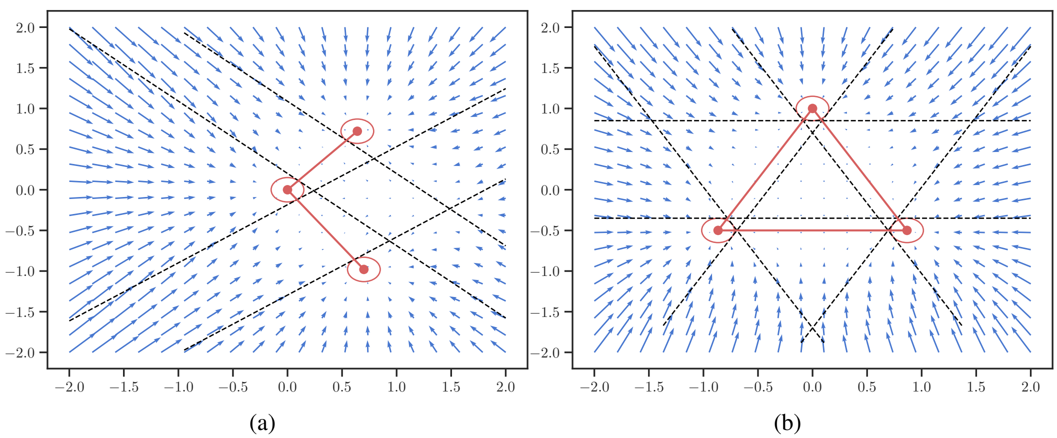


Figure 6: **The score function of obtuse and acute simplex**. The red dots are the training points $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$. The black lines are the ReLU boundaries. In figure (a) we plot the score function of obtuse simplex (Proposition 3). In figure (b) we plot acute simplex (Proposition 4)

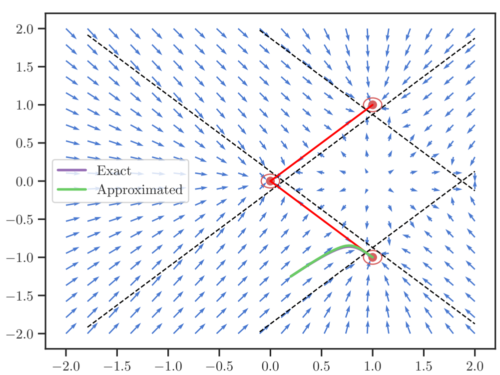


Figure 7: **The score function of orthogonal dataset**. The purple line is the trajectory of the score flow of the exact score function, and the green line is the trajectory of the score flow of the approximated score function (equation 18) in the case where $\sigma = 0.03, \rho = 0.09$. Both trajectories are very similar.

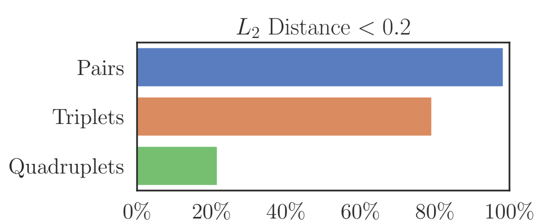


Figure 8: **Existence of stable virtual training points.** We run fixed-point iterations on a single denoiser, starting from all possible pair-wise, triplet-wise, and quadruplet-wise combinations of training samples. The plot shows the percentage of points that converged within an L_2 distance of 0.2 to the original, virtual, input point.

