Data Generation without Function Estimation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Estimating the score function—or other population-density-dependent functions—is a fundamental component of most generative models. However, such function estimation is computationally and statistically challenging. *Can we avoid function estimation for data generation?* We propose an **estimation-free** generative method: *A set of points* whose locations are *deterministically* updated with (inverse) *gradient descent* can transport a uniform distribution to arbitrary data distribution, in the mean field regime, **without function estimation, training neural networks, and even noise injection**. The proposed method is built upon recent advances in the physics of interacting particles. Leveraging recent advances in mathematical physics, we prove that the proposed method samples from the true underlying data distribution in the asymptotic regime, without making any structural assumptions on the distribution.

1 Introduction

Given i.i.d. samples from an unknown distribution, how to generate a new sample from the distribution? Existing generative models often rely on estimating functions depending on population density such as the score function. Such an estimation is both statistically and computationally challenging. Computationally, estimating the score function even for a simple Gaussian mixture model with maximum likelihood estimation of parameters is NP-hard (Arora et al., 2001). Statistically, score estimation suffers the curse of dimensionality (Wibisono et al., 2024). This raises a fundamental question: can we avoid function estimation for data generation? Ultimately, what is typically available during training is an empirical distribution over i.i.d. samples, for which the score function is not even defined. Can we generate new samples directly from this discrete empirical distribution, avoiding (score) function estimation?

We demonstrate that data generation can be achieved without function estimation. Our proposed approach is a novel generative method operating entirely on the empirical distribution of a finite set of training samples. We construct an interactive system that iteratively updates the positions of these data points in two phases: (1) applying standard gradient descent to shape data points (2) performing an inverse gradient descent step to generate new samples. This method builds upon recent advances in the theoretical study of systems of interacting particles (Duerinckx and Serfaty, 2020; Frank and Matzke, 2025). Leveraging this rich literature, we prove that the proposed estimation-free method can transport a uniform measure over a finite ball/sphere to an arbitrary data distribution in the mean-field regime, where the number of samples tends to infinity. Finally, we experimentally validate estimation-free data generation using a finite number of points.

Submitted to NeurIPS 2025 Workshop on Frontiers in Probabilistic Inference: Sampling Meets Learning.

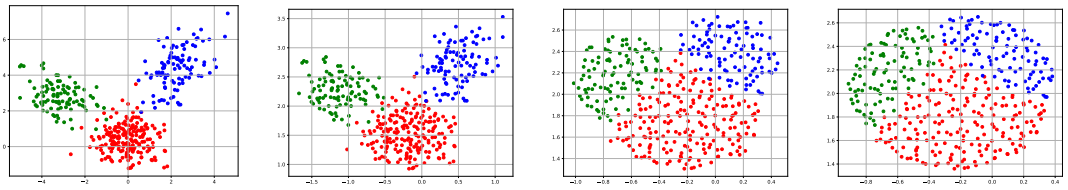


Figure 1: **Convergence of the empirical distribution with optimization.** Scatter plot of gradient descent iterates $x_i^{(k)}$, defined in (3), initialized at $x_i^{(0)}$ drawn from a Gaussian mixture distribution. Different mixture components (modes) are distinguished by color. As k evolves, the points become uniformly distributed on the circle.

2 Method

Given the support of an empirical distribution over points $x_1, \ldots, x_n \in \mathbb{R}^d$, consider the following optimization problem:

$$x_1^*, \ldots, x_n^* := \arg \min_{x_1,\ldots,x_n \in \mathbb{R}^d} \left(E_n^{(\epsilon)}(x_1, \ldots, x_n) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} W_\epsilon^{(s)}(x_i - x_j) \right), \quad (1)$$

where $W_\epsilon^{(s)}(x-y) = \frac{\|x-y\|^2}{2} + \frac{1}{s(\|x-y\|^2+\epsilon)^{s/2}}$ is called a "attractive-repulsive power-law interaction potential" (Balagué et al., 2013; Shu, 2025). The squared-norm term encourages attraction between particles, while the inverse-norm term induces repulsion, preventing collapse[1]. These competing forces lead to a striking structure in the distribution of the optimized points $x_1^*, \ldots, x_n^*$ in the asymptotic regime as $n \to \infty$, where the limiting distribution is characterized by

$$\arg\min_{p \in \Omega} \left(E(p) := \int W^{(s)}(x-y)\, p(x)\, p(y)\, dx\, dy \right), \Omega := \{\text{probability measures over } \mathbb{R}^d\}, \quad (2)$$

where $W^{(s)} := W_0^{(s)}$. The minimizer is unique up to translation and is uniform over

$$\begin{cases} \text{a ball (Carrillo and Shu, 2023)} & \text{if } s = d-2, \\ \text{a sphere (Frank et al., 2025)} & \text{if } 2 < d \text{ and } -2 \leq s < d-4, \end{cases}$$

with a radius that is finite and computable in closed form. By optimizing the locations $x_1, \ldots, x_n$, one can asymptotically transport any empirical distribution to a simple uniform distribution over a compact ball or sphere. This optimization can be performed via standard gradient descent (GD):

$$\forall i: \quad x_i^{(k+1)} = x_i^{(k)} - \gamma \nabla_{x_i} E_n^{(\epsilon)}(x_1^{(k)}, \ldots, x_n^{(k)}). \quad (3)$$

Figure 1 plots the time evaluation of points $\{x_i^{(k)}\}$ indexed by k starting from the initial points $x_i^{(0)}$ drawn i.i.d. from Gaussian mixture. We observe that the distribution becomes uniform as k increases. Although the distribution is uniform, it contains very interesting information about the initial Gaussian mixture distribution. For example, the ratio of area of different colors are proportional to the number of points in each cluster. This structure indicates that it may be possible to recover original data distribution from the final state. Whereas in the initial state $x_i^{(0)}$ the data points are i.i.d., evolving according to (3) induces strong correlations and entanglement among them under the gradient descent dynamics. Thus, in the limit $k \to \infty$, although the marginal distribution of each data point becomes uniform, their joint distribution still conveys potentially useful information about the original distribution.

Even more interesting is the inverse of gradient descent: assuming GD converges to the minimizer of E_n, inverse GD maps points uniformly distributed on a sphere (or disk) to an arbitrary empirical distribution supported on a finite set of points. The Proposition 11 in the Appendix guarantees that this inverse exists and is computable under mild assumptions. Indeed, the inverse process can be

[1]For $s = 0$, the repulsive term is replaced by the logarithm of the norm, as given by $s \to 0$ (Shu, 2025).

efficiently computed using gradient descent on a convex function, without requiring the score function or any other statistical information about the population density. Combining gradient descent (GD) and its inverse can deterministically transport an empirical distribution to a uniform distribution and vice versa. This insight enables a three-step sampling method that operates on a finite set of points: (1) apply GD to the n available samples from the target distribution; (2) add a new point drawn uniformly at random from a sphere or ball; and (3) apply inverse GD. We call this three step method **estimation-free sampling (EFS)**.

(1) Forward optimization: Apply gradient descent to $E_n^{(\epsilon)}$ (see (3)) starting from i.i.d. samples $x_1^{(0)}, \ldots, x_n^{(0)}$ drawn from the target distribution. In the next section, we show that this optimization acts as a transport map, pushing the empirical distribution toward a uniform distribution in the asymptotic regime.

(2) Augmentation: Add a new point $y^{(k)}$, sampled from a uniform distribution, to the set $x_0^{(k)}, \ldots, x_n^{(k)}$. To generate $y^{(k)}$, we first estimate the center and radius of the ball or sphere enclosing the points $x_i^{(k)}$. $y^{(k)}$ can also be generated by interpolating between two points in the set, i.e., $y^{(k)} = (1-t)x_i^{(k)} + tx_j^{(k)}$ for some $i \neq j$ and $t \in (0,1)$.

(3) Backward optimization: Compute the inverse GD for the newly generated point $y^{(k)}$ by optimizing a convex function (defined in Proposition 11), using efficient gradient descent with a constant step size. While the forward process involves n points, the backward step involves $n+1$ points, consisting of the n points from the forward step and the newly generated point from step (2). Find Algorithm details in Appendix A.

3 Theory

We demonstrate that the proposed estimation-free sampling (EFS) method can generate new samples from a distribution in the asymptotic regime. As established in Proposition 11, backward step is convex problem with well-established theoretical guarantees. In contrast, the forward optimization is non-convex. Nevertheless, its global convergence can be analyzed using recent advances in Wasserstein gradient flows on interactive-repulsive energy (Duerinckx and Serfaty, 2020; Carrillo and Shu, 2023). As the potential function is shift-invariant, *all statements hold up to translation*, and we refrain from repeating "up to translation" for simplicity.

3.1 Forward Transport to Uniform Distribution

While our ultimate goal is to analyze the backward step of EFS, which generates new samples, the forward step also plays a critical role in data generation. Here, we show that the forward step of EFS *transports* the data distribution to a uniform distribution[2].

Consider a continuous-time model where the samples $x_i^{(k)}$ updated by gradient descent, modeled by the following ordinary differential equation (ODE):

$$\frac{dx_i}{dt} = -\frac{1}{n}\sum_{j\neq i} \nabla W^{(s)}(x_i - x_j), \tag{4}$$

ODEs have been widely used to describe the limiting behavior of gradient descent with an infinitesimally small step size (Su et al., 2016; Zhang et al., 2021a,b; Chizat and Bach, 2018). Analyzing the above system becomes challenging due to the coupling between the variables x_i, as the complexity increases with the number of particles n. A common approach to address this challenge is to analyze the evolution of the empirical distribution of the particles at a macroscopic level, rather than tracking their individual trajectories. Define the empirical distribution as

$$\mu_t^{(n)} := \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i(t)}, \tag{5}$$

where δ_x denotes the Dirac measure centered at x. Consider the following PDE:

$$\frac{d\mu}{dt} = \operatorname{div}\left(\mu_t(x)\int \nabla W^{(s)}(x-y)\mu_t(y)\,dy\right), \tag{6}$$

[2]A map f transports measure μ to ν if the distribution of $f(x)$ is ν when $x \sim \mu$ (Villani et al., 2008).

where div denotes the divergence operator acting on the associated vector field. Duerinckx and Serfaty (2020) prove $\mu_t^{(n)}$ converges in the weak sense to μ_t, the solution of the PDE above.

