SIAM J. MATH. DATA SCI.
Vol. 6, No. 4, pp. 1205–1235

Lipschitz-Regularized Gradient Flows and Generative Particle Algorithms for High-Dimensional Scarce Data[*]

Hyemin Gu[†], Panagiota Birmpa[‡], Yannis Pantazis[§], Luc Rey-Bellet[†], and Markos A. Katsoulakis[†]

Abstract. We have developed a new class of generative algorithms capable of efficiently learning arbitrary target distributions from possibly scarce, high-dimensional data and subsequently generating new samples. These particle-based generative algorithms are constructed as gradient flows of Lipschitz-regularized Kullback–Leibler or other f-divergences. In this framework, data from a source distribution can be stably transported as particles towards the vicinity of the target distribution. As a notable result in data integration, we demonstrate that the proposed algorithms accurately transport gene expression data points with dimensions exceeding 54K, even though the sample size is typically only in the hundreds.

Key words. gradient flow, generative modeling, information theory, optimal transport, particle algorithms, data integration

MSC codes. 35Q84, 49Q22, 62B10, 65C35, 68T07, 94A17

DOI. 10.1137/23M1587841

1. Introduction and main results.

We construct new algorithms that are capable of efficiently transporting samples from a source distribution to a target data set. The transportation mechanism is built as the gradient flow (in probability space) for Lipschitz-regularized divergences, [16, 5, 7]. Samples are viewed as particles and are transported along the gradient of the discriminator of the divergence towards the target data set. Lipschitz regularized f-divergences interpolate between the Wasserstein metric and f-divergences and provide a flexible family of loss functions to compare nonabsolutely continuous probability measures. In machine learning one needs to build algorithms to handle target distributions Q which

[*] Received by the editors July 19, 2023; accepted for publication (in revised form) June 11, 2024; published electronically December 9, 2024.
https://doi.org/10.1137/23M1587841
Funding: This research was partially supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-21-1-0354. The research of the first, fourth, and fifth authors was partially supported by the National Science Foundation (NSF) under grants DMS-2008970, DMS-2307115, and TRIPODS CISE-1934846. The research of the second author was partially supported by the National Science Foundation (NSF) under grant DMS-2008970. The work of the third author was partially supported by the Hellenic Foundation for Research and Innovation (HFRI) under project 4753.

[†] Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, MA 01003 USA (hgu@umass.edu, luc@umass.edu, markos@umass.edu).
[‡] Department of Actuarial Mathematics and Statistics, Heriot-Watt University, Edinburgh EH14 4AS, Scotland (P.Birmpa@hw.ac.uk).
[§] Institute of Applied and Computational Mathematics, FORTH, GR 700 13 Heraklion, Crete, Greece (pantazis@iacm.forth.gr).

are singular, either by their intrinsic nature such as probability densities concentrated on low-dimensional structures and/or because Q is usually only known through N samples. The Lipschitz regularization also provides numerically stable, mesh free, particle algorithms that can act as a generative model for high-dimensional target distributions. The proposed generative approach is validated on a wide variety of datasets and applications ranging from heavy-tailed distributions and image generation to gene expression data integration, including problems in very high dimensions and with scarce target data. In this introduction we provide an outline of our main results, background material, and related prior work.

Generative modeling. In generative modeling, which is a form of unsupervised learning, a data set $(X^{(i)})_{i=1}^N$ from an unknown "target" distribution Q is given and the goal is to construct an approximating model in the form of a distribution $P \approx Q$ which is easy to simulate, with the goal to generate additional, inexpensive, approximate samples from the distribution Q. Succinctly, the goal of generative modeling is to learn the target distribution Q from input data $(X^{(i)})_{i=1}^N$. This is partly in contrast to sampling, where typically Q is known up to normalization. In the last 10 years, generative modeling has been revolutionized by new innovative algorithms taking advantage of neural networks (NNs) and more generally deep learning. On one hand NNs provide enormous flexibility to parametrize functions and probabilities and on the other, lead to efficient optimization algorithms in function spaces. Generative adversarial networks (GANs) [22, 4], for example, are able to generate complex distributions and are quickly becoming a standard tool in image analysis, medical data, cosmology, computational chemistry, materials science, and so on. Many other algorithms have been proposed since, such as normalizing flows [33, 13], diffusion models [54, 27], score-based generative flows [57, 58], variational autoencoders [31], and energy-based methods [34].

Information theory, divergences, and optimal transport. Divergences such as Kullback–Leibler (KL) and f-divergences, and probability metrics such as Wasserstein, provide a notion of "distance" between probability distributions, thus allowing for comparison of models with one another and with data. Divergences and metrics are used in many theoretical and practical problems in mathematics, engineering, and the natural sciences, ranging from statistical physics, large deviations theory, uncertainty quantification, partial differential equations (PDE) and statistics to information theory, communication theory, and machine learning. In particular, in the context of GANs, the choice of objective functional (in the form of a probability divergence plus a suitable regularization) plays a central role.

A very flexible family of divergences, the (f,Γ)-divergences, were introduced in [5]. These new divergences interpolate between f-divergences (e.g., KL, α-divergence, Shannon–Jensen) and Γ-Integral Probability Metrics (IPM) like 1-Wasserstein and MMD distances (where Γ is the 1-Lipschitz functions or an Reproducing Kernel Hilbert Space (RKHS) 1-ball, respectively). Another way to think of Γ is as a regularization to avoid overfitting, built directly in the divergenc;e see, for instance, structure-preserving GANs [7]. In this paper, we focus on one specific family which we view as a Lipschitz regularization of the KL-divergence (or f-divergences) or as an entropic regularization of the 1-Wasserstein metric. In this context, the interpolation is mathematically described by the Infimal Convolution formula

$$D_f^{\Gamma_L}(P\|Q) = \inf_{\gamma\in\mathcal{P}(\mathbb{R}^d)} \left\{ L \cdot W^{\Gamma_1}(P,\gamma) + D_f(\gamma\|Q) \right\} , \tag{1.1}$$

where $\mathcal{P}(\mathbb{R}^d)$ is the space of all Borel probability measures on $\mathbb{R}^d$ and $\Gamma_L = \{\phi : \mathbb{R}^d \to \mathbb{R} : |\phi(x) - \phi(y)| \leq L|x-y| \text{ for all } x, y\}$ is the space of Lipschitz continuous functions with Lipschitz constant bounded by L (note that $L\Gamma_1 = \Gamma_L$). Furthermore, $W^{\Gamma_1}(P,Q)$ denotes the 1-Wasserstein metric with transport cost $|x-y|$ which is an integral probability metric, and has the dual representation

$$W^{\Gamma_1}(P,Q) = \sup_{\phi\in\Gamma_1} \{E_P[\phi] - E_Q[\phi]\} . \tag{1.2}$$

Finally, if $f : [0,\infty) \to \mathbb{R}$ is strictly convex and lower-semicontinuous with $f(1) = 0$ the f-divergence of P with respect to Q is defined by $D_f(P\|Q) = E_Q[f(\frac{dP}{dQ})]$ if $P \ll Q$ and set to be $+\infty$ otherwise. The new divergences inherit desirable properties from both objects, e.g.,

$$0 \leq D_f^{\Gamma_L}(P\|Q) \leq \min\left\{D_f(P\|Q), L \cdot W^{\Gamma_1}(P,Q)\right\} . \tag{1.3}$$

The Lipschitz-regularized f-divergences (1.1) admit a dual variational representation,

$$D_f^{\Gamma_L}(P\|Q) := \sup_{\phi\in\Gamma_L} \left\{ E_P[\phi] - \inf_{\nu\in\mathbb{R}} \{\nu + E_Q[f^*(\phi-\nu)]\} \right\} , \tag{1.4}$$

where f^* is the Legendre transform of f. Some of the important properties of Lipschitz regularized f-divergences, which summarizes results from [16, 5] are given in supplementary material section SM1. Typical examples of f-divergences include the KL-divergence with $f_{\text{KL}}(x) = x\log x$, and the α-divergences with $f_\alpha(x) = \frac{x^\alpha - 1}{\alpha(\alpha-1)}$. The corresponding Legendre transforms are $f^*_{\text{KL}}(y) = e^{y-1}$ and $f^*_\alpha \propto y^{\frac{\alpha}{(\alpha-1)}}$. In the KL case the infimum over ν can be solved analytically and yields the Lipschitz-regularized Donsker–Varadhan formula with a $\log E_Q[e^\phi]$ term; see [6] for more on variational representations.

Gradient flows in probability space. The groundbreaking work of [30, 47] recasted the Fokker–Planck (FP) and the porous media equations as gradient flows in the 2-Wasserstein space of probability measures. More specifically, the FP equation can be thought as the gradient flow of the KL divergence

$$\partial_t p_t = \nabla \cdot \left(p_t \nabla \frac{\delta D_{KL}(p_t\|q)}{\delta p_t} \right) = \nabla \cdot \left(p_t \nabla \log\left(\frac{p_t}{q}\right)\right) , \tag{1.5}$$

where p_t and q are the densities at time t and the stationary density, respectively. A similar result relates weighted porous media equation and gradient flows for f divergences [47]. This probabilistic formulation allowed the use of such gradient flows and related perspectives to build new Machine Learning concepts and tools. For instance, the FP equation plays a key role in both generative modeling and in sampling.

In the remaining part of this introduction we provide an outline of our main results, as well as a discussion of related prior work.

Lipschitz-regularized gradient flows in probability space. From a generative modeling perspective, where Q is known only through samples–-and may not have a density, especially if Q is concentrated on a low-dimensional structure–-one cannot use gradient flows such as (1.5) without further regularization. For instance, related generative methods such as score matching and diffusion models regularize data by adding noise [57, 58]. Here we propose a different and complementary approach by regularizing the divergence directly and without adding noise to the data. We propose gradient flows for the Lipschitz-regularized divergences (1.4) of the form

$$\partial_t P_t = \operatorname{div}\left(P_t \nabla \frac{\delta D_f^{\Gamma_L}(P_t\|Q)}{\delta P_t} \right), \tag{1.6}$$

for an initial (source) probability measure P_0 and an equilibrium (target) measure Q, for P_0, Q in the Wasserstein space $\mathcal{P}_1(\mathbb{R}^d) = \left\{P \in \mathcal{P}(\mathbb{R}^d) : \int |x| dP(x) < \infty\right\}$. We want to emphasize that $\mathcal{P}_1(\mathbb{R}^d)$ includes singular measures such as empirical distributions constructed from data. In section 2 we prove the first variation formula

$$\frac{\delta D_f^{\Gamma_L}(P\|Q)}{\delta P} = \phi^{L,*} = \underset{\phi\in\Gamma_L}{\operatorname{argmax}} \left\{ E_P[\phi] - \inf_{\nu\in\mathbb{R}} (\nu + E_Q[f^*(\phi-\nu)]) \right\}. \tag{1.7}$$