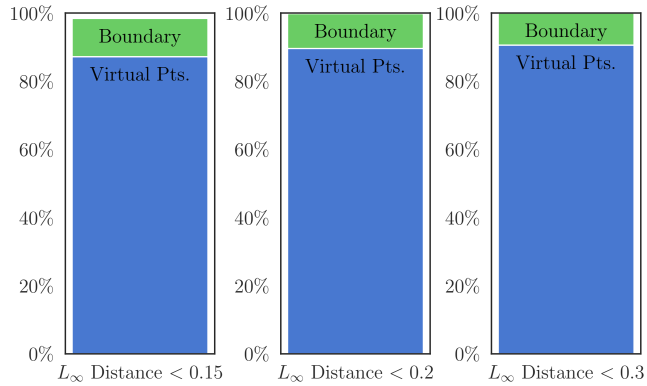


Figure 9: **Convergence types frequency of randomly sampled points for score flow based on L_∞ proximity.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$ for sampling using the score flow. We plot the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points, based on their L_∞ proximity for different thresholds.

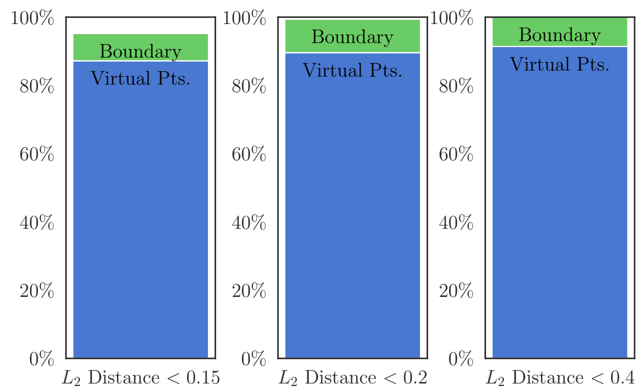


Figure 10: **Convergence types frequency of randomly sampled points for score flow based on L_2 proximity.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$ for sampling using the score flow. We plot the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points, based on their L_2 proximity for different thresholds.

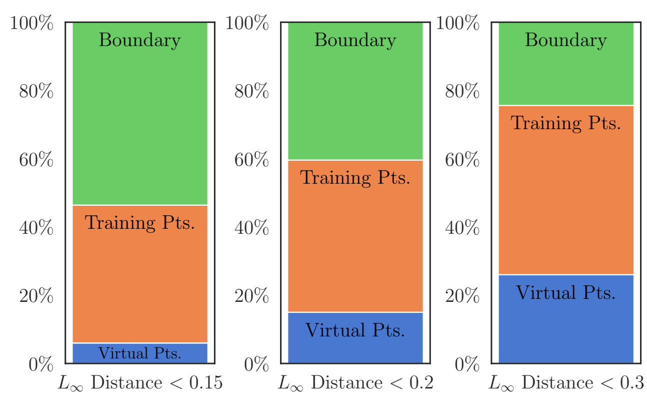


Figure 11: **Convergence types frequency of randomly sampled points for probability flow based on L_∞ proximity.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$ for probability flow. We plot the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points, based on their L_∞ proximity for different thresholds.

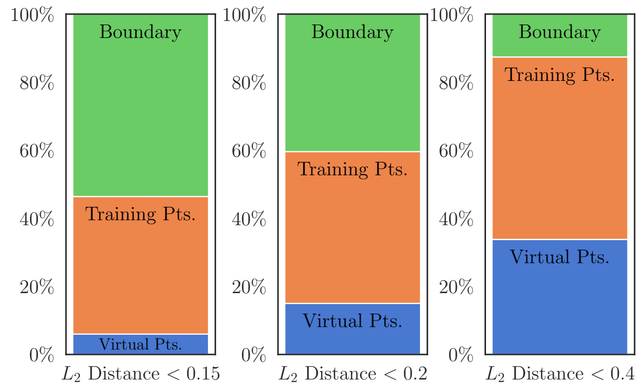


Figure 12: **Convergence types frequency of randomly sampled points for probability flow based on L_2 proximity.** We run the discrete ODE formulation of equation 21 for 500 randomly sampled points from $\mathbb{R}^{30}$ for probability flow. We plot the percentage of points that converged to either a virtual point, a training point, or to the boundaries of the hyperbox, out of all points, based on their L_2 proximity for different thresholds.