Theorem 1 (Duerinckx and Serfaty (2020))**.** *If $\mu_0^{(n)}$ converges to a regular measure μ_0 in Wasserstein distance, then $\mu_t^{(n)}$ converges weakly to μ_t, provided $d-2 \leq s < d$.*

The macroscopic measure μ_t is significantly easier to analyze than the microscopic trajectories $x_i(t)$. While the energy E_n is generally non-convex in $x_i(t)$, the limiting energy E obeys linear interpolation convexity in μ (Shu, 2025). This convexity can be exploited to characterize the asymptotic behavior of μ_t as $t \to \infty$ (Shu, 2025).

Theorem 2 (Frank and Matzke (2025); Carrillo and Shu (2023))**.** *The steady state of μ_t is the global minimizer of E (up to translation) which is the uniform measure over:* $\begin{cases} \text{a ball (Carrillo and Shu, 2023)} & \text{if } d-2 \leq s < d, \\ \text{a sphere (Frank et al., 2025)} & \text{if } 2 < d \text{ and } -2 \leq s < d-4, \end{cases}$ *with finite radius.*

Notably, proving that a steady state (i.e., a local minimizer) is in fact a global minimizer required decades of mathematical development, beginning with the foundational work of Frostman (1935), and has only recently been fully resolved in certain parameter regimes (Frank and Matzke, 2025). These advances allow us to analyze the asymptotic dynamics of the forward step in EFS.

Corollary 3. *If $\mu_0^{(n)}$ converges to μ_0 in Wasserstein-2 distance, then the measure $\mu_t^{(n)}$ converges in the weak sense to the uniform distribution over a ball of finite radius as $n \to \infty$ and $t \to \infty$, for $s = d-2$.*

3.2 Backward Transport to Target Distribution

It is straightforward to verify that the backward optimization can exactly recover the original training data $x_k^{(i)}$ by initializing EFS with $y_k = x_k^{(i)}$. However, our goal is not to reconstruct existing data points, but to generate new samples from the underlying data distribution. We show that this form of generalization is achievable in the asymptotic regime.

Consider the continuous-time formulation of the backward optimization process:

$$\frac{dy^{(n)}}{d\bar{t}} = \frac{1}{n}\sum_{i=1}^{n} \nabla W^{(s)}\left(y_t^{(n)} - x_i(t)\right), \quad \text{where } x_i(t) \text{ is defined in (4).} \tag{7}$$

Here, the differential $d\bar{t}$ indicates that the process is reversed time (Anderson, 1982). This dynamics corresponds to EFS in the limit of an infinitesimally small step size γ and a large terminal time $T \to \infty$. We prove this dynamics *transport* μ_t back to μ_0 in the asymptotic regime as $n \to \infty$. In other words, the distribution of $y_0^{(n)}$ converges to μ_0 as $n \to \infty$.

Theorem 4. *Assume $\mu_0^{(n)}$ converges to μ_0 in Wasserstein-2 distance. Suppose that $y_t^{(n)}$ is a random variable with law μ_t, where μ_t is the solution to the continuity equation* (6)*. Then, the distribution of $y_0^{(n)}$—obtained from the reversed-time ODE* (7)*—converges to μ_0 as $n \to \infty$, for continuous measures μ_0 and μ_t, and for $s = d-2$.*

To generate new samples, we assume access to i.i.d. samples from the target distribution. Since the empirical measure $\mu_0^{(n)}$ converges to μ_0 in Wasserstein distance (Villani et al., 2008) which ensures the assumption holds in the last theorem. According to Corollary 3, the measure μ_t converges to a uniform distribution as $t \to \infty$. Consequently, sampling from a nearly uniform distribution allows for effective recovery of the initial data distribution μ_0.

Thus, a set of points whose locations are updated via *(inverse) gradient descent*, can provably generate new samples from an arbitrary distribution μ_0—without function estimation, training neural networks, or injecting noise. Avoiding explicit function estimation enables us to establish asymptotic guarantees for EFS without any structural assumptions on the data distribution. In comparison, methods such as Langevin dynamics depend on the log-concavity of the distribution and its explicit form $\mu_0 = \frac{e^{-U(x)}}{\int e^{-U(x)}\,dx}$. Instead, EFS requires only i.i.d. samples. For more details on related works, please see Section B in the Appendix.

A Note on Proofs and Experimental Results

Due to space constraints, we defer the proofs, experimental results, related work, and discussion of limitations to the Appendix.

References

Sanjeev Arora, Ravi Kannan, and Ankur Moitra. Learning mixtures of gaussians is np-hard. In *Proceedings 34th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2001.

Andre Wibisono, Yihong Wu, and Kaylee Yingxi Yang. Optimal score estimation via empirical bayes smoothing. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 4958–4991. PMLR, 2024.

Mitia Duerinckx and Sylvia Serfaty. Mean field limit for coulomb-type flows. *Duke Mathematical Journal*, 169(15):2887–2935, 2020.

Rupert L Frank and Ryan W Matzke. Minimizers for an aggregation model with attractive–repulsive interaction. *Archive for Rational Mechanics and Analysis*, 249(2):15, 2025.

Daniel Balagué, José A Carrillo, Thomas Laurent, and Gaël Raoul. Nonlocal interactions by repulsive–attractive potentials: radial ins/stability. *Physica D: Nonlinear Phenomena*, 260:5–25, 2013.

Ruiwen Shu. A family of explicit minimizers for interaction energies. *arXiv preprint arXiv:2501.14666*, 2025.

José A Carrillo and Ruiwen Shu. From radial symmetry to fractal behavior of aggregation equilibria for repulsive–attractive potentials. *Calculus of Variations and Partial Differential Equations*, 62 (1):28, 2023.

Rupert L Frank, Rupert L Frank, Rupert L Frank, and Ryan W Matzke. Minimizers for an aggregation model with attractive–repulsive interaction. *Archive for Rational Mechanics and Analysis*, 249(2): 15, 2025.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

Weijie Su, Stephen Boyd, and Emmanuel J Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17 (153):1–43, 2016.

Peiyuan Zhang, Antonio Orvieto, and Hadi Daneshmand. Rethinking the variational interpretation of accelerated optimization methods. *Advances in Neural Information Processing Systems*, 34: 14396–14406, 2021a.

Peiyuan Zhang, Antonio Orvieto, Hadi Daneshmand, Thomas Hofmann, and Roy S Smith. Revisiting the role of euler numerical integration on acceleration and stability in convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3979–3987. PMLR, 2021b.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

O Frostman. Potentiel d'equilibre et theorie des ensembles. *thesis*, 1935.

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Keith Ball et al. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31 (1-58):26, 1997.

George EP Box and Mervin E Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Andrei Kolmogoroff. Über die analytischen methoden in der wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104:415–458, 1931.

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

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

Min Jae Song. Cryptographic hardness of score estimation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 2017.

Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32, 2019.

Youssef Mroueh, Tom Sercu, and Anant Raj. Sobolev descent. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2976–2985. PMLR, 2019.

Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, and Evgeny Burnaev. Large-scale wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34:15243–15256, 2021.

David Alvarez-Melis, Yair Schiff, and Youssef Mroueh. Optimizing functionals on the space of probabilities with input convex neural network. In *Annual Conference on Neural Information Processing Systems*, 2021.

Jiaojiao Fan, Qinsheng Zhang, Amirhossein Taghvaei, and Yongxin Chen. Variational wasserstein gradient flow. In *International Conference on Machine Learning*, pages 6185–6215. PMLR, 2022.

Clément Bonet, Nicolas Courty, François Septier, and Lucas Drumetz. Efficient gradient flows in sliced-wasserstein space. *Transactions on Machine Learning Research Journal*, 2022.

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 6511–6528. PMLR, 2022.

Fabian Altekrüger, Johannes Hertrich, and Gabriele Steidl. Neural wasserstein gradient flows for discrepancies with riesz kernels. In *International Conference on Machine Learning*, pages 664–690. PMLR, 2023.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International conference on machine learning*, pages 146–155. PMLR, 2017.

Atsushi Nitanda and Taiji Suzuki. Stochastic particle gradient descent for infinite ensembles. *arXiv preprint arXiv:1712.05438*, 2017.

Hadi Daneshmand, Jason D Lee, and Chi Jin. Efficient displacement convex optimization with particle gradient descent. In *International Conference on Machine Learning*, pages 6836–6854. PMLR, 2023.

Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022.

Yilun Xu, Ziming Liu, Yonglong Tian, Shangyuan Tong, Max Tegmark, and Tommi Jaakkola. Pfgm++: Unlocking the potential of physics-inspired generative models. In *International Conference on Machine Learning*, pages 38566–38591. PMLR, 2023.

Alexander Kolesov, Manukhov Stepan, Vladimir V Palyulin, and Alexander Korotin. Field matching: an electrostatic paradigm to generate and transfer data. *arXiv preprint arXiv:2502.02367*, 2025.

Ruiwen Shu and Jiuya Wang. Generalized erdős-turán inequalities and stability of energy minimizers. *arXiv e-prints*, pages arXiv–2110, 2021a.

Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 2017.

Ruiwen Shu and Jiuya Wang. The sharp erdős-turán inequality. *arXiv preprint arXiv:2109.11006*, 2021b.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 8024–8035, 2019. URL `https://papers.nips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jonathan Masci, Ueli Meier, Dan Ciresan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.

Appendix

A Algorithm

For completeness, we present the EFS algorithm in Algorithm 1 and its forward and backward optimization subroutines in Algorithm 2 and Algorithm 3, respectively. Remarkably, even when the distribution is uniform over a high-dimensional ball, the samples are effectively drawn uniformly from the unit sphere, which is denoted by $\mathbb{S}^{d-1}$, as almost all the volume concentrates near the boundary (Ball et al., 1997).