The optimal $\phi^{L,*}$ in (1.7) (called the discriminator in the GAN literature) in the variational representation of the divergence (1.4) serves as a potential to transport probability measures, leading to the *transport/variational* PDE reformulation of (1.6):

$$\begin{aligned} &\partial_t P_t + \operatorname{div}(P_t v_t^L) = 0\,, \quad P_0 = P \in \mathcal{P}_1(\mathbb{R}^d)\,, \\ &v_t^L = -\nabla \phi_t^{L,*}\,, \quad \phi_t^{L,*} = \underset{\phi\in\Gamma_L}{\operatorname{argmax}} \left\{ E_{P_t}[\phi] - \inf_{\nu\in\mathbb{R}} (\nu + E_Q[f^*(\phi-\nu)]) \right\}, \end{aligned} \tag{1.8}$$

where we remind that $\Gamma_L = \{\phi : \mathbb{R}^d \to \mathbb{R} : |\phi(x) - \phi(y)| \leq L|x-y| \text{ for all } x, y\}$. This transport/variational PDE should be understood in a weak sense since P_t and Q are not necessarily assumed to have densities. However, the purpose of this paper is not to develop the PDE theory for this new gradient flow but rather to first establish its computational feasibility through associated particle algorithms, explore its usefulness in generative modeling for problems with high-dimensional scarce data, and overall computational efficiency and scalability. Given sufficient regularity, along a trajectory of a smooth solution P_t of (1.8) we have the following dissipation identity:

$$\frac{d}{dt} D_f^{\Gamma_L}(P_t\|Q) = -I_f^{\Gamma_L}(P_t\|Q) \leq 0, \quad \text{where} \quad I_f^{\Gamma_L}(P_t\|Q) = E_{P_t}\left[|\nabla \phi_t^{L,*}|^2\right] \tag{1.9}$$

and $I_f^{\Gamma_L}(P\|Q)$ is a Lipschitz-regularized version of the Fisher Information. Due to the transport/variational PDE (1.8) $I_f^{\Gamma_L}(P\|Q)$ can be interpreted as a total kinetic energy; see section 2, and section 3 for its practical importance in the particle algorithms introduced next.

Lipschitz-regularized generative particle algorithms (GPA). In the context of generative models, the target Q and the generative model P_t in (1.6) are available only through their samples and associated empirical distributions. However, as it can be seen from (1.3) the divergence $D_f^{\Gamma_L}(P\|Q)$ can compare directly singular distributions (e.g., empirical measures) without need for extra regularization such as adding noise to our models. For precisely this reason the proposed gradient flow (1.6) is a natural mathematical object to consider as a generative model.

From a computational perspective, it becomes feasible to solve high-dimensional transport PDE such as (1.6) when considering the Lagrangian formulation of the transport PDE in (1.8), i.e., the ODE/variational problem

$$\begin{aligned} &\frac{d}{dt}Y_t = v_t^L(Y_t) = -\nabla\phi_t^{L,*}(Y_t)\,, \quad Y_0 \sim P\,, \\ &\phi_t^{L,*} = \underset{\phi\in\Gamma_L}{\operatorname{argmax}}\left\{E_{P_t}[\phi] - \inf_{\nu\in\mathbb{R}}\{\nu + E_Q[f^*(\phi-\nu)]\}\right\}. \end{aligned} \tag{1.10}$$

In order to turn (1.10) into a particle algorithm we need the following ingredients:

- Consider samples $(X^{(i)})_{i=1}^N$ from the target Q and $(Y^{(i)})_{i=1}^M$ samples from an initial (source) distribution $P = P_0$. In this case for the corresponding empirical measures $\widehat{Q}^N$ and $\widehat{P}^M$ we will consider the gradient flow (1.6) for $D_f^{\Gamma_L}(\widehat{P}^M\|\widehat{Q}^N)$. A key observation in our algorithms is that the divergence $D_f^{\Gamma_L}(\widehat{P}^M\|\widehat{Q}^N)$ is always well-defined and finite due to Lipschitz regularization and (1.3).
- Corresponding estimators for the objective functional in the variational representation of the divergence $D_f^{\Gamma_L}(\widehat{P}^M\|\widehat{Q}^N)$ (see (1.4) and also (1.10)):

$$E_{\widehat{P}^M}[\phi] - \inf_{\nu}(\nu + E_{\widehat{Q}^N}[f^*(\phi-\nu)]) = \frac{\sum_{i=1}^M \phi(Y_n^{(i)})}{M} - \inf_{\nu\in\mathbb{R}}\left\{\nu + \frac{\sum_{i=1}^N f^*(\phi(X^{(i)})-\nu)}{N}\right\}.$$

- The function space Γ_L in (1.10) is approximated by a space of neural network approximations Γ_L^{NN}. The Lipschitz condition can be implemented via neural network spectral normalization as discussed in section 3.
- The transport ODE in (1.10) is discretized in time using an Euler or a higher order scheme; see section 3. Furthermore, the gradient $\nabla\phi_t^{L,*}$ is evaluated by automatic differentiation of NNs at the positions of the particles.

By incorporating these approximations we derive from (1.10), upon Euler time discretization the *Lipschitz-regularized GPA*:

$$\begin{aligned} &Y_{n+1}^{(i)} = Y_n^{(i)} - \Delta t\nabla\phi_n^{L,*}(Y_n^{(i)})\,, \quad Y_0^{(i)} = Y^{(i)}\,, Y^{(i)} \sim P\,, \quad i = 1,\ldots,M, \\ &\phi_n^{L,*} = \underset{\phi\in\Gamma_L^{NN}}{\operatorname{argmax}}\left\{\frac{\sum_{i=1}^M \phi(Y_n^{(i)})}{M} - \inf_{\nu\in\mathbb{R}}\left\{\nu + \frac{\sum_{i=1}^N f^*(\phi(X^{(i)})-\nu)}{N}\right\}\right\}, \end{aligned} \tag{1.11}$$

Besides the transport aspect of (1.11), it can be also viewed as a new generative algorithm, where the input is samples $(X^{(i)})_{i=1}^N$ from the "target" Q. Initial data, usually referred to as "source" data, $(Y_0^{(i)})_{i=1}^M$ from P are transported via (1.11), after time $T = n_T\Delta t$, where n_T is the total number of steps, to a new set of generated data $(Y_{n_T}^{(i)})_{i=1}^M$ that approximate samples from Q. See, for instance, the demonstration in Figure 1.

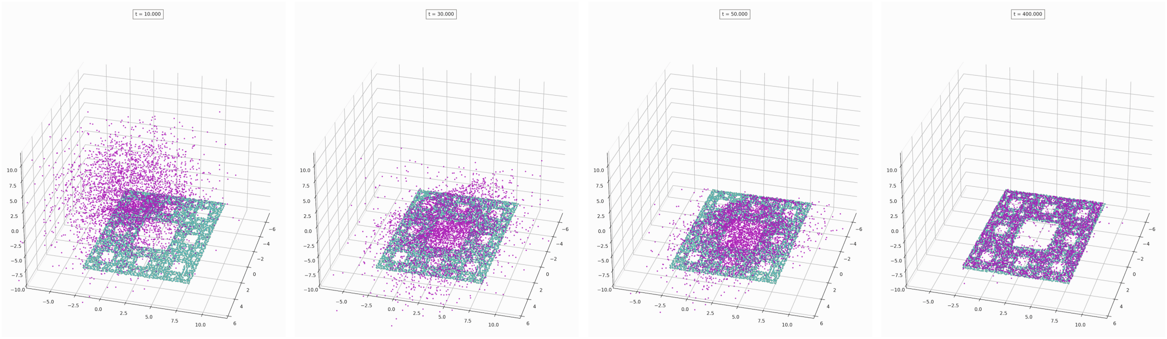


Figure 1. *Sierpinski carpet embedded in three dimensions. Source data (purple particles) are transported via GPA close to the target data (cyan particles). The target particles were sampled from a Sierpinski carpet of level* 4 *by omitting all finer scales. See Figure* 7 *for a related two-dimensional (*2D*) demonstration and a comparison to GANs.*

In analogy to (1.10), this Lagrangian point of view has been recently introduced to write the solution of the FP equation (1.5) as the density of particles evolving according to its Lagrangian formulation, [41],

$$\frac{d}{dt}Y_t = v_t(Y_t) = \nabla \log q(Y_t) - \nabla \log p_t(Y_t)\,, \quad \text{where } Y_t \sim P_t\,. \tag{1.12}$$

In fact, in [58], the authors proposed the deterministic probability flow (1.12) as an alternative to generative stochastic samplers for score generative models due to advantages related to obtaining better statistical estimators. We note here that the score term $\nabla \log p_t(Y_t)$ in (1.12) is not a priori known and can be estimated by score-based methods [28]. In practice, these Lagrangian tools are used both for generation [58] as well as sampling [50, 9].

Main contributions. As discussed earlier, the purpose of this paper is to introduce the new Lipschitz-regularized gradient flow (1.6), in section 2, and subsequently establish its computational feasibility through associated particle algorithms, its computational efficiency and scalability, and explore its usefulness in generative modeling for problems with high-dimensional scarce data. Towards these goals our main findings can be summarized as follows.

1. *GPA for generative modeling with scarce data.* We demonstrate that our proposed GPA, introduced in section 3, can learn distributions from very small data sets, including MNIST and other benchmarks, often supported on low-dimensional structures; see Figure 1. In section 4 we discuss generalization properties of GPA and strategies for mitigating memorization of target data, which has proved to be a significant and ongoing challenge in generative modeling. In section 8 we compare GPA to GANs and score-based generative models (SGM) in a series of examples and show GPA to be an effective data-augmentation tool.
2. *Lipschitz-regularization.* We demonstrate that Lipschitz-regularized divergences provide a well-behaved pseudo-metric between models and data or data and data. They remain finite under very broad conditions, making the training of GPA (1.11) on data

always well-defined and numerically stable. In fact, Lipschitz regularization corresponds to effectively imposing an advection-type Courant–Friedrichs–Lewy (CFL) numerical stability condition on the FP PDE (1.5) through the Lipschitz-regularization parameter L in (1.6). The example in section 6 demonstrates empirically that the selection of L is important.

3. *Choice of f-divergence in* (1.6). Although KL is often a natural choice, a careful selection of f-divergences, for example the family of α-divergences where $f_\alpha = \frac{x^\alpha - 1}{\alpha(\alpha-1)}$, will allow for training that is numerically stable, including examples with heavy-tailed data; see section 7.
4. *Latent-space GPA for very high-dimensional problems.* GPA can be effective even for scarce data sets in high dimensions. We provide a demonstration where we integrate (real) gene expression data sets exceeding 50,000 dimensions. The goal of data transportation in this context is to mitigate batch effects between studies of different groups of patients; see section 9. From a practical perspective, to be able to operate in such high-dimensions we need a latent-space representation of the data and subsequently we use GPA to transport particles in the latent space. In section 5 we provide related *performance guarantees* using a new Data Processing Inequality (DPI) for Lipschitz-regularized divergences.