Algorithm 1: Estimation Free Sampling (*EFS*)

Input: I.I.D samples $\{x_1^{(0)}, \ldots, x_n^{(0)}\}$

Parameters: Step size γ, number of forward iterations k, learning rate of backward (proximal step) β, number of iterations for each backward proximal step T, potential parameters s and ϵ

`/* Forward the training data to the sphere by gradient descent */`

Set $\{x_0^{(j)}, \ldots, x_n^{(j)}\}_{j=1}^k \leftarrow$ `Forward`$(\{x_1^{(0)}, \ldots, x_n^{(0)}\}, \gamma, k, s, \epsilon)$

`/* Draw a new random point uniformly from the sphere */`

Set $c = \frac{1}{n}\sum_{i=1}^n x_i^{(k)}, r = \frac{1}{n}\sum_{i=1}^n \|c - x_i^{(k)}\|$

Draw $\nu \sim \mathbb{S}^{d-1}(c, r)$ `// Sphere with center` c `and radius` r

`/* Backward the new data point to the original space */`

Set $y^0 \leftarrow$ `Backward`$(\{x_1^{(j)}, \ldots, x_n^{(j)}\}_{j=1}^k, y^{(k)}, \gamma, k, \beta, T, s, \epsilon)$

Output: Generated sample $y^{(0)}$

Algorithm 2: Forward Optimization (*Forward*)

Input: Training data $\{x_1^{(0)}, \ldots, x_n^{(0)}\}$

Parameters: step size γ, number of forward iterations k, potential parameters s and ϵ

for $j \leftarrow 0$ **to** $k-1$ **do**
  for $i \leftarrow 1$ **to** n **do**
    $\Delta_i = \frac{1}{n-1}\sum_{a\in[n], a\neq i} \nabla W_\epsilon^{(s)}(x_i^{(j)} - x_a^{(j)})$
    $x_i^{(j+1)} = x_i^{(j)} - \gamma\Delta_i$
  end
end

Output: $\{x_0^{(j)}, \ldots, x_n^{(j)}\}_{j=1}^k$

Algorithm 3: Backward Optimization (*Backward*)

Input: Data $\{x_1^{(j)}, \ldots, x_n^{(j)}\}_{j=1}^k$, new sample $y^{(k)}$

Parameters: Step size γ, number of forward iterations k, learning rate of backward (proximal step) β, number of iterations for each backward proximal step T, potential parameters s and ϵ

Set: $j = k$

while $j \geq 0$ **do**
  Set: $v_0 = y^{(j)}$
  for $t \leftarrow 0$ **to** T **do**
    $\nabla = \frac{\gamma}{n}\sum_{i\in[n]} \nabla W_\epsilon^{(s)}(v_t - x_i^{(j)})$
    $\Delta = v_t - y^{(j)} - \nabla$
    $v_{t+1} = v_t - \beta\Delta$
  end
  $j = j - 1$ and $y^{(j)} = v_T$
end

Output: $y^{(0)}$

B Related works

Data generation with estimation. Generative models traditionally rely on transporting simple to complex distributions. A classical example is the Box–Muller transform: given independent random variables $\theta \sim \text{Uniform}[0, 2\pi]$ and $r \sim \text{Exponential}$, the random vector $v = r(\cos\theta, \sin\theta)$ follows a standard two-dimensional Normal distribution (Box and Muller, 1958). This illustrates a fundamental idea: sampling from a complex distribution can be achieved via a nonlinear transformation of samples drawn from a simpler distribution.

Modern generative models adopt this principle using function approximation: Given latent variable x sampled from a Gaussian distribution, the goal is to find a parametric function f_θ such that the distribution of $f_\theta(x)$ approximates the data distribution. In generative adversarial networks (GANs), f_θ is implemented by a neural network and trained via adversarial objectives (Goodfellow et al., 2014). In normalizing flows, f_θ is a sequence of invertible and differentiable transformations optimized via maximum likelihood (Rezende and Mohamed, 2015). Score-based diffusion models, gradually transform data using estimated score functions of diffusion density (Song et al., 2020). In contrast to these methods, our proposed approach computes a non-linear transformation using a set of interacting points, thereby avoiding the estimation of transport map f_θ.

Varational perspective towards generative models. Despite its distinct mechanism, EFS shares conceptual parallels with diffusion models. Both can be interpreted as discretizations of different *"Wasserstein gradient flows"* (Ambrosio et al., 2008). This connection arises from a common variational principle: many simple distributions from which i.i.d. samples are easily drawn, such as the Gaussian distribution or the uniform distribution on a sphere, can be viewed as equilibrium states of physical systems with minimum energy. For example, Gaussian distribution is the minimizer of the negative entropy functional over the space of probability measures (Jordan et al., 1998), while the uniform distribution on the unit sphere minimizes a Riesz-type interaction energy (Frank and Matzke, 2025). More generally, such distributions can be described as

$$p^* = \arg\min_{p \in \Omega} F(p), \quad \Omega := \{\text{probability measures over } \mathbb{R}^d\}, \tag{8}$$

where F is an appropriate energy functional.

This variational formulation provides a principled mechanism to transport any distribution to the minimum energy state by gradient descent on F in the space of probability densities. Diffusion models, optimize $F(p) = \int \log(p(x))p(x)dx$ using Wasserstein gradient flow (Jordan et al., 1998). More interesting observation is that the reverse of gradient flow which can transport Gaussian distribution to any distribution. For entropy, inverse Wasserstein gradient flow is implemented by the Kolmogorov backward equation (Kolmogoroff, 1931), enabling transformation from a Gaussian to arbitrary target distributions. This key result forms the basis of modern generative diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020), where the reverse-time stochastic process includes a drift term proportional to the score function of intermediate densities (Anderson, 1982). However, estimating this score function for arbitrary distributions remains statistically and computationally challenging. In this vein, Wibisono et al. (2024) recently proved that estimating the score function of sub-Gaussian distributions with Lipschitz-continuous scores suffers from a curse of dimensionality in the required sample complexity. Song (2024) further demonstrated that, under lattice-based cryptographic hardness assumptions, score estimation remains computationally intractable even when the sample complexity is polynomial in the relevant parameters.To circumvent these challenges, our method replaces the entropy-based potential with a Riesz-type interaction energy and optimizes it directly over discrete empirical measures.

Inspired by Formulation (8), one may attempt to optimize functionals of the probability measure p that quantify its distance to a target distribution p^*. For example, the celebrated work of Arjovsky et al. (2017) introduce the Wasserstein GAN, which optimizes the Wasserstein-1 distance between the model distribution p and the target distribution p^* by solving its Kantorovich–Rubinstein dual via an adversarial training objective over both the generator (which parameterizes p) and the critic (which approximates the dual). More closely related to gradient flows on the space of probability measures, Arbel et al. (2019) consider the Wasserstein gradient flow of the Maximum Mean Discrepancy (MMD) functional and provide particle-based implementations. Additionally, Mroueh et al. (2019) propose Sobolev Descent, deriving a particle algorithm from the Sobolev integral probability metric that transports samples along smooth descent paths. More recently, there is been a surge in using

the seminal proximal point method of Jordan, Kinderlehrer, and Otto (JKO) (Jordan et al., 1998), to discretisize the backward Wasserstein gradient flows over the chosen energy function F (Mokrov et al., 2021; Alvarez-Melis et al., 2021; Fan et al., 2022; Bonet et al., 2022; Bunne et al., 2022; Altekrüger et al., 2023). Central to almost all of these works, is the estimation of the transport map in JKO proximal step with input convex neural networks (Amos et al., 2017). In contrast to these works, our approach does not require parameterizing the transport map, the functionals that solve the variational formulations, or the underlying ODE/PDE induced by the continuity equation. Instead, it generates data via direct (inverse) gradient descent on a finite set of points without function estimation.

Particle gradient descent. Nitanda and Suzuki (2017) introduce "particle gradient descent" for optimizing a given energy function F over sparse measures as

$$\min_{x_1,\dots,x_n} F\left(\frac{1}{n}\sum_{i=1}^{n}\delta(x_i)\right), \quad \delta(x) : \text{the Dirac measure at } x. \tag{9}$$

Particle gradient descent is the gradient descent optimizing the location of $x_1, \dots, x_n$. Single layer neural networks can be viewed as particle gradient descent on a specific energy functions. Motivated by this connection, Chizat and Bach (2018) study the connection between gradient descent on particles and gradient flow in the space of probability measure equipped with Wasserstein-2 metric in asymptotic regime $n \to \infty$. Daneshmand et al. (2023) establishes a non-asymptotic convergence analysis for particle gradient descent on displacement convex functions. While primary focus of these studies is on analyzing single-layer neural networks, we leverage (inverse) particle gradient descent to develop an estimation-free generative method.

Physics-inspired methods. Xu et al. (2022) developed a generative model that maps samples from a uniform distribution over an infinite-radius hemisphere to an arbitrary target distribution. Their model relies on estimating the *Poisson vector field* parameterized by neural networks. This idea, primarily inspired by physical systems—especially electrostatic theory—led to further developments in subsequent works (Xu et al., 2023; Kolesov et al., 2025).

To cope with the challenges of sampling from an infinite-radius hemisphere, Xu et al. (2022) simulate the corresponding ODE by perturbing the training data and then estimate the negative normalized field from these perturbed samples. This contrasts with our approach, in which the primitive distribution is uniform on a finite-dimensional compact manifold (the sphere), and is therefore easy to sample from. Although we also transport samples from a uniform distribution to an arbitrary data distribution, our transport mechanism does not rely on function estimation; rather, it is purely based on (inverse) gradient descent over a finite set of points.

C Discussions

We demonstrate, both theoretically and experimentally, that it is possible to transport a uniform distribution to a target distribution without estimating a score function—using only i.i.d. samples from the target. Our method, Estimation-Free Sampling (EFS), avoids both noise injection and function estimation. Instead, it introduces interactions between samples by optimizing an energy functional, thereby shaping the empirical distribution of samples. EFS lies at the intersection of three foundational fields—mathematical optimization, potential theory, and generative AI —offering rich opportunities for interdisciplinary research.

Optimization for Sampling. While most generative models are built upon stochastic differential equations, EFS is purely a *deterministic optimization method*, opening new avenues for the optimization community to use powerful optimization techniques—such as accelerated methods, higher-order optimization methods, and efficient stochastic techniques—to generate data.

Notably, EFS involves both convex and non-convex optimization. Linking the non-convex gradient descent dynamics to a well-studied Wasserstein gradient flow, we establish asymptotic convergence guarantees. However, understanding the behavior of gradient descent in non-asymptotic regimes remains an open challenge. We believe the introduced asymptotic results provides a solid foundation for future non-asymptotic analyses.

Potential Theory. EFS induces attractive-repulsive interactions between training samples using a well-known potential function studied extensively in fractional potential theory (Shu and Wang, 2021a; Frank et al., 2025; Duerinckx and Serfaty, 2020). By linking generative modeling to this rich theoretical framework, we gain access to powerful analytical tools for understanding and designing generative models. In particular, our analysis leverages results on Wasserstein gradient flows of attractive-repulsive energies (Duerinckx and Serfaty, 2020), as well as variational analysis of these energy functionals (Carrillo and Shu, 2023; Frank and Matzke, 2025). This bridge between theory and practice opens new directions for developing principled and reliable generative models with theoretical guarantees.

Potential theory can design new interaction potentials tailored to data generation and practical applications. As noted in our experiments, one major challenge was the choice of the power s in the potential function, which led to numerical instability in high dimensions. Investigating alternative potentials that avoid such issues—particularly those that do not involve dimension-dependent blow-up—may help resolve computational challenges of EFS.

Generative AI. EFS opens up several promising directions for future research, particularly in scaling to large-scale machine learning applications. A key practical challenge lies in the quadratic time complexity of the forward optimization step with respect to the number of training samples. We conjecture that stochastic optimization techniques could mitigate this issue and significantly reduce computational cost.

D Proofs

Notations. $\mathbb{S}^{d-1}(c,r)$ denotes sphere with center $c \in \mathbb{R}^d$ and radius r. Ω denotes the set of probability measure over $\mathbb{R}^d$ with Wasserstein-2 metric, which is denoted by $\mathcal{W}_2(\mu,\nu)$ where $\mu,\nu \in \Omega$. Given vector function $v : \mathbb{R}^d \to \mathbb{R}^d$, its *divergence* is defined as $\mathrm{div}(v) = \sum_{i=1}^d \frac{dv}{dx_i}$. Suppose $E : \Omega \to \mathbb{R}$. Then, the *first variation* of E with respect to μ is denoted by $\frac{dE}{d\mu}$ (Santambrogio, 2017). Function $f : \mathbb{R}^d \to \mathbb{R}^d$ transport $\mu \in \Omega$ to $\nu \in \Omega$ if for a random vector $x \in \mathbb{R}^d$ drawn from μ, $f(x)$ has density ν. Recall $W_\epsilon^{(s)}$ is the potential function defined as

$$W_\epsilon^{(s)}(x-y) = \frac{\|x-y\|^2}{2} + \frac{1}{s(\|x-y\|^2+\epsilon)^{s/2}}. \tag{10}$$

For simplicity, we use the shorthand notation $W^{(s)} = W_0^{(s)}$ and $W^{(s)}$ by W. The sign $*$ denotes the standard convolution as

$$f * g(y) = \int f(y-x)g(x)dx \tag{11}$$

$\|f\|_{L^2}$ is the L_2 functional norm defined as

$$\|f\|_{L_2}^2 = \int \|f(x)\|^2 dx \tag{12}$$

Weak/Strong convergence. A sequence of measures $\mu^{(n)} \in \Omega$ is said to converge weakly to μ if, for all bounded continuous test functions $f : \mathbb{R}^d \to \mathbb{R}$, the integrals $\int f(x)\mu^{(n)}(x)\,dx$ converge to $\int f(x)\mu(x)\,dx$. A stronger notion is convergence in the $L^2(\nu)$ norm, which requires that

$$\lim_{n\to\infty} \int \|\mu^{(n)}(x) - \mu(x)\|^2 \nu(x) dx = 0.$$

MMD. Define MMD : $\Omega \times \Omega \to \mathbb{R}_+$ as

$$\mathrm{MMD}^2(\mu,\nu) = \int K(x-y)(\mu(x)-\nu(x))dx(\mu(y)-\nu(y))dy, \quad \text{with } K(\Delta) \coloneqq \frac{1}{s\|\Delta\|^s} \tag{13}$$

where MMD denotes the Maximum Mean Discrepancy between two measures μ and ν.

Suppose that $\widehat{K}$ and $\widehat{\Delta}$ denote the Fourier transforms of the functions $K(x)$ and $\Delta(x) \coloneqq \mu(x) - \nu(x)$, respectively. Viewing MMD as a convolution and Plancherel's theorem allow us to write MMD as (Shu and Wang, 2021b)

$$\text{MMD}^2(\mu, \nu) = \int \widehat{K}(w)|\widehat{\Delta}(w)|^2 dw, \tag{14}$$

where $\widehat{K}(w) = C\|w\|^{-d+s}$ with a constant C depending only on d and s (Frank and Matzke, 2025). Substituting the Fourier transform into the equation above establishes that $\text{MMD}(\mu, \nu) = 0$ implies $\mu = \nu$. Notably, it is sufficient for the Fourier transform $\widehat{K}$ to be strictly positive almost everywhere to guarantee that K is a *universal kernel* (Gretton et al., 2012).

To avoid potential confusion, note that the MMD associated with the kernel K defined above is not a metric. While machine learning often relies on positive semi-definite kernels to ensure that MMD defines a valid metric, the kernel K is not positive semi-definite. As a result, the corresponding MMD does not obey the triangle inequality.

Gradient dominance of E. We establish an important property of the energy function E. Define functional $F : \Omega \to L_2$ as

$$F(\mu)(y) = \nabla \frac{dE}{d\mu} = \int W(y - x)\mu(x)dx. \tag{15}$$

Remarkably, μ is a steady state of the Wasserstein gradient flow on E if $F(\mu) = 0$. The next theorem represent MMD using F.

Theorem 5. *Suppose that $\mu \in \Omega$ and $\nu \in \Omega$ have the same first-moment, then*

$$\|F(\mu) - F(\nu)\|_{L_2}^2 = \textit{MMD}^2(\mu, \nu)$$

holds for $s = d - 2$.

An application of the above theorem recovers the result of Carrillo and Shu (2023): all steady states μ satisfying $F(\mu) = 0$ are global minimizers of E (up to translation). More precisely, the theorem implies that for all μ, ν such that $F(\mu) = F(\nu) = 0$, the following holds:

$$0 = \|F(\mu) - F(\nu)\|_{L_2}^2 = \text{MMD}(\mu, \nu). \tag{16}$$

Beyond recovering existing results, the theorem will also be used to analyze the backward optimization for EFS in Section D.1.

Proof. Let $\widehat{f}(w)$ denote the Fourier transform of function $f(x)$. Define $\widehat{\Delta} \coloneqq \widehat{\mu} - \widehat{\nu}$ which is equivalent to the Fourier transform of $\Delta \coloneqq \mu - \nu$. As discussed in Section D, MMD can be written as

$$\text{MMD}^2(\nu, \mu) = \int \widehat{K}(w)|\widehat{\Delta}(w)|^2 dw \tag{17}$$

According to the definition,

$$\nabla W(z) = \nabla K(z) + \text{iden}(z), \quad \text{iden}(z) \coloneqq z \tag{18}$$

Replacing the above formula into the definition of F obtains

$$\|F(\mu) - F(\nu)\|_{L_2}^2 = \|(\nabla K + \text{iden}) * \mu - (\nabla K + \text{iden}) * \nu\|_{L^2}^2, \tag{19}$$

where $*$ denotes convolution defined in (11) and L_2 is the norm 2 for functions defined in (12). It is easy to check that

$$(\text{iden} * \mu)(y) = y \underbrace{\int \mu(x)dx}_{=1} - \int x\mu(x)dx \tag{20}$$

$$= y \underbrace{\int \nu(x)dx}_{=1} - \underbrace{\int x\mu(x)dx}_{\mathbb{E}_\mu[x]} \tag{21}$$

$$= y \int \nu(x)dx - \underbrace{\int x\nu(x)dx}_{\mathbb{E}_\nu[x]} = (\text{iden} * \nu)(y). \tag{22}$$

The last equation holds because the first moments of μ and ν are equal. This observation allows us to significantly simplify the expression for the quantity of interest as follows:

$$\|F(\mu) - F(\nu)\|_{L_2}^2 = \|\nabla K * \mu - \nabla K * \nu\|_{L_2}^2 \tag{23}$$

Recall two fundamental properties of Fourier transform as: $\begin{cases} \widehat{f * g} = \widehat{f}\widehat{g} \\ \widehat{\partial f} = iw\widehat{f} \end{cases}$. These properties together with Parseval's theorem yield

$$\|F(\mu) - F(\nu)\|_{L_2}^2 = \int \|w\|^2 \left(\widehat{K}(w)\right)^2 |\widehat{\Delta}(w)|^2 dw \tag{24}$$

For $s = d - 2$, it is easy to check that

$$\|w\|^2 (\widehat{K}(w))^2 = \widehat{K}(w) \tag{25}$$

holds given $\widehat{K}(w) = C\|w\|^{-2}$ (Frank and Matzke, 2025). Combining all the results completes the proof:

$$\|F(\mu) - F(\nu)\|_{L_2}^2 \overset{(24)}{=} \int \|w\|^2 \left(\widehat{W}(w)\right)^2 |\widehat{\Delta}(w)|^2 dw \tag{26}$$

$$\overset{(25)}{=} \int \widehat{W}(w) |\widehat{\Delta}(w)|^2 dw \tag{27}$$

$$\overset{(17)}{=} \text{MMD}^2(\mu, \nu) \tag{28}$$

□

A convergence result for ODEs. Consider the following two differential equations

$$\begin{cases} \frac{dy}{dt} = F_t(y_t) \\ \frac{dy^{(n)}}{dt} = F_t^{(n)}(y^{(n)}) \end{cases}, \quad y_0 = y_0^{(n)} \tag{29}$$

where $F_t, F_t^{(n)} : \mathbb{R}^d \to \mathbb{R}^d$ and $y_t, y_t^{(n)} \in \mathbb{R}^d$. If $F_t^{(n)}$ converges to F_t as $n \to \infty$, then $y_t^{(n)}$ converges to y_t in interval $t \in [0, T)$.