Related work. Our approach is inspired by the MMD and KALE gradient flows from [3, 21] based on an entropic regularization of the MMD metrics, and related work using the Kernelized Sobolev Discrepancy [44]. Furthermore, the recent work of [16, 5] built the mathematical foundations for a large class of new divergences which contains the Lipschitz regularized f-divergences and used them to construct GANs, and in particular, symmetry preserving GANs [7]. Also related is the Sinkhorn divergence [19] which is a different entropic regularization of the 2-Wasserstein metrics. Lipschitz regularizations and the related spectral normalization have been shown to improve the stability of GANs [43, 4, 24]. Our particle algorithms share similarities with GANs [22, 4], sharing the same discriminator but having a different generator step. They are also broadly related to continuous-time generative algorithms, such as continuous-time normalizing flows (NF) [12, 33, 13], diffusion models [54, 27] and score-based generative flows [57, 58]. However, the aforementioned continuous-time models, along with variational autoencoders [31] and energy based methods [34], are mostly KL/likelihood-based.

On the other hand, particle gradient flows such as the ones proposed here can be classified as a separate class within implicit generative models. Within such generative models that include GANs, there is more flexibility in selecting the loss function in terms of a suitable divergence or probability metric, enabling the direct comparison of even mutually singular distributions, e.g., [4, 24]. Gradient flows in probability spaces related to the KL divergence, such as the FP equations and Langevin dynamics [51, 17] or Stein variational gradient descent [38, 37, 39], form the basis of a variety of sampling algorithms when the target distribution Q has a known density (up to normalization). The weighted porous media equations form another family of gradient flows based on α-divergences, e.g., [47, 1, 15, 61] which are very useful in the presence of heavy tails. Our gradient flows are Lipschitz-regularizations of such classical PDE's (FP and porous medium equations). Finally, deterministic particle methods and associated probabilistic flows of ODEs such as the ones derived here for Lipschitz-regularized gradient flows, were considered in recent works for classical KL-divergences and associated FP equations as sampling tools [41, 9], for Bayesian inference [50] and as generative models [58].

2. Lipschitz-regularized gradient flows.

In this section we introduce the concept of Lipschitz-regularized gradient flows in probability space, including the key computation of the first variation of Lipschitz-regularized divergences. This will allow us to build effective particle-based algorithms in section 3. Indeed, given a target probability measure Q, we build an evolution equation for probability measures based on the Lipschitz regularized f-divergences $D_f^{\Gamma_L}(P\|Q)$ in (1.4), by considering the PDE

$$\partial_t P_t = \operatorname{div}\left(P_t \nabla \frac{\delta D_f^{\Gamma_L}(P_t\|Q)}{\delta P_t}\right), \quad \text{with initial condition} \quad P_0 \in \mathcal{P}_1(\mathbb{R}^d), \tag{2.1}$$

where $\frac{\delta D_f^{\Gamma_L}(P\|Q)}{\delta P}$ is the first variation of $D_f^{\Gamma_L}(P\|Q)$, to be discussed below in Theorem 2.1. An advantage of the Lipschitz regularized f-divergences is its ability to compare singular measures and thus (2.1) needs to be understood in a weak sense. For this reason we use the probability measure P_t notation in (2.1), instead of density notation p_t as in the FP equation (1.5). In the formal asymptotic limit $L \to \infty$ and if $P \ll Q$, (2.1) yields the FP equation (1.5) (for KL divergence) and the weighted porous medium equation (for α-divergences) [47, 15]; see Remark 2.6. Note that the purpose of this paper is not to develop the PDE theory for (2.1) but rather to first establish its computational feasibility through associated particle algorithms and demonstrate its usefulness in generative modeling.

Theorem 2.1 (first variation of Lipschitz regularized f-divergences). *Assume f is superlinear, strictly convex and $P, Q \in \mathcal{P}_1(\mathbb{R}^d)$. We define*

$$\phi^{L,*} := \operatorname*{argmax}_{\phi\in\Gamma_L}\left\{E_P[\phi] - \inf_{\nu\in\mathbb{R}}\{\nu + E_Q[f^*(\phi-\nu)]\}\right\}, \tag{2.2}$$

where the optimizer $\phi^{L,*} \in \Gamma_L$ exists, is defined on supp$(P) \cup$ supp(Q), and is unique up to a constant. Subsequently, we extend $\phi^{L,*}$ in all of $\mathbb{R}^d$ using (2.6). Let ρ be a signed measure of total mass 0, and let $\rho = \rho_+ - \rho_-$, where $\rho_\pm \in \mathcal{P}_1(\mathbb{R}^d)$ are mutually singular, i.e., there exist two disjoint sets $X_\pm$ such that $\rho_\pm(A) = \rho_\pm(A \cap X_\pm)$ for all measurable sets A.
If $P + \epsilon\rho \in \mathcal{P}_1(\mathbb{R}^d)$ for sufficiently small $\epsilon > 0$, then

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\left(D_f^{\Gamma_L}(P+\epsilon\rho\|Q) - D_f^{\Gamma_L}(P\|Q)\right) = \int \phi^{L,*} d\rho\,. \tag{2.3}$$

Then we write

$$\frac{\delta D_f^{\Gamma_L}(P\|Q)}{\delta P}(P) = \phi^{L,*}\,. \tag{2.4}$$

Remark 2.2. The first variation of the Lipschitz-regularized KL divergence given in Theorem 2.1 is defined on $\mathcal{P}_1(\mathbb{R}^d)$ which includes singular measures such as empirical distributions.

On the other hand, the classical FP (1.5) (where $L=\infty$) can be rewritten in a gradient flow formulation,

$$\begin{aligned}&\partial_t p_t = \nabla\cdot(\nabla\phi^*(x,t)p_t)\,, \quad \text{where}\\ &\phi_t^* = \log\frac{p_t(x)}{q(x)} = \underset{\phi\in C_b(\mathbb{R}^d)}{\operatorname{argmax}}\left\{E_{P_t}[\phi] - \inf_{\nu\in\mathbb{R}}\left\{\nu + E_Q[e^{\phi-\nu-1}]\right\}\right\}\end{aligned}\tag{2.5}$$

is built on the first variation of the (unregularized) KL divergence given by

$$\frac{\delta D_{KL}(P\|Q)}{\delta P} = \log\frac{dP}{dQ} = \phi^* = \underset{\phi\in C_b(\mathbb{R}^d)}{\operatorname{argmax}}\left\{E_P[\phi] - \inf_{\nu\in\mathbb{R}}\left\{\nu + E_Q[e^{\phi-\nu-1}]\right\}\right\},$$

where $C_b(\mathbb{R}^d)$ is the space of all bounded continuous functions on $\mathbb{R}^d$. In this case, the first variation is defined on the space of probability measures which are absolutely continuous with respect to Q.

The proof of Theorem 2.1 is partly based on the next lemma (proof in supplementary material section SM2.1).

Lemma 2.3. *Let f be superlinear and strictly convex, and let $P,Q\in\mathcal{P}_1(\mathbb{R}^d)$. For $y\notin \operatorname{supp}(P)\cup\operatorname{supp}(Q)$, we define*

$$\phi^{L,*}(y) = \sup_{x\in\operatorname{supp}(Q)}\left\{\phi^{L,*}(x) + L|x-y|\right\}.\tag{2.6}$$

Then $\phi^{L,*}$ is Lipschitz continuous on $\mathbb{R}^d$ with Lipschitz constant L and $\phi^{L,*} = \sup\{h(x): h\in\Gamma_L,\ h(y)=\phi^{L,*}(y), \text{for every } y\in\operatorname{supp}(Q)\}$.

See Remark 2.4, part (b) for the algorithmic intepretation of this lemma.

Proof of Theorem 2.1. If $\rho=\rho_+-\rho_-$, we may assume (Jordan decomposition) that $\rho_\pm\in\mathcal{P}(X)$ are mutually singular so there exist two disjoint sets $X_\pm$ such that $\rho_\pm(A)=\rho_\pm(A\cap X_\pm)$ for all measurable sets A. The measure $P+\epsilon(\rho_+-\rho_-)$ has total mass 1 but to be a probability measure we need that $\epsilon\rho_-(A)\le(P+\epsilon\rho_+)(A)$ holds for all A. This implies that ρ_- is absolutely continuous with respect to P. Indeed if $P(A)=0$, then

$$\epsilon\rho_-(A) = \epsilon\rho_-(A\cap X_-) \le P(A\cap X_-) + \epsilon\rho_+(A\cap X_-) \le P(A) = 0.\tag{2.7}$$

If $P+\epsilon\rho\in\mathcal{P}_1(\mathbb{R}^d)$, the divergence is finite and thus by (1.4)

$$\begin{aligned}D_f^{\Gamma_L}(P+\epsilon\rho\|Q) &= \sup_{\phi\in\Gamma_L}\left\{E_{P+\epsilon\rho}[\phi] - \inf_{\nu\in\mathbb{R}}\{\nu + E_Q[f^*(\phi-\nu)]\}\right\}\\ &\ge \int\phi^{L,*}\,d(P+\epsilon\rho) - \inf_{\nu\in\mathbb{R}}\left\{\nu + \int f^*(\phi^{L,*}-\nu)dQ\right\}\\ &= \epsilon\int\phi^{L,*}d\rho + D_f^{\Gamma_L}(P\|Q).\end{aligned}\tag{2.8}$$

Thus

$$\liminf_{\epsilon\to0^+}\frac{1}{\epsilon}\left(D_f^{\Gamma_L}(P+\epsilon\rho\|Q) - D_f^{\Gamma_L}(P\|Q)\right) \ge \int\phi^{L,*}d\rho.\tag{2.9}$$

For the other direction let us define $F(\epsilon) = D_f^{\Gamma_L}(P+\epsilon\rho\|Q)$. By Theorem SM1.1 in the supplementary material $F(\epsilon)$ is convex, lower semicontinuous, and finite on $[0,\epsilon_0]$. Due to the convexity of F, it is differentiable on $(0,\epsilon_0)$ except for a countable number of points. If $\phi_\epsilon^{L,*}$ is the optimizer for $D_f^{\Gamma_L}(P+\epsilon\rho\|Q)$ we have, using the same argument as before,

$$D_f^{\Gamma_L}(P+(\epsilon+\delta)\rho\|Q) - D_f^{\Gamma_L}(P+\epsilon\rho\|Q) \geq \delta \int \phi_\epsilon^{L,*} d\rho, \tag{2.10}$$

$$D_f^{\Gamma_L}(P+(\epsilon-\delta)\rho\|Q) - D_f^{\Gamma_L}(P+\epsilon\rho\|Q) \geq -\delta \int \phi_\epsilon^{L,*} d\rho. \tag{2.11}$$

If F is differentiable at ϵ, this implies that

$$\begin{aligned}\int \phi_\epsilon^{L,*} d\rho &\leq \lim_{\delta\to 0} \frac{1}{\delta}\left(D_f^{\Gamma_L}(P+(\epsilon+\delta)\rho\|Q) - D_f^{\Gamma_L}(P+\epsilon\rho\|Q)\right) = F'(\epsilon)\\ &= \lim_{\delta\to 0} \frac{1}{\delta}\left(D_f^{\Gamma_L}(P+\epsilon\rho\|Q) - D_f^{\Gamma_L}(P+(\epsilon-\delta)\rho\|Q)\right) \leq \int \phi_\epsilon^{L,*} d\rho\,. \end{aligned}\tag{2.12}$$