Lemma 6. *Define*

$$\epsilon_n \coloneqq \int \|F_t(y) - F_t^{(n)}(y)\|^2 dy.$$

Suppose $F_t(y)$ is almost surely L-Lipschitz in y. If $\lim_{n\to\infty} \epsilon_n = 0$, then $\int_0^T \|y_t - y_t^{(n)}\|^2 dt$ converges to 0 as $n \to \infty$.

Proof. We start by writing down ODEs in the integral form and consider their difference,

$$\begin{aligned} \|y^{(n)}(t) - y(t)\| &= \left\| \int_0^t \left[F_s^{(n)}(y^{(n)}) - F_s(y) \right] ds \right\| \\ &= \left\| \int_0^t \left[F_s^{(n)}(y^{(n)}) - F_s(y^{(n)}) \right] ds + \int_0^t \left[F_s(y^{(n)}) - F_s(y) \right] ds \right\| \\ &\leq t \sup_y \left\| F_s^{(n)}(y) - F_s(y) \right\| + L \int_0^t \left\| y^{(n)}(s) - y(s) \right\| ds \\ &\leq t\epsilon_n + L \int_0^t \left\| y^{(n)}(s) - y(s) \right\| ds \end{aligned} \tag{30}$$

The last inequality was followed by convergence of $F_s^{(n)}$ to F_s, and boundedness of L^∞ norm by L^2, the first term is converging to zero, and bounded by $\epsilon_n \xrightarrow{n\to\infty} 0$.

Define $\Delta_n(t) \coloneqq \sup_{s \leq t} \left\| y^{(n)}(s) - y(s) \right\|$. By taking supremum from (30),

$$\Delta_n(t) \leq t\epsilon_n + L \int_0^t \Delta_n(s) ds.$$

In turn, applying Grönwall's inequality yields,

$$\Delta_n(t) \leq t\epsilon_n e^{Lt} \xrightarrow{n\to\infty} 0.$$

Thus,

$$\left\| y^{(n)}(s) - y(s) \right\| \xrightarrow{n\to\infty} 0,$$

uniformly for all $s \leq t$ and proof is completed. □

Continuity equation and transporting distributions. Let $v_t : \mathbb{R}^d \to \mathbb{R}^d$ be a general vector field. $\mu_t \in \Omega$ is a solution of continuity equation associated with v_t if it obeys

$$\frac{d\mu_t}{dt} = -\mathrm{div}(\mu_t v_t) \tag{31}$$

Given the vector field, we define the following ODE

$$\frac{dy_t}{dt} = v_t(y_t) \tag{32}$$

The above ODE transports μ_0 to μ_t as stated in the following lemma.

Lemma 7 (Lemma 8.1.6. of (Ambrosio et al., 2008))**.** *Suppose that the vector field v_t obeys the following 3 conditions*

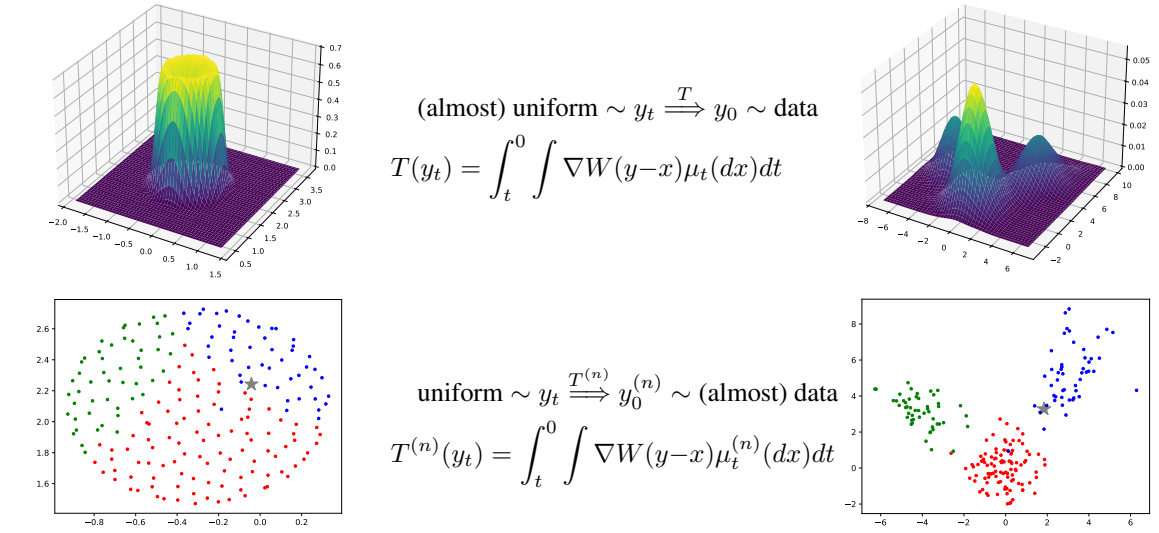
(1) $\int |v_t(x)| d\mu_t(x) dx < \infty$.

(2) For every compact subset $B \subset \mathbb{R}^d$, $\sup_{x\in B} |v_t(x)| < \infty$.

(3) $v_t(x)$ *is Lipschitz*

If y_0 is a random variable with distribution μ_0, then y_t is a random variable with distribution μ_t for a finite t.

D.1 Proof of Theorem 4



Proof sketch for Theorem 4. Given the solution to the continuity equation (6), we define the map T above, which provably transports μ_t to data distribution, as stated in Lemma 8. Recall that Theorem 2 states that μ_t converges to the uniform distribution, from which new samples can be drawn. T is not implementable as it requires μ_t, the solution of the continuity equation (6). We show that the empirical distribution $\mu_t^{(n)}$, defined over the point set $\{x_1(t), \ldots, x_n(t)\}$, yields a transport map $T^{(n)}$ (as defined above) that converges to T as $n \to \infty$. The $\star$ symbol in the second row indicates y_t (left) and $y_0^{(n)}$ (right). Colored points represent $\{x_1(t), \ldots, x_n(t)\}$ on the left and $\{x_1(0), \ldots, x_n(0)\}$ on the right.

Recap. Recall that $\mu_t^{(n)}$ defined as $\mu_t^{(n)} = \frac{1}{n}\sum_{i=1}^n \delta_{x_i(t)}$ where δ is the Dirac measure and $x_i(t)$ obeys

$$\frac{dx_i}{dt} = -\frac{1}{n}\sum_{j\neq i} \nabla W^{(s)}(x_i - x_j). \tag{33}$$

Using $x_i(t)$, we define $y_t^{(n)}$ which obeys

$$\frac{dy^{(n)}}{d\bar{t}} = F_t^{(n)}(y_t^{(n)}), \quad F_t^{(n)}(y) \coloneqq \int \nabla W\left(y - x\right) \mu_t^{(n)}(x)dx, \tag{34}$$

where $d\bar{t}$ indicates that the process is reversed time (Anderson, 1982).

Reversed-time transport. Recall μ_t as the solution of continuity equation for Wasserstein-2 gradient flow:

$$\frac{d\mu}{dt} = \operatorname{div}\left(\mu(x)\int \nabla W^{(s)}(x-y)\mu_t(y)\,dy\right), \tag{35}$$

Notably, $F_t(y) \coloneqq \int \nabla W(y-x)\mu_t(x)dx$ represents the vector field associated with the above dynamics. Given this vector field, define the following ODE

$$\frac{dy}{dt} = -F_t(y_t), \quad \text{where } F_t(y) \coloneqq \int \nabla W(y-x)\mu_t(x)dx \tag{36}$$

The above ODE transports μ_t backward to μ_0 as stated in the following lemma.

Lemma 8. *Let y_0 be a random vector drawn from μ_0. Then, the distribution of y_t is μ_0.*

The above result together with a simple change of variables obtains the inverse transport from μ_t to μ_0.

Corollary 9. *Consider the following reversed-time dynamics:*

$$\frac{dy_t}{d\bar{t}} = F_t(y_t) \tag{37}$$

The above dynamics transports μ_t to μ_0 as long as μ_t is not a steady state of (35)*.*

Proof. According to definition

$$\lim_{\epsilon\to 0}\frac{y_{t+\epsilon}-y_t}{\epsilon} = -F_t(y_t) \implies \frac{dy}{d\bar{t}} = \lim_{\epsilon\to 0}\frac{y_{t-\epsilon}-y_t}{\epsilon} = \lim_{\epsilon\to 0} F_{t-\epsilon}(y_{t-\epsilon}) \tag{38}$$

where $F_{t-\epsilon}(y) = \int \nabla W(y-x)\mu_{t-\epsilon}(x)dx$. Lemma 10 implies F_t is Lipschitz. Thus, invoking Lemma 6 concludes the statement. □

While the flow y_t transports μ_t to the data distribution, its construction relies on access to μ_t, which is defined as the solution to a PDE. Since μ_t is not feasible to compute, we cannot implement the reverse-time ODE (36) to recover y_0. However, Theorem 4 restated bellow ensures that EFS can reconstruct the backward dynamic as $n \to \infty$.