Consequently,

$$F'(\epsilon) = \int \phi_\epsilon^{L,*} d\rho\,. \tag{2.13}$$

Let $F'_+(0)$ be the right derivative at $\epsilon=0$, i.e. $F'_+(0) = \lim_{\epsilon\to 0^+} \frac{1}{\epsilon}(F(\epsilon)-F(0))$. By convexity, for any sequence ϵ_n such that F is differentiable at ϵ_n and $\epsilon_n \searrow 0$, we have

$$F'_+(0) = \lim_{n\to\infty} F'(\epsilon_n) = \lim_{n\to\infty} \int \phi_{\epsilon_n}^{L,*} d\rho\,.$$

We write $\mathbb{R}^d = \cup_{m\in\mathbb{N}} K_m$ with $K_m \subset \mathbb{R}^d$ being compact set and $K_m \subset K_{m+1}$. The optimizer $\phi_{\epsilon_n}^{L,*}$ is unique up to constant which we choose now such that $\phi_{\epsilon_n}^{L,*}(0) = 0$. The Lipschitz condition implies that the sequence $\phi_{\epsilon_n}^{L,*}$ is equibounded and equicontinuous on K_m. By the Arzelà–Ascoli theorem, there exists a subsequence of $\phi_{\epsilon_n}^{L,*}$ that converges uniformly in K_m. Using diagonal argument, by taking subsequences sequentially along $\{K_m\}_{m\in\mathbb{N}}$ we conclude there exists a subsequence such that $\phi_{\epsilon_{n_k}}^{L,*}$ converges uniformly in any K_m and thus $\phi_{\epsilon_{n_k}}^{L,*}$ converges pointwise in $\mathbb{R}^d$. Let $\phi_0^{L,*} \in \mathrm{Lip}^L(\mathbb{R}^d)$ be the limit and for simplicity we also denote by $\phi_{\epsilon_n}^{L,*}$ the convergent subsequence. The choice $\phi_{\epsilon_n}^{L,*}(0) = 0$ and the Lipschitz condition implies that $|\phi_{\epsilon_n}^{L,*}(x)| \leq L|x|$ which is integrable with respect to ρ since $\rho_\pm \in \mathcal{P}_1(X)$. Thus by dominated convergence

$$F'_+(0) = \lim_{n\to\infty} \int \phi_{\epsilon_n}^{L,*} d\rho = \int \phi_0^* d\rho\,.$$

By the lower semicontinuity of $D_f^{\Gamma_L}(\cdot\|Q)$ (see Theorem SM1.1 in the supplementary material), we have

$$\begin{aligned} D_f^{\Gamma_L}(P\|Q) &\leq \liminf_{n\to\infty} D_f^{\Gamma_L}(P+\epsilon_n\rho\|Q)\\ &= \liminf_{n\to\infty} \left\{ E_{P+\epsilon_n\rho}[\phi_{\epsilon_n}^{L,*}] - \inf_{\nu\in\mathbb{R}} \left\{\nu + E_Q[f^*(\phi_{\epsilon_n}^{L,*} - \nu)]\right\}\right\}\\ &= \liminf_{n\to\infty} E_{P+\epsilon_n\rho}[\phi_{\epsilon_n}^{L,*}] - \limsup_{n\to\infty} \inf_{\nu\in\mathbb{R}} \left\{\nu + E_Q[f^*(\phi_{\epsilon_n}^{L,*} - \nu)]\right\}\\ &\leq E_P[\phi_0^{L,*}] - \inf_{\nu\in\mathbb{R}} \left\{\nu + E_Q[f^*(\phi_0^{L,*} - \nu)]\right\} \leq D_f^{\Gamma_L}(P\|Q), \end{aligned}$$

where for the second inequality we use the dominated convergence theorem, (2.13) and that by Fatou's lemma (using that $f^*(x) \geq x$ and that $|\phi_{\epsilon_n}^{L,*}(x)| \leq L|x|$),

$$\limsup_{n\to\infty} \int f^*(\phi_{\epsilon_n}^{L,*})dQ \geq \liminf_{n\to\infty} \int f^*(\phi_{\epsilon_n}^{L,*})dQ \geq \int f^*(\phi_0^{L,*})dQ\,.$$

From (2.14) we conclude that $\phi_0^{L,*}$ must be an optimizer, and thus $\phi_0^{L,*}(x) = \phi^{L,*}(x)$, P a.s., and $\phi_0^{L,*}(x) \leq \phi^{L,*}(x)$ for all x (see Lemma 2.3). Using that ρ_- is absolutely continuous with respect to P we have then

$$F'_+(0) = \int \phi_0^{L,*} d\rho = \int \phi_0^{L,*} d\rho_+ - \int \phi_0^{L,*} d\rho_- = \int \phi_0^{L,*} d\rho_+ - \int \phi^{L,*} d\rho_- \leq \int \phi^{L,*} d\rho. \tag{2.14}$$

Combining with (2.9) implies that $F'_+(0) = \int \phi^{L,*} d\rho$. ■

Remark 2.4 (algorithmic perspectives and related results). The statement and the proof of Theorem 2.1 contain certain key algorithmic elements that will become relevant in later sections: (a) A version of Theorem 2.1 was proved in [16] for the special case of KL divergence. In Theorem 2.1 our results are proved for general f-divergences. This generality is necessary in generative modeling based on both past experience in GANs [45, 40, 5, 7], as well as the demonstration examples with heavy tails considered here. (b) In Theorem 2.1, the maximizer $\phi^{L,*} \in \Gamma_L$ defined on $\mathrm{supp}(P) \cup \mathrm{supp}(Q)$, is maximally extended as an L-Lipschitz function to all of $\mathbb{R}^d$; see Lemma 2.3. Notice that in our algorithm in section 3, we also allow for L-Lipschitz extensions which are constructed algorithmically simply by optimization in the space of L-Lipschitz NNs; see Algorithm 3.1. (c) The derived (not assumed!) absolute continuity of the perturbation ρ in (2.7), captures some important intuition about the nature of $P + \epsilon\rho$ when P is an empirical measure, e.g., when it is built from particles as in Algorithm 3.1: in this perturbation, existing particles can be removed from P according to ρ_-, corresponding to the absolute continuity (2.7), while new particles can be created anywhere according to ρ_+, the latter not requiring absolute continuity. These perturbations/variations of empirical measures are precisely the ones arising in the particle algorithm (3.2).

Using Theorem 2.1 we can now rewrite (2.1) as a *transport/variational* PDE,

$$\begin{aligned} &\partial_t P_t + \mathrm{div}(P_t v_t^L) = 0\,, \quad P_0 = P \in \mathcal{P}_1(\mathbb{R}^d)\,, \\ &v_t^L = -\nabla\phi_t^{L,*}\,, \quad \phi_t^{L,*} = \operatorname*{argmax}_{\phi\in\Gamma_L} \left\{ E_{P_t}[\phi] - \inf_{\nu\in\mathbb{R}} (\nu + E_Q[f^*(\phi - \nu)]) \right\}. \end{aligned} \tag{2.15}$$

The transport/variational reformulation (2.15) is the starting point for developing our generative particle algorithms in section 3 based on data, when P and Q are replaced by their empirical measures $\hat{P}^M$, $\hat{Q}^N$ based on M and N i.i.d. samples, respectively. Furthermore, (2.15) provides a numerical stability perspective on the Lipschitz regularization (2.1) In particular, the Lipschitz condition on $\phi \in \Gamma_L$ enforces a finite speed of propagation of at most L in the transport equation in (2.15). This is in sharp contrast with the FP equation (1.5), which is a diffusion equation and has infinite speed of propagation. We refer to section 6 for connections to the CFL stability condition.

The gradient flow structure of (2.1) is reflected in dissipation estimates, namely an equation for the rate of change (dissipation) of the divergence along smooth solutions P_t of (2.1).

Theorem 2.5 (Lipschitz-regularized dissipation). *Along a trajectory of a smooth solution* $\{P_t\}_{t\geq 0}$ *of* (2.15) *with source probability* $P_0 = P$ *we have the rate of decay identity*

$$\frac{d}{dt} D_f^{\Gamma_L}(P_t \| Q) = -I_f^{\Gamma_L}(P_t \| Q) \leq 0, \tag{2.16}$$

where we define the Lipschitz-regularized Fisher Information as

$$I_f^{\Gamma_L}(P_t \| Q) = E_{P_t}\left[|\nabla \phi^{L,*}|^2 \right] . \tag{2.17}$$

Consequently, for any $T \geq 0$, *we have* $D_f^{\Gamma_L}(P_T \| Q) = D_f^{\Gamma_L}(P \| Q) - \int_0^T I_f^{\Gamma_L}(P_s \| Q) ds$.

The proof can be found in supplementary material section SM2.2. For the generative particle algorithms of section 3 the Lipschitz-regularized Fisher Information will be interpreted as the total kinetic energy of the particles (3.3).

Remark 2.6 (formal asymptotics of Lipschitz-regularized gradient flows). The rigorous $(L \to \infty)$-asymptotic results of the limit of the Lipschitz-regularized f-divergences to (unregularized) f-divergences presented in [16, 5] (see also Theorem SM1.1 in the supplementary material) motivates a discussion on the formal asymptotics of the corresponding gradient flows. In particular, the Lipschitz-regularization $L \to \infty$ asymptotics towards the (unregularized) gradient flows can be formally obtained as the limit of the transport/variational PDEs (2.15), i.e.,

$$\underbrace{\partial_t P_t = \operatorname{div}\left(P_t \nabla \phi_t^{L,*} \right)}_{\text{Lip. regularized} f\text{-divergence flow}} \quad \xrightarrow[L\to\infty]{} \quad \underbrace{\partial_t P_t = \operatorname{div}\left(P_t \nabla \phi_t^{*} \right)}_{f\text{-divergence flow}}, \text{ where } \phi_t^* = f'\left(\frac{dP_t}{dQ} \right). \tag{2.18}$$

When p_t, q are the probability densities of P_t and Q, respectively, and $f(x) = f_{\mathrm{KL}}(x) = x\log(x)$ and $f_\alpha(x) = \frac{x^\alpha - 1}{\alpha(\alpha-1)}$, the Lipschitz regularized f-divergence flow in (2.18) converges to the classical FP equation given by $\partial_t p_t = \operatorname{div}(p_t \nabla \log(\frac{p_t}{q}))$ and Weighted Porous Medium equation given by $\partial_t p_t = \frac{1}{\alpha-1}\operatorname{div}(p_t \nabla(\frac{p_t}{q})^{\alpha-1})$, respectively. Similarly, when $f = f_{\mathrm{KL}}$, as $L \to \infty$, we formally recover from (2.17) the usual Fisher information $I_f^{\Gamma}(P\|Q) = E_P[|\nabla \log(\frac{p}{q})|^2]$.

Some PDE questions for Lipschitz-regularization. A rigorous analysis encompassing aspects such as well-posedness, stability, regularity, and convergence to equilibrium Q, remains to be explored. For example, the DiPerna–Lions theory [2, 14] for transport equations with rough velocity fields and its more recent variants could be useful for proving well-posedness. Additionally, functional inequalities tailored for porous medium and FP equations contribute to proving convergence of a PDE to its equilibrium such as exponential or polynomial convergence. Classical examples of such inequalities are Poincaré and Logarithmic Sobolev-type inequalities, and generalizations thereof for FP and porous medium equations [1, 48, 15]. However, convergence of the new class of PDE gradient flows (2.1) to their equilibrium states, will require new functional inequalities entailing the Lipschitz-regularized Fisher Information and probability measures Q which may not have densities.

3. Generative particle algorithms.

In this section we build a numerical algorithm to solve the transport/discriminator gradient flow (2.15) when N i.i.d. samples from the target distribution Q are given. We first discretize the system in time using a forward-Euler scheme,

$$\begin{aligned} &P_{n+1} = \left(I - \Delta t \nabla \phi_n^{L,*}\right)_{\#} P_n, \quad \text{where } P_0 = P, \\ &\phi_n^{L,*} = \underset{\phi \in \Gamma_L}{\arg\max} \left\{ E_{P_n}[\phi] - \inf_{\nu \in \mathbb{R}} \{\nu + E_Q[f^*(\phi - \nu)]\} \right\}. \end{aligned} \tag{3.1}$$

Here, the pushforward measure for a map $T : \mathbb{R}^d \to \mathbb{R}^d$ and $P \in \mathcal{P}(\mathbb{R}^d)$ is denoted by $T_{\#}P$ (i.e., $T_{\#}P(A) = P(T^{-1}(A))$). Next, given N i.i.d. samples $\{X^{(i)}\}_{i=1}^N$ from the target distribution Q, we consider the empirical measure $\hat{Q}^N = N^{-1} \sum_{i=1}^N \delta_{X^{(i)}}$. Likewise, given M i.i.d. samples $\{Y_0^{(i)}\}_{i=1}^M$ from a known initial (source) probability measure P and consider the empirical measure $\hat{P}^M = M^{-1} \sum_{i=1}^M \delta_{Y_0^{(i)}}$. By replacing the measures P and Q in (3.1) by their empirical measures $\hat{P}^M$ and $\hat{Q}^N$ we obtain the following particle system:

$$\begin{aligned} &Y_{n+1}^{(i)} = Y_n^{(i)} - \Delta t \nabla \phi_n^{L,*}(Y_n^{(i)}), \quad Y_0^{(i)} = Y^{(i)}, Y^{(i)} \sim P, \quad i = 1, \dots, M \\ &\phi_n^{L,*} = \underset{\phi \in \Gamma_L^{NN}}{\operatorname{argmax}} \left\{ \frac{\sum_{i=1}^M \phi(Y_n^{(i)})}{M} - \inf_{\nu \in \mathbb{R}} \left\{ \nu + \frac{\sum_{i=1}^N f^*(\phi(X^{(i)}) - \nu)}{N} \right\} \right\}, \end{aligned} \tag{3.2}$$

where the function space Γ_L in (3.1) is approximated by a space of NN approximations Γ_L^{NN}. We will refer to this particle algorithm as (f, Γ_L)-GPA or simply GPA. The transport mechanism given by (3.2) corresponds to a linear transport PDE in (2.15). However, between particles nonlinear interactions are introduced via the discriminator $\phi_n^{L,*}$ which in turn depends on all particles in (3.2) at step n of the algorithm, namely the generated particles $(Y_n^{(i)})_{i=1}^M$, as well as the "target" particles $(X^{(i)})_{i=1}^N$. Notice that $\phi_n^{L,*}$ discriminates the generated samples at time n from the target data using the second equation of (3.2), and is not directly using the generated data of the previous steps up to step $n-1$. Moreover the gradient of the discriminator is computed only at the positions of the particles.

Overall, (3.2) is an approximation scheme of the Lagrangian formulation (1.10) of the Lipschitz-regularized gradient flow (1.6), where we have (a) discretized time, (b) approximated the function space Γ_L in terms of neural networks, and (c) used empirical distributions/particles to build approximations of the target Q, (d) used gradient-based optimization methods to approximate the discriminator $\phi_n^{L,*}$ such as stochastic gradient descent or the Adam optimizer. All these elements are combined in Algorithm 3.1.

Remark 3.1 (Lipschitz regularization for GPA). Lipschitz regularized f-divergences are practically advantageous since they allow one to calculate divergences between arbitrary empirical measures with nonoverlapping supports. Indeed, given a Lipschitz constant L, the L-Lipschitz regularized f-divergence is bounded by L times the 1-Wasserstein metric as stated in (1.3) and discussed in more detail in [5]. Therefore, a suitable choice of L depending on data offers numerical tractability for the particle system in (3.2) and Algorithm 3.1. Without proper Lipschitz regularization, GPA diverges or produces inaccurate solutions as illustrated in Figure 3. In our implementation, the Lipschitz regularization is enforced via Spectral Normalization (SN) for NNs, [43]. Despite its clear numerical benefits, SN incurs a relatively modest computational cost. Applying SN in an experiment leads to a 10% increase in computational time compared to a nonregularized counterpart. Another way to impose Lipschitz regularization for NNs is to add a gradient penalty to the loss [24, 5].

Remark 3.2 (improved accuracy and higher-order schemes). Replacing the forward Euler in (3.2) or line 10 in Algorithm 3.1 with Heun's predictor/corrector method is observed to lead to a significant improvement in the accuracy of the GPA for several examples; see, for instance, Figure SM1. In addition, adopting a smaller Δt in (3.2) and Algorithm 3.1 may contribute to enhanced accuracy in GPA outcomes. Employing a smaller Δt often requires a smoother discriminator, achieved by substituting the ReLU activation function with a smoothed ReLU. We refer to supplementary material section SM3.2 for details.

GPA kinetic energy and Lipschitz-regularized Fisher information. Theorem 2.5 suggests the empirical Lipschitz-regularized Fisher Information,

$$I_f^{\Gamma_L}(\hat{P}_n^M \| \hat{Q}^N) = \int |\nabla \phi_n^{L,*}|^2 \hat{P}_n^M(dx) = \frac{1}{M} \sum_{i=1}^{M} |\nabla \phi_n^{L,*}(Y_n^{(i)})|^2 , \tag{3.3}$$

as a quantity of interest to monitor the convergence of GPA (3.2). Here $\hat{P}_n^M$ denotes the empirical distribution of the generative particles $(Y_n^{(i)})_{i=1}^M$. Indeed, $I_f^{\Gamma_L}(\hat{P}_n^M \| \hat{Q}^N)$ is the total kinetic energy of the generative particles since $\nabla \phi_n^{L,*}(Y_n^{(i)})$ is the velocity of the ith particle at time step n. The algorithm will stop when the total kinetic energy $I_f^{\Gamma_L}(\hat{P}_n^M \| \hat{Q}^N) \approx 0$.

Algorithm 3.1. $[(f,\Gamma_L)$-GPA] Lipschitz regularized generative particles algorithm.

```
Require: $f$ for the choice of $f$-divergence and its Legendre conjugate $f^*$, $L$: Lipschitz constant,
    $n_{\max}$: number of updates for the particles, $\Delta t$: time step size, $M$: number of initial
    particles, $N$: number of target particles
Require: $W = \{W^l\}_{l=1}^D$: parameters for the neural networks (NN) $\phi : \mathbb{R}^d \to \mathbb{R}$, $D$: depth of
    the NN, $\delta$: learning rate of the NN, $m_{\max}$: number of updates for the NN.
Result: $\{Y_{n_{\max}}^{(i)}\}_{i=1}^M$
 1: Sample $\{Y_0^{(i)}\}_{i=1}^M \sim P_0 = P$, a batch of prior samples
 2: Sample $\{X^{(j)}\}_{j=1}^N \sim Q$, a batch from the real data
 3: Initialize $\nu \leftarrow 0$
 4: Initialize $W$ randomly and $W^l \leftarrow L^{1/D} * W^l / \|W^l\|_2$, $l = 1, \dots, D$        ▷ $\phi_0^L(\cdot; W) \in \Gamma_L$
 5: for $n = 0$ to $(n_{\max} - 1)$ do
 6:     for $m = 0$ to $m_{\max} - 1$ do
 7:         $grad_{W,\nu} \leftarrow \nabla_{W,\nu} \left[ M^{-1} \sum_{i=1}^M \phi_n^L(Y_n^{(i)}; W) - N^{-1} \sum_{j=1}^N f^*(\phi_n^L(X^{(j)}; W) - \nu) + \nu \right]$
 8:         $(\nu, W) \leftarrow (\nu, W) + \delta \cdot optimizer(grad_\nu, grad_W)$
 9:         $W^l \leftarrow L^{1/D} * W^l / \|W^l\|_2$, $l = 1, \dots, D$
10:     end for                                                      ▷ $\phi_n^{L,*}(\cdot; W) \in \Gamma_L$
11:     $Y_{n+1}^{(i)} \leftarrow Y_n^{(i)} - \Delta t \nabla \phi_n^{L,*}(Y_n^{(i)}; W)$, $i = 1, \dots, M$     ▷ forward Euler
12: end for
    $L$-Lipschitz continuity is imposed by $W^l \leftarrow L^{1/D} * W^l / \|W^l\|_2$, $l = 1, \dots, D$.
```

Overall, Algorithm 3.1 estimates two natural quantities of interest: the Lipschitz regularized f-divergence $M^{-1}\sum_{i=1}^{M}\phi_n^{L,*}(Y_n^{(i)};W) - N^{-1}\sum_{j=1}^{N} f^*(\phi_n^{L,*}(X^{(j)};W) - \nu^*) + \nu^*$ and the Lipschitz regularized Fisher information (3.3). These quantities are used to track the progress and terminate the simulations.

4. Generalization properties of GPA.

The transport/discriminator formulation in (3.1) is the core mechanism in GPA, facilitating sample generation by transporting particles through time-dependent vector fields obtained by iteratively solving (3.2) over time. Ensuring the diversity of generated samples and avoiding "memorization" of the target data, is a critical challenge in generative modeling, as discussed extensively in recent publications, for instance in the context of diffusion models, [49, 55, 56, 23, 36, 10], including empirical [56] and theory-based mitigation strategies [63]. In GPA as well, there is the theoretical possibility, based on the gradient flow dynamics and the dissipation estimate in Theorem 2.5, that with a rich enough NN to learn the discriminator, suitable learning rates, and long enough runs, Algorithm 3.1 may reproduce the empirical distribution of the target data, especially when $M = N$. This phenomenon can be observed for the MNIST data set in Figure SM4. To mitigate these challenges and ensure better generalization for the proposed GPA algorithms, we explore three distinct strategies:

1. *From training particles to generated particles.* In this approach we use M training particles from an initial distribution P_0 and N target particles to learn the time-dependent vector fields given by Algorithm 3.1. This vector field is constructed as an NN on the *entire* space. Therefore, we can transport (e.g., simultaneously) any additional number of particles sampled from P_0 using this, already learned, vector field. We refer to the latter type of particles as "generated particles". See Figure 5 and also Figure SM6 in the supplementary material for practical demonstrations of such generated particles.