Theorem 4. *Assume $\mu_0^{(n)}$ converges to μ_0 in Wasserstein-2 distance. Suppose that $y_t^{(n)}$ is a random variable with law μ_t, where μ_t is the solution to the continuity equation* (6)*. Then, the distribution of $y_0^{(n)}$—obtained from the reversed-time ODE* (7)*,*

$$\frac{dy^{(n)}}{d\bar{t}} = \frac{1}{n}\sum_{i=1}^{n}\nabla W^{(s)}\left(y^{(n)} - x_i\right)$$

converges to μ_0 as $n \to \infty$, for regular measures μ_0 and μ_t, and for $s = d-2 > 0$.

Proof. We prove that $y_0^{(n)}$ converges to y_0, whose distribution is μ_0, according to Lemma 8. To establish this convergence, we use Theorem 1 of (Duerinckx and Serfaty, 2020), which shows that $\mu_t^{(n)}$ converges to μ_t in MMD as $n \to \infty$, namely

$$\lim_{n\to\infty} \text{MMD}(\mu_t, \mu_t^{(n)}) = 0, \quad \text{for } \beta > 0. \tag{39}$$

As proven in the next lemma, $F_t(y)$ is Lipschitz. This Lipschitz continuity, together with the result above, ensures that the conditions in Lemma 6 are satisfied. Thus, $y_0^{(n)}$ converges to y_0, as guaranteed by the lemma. □

Lemma 10. *Recall vector function $F_t(y) = \int \nabla W^{(d-2)}(y-x)\mu_t(x)dx$ where μ_t is the solution of PDE* (35) *starting from μ_0. Then, $F_t(y)$ is L-Lipschitz for $L = c_1 + c_2(E(\mu_0)+1)^{\frac{2d+2}{d}}$ where constants c_1 and s_2 depend only on d.*

Proof. The first step is to cast Lipschitzness to the boundedness of the following integral

$$\int \frac{1}{\|y-x\|^d}\mu_t(x)dx \le L \implies F_t \text{ is } L\text{-Lipschitz} \tag{40}$$

Then, we use properties of Wasserstein gradient flow to prove

$$\int \frac{1}{\|y-x\|^d}\mu_t(x)dx \le c_1 + c_2 E(\mu_0), \tag{41}$$

where c_1 and c_2 are constants that only depend on d. For the similarity, we introduce notation $a \lesssim b$ if there are constants c_1 and c_2 that only depends on d such that $a \le c_1 + c_2 b$

Step I. A Sufficient Condition for Lipschitz-ness. According to the definition, we have

$$\|F_t(y) - F_t(y')\|_2 = \left\| \int \nabla K(y-x)\mu_t(dx) - \int \nabla K(y'-x)\mu_t(dx) \right\|_2 \tag{42}$$

$$\leq \|(y-y') \int_0^1 \nabla^2 K(\gamma y + (1-\gamma)y' - x)\mu_t(dx)d\gamma\|_2 \tag{43}$$

$$\leq \|y-y'\|_2 \max_y \| \int \nabla^2 K(y-x)\mu_t(x)dx\|_2 \tag{44}$$

To prove that the right side is bounded, we bound the norm of $\nabla^2 K$ as

$$\| \int \nabla^2 K(y-x)\mu_t(x)dx\| = \left\| \int -\frac{I_d}{\|y-x\|^d} + (d)\frac{(y-x)(y-x)^\top}{\|y-x\|^{d+2}}\mu_t(dx) \right\| \tag{45}$$

$$\leq (d) \int \frac{1}{\|y-x\|^d}\mu_t(x)dx \tag{46}$$

Thus, the above integral bounds the Lipschitz constant of F_t.

Step II. Gradient flow property. Since μ_t is a gradient flow, the following holds

$$E(\mu_t) \leq E(\mu_0) \tag{47}$$

We establish an important consequence of the above inequality, namely the following bound

$$\sup_y \left| \int \frac{1}{\|y-x\|^{d-1}}\mu_t(x)dx \right| \lesssim (1 + E(\mu_0))^2 \tag{48}$$

holds. Using Jensen's inequality, we have

$$\int \frac{1}{\|x-y\|}d\mu_t(x) \leq \left(\int \frac{1}{\|x-y\|^{d-1}}d\mu_t(x) \right)^{\frac{1}{d-1}} \tag{49}$$

Combining the last two bounds obtains

$$\int \frac{1}{\|y-x\|^d}\mu_t(x)dx \leq \left| \int \frac{1}{\|y-x\|^{d-1}}\mu_t(x)dx \right| \left| \int \frac{1}{\|y-x\|}\mu_t(x)dx \right| \lesssim (1 + E(\mu_0))^{\frac{2d+2}{d}} \tag{50}$$

The above equation concludes that F_t is Lipschitz according to step I. To complete the proof, we need to prove equation (48).

We first establish a consequence of $E(\mu_t) \leq E(\mu_0)$ for μ_t. Let $\widehat{\mu}(w)$ denote Fourier transform of $\mu_t(x)$. It is easy to check that

$$|\widehat{\mu}(w)| \leq \int |\mu_t(x)|dx = 1 \tag{51}$$

Without of loss of generality, we can assume that $\int x\mu_t(x)dx = 0$. An application of Jensen's inequality yields

$$\int \|x\|^2 d\mu_t(x) = \int \|x - \int y\mu_t(y)dy\|^2 d\mu_t(x) \tag{52}$$

$$\leq \int \|x-y\|^2 d\mu_t(x)d\mu_t(y)dxdy \tag{53}$$

$$\lesssim E(\mu_t) \lesssim E(\mu_0) \tag{54}$$

According properties of Fourier transform, we have

$$|\frac{\partial \widehat{\mu}}{\partial w_i \partial w_j}| = \int |x_i x_j|\mu_t(x)dx \tag{55}$$

$$\leq \frac{1}{2}\left(\int x_i^2 \mu_t(x)dx + \int x_j^2 \mu_t(x)dx \right) \tag{56}$$

$$\leq \int \|x\|^2 \mu_t(x) \tag{57}$$

$$\overset{(54)}{\leq} E(\mu_0) \tag{58}$$

Gagliardo–Nirenberg interpolation inequality yields

$$\|\widehat{\mu}\|_{L^1} \lesssim (\|\partial^2\widehat{\mu}\|_{L^\infty} + 1)(\|\widehat{\mu}\|_{L^\infty} + 1) \tag{59}$$

$$\overset{(58)}{\lesssim} (1 + E(\mu_0))(\|\widehat{\mu}\|_{L^\infty} + 1) \tag{60}$$

$$\overset{(51)}{\lesssim} (1 + E(\mu_0))^2 \tag{61}$$

We will use the above bound to complete the proof.

Recall the Fourier transform of the radial function $\|x\|^{-s}$ is $C\|w\|^{-d+s})$ for $0 < s < d$ where C is a constant depending on d and s (Frank and Matzke, 2025). Using the Parseval's theorem, (47) translates to the following inequality in the complex (Fourier) domain

$$C\int \frac{|\widehat{\mu}(w)|^2}{\|w\|^2}dw = \int \frac{1}{\|y-x\|^{d-2}}d\mu_t(x)d\mu_t(y) \leq (d-2)E(\mu_0) \tag{62}$$

holds. Similarly, we can write (48) as

$$\int \frac{1}{\|w\|}|\widehat{\mu}(w)|dw \lesssim (1 + E(\mu_0))^2 \tag{63}$$

Define set $A = \{w | \|w\| < 1\}$, a straightforward application of Cauchy-Schwarz obtains

$$\left(\int_A \|w\|^{-1}|\widehat{\mu}(w)|dw\right)^2 \leq \int_A \|w\|^{-2}|\widehat{\mu}(w)|^2dw \int_A dw \tag{64}$$

$$\overset{(62)}{\lesssim} E(\mu_0) \tag{65}$$

To establish (63), we need to bound the above integral taken over the complement of A denoted by A^c:

$$\int_{A^c} \frac{1}{\|w\|}|\widehat{\mu}(w)|dw \leq \int |\widehat{\mu}(w)|dw \overset{(61)}{\lesssim} (1 + E(\mu_0))^2 \tag{66}$$

Combining the last two inequality concludes (63), hence the proof is complete.

□

D.2 Proof of the Auxiliary Lemma 8

Lemma 8. *Let y_0 be a random vector drawn from μ_0 such that $E(\mu_0) < \infty$. Consider the following ODE*

$$\frac{dy}{dt} = -F_t(y_t), \quad F_t(y) \coloneqq \int \nabla W(y-x)\mu_t(x)dxdt$$

Then, the distribution of y_t is μ_t.

Proof. If the vector field F_t obeys 3 conditions in Lemma 7, then invoking the lemma concludes the proof. It remains to to validate necessary conditions for the vector field. Since μ_t is a gradient flow, it obeys

$$\int \frac{1}{(d-2)\|x-y\|^{d-2}}d\mu_t(x)d\mu_t(y) + \frac{1}{2}\int \|x-y\|^2 d\mu_t(x)d\mu_t(y) = E(\mu_t) \leq E(\mu_t) \tag{67}$$

Condition 1: $\int \|F_t(x)\|\mu_t(x)dx < \infty$. According to the definition, we have

$$\|F_t(x)\|\mu_t(x)dx = \|\int \frac{x-y}{\|x-y\|^d} + (x-y)d\mu_t(y)\|\mu_t(x)dx \tag{68}$$

$$\leq \int \frac{1}{\|x-y\|^{d-1}}d\mu_t(x)d\mu_t(y) + \int \|x-y\|d\mu_t(x)d\mu_t(y) \tag{69}$$

We bound each term in the above upper-bound. The first term can be bounded using Jensen's inequality as

$$\int \|x-y\| d\mu_t(x) d\mu_t(y) \leq \sqrt{\int \|x-y\|^2 d\mu_t(x) d\mu_t(y)} \overset{(67)}{\leq} \sqrt{E(\mu_0)} \tag{70}$$

Similarly, we get

$$\int \frac{1}{\|x-y\|} d\mu_t(x) d\mu_t(y) \leq \left(\int \|x-y\|^2 d\mu_t(x) d\mu_t(y) \right)^{1/(d-2)} \overset{(67)}{\leq} (E(\mu_0))^{1/(d-2)} \tag{71}$$

The above inequality yields

$$\int \frac{1}{\|x-y\|^{d-1}} d\mu_t(x) d\mu_t(y) \leq \left(\int \frac{1}{\|x-y\|^{d-2}} d\mu_t(x) d\mu_t(y) \right) \int \frac{1}{\|x-y\|} d\mu_t(x) d\mu_t(y) \tag{72}$$

$$\leq E(\mu_0)^{(d-1)/(d-2)} \tag{73}$$

Condition 2. $\|F_t(x)\| < \infty$ for a bounded x. Since all statements hold up to translation, we can assume $\int y \mu_t(y) dy = 0$. This allows us to simplify the expression for the vector field as

$$\|F_t(x)\| = \| \frac{x-y}{\|x-y\|^d} d\mu_t(y) + x - \underbrace{\int y \mu_t(y)}_{=0} \| \tag{74}$$

$$\leq \int \frac{1}{\|x-y\|^{d-1}} d\mu_t(y) + \|x\| \tag{75}$$

$$\overset{(48)}{\leq} c_1 + c_2(1 + E(\mu_0)^2) + \|x\|, \tag{76}$$

where c_1 and c_2 are constants independent from t.

Condition 3. Lipschitzness. Lemma 10 ensures F_t is Lipschitz. □

E Inverse Gradient Descent

Proposition 11. *Consider the following proximal optimization problem*

$$(y_1^*, y_2^*, \cdots, y_n^*) = \arg\min_{y_1, y_2, \cdots, y_n \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{d} \|y_i - x_i^{(k)}\|^2 - \gamma E_n^{(\epsilon)}(y_1, y_2, \cdots, y_n) \tag{77}$$

The above optimization is convex with solution $y_i^* = x_i^{(k-1)}$ for all $i \in [n]$, as long as the learning rate γ is sufficiently small.

Proof. Let the objective of the proximal step (77) defined as,

$$H(y_1, \cdots, y_n) \coloneqq \frac{1}{2} \sum_{i=1}^{n} \|y_i - x_i^{(k)}\|^2 - \gamma E_n^{(\epsilon)}(y_1, \cdots, y_n). \tag{78}$$

And for ease of notation, let $Y = (y_1, \cdots, y_n) \in \left(\mathbb{R}^d\right)^n$. It is easy to observe that,

$$\nabla^2 H(Y) = I_{n \times d} - \gamma \nabla^2 E_n^{(\epsilon)}(Y).$$

Thus, the optimization problem (11) is convex as long as we can prove an upper bound on the spectral norm of $\nabla^2 E_n^{(\epsilon)}(Y)$ and choose a small learning rate for the inverse gradient map γ. From this point onward, we drop the subscript/superscript ϵ to streamline the notation. Thus W is defined as

$$W(z) = -\frac{m}{(\|z\|^2 + \epsilon)^{m/2}} + \frac{1}{2}\|z\|^2$$

with direct calculations we have,

$$\nabla W(z) = \left(1 + m^2 (r^2 + \epsilon)^{-\frac{m}{2}-1}\right) z,$$

where $r \coloneqq \|z\|$. Again with differentiation,

$$\nabla^2 W(z) = \left(1 + m^2 (r^2 + \epsilon)^{-\frac{m}{2}-1}\right) I_d - m^2 (m+2)(r^2 + \epsilon)^{-\frac{m}{2}-2} z z^\top.$$

Hence, for every unit vector $\nu \in \mathbb{S}^d$,

$$\begin{aligned} \nu^\top \nabla^2 W(z) \nu &\leq 1 + m^2 (r^2 + \epsilon)^{-\frac{m}{2}-1} \\ &\leq 1 + m^2 \epsilon^{-\frac{m}{2}-1}. \end{aligned} \tag{79}$$

Recall that,

$$E_n(Y) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} W(y_i - y_j).$$

Taking derivatives, for each i we have,

$$\nabla_{y_i} E_n(Y) = \frac{1}{n(n-1)} \sum_{j \neq i} \nabla W(y_i - y_j).$$

Next, for $j \neq i$,

$$\nabla^2_{y_i, y_j} E_n(Y) = \frac{-1}{n(n-1)} \nabla^2 W(y_i - y_j)$$

and,

$$\nabla^2_{y_i} E_n(Y) = \frac{1}{n(n-1)} \sum_{j \neq i} \nabla^2 W(y_i - y_j).$$

Taking an arbitrary block vector $\nu \in \left(\mathbb{R}^d\right)^n$,

$$\begin{aligned}\langle \nu, \nabla^2 E_n(Y)\nu\rangle &= \frac{1}{n(n-1)}\left(\sum_{i=1}^n \nu_i^\top \left(\sum_{j\neq i}\nabla^2 W(y_i-y_j)\right)\nu_i + \sum_{i\neq j}\nu_i^\top\left(-\nabla^2 W(y_i-y_j)\right)\nu_i\right)\\ &= \frac{1}{n(n-1)}\left(\frac{1}{2}\sum_{i\neq j}(\nu_i-\nu_j)^\top \nabla^2 W(y_i-y_j)(\nu_i-\nu_j)\right).\end{aligned}$$

In turn, by spectral bound on each block in (79),

$$\begin{aligned}\langle \nu, \nabla^2 E_n(Y)\nu\rangle &\leq \frac{1}{2n(n-1)}\left(\sum_{i\neq j}\left(\left(1+m^2(\|y_i-y_j\|^2+\epsilon)^{-\frac{m}{2}-1}\right)\right)\|\nu_i-\nu_j\|^2\right)\\ &\leq \frac{1}{2n(n-1)}\left(\sum_{i\neq j}\left(1+m^2\epsilon^{-\frac{m}{2}-1}\right)\|\nu_i-\nu_j\|^2\right) \qquad (80)\\ &= \frac{\left(1+m^2\epsilon^{-\frac{m}{2}-1}\right)}{2n(n-1)}\left(\sum_{i\neq j}\|\nu_i-\nu_j\|^2\right)\\ &\leq \frac{\left(1+m^2\epsilon^{-\frac{m}{2}-1}\right)}{2n(n-1)}\left(2n\sum_i\|\nu_i\|^2\right)\\ &= \frac{1+m^2\epsilon^{-\frac{m}{2}-1}}{n-1}\|\nu\|^2,\end{aligned}$$

where in (80) we used the (79.

Hence, for the operator norm of $\nabla^2 E_n(Y)$ we infer that,

$$\|\nabla^2 E_n(Y)\|_2 \leq \frac{1+m^2\epsilon^{-\frac{m}{2}-1}}{n-1}.$$

Thus, by choosing any sufficiently small $\gamma \in (0, \frac{n-1}{1+m^2\epsilon^{-\frac{m}{2}-1}})$,

$$\nabla^2 H(Y) \succ 0,$$

and objective of the proximal step (78) is strongly convex and hence is uniquely minimized.

Considering the optimal solution $(y_1^*, y_2^*, \cdots, y_n^*)$ to (78), by first order optimality condition,

$$\nabla_{y_i} H(y_1^*, y_2^*, \cdots, y_n^*) = y_i^* - x_i^{(k)} - \gamma \nabla_{y_i} E_n(y_1^*, y_2^*, \cdots, y_n^*) = 0.$$

Thus,

$$x_i^{(k)} = y_i^* - \gamma \nabla_{y_i} E_n(y_1^*, y_2^*, \cdots, y_n^*),$$

and hence for all $i \in [n]$,

$$y_i^* = x_i^{(k-1)},$$

and proof is concluded. □

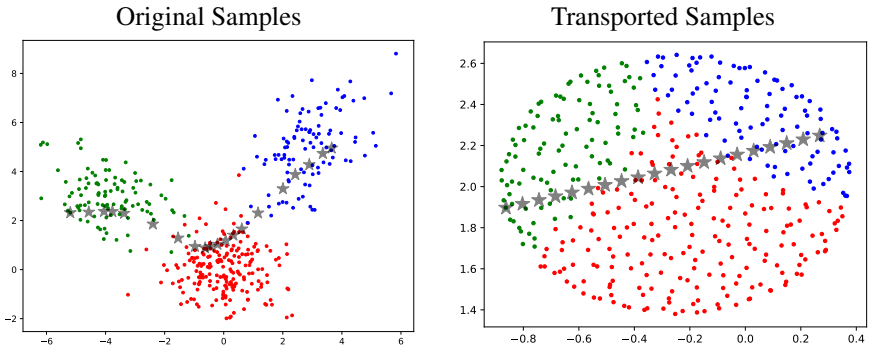


Figure 2: **Interpolation in Latent Space.** *Colored points* are samples drawn from a Gaussian mixture model (left) and their positions after forward optimization (right). The $\star$s are the generated sample y in EFS (right), and their location after backward optimization in EFS (left). The straight line in the latent space (right) is transformed into a curved trajectory (left) that aligns with the data distribution. Colors correspond to different mixture components.

F Experiments

While we have proven EFS can transport an almost uniform distribution to an arbitrary target distribution, our result is asymptotic—it holds in the limit as $n \to \infty$. This asymptotic regime greatly simplifies the theoretical analysis. However, practical applications are inherently constrained to finite n. We bridge this gap with experiments.

As the first estimation-free generative algorithm, EFS needs further research for a comprehensive benchmarking. Our experiments underscore both the practical challenges and the promising potential of this novel generative method. The code is implemented in PyTorch (Paszke et al., 2019) and was executed on a single NVIDIA RTX 6000 GPU with 48GB memory. The total execution time is under 15 minutes.