   This approach, which is based on learning a time-dependent vector field, aligns with other flow-based generative models such as SGM [58], and normalizing flows [13]. However, the latter methods are more efficient in learning their time-dependent vector field by employing a corresponding space/time objective functional. We believe that a similar formulation can be built for GPA, by using the mean-field game functionals for Wasserstein gradient flows in [62]. We plan to explore this space/time approach in a follow-up work.
2. *Imbalanced sample sizes.* In this strategy we choose $M \gg N$ in Algorithm 3.1. First, we empirically found strong evidence of overfitting and memorization in the $M = N$ case, i.e., training particles eventually match the target particles. However, in the setting of the imbalanced sample sizes $M \gg N$ particles maintain their sample diversity. See Figure SM3 in the supplementary material. These different behaviors are captured and quantified by the two estimators (divergence and kinetic energy) in Algorithm 3.1; compare the findings in parts (c, e) of Figure SM3 in the supplementary material.
3. *GPA for data augmentation.* Lastly, we demonstrate that GPA can serve as a data augmentation tool to train other generative models particularly those requiring large sample sizes. For instance, the examples in Figures 6 and 9 showcase the effectiveness of GPA-based data augmentation for GANs.

Overall, GPA learns from target data and training particles, a time-dependent vector field represented by Lipschitz NNs defined on the entire space. In this sense, GPA is expected to gain in *extrapolation* properties since the learned vector field can be used to move arbitrary new particles towards the target data.

5. Data processing inequality and latent space GPA.

Performance degradation is a common challenge for all generative models in high-dimensional settings, a problem that becomes more pronounced in regimes with low sample sizes. For GPA, the optimization of the discriminator within the NN space exhibits superior scalability, particularly in regimes of hundreds of dimensions, compared to optimization in RKHS which typically performs well in lower dimensions. However, similarly to other neural-based generative models, GPA faces challenges in really high-dimensional problems. To overcome this type of scalability constraints, we can take advantage of latent space formulations used in recent papers in generative flows, e.g., [60, 52, 46], to complement and scale-up score-based models, diffusion models, and normalizing flows. The key idea is simple and powerful as demonstrated in these earlier works: a pretrained auto-encoder first projects the high-dimensional real space to a lower-dimensional latent space and then a generative model is trained in the compressed latent space. Subsequently, the decoder of the auto-encoder allows one to map the data generated in the latent space back to the original high-dimensional space.

In Theorem 5.1, we demonstrate that operating in the latent space can be understood in light of a suitable Data Processing Inequality (DPI) and we provide conditions which guarantee that the error induced by the transportation of a high-dimensional data distribution via combined encoding/decoding and particle transportation in a lower-dimensional latent space is controlled by the error only in the (much more tractable) latent space. More specifically, we consider the following mathematical setting: (i) a probability $Q = Q^{\mathcal{Y}}$, defined on the original, high-dimensional space $\mathcal{Y}$, typically supported on some low-dimensional set $S \subset \mathcal{Y} = \mathbb{R}^d$; (ii) an encoder map $\mathcal{E} : \mathcal{Y} \to \mathcal{Z}$ where $\mathcal{Z} \subset \mathbb{R}^{d'}$, $d' < d$, and a decoder map $\mathcal{D} : \mathcal{Z} \to \mathcal{Y}$ which are invertible in S, i.e., $\mathcal{D} \circ \mathcal{E}(S) = S$. Let $\mathcal{E}_\# Q^{\mathcal{Y}}$ denote the image of the measure $Q^{\mathcal{Y}}$ by the map $\mathcal{E}$, i.e., for $A \subset \mathcal{Z}$, $\mathcal{E}_\# Q^{\mathcal{Y}}(A) := Q^{\mathcal{Y}}(\mathcal{E}^{-1}(A))$. Similarly, we define $\mathcal{D}_\# P^{\mathcal{Z}}$ as the combination of the encoding/decoding and particle transportation $\mathcal{T}^n$ in a lower-dimensional latent space where $P^{\mathcal{Z}} := \mathcal{T}^n_\# \mathcal{E}_\# P_0$. The fidelity of the approximation $Q^{\mathcal{Y}} \approx \mathcal{D}_\# P^{\mathcal{Z}}$ of the target measure $Q^{\mathcal{Y}}$ in the original space $\mathcal{Y}$ will be then guaranteed by the *a posteriori* estimate in Theorem 5.1, interpreted in the sense of numerical analysis, where the approximation in the compressed latent space $\mathcal{Z}$ bounds the error in the original space $\mathcal{Y}$. Its proof is a consequence of a new, tighter data processing inequality derived in [5]; see also Theorem SM1.2 in the supplementary material that involves both transformation of probabilities and discriminator space Γ.

Theorem 5.1 (autoencoder performance guarantees). *For $Q^{\mathcal{Y}} \in \mathcal{P}(\mathcal{Y})$, suppose that there is a exact encoder/decoder with encoder $\mathcal{E} : \mathbb{R}^d \to \mathbb{R}^{d'}$ and decoder $\mathcal{D} : \mathbb{R}^{d'} \to \mathbb{R}^d$, where exact means perfect reconstruction $\mathcal{D}_\# \mathcal{E}_\# Q^{\mathcal{Y}} = Q^{\mathcal{Y}}$. Furthermore, assume the decoder is Lipschitz continuous with Lipschitz constant $a_{\mathcal{D}}$. Then, for any $P^{\mathcal{Z}} \in \mathcal{P}_1(\mathcal{Z})$ we have*

$$D_f^{\Gamma_L}(\mathcal{D}_\# P^{\mathcal{Z}} \| Q^{\mathcal{Y}}) \le D_f^{a_{\mathcal{D}}\Gamma_L}(P^{\mathcal{Z}} \| \mathcal{E}_\# Q^{\mathcal{Y}}). \tag{5.1}$$

Proof. From the data processing inequality Theorem SM1.2 in the supplementary material and using that the composition of Lipschitz functions with Lipschitz constants L_1, L_2 is $L_1 L_2$-Lipschitz, we have

$$D_f^{\Gamma_L}(\mathcal{D}_\# P^{\mathcal{Z}} \| \mathcal{D}_\# \mathcal{E}_\# Q^{\mathcal{Y}}) \le D_f^{a_{\mathcal{D}}\Gamma_L}(P^{\mathcal{Z}} \| \mathcal{E}_\# Q^{\mathcal{Y}}). \tag{5.2}$$

Since the encoder $\mathcal{E}$ and the decoder $\mathcal{D}$ perfectly reconstruct $Q^{\mathcal{Y}}$, namely $\mathcal{D}_\# \mathcal{E}_\# Q^{\mathcal{Y}} = Q^{\mathcal{Y}}$, we obtain that

$$D_f^{\Gamma_L}(\mathcal{D}_\# P^{\mathcal{Z}} \| Q^{\mathcal{Y}}) \le D_f^{a_{\mathcal{D}}\Gamma_L}(P^{\mathcal{Z}} \| \mathcal{E}_\# Q^{\mathcal{Y}}). \tag{5.3}$$

Note also that if $a_{\mathcal{D}} \le 1$, $D_f^{\Gamma_L}(\mathcal{D}_\# P^{\mathcal{Z}} \| \mathcal{D}_\# \mathcal{E}_\# Q^{\mathcal{Y}}) \le D_f^{\Gamma_L}(P^{\mathcal{Z}} \| \mathcal{E}_\# Q^{\mathcal{Y}})$. ■

We apply this result in section 9 where the merging (transporting) of high-dimensional gene expression data sets with dimension exceeding 54K in performed in a latent space which is constructed via Principal Component Analysis (PCA), i.e., a linear auto-encoder.

Remark 5.2 (autoencoder guarantees in generative modeling). It is clear that Theorem 5.1 is a result about autoencoders and it is independent of the choice of any specific transport/generation algorithm in the latent space. In this sense our conclusions from Theorem 5.1 are generally applicable to other latent space methods for generative modeling, such as GANs.

6. Lipschitz regularization and numerical stability.

In this section, we discuss the numerical stability of GPA induced by Lipschitz regularization. The Lipschitz bound L on the discriminator space implies a pointwise bound $|\nabla \phi_n^{L,*}(Y_n^{(i)})| \le L$. Hence the Lipschitz regularization imposes a speed limit L on the particles, ensuring the stability of the algorithm for suitable choices of L, as we will discuss next.

We first illustrate how Lipschitz regularization works in GPA Algorithm 3.1 in a mixture of 2D Gaussians. We explore the influence of the Lipschitz regularization constant L by monitoring the Lipschitz regularized Fisher information (3.3) (i.e., kinetic energy of particles). In Figure 2 we track this quantity in time. We empirically observe that a proper choice of L enables the particles slow down and eventually stop near the target particles, using (3.3) as a convergence indicator. Time trajectories of particles are displayed in Figure 3. Individual curves in Figure 2 result from the Lipschitz regularized $(f_{\mathrm{KL}}, \Gamma_L)$-GPA with $L = 1$, 10, 100, ∞. We fix all other parameters including time step Δt, focusing on the influence of the Lipschitz constant L. For $L = 1, 10$, the kinetic energy decreases and particles eventually stop. However, without Lipschitz regularization, the particles keep (relatively) high speeds of propagation. Figure 3 verifies that in this case ($L = \infty$) the algorithm fails to converge.

Numerical stability of GPA. Based on these empirical findings, we observe a close relationship between a finite propagation speed L and numerical stability of the algorithm. Indeed, from a numerical analysis point of view, (3.1) is a particle-based explicit scheme for the PDE (2.15). In this context, the CFL condition for stability of discrete schemes for transport PDEs such as the first equation in (2.15) becomes $\sup_x |\nabla \phi_t^{L,*}(x)| \frac{\Delta t}{\Delta x} \le 1$, [35]. Clearly, the Lipschitz regularization $|\nabla \phi_t^{L,*}(x)| \le L$ enforces a CFL type condition with a learning rate Δt proportional to the inverse of L. It remains an open question how to rigorously extend these CFL-based heuristics to particle-based algorithms, we also refer to some related

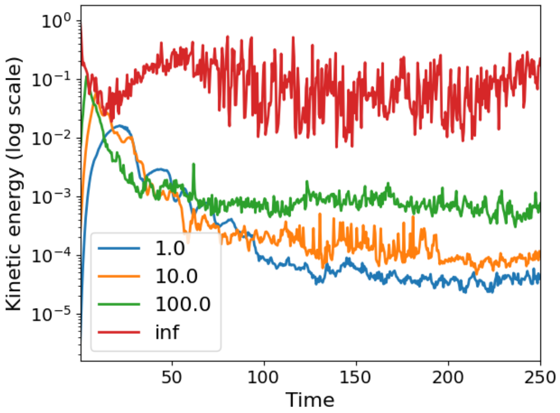


Figure 2. *(2D mixture of Gaussians) Kinetic energy of particles* (3.3) *for* $(f_{\mathrm{KL}}, \Gamma_L)$*-GPA with different* L*'s. Theorem* 2.5 *suggests that particles need to slow down and practically stop when they reach the "vicinity" of the target particles.*