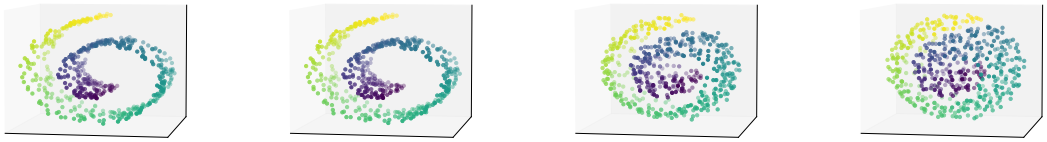
Mixture of Gaussians: Capturing Data Geometry. EFS effectively recovers the underlying data distribution even after transforming into a set of points uniformly distributed on a circle. Figure 2 displays the original samples drawn from a Gaussian mixture (left) and their transformed positions after the EFS forward optimization step (right), where they are mapped onto the circle. New samples are then generated along a straight line on circle. Backward optimization maps these linear samples onto a curved trajectory that closely follows the geometry of the original data distribution. The backward process adaptively distorts pairwise distances to avoid low-density regions. Furthermore, we observe that EFS creatively generate unseen new samples.

Generating samples from Swiss roll. We repeat the experiments on the Swiss roll dataset, similar to those on the Gaussian mixture in Figures 1, and 2. Table 1 summarizes the parameter choices for this dataset. We evaluate: (i) convergence to a uniform distribution via forward optimization, (ii) generation of new samples via backward optimization, and (iii) preservation of data geometry through interpolation.

(i) *Forward optimization:*In Figure 3, we observe that the distribution of data points becomes uniform over the unit ball. Recall that Corollary 3 proves this result in the asymptotic regime as $n \to \infty$. Figure 3 shows that this asymptotic result provides a good approximation even when $n = 500$.

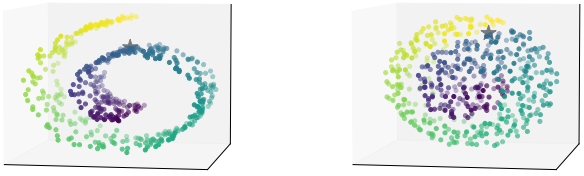
(ii) *Backward optimization:*We assert that EFS can generate new samples from the Swiss roll dataset using a given set of i.i.d. samples. While Theorem 4 establishes that EFS draws samples from the data distribution in the mean-field regime, Figure 4 supports this result in the non-asymptotic setting.

(iii) *Interpolation:* Figure 5 illustrates how backward optimization can generate new samples from the Swiss roll dataset via interpolation. Similar to the Gaussian mixture case shown in



$\{x_i^{(0)}\} \Rightarrow$ $\{x_i^{(36)}\} \Rightarrow$ $\{x_i^{(72)}\} \Rightarrow$ $\{x_i^{(108)}\}$

Figure 3: **Forward convergence for Swiss roll.** Scatter plot of gradient descent iterates $x_i^{(k)}$, defined in (3), initialized at $x_i^{(0)}$ drawn from a Swiss roll with noise level 0.2. As k evolves, the points become uniformly distributed akin to 2d example of Gaussian mixture in Figure 1. Notably, Corollary 3 establishes this convergence in the mean-field regime as $n \to \infty$.



$\{x_i^{(0)}\} \cup \{y^{(0)} = \star\}$ $\{x_i^{(k)}\} \cup \{y^{(k)} = \star\}$

Figure 4: **Sampling:** The black $\star$ is the generated sample $y^{(0)}$; Left: $x_i^{(0)} \overset{i.i.d.}{\sim}$ Swiss roll with noise level 0.2; Right: $x_i^{(k)}$ obtained by forward optimization. Colors: mixture components. The location of the new sample is updated by Backward Optimization in EFS

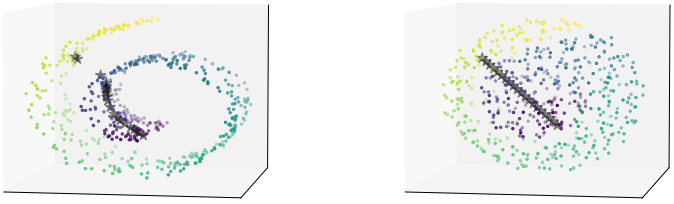


Figure 5: **Interpolation in Latent Space.** *Colored points* are samples drawn from a Swiss roll (left) and their positions after forward optimization (right). The $\star$s are the generated sample $y^{(0)}$ in EFS (right), and their location after applying backward step of EFS (left). The straight line in the latent space (right) is transformed into a curved trajectory (left) that aligns with the data distribution. We observe that the algorithm avoids generating data from no density regions.

Figure 2, we observe that EFS respects the data density and avoids generating samples in low-density regions.

MNIST: Generating New Samples. EFS can generate new MNIST images (LeCun et al., 1998). Theorem 4 requires high exponents in the range $d - 2 \leq m < d$. For MNIST with $d = 784$, this exponent leads to numerical overflow due to limited floating-point precision. To avoid this issue, we reduce the data dimension before applying EFS. Specifically, we use a convolutional autoencoder (Masci et al., 2011) to compress the data into a 15-dimensional latent space (see Appendix for architectural and training details). We then apply encoding $\rightarrow$ EFS $\rightarrow$ decoding to generate new samples. Figure 6 shows generated samples alongside their nearest neighbors (in Euclidean distance) from decoded original samples $x_1^{(0)}, \ldots, x_n^{(0)}$. The generated digits exhibit stylistic variations distinct from their closest training examples, with minimum Euclidean distance of 0.7508. To emphasize subtle differences, we include an animated image that alternates between the generated sample and its nearest neighbor (open with Adobe Reader). See Table 1 for details.

Figure 6: **Generative MNIST.** *First row:* Images generated by EFS. These are obtained by applying forward and backward optimization in the latent space of an autoencoder, followed by decoding into the image space. *Second row:* The closest decoded training samples $\{x_i^{(0)}\}_{i=1}^n$ for each generated image, determined by Euclidean distance. *Third row:* An animation illustrating subtle differences between the generated images and their nearest neighbors (use Adobe Reader to see). *Note:* The minimum Euclidean distance between each generated image and its nearest neighbor is 0.75.

MNIST: Interpolation Comparison. Recall that in the previous experiment, we used an autoencoder for dimensionality reduction. Notably, autoencoders can generate samples with interpolation in the latent space. To highlight differences, we compare the quality of the autoencoder samples with those produced by EFS.

For EFS, we generate new samples by linearly interpolating between uniformly distributed samples after forward optimization as $y^{(k)} = (1-t)x_i^{(k)} + tx_j^{(k)}$. Then, we apply the backward optimization to $y^{(k)}$. Similarly, autoencoder samples are obtain by decoding $(1-t)x_i^{(0)} + tx_j^{(0)}$ where $x_i^{(0)}$ are encoded samples in the latent space. Figure 7 shows generate samples with these two different strategies. Observe interpolation in the autoencoder's latent space often leads to unrealistic outputs that do not resemble valid digits. As illustrated in the cartoon of Figure 7, such interpolation may cross regions of low data density. In stark contrast, EFS effectively avoids generating samples from these low-density regions, as similarly observed for Gaussian mixtures in Figure 6.

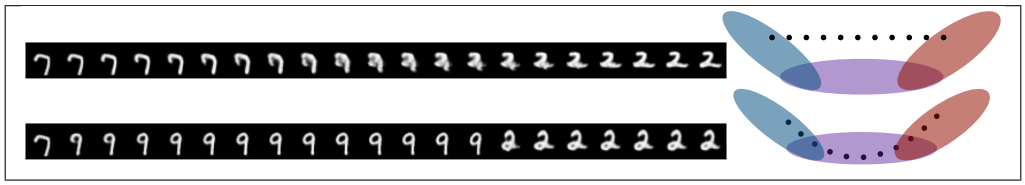


Figure 7: **Interpolation Comparison.** In the interpolation between images 7 and 2, the intermediate autoencoder outputs do not resemble valid digits (1st row), whereas our method produces a transition via $7 \rightarrow 9 \rightarrow 2$ (2nd row). On the right, we visualize the interpolation paths cross low density regions for autoencoder, while moves along the underlying digit clusters for EFS. Moreover, for EFS, it can be observed that different images are generated for each digit, illustrating that interpolation paths evolve smoothly within clusters.

F.1 Autoencoder details

As previously mentioned, the experiments in Section F employ an autoencoder. In this section, we provide further details on the architecture and training procedure of the autoencoder. The model is

	γ	k	T	β	ϵ	s	n
Gaussian mixtures	0.1	31	300	0.1	0.001	1	400
MNIST	0.05	120	300	0.01	0.001	$d-2$	15000
Swiss roll	0.05	120	300	0.1	0.001	$d-2$	500

Table 1: Parameters of EFS

a standard convolutional autoencoder consisting of two encoding layers followed by two decoding layers. The structure of the encoder is summarized in Listing 1. Importantly, the encoder serves as a dimensionality reduction mechanism. Without this reduction, computing $W^{(s)}$, as required in EFS, becomes numerically infeasible.

F.2 Training Protocol

We trained the autoencoder on the MNIST dataset using the Adam optimizer with a learning rate of 10^{-3}, a batch size of 250, and for 120 epochs. The training loss, measured by mean squared error (MSE), decreased to 0.008 after 120 epochs. Training was conducted on an NVIDIA RTX 6000 GPU with 48 GB of memory. The implementation was done in PyTorch (Paszke et al., 2019) (see the notebook `neurips-figures4-5.ipynb` for details). No preprocessing was applied to the data.

```
Autoencoder(
  (encoder): Sequential(
    (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1,
        1))
    (1): ReLU(inplace=True)
    (2): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1,
         1))
    (3): ReLU(inplace=True)
    (4): Conv2d(32, 15, kernel_size=(7, 7), stride=(1, 1))
  )
  (decoder): Sequential(
    (0): ConvTranspose2d(15, 32, kernel_size=(7, 7), stride=(1, 1))
    (1): ReLU(inplace=True)
    (2): ConvTranspose2d(32, 16, kernel_size=(3, 3), stride=(2, 2),
        padding=(1, 1), output_padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): ConvTranspose2d(16, 1, kernel_size=(3, 3), stride=(2, 2),
        padding=(1, 1), output_padding=(1, 1))
    (5): Sigmoid()
  )
)
```

Listing 1: Autoencoder