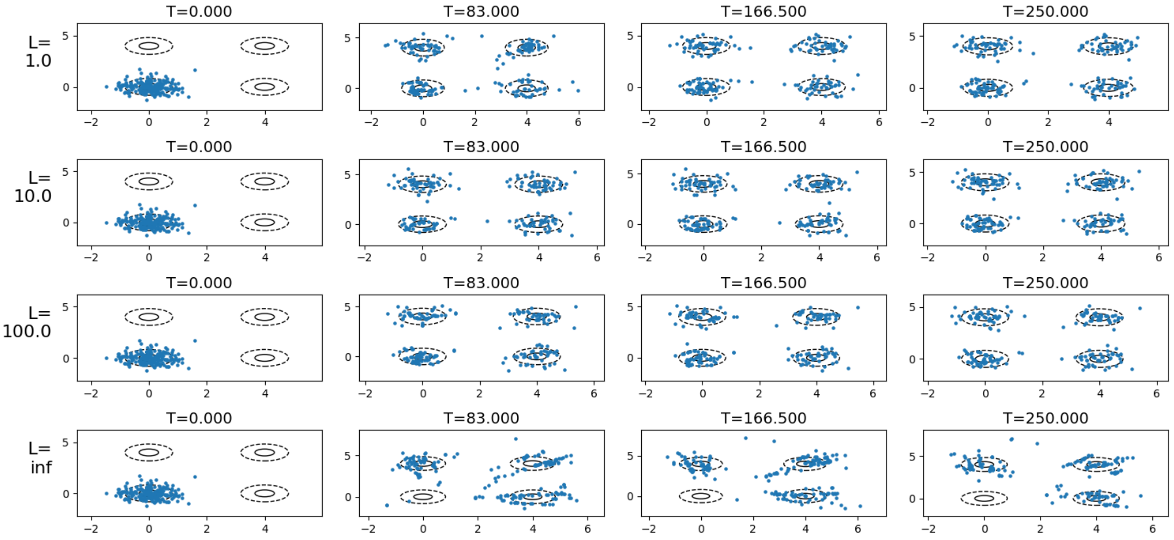


Figure 3. *(2D mixture of Gaussians) We empirically observe that Lipschitz constant* L *controls the propagation speed of* $(f_{\mathrm{KL}}, \Gamma_L)$*-GPA with different* L*'s. For* $L < \infty$*, the particles are propagated to the four wells. As* L *gets larger, the algorithm becomes more unstable. For* $L = \infty$ *(unregularized KL), GPA fails to capture the target.*

questions in [11]. However, in the context of (3.2), the speed constraint L on the particles induces an implicit spatial discretization grid Δx where particles are transported for each Δt by at most $\Delta x = L\Delta t$. Intuitively, this implicit spatio-temporal discretization suggests that $\sup_x |\nabla\phi_t^{L,*}(x)| \frac{\Delta t}{\Delta x} = \frac{\sup_x |\nabla\phi_t^{L,*}(x)|}{L} \leq 1$. Hence (3.2) or Algorithm 3.1 are expected to satisfy the same CFL condition for the transport PDE in (2.15). Based on these CFL heuristics for particles, here we keep the inversely proportional relation between L and Δt as a criterion for tuning the learning rate Δt. Finally, these CFL-based bounds and the empirical findings in Figure 3 suggest that a time-dependent "schedule" for L could enhance the stability and convergence properties of GPAs, as the quantity $\sup_x |\nabla \phi_t^{L,*}(x)|$ could serve as (or inspire) an indicator of proximity to the target distribution. However, in this paper we do not explore further such time-adaptive strategies for L.

7. Generative particle algorithms for heavy-tailed data.

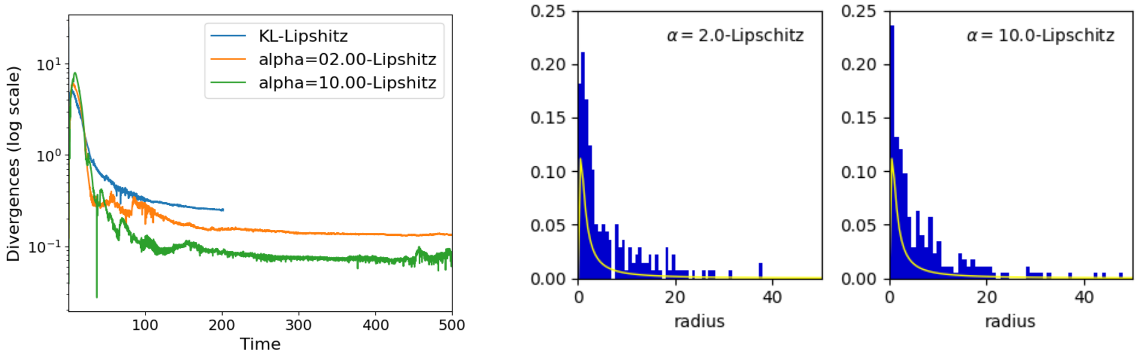
Lipschitz regularized gradient flows in section 2 and GPA in section 3 are built on a family of f-divergences as discussed in section 1. Here we study the choice of f_{KL} versus f_α on GPA for samples from distributions with various tails, e.g., Gaussian, stretched exponential, or polynomial. This exploration rests on the intuition that transporting a Gaussian to a heavy-tailed distribution and vice-versa is a nontrivial task. This is due to the fact that a significant amount of mass deep in the tail needs to be transported to and from a (light-tailed) Gaussian. Furthermore, for heavy tailed distributions, KL divergence may become infinity, and thus cannot be trained, while in the f_α divergence we have flexibility to accommodate heavy tails using the parameter α. However, even with the use of an f_α divergence, transporting particles deep into the heavy tails takes a considerable amount of time due to the speed restriction L of Lipschitz regularization; see section 6. Therefore, in our experiments, we are less focused on "perfect" transportation and more on "numerically stable" transportation of moderately heavy-tailed distributions.

Indeed, in our first experiment we observe the following. The choice of f_{KL} for heavy-tailed data renders the function optimization step in (3.2) numerically unstable and eventually leads to the collapse of the algorithm. On the other hand, the choice of f_α with $\alpha > 1$ makes the algorithm stable. The different behaviors of f_{KL} and f_α on heavy-tailed data is illustrated in Figure 4 and also Figure SM2 in the supplementary material.

Next, we explore the performance of GPA for several distributions with varying degrees of heavy-tailed structure. Initial distributions P_0 are chosen as heavy-tailed distributions in cases 1–4 in Table 1, whereas target distribution Q are chosen as heavy-tailed distributions in cases 5–8. We chose Generalized Gaussian distribution (Stretched exponential distribution, $GMM(\beta) \propto \exp(-|x|^\beta)$) with $\beta = 0.5$ as a heavy-tailed distribution because it fails to be subexponential. But it has finite moments of all orders. On the other hand, Student-t distributions with degree of freedom ν ($Student-t(\nu)$) have polynomial tails. Among them, $Student-t(3.0)$ has a finite second moment, $Student-t(1.5)$ has an infinite second moment but has a finite first moment, and $Student-t(0.5)$ has an infinite second moment but its first moment is undefined. In all cases in Table 1 we use the Gaussian distribution $N((10,10), I)$ as either source or target. Table 1 displays the summary of the transportation of particles for different cases. Overall, with the exception of especially heavy-tailed distributions in cases 3 and 4 (both with infinite second moments and thus very heavy tails), KL and/or α-divergences work reasonably well. We also note that α-divergences in GANs for images can provide superior performance to KL and related divergences, even in the abscence of heavy tails [45, 40, 5, 7].

8. Learning from scarce data.

In this section, we empirically demonstrate that GPA can be an effective generative model when only scarce target data is available. We analyze three types of problems: GPA for generating images in a high-dimensional space given scarce



(a) (f, Γ_1)-divergences

(b) The radii of transported samples (blue), and the corresponding radial distribution function (yellow).

Figure 4. *(Gaussian to student-t with $\nu = 0.5$ in two dimensions) We consider 200 initial samples from $N((10,10), 0.5^2 I)$, transported towards 200 target samples from Student $- t(\nu)$ with $\nu = 0.5$ using (f,Γ_1)-GPA's for $f = f_{\mathrm{KL}}$ and $f = f_\alpha$ with $\alpha = 2, 10$.* (a) *(f,Γ_1)-divergences are computed by the corresponding estimator in* (3.2). *$(f_{\mathrm{KL}},\Gamma_1)$-GPA collapses at around $t = 202$ as the function optimization step with f_{KL} is numerically unstable on heavy-tailed data while (f_α,Γ_1)-GPA with $\alpha = 2, 10$ propagate particles stably during the entire simulation window. See Figure* SM2 *for details. However, GPA still appears to take a long time to transport particles deep into the heavy tails due to the speed restriction of the Lipschitz regularization. Stability in performance that lacks in accuracy is manifested in the relatively large size of the α-divergences.* (b) *We observed that (f_α,Γ_1)-GPA with $\alpha = 10$ transports particles further and deeper into the tails than (f_α,Γ_1)-GPA with $\alpha = 2$.*

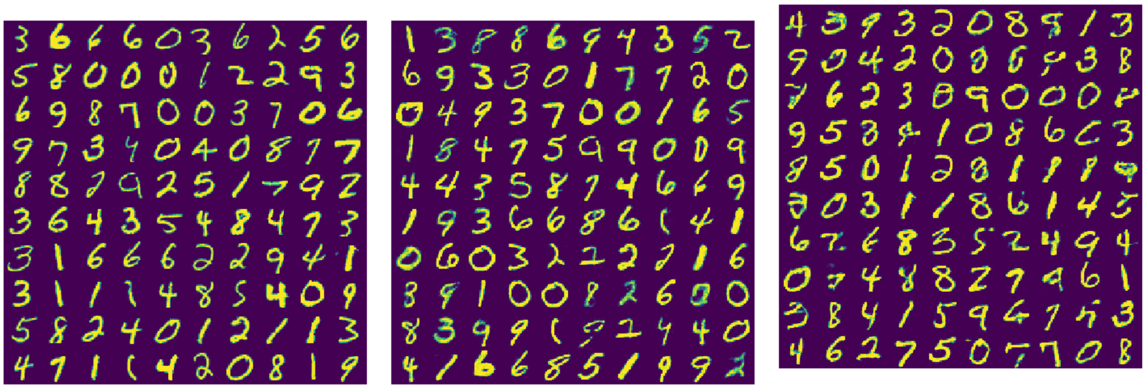
Table 1

Transportation of heavy-tails to Gaussian (cases 1–4) and Gaussian to heavy-tails (cases 5–8) by (f,Γ_1)-GPA with f_{KL} and f_α with $\alpha = 2$. When the algorithm collapses, the corresponding time is reported. In other cases, the converged $D_f^{\Gamma_1}(P_T\|Q)$'s are reported.

Case	GPA source P_0	GPA target Q	$D_{\mathrm{KL}}^{\Gamma_1}$	$D_\alpha^{\Gamma_1}$ with $\alpha = 2$
1	$GGM(0.5)$	$\mathcal{N}((10,10), 0.5^2 I)$	$O(10^{-6})$	$O(10^{-6})$
2	$Student - t(3)$	$\mathcal{N}((10,10), 0.5^2 I)$	$O(10^{-4})$	$O(10^{-4})$
3	$Student - t(1.5)$	$\mathcal{N}((10,10), 0.5^2 I)$	**diverged at** $t = 0$	$O(10^{0})$
4	$Student - t(0.5)$	$\mathcal{N}((10,10), 0.5^2 I)$	**diverged at** $t = 0$	$O(10^{7})$
5	$\mathcal{N}((10,10), 0.5^2 I)$	$GGM(0.5)$	$O(10^{-6})$	$O(10^{-3})$
6	$\mathcal{N}((10,10), 0.5^2 I)$	$Student - t(3)$	$O(10^{-6})$	$O(10^{-4})$
7	$\mathcal{N}((10,10), 0.5^2 I)$	$Student - t(1.5)$	$O(10^{-3})$	$O(10^{-3})$
8	$\mathcal{N}((10,10), 0.5^2 I)$	$Student - t(0.5)$	**diverged at** $t = 202$	$O(10^{-1})$

target data, GPA for data augmentation, and GPA for approximating a multiscale distribution represented by scarce data. Experiments for the first two applications are conducted following the strategies outlined in section 4 to uphold the generalization properties of GPA.

GPA for image generation given scarce target data. Here we consider the example of MNIST image generation using GPA, given a target data set that is relatively sparse compared to the corresponding spatial dimensionality. Recall the entire MNIST data set has 60,000 images. We demonstrate an example of generating images for MNIST in $\mathbb{R}^{784}$ from 200 target samples in Figure 5. We showcase results from our first two strategies in section 4 to ensure the generalization property of GPA: (i) the imbalanced sample sizes $M \gg N$ (Figure 5b) and



(a) Fixed target samples with sample size $N = 200$

(b) $M = 600$ transported particles from $(f_{\mathrm{KL}}, \Gamma_5)$-GPA

(c) 600 generated particles that are simultaneously transported from $(f_{\mathrm{KL}}, \Gamma_5)$-GPA

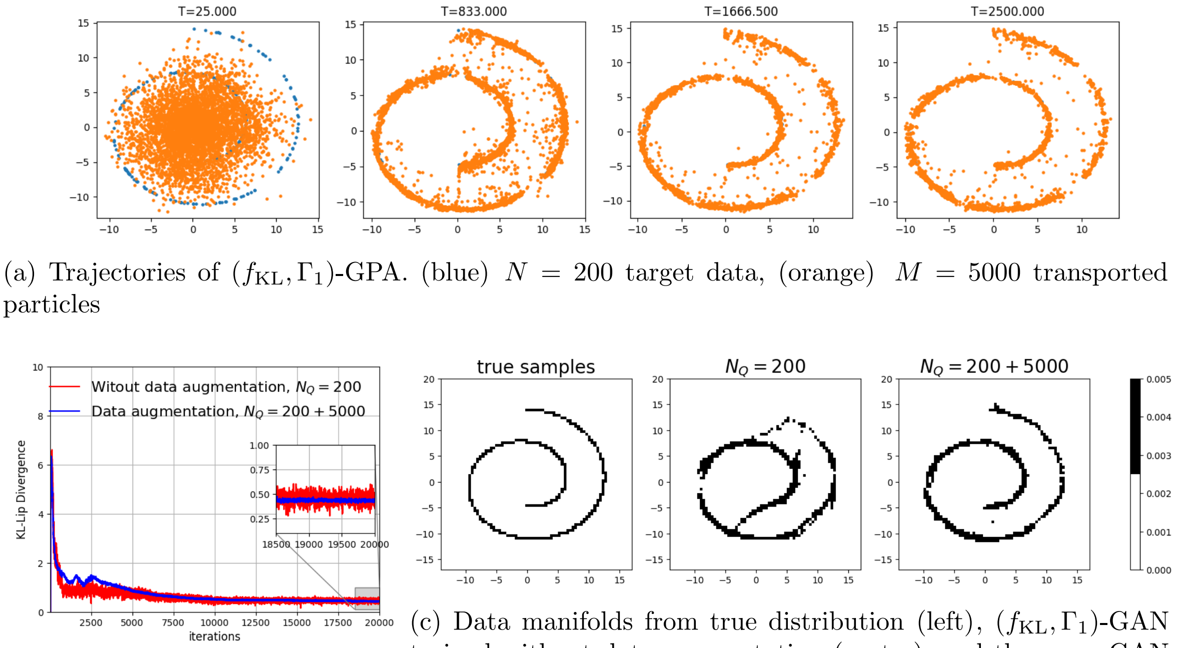
Figure 5. *(MNIST) GPA for image generation given scarce target data.* (a) *A subset of the* $N = 200$ *target samples. Results in* (b)–(c) *are generated by* $(f_{\mathrm{KL}}, \Gamma_5)$*-GPA based on the first two strategies in section* 4. *We report GPA results with* $L = 5$*, which was empirically found to generate samples stably and in a reasonable amount of time.* (b) $M = 600$ *initial particles from* $Unif([0,1]^{784})$ *were transported toward the target in the setting of* $M \gg N$*, which promotes sample diversity. See Figure* SM5 *in the supplementary material for details.* (c) *A new set of* 600 *initial particles from* $Unif([0,1]^{784})$ *were transported through the previously learned vector fields. These transported samples are referred to as generated particles, as explained in section* 4. *Training time:* 5000 *time steps (*$T = 2500$*) or* 48 *minutes in the setting in supplementary material section* SM3.1.

(ii) the generated particles that are simulataneously transported with M training particles (Figure 5c). In addition, we highlight the efficiency of GPA in training time and target sample size by comparing GPA against WGAN [4] and SGM [58] in Figure 10, in a scarce data regime. On the other hand, for a demonstration of scalability of GPA in the number of data, we refer to Figure SM6.

GPA for data augmentation. Here, we further verify the capabilities of GPA to learn from scarce target data in low- and high-dimensional examples such as Figures 6 and 9. Specifically, GPA can serve as a data augmentation tool for GANs or other generative models, including variational autoencoders [31], autoencoders, and conditional generative models. These models often require a substantial amount of target data in order to enable effective learning of generators. GPA provides augmented data needed for the proper training of the generative model with both sample diversity and quality, as depicted in Figures 6 and 9. An additional advantage of GPA augmentation is that proximity between the augmented data and the original data can be monitored and controlled by the GPA termination time T. Indeed, the (f, Γ_L)-divergence, one of the estimators of GPA in Algorithm 3.1, ensures that the divergence between these datasets remains below the tolerance error ϵ_{TOL}:

$$D^{\Gamma_1}_{f_{\mathrm{KL}}}(P_T \| Q) \leq \epsilon_{TOL}. \tag{8.1}$$

Other data augmentation techniques, such as small noise injection or transformations, do not inherently ensure the proximity to the target distribution, as captured in (8.1). Here



(a) Trajectories of $(f_{\mathrm{KL}}, \Gamma_1)$-GPA. (blue) $N = 200$ target data, (orange) $M = 5000$ transported particles

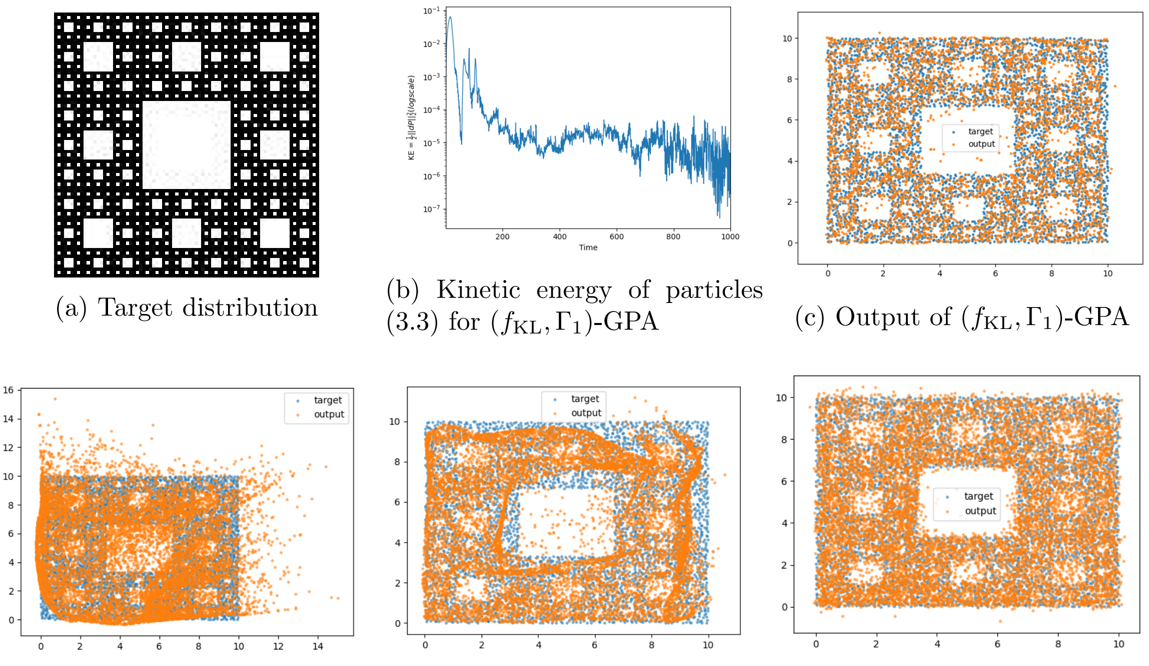
(b) $(f_{\mathrm{KL}}, \Gamma_1)$-divergences

(c) Data manifolds from true distribution (left), $(f_{\mathrm{KL}}, \Gamma_1)$-GAN trained without data augmentation (center) and the same GAN trained with data augmentation (right).

Figure 6. *(Swiss roll) Data augmentation using GPA.* (a) *Given $N = 200$ samples from the Swiss roll uniform distribution Q, $M = 5000$ additional samples are generated by transporting initial samples from $P_0 = \mathcal{N}(0, 3^2 I)$ using $(f_{\mathrm{KL}}, \Gamma_1)$-GPA. Imbalanced sample sizes $M \gg N$ strategy in section 4 is used to ensure sample diversity. Particles at $T = 2500$ with $D^{\Gamma_1}_{f_{\mathrm{KL}}}(P_T \| Q) \leq 1.07 * 10^{-4}$ are used as the augmented data.* (b) *When $(f_{\mathrm{KL}}, \Gamma_1)$-GAN is trained from 200 original samples* (red)*, the loss (divergence) oscillates; see inset in* (b)*. To improve the GAN, we train it with 200 original + 5000 augmented samples. By GPA-data augmentation, the GAN loss decreases stably; see inset in* (b)*.* (c) *GPA-augmented data significantly enhanced the learning of the manifold when using a GAN on the 5200 samples.*

we present two examples for this purpose. First, we use a Swiss roll example in Figure 6 to illustrate the procedure and features of GPA augmentation. Furthermore, in Figure 9, we showcase a high-dimensional and consequently more intriguing example of data augmentation for the MNIST dataset. This illustration demonstrates that a WGAN trained with GPA augmented data performs similarly to one trained with original, real data of the same size. In conclusion, we demonstrated how to employ GPA for data augmentation as another strategy for acquiring the generalization properties discussed in section 4.

GPA for multiscale distribution. We consider a target distribution with a multiscale (fractal) structure such as a Sierpinski carpet of level 4. Namely, this uniform distribution is constructed from a fractal set by keeping the four largest scales and truncating all finer scales. We refer to Figure 7a where we consider 4096 target particles in $[0, 10] \times [0, 10]$. Each target particle is random-sampled only once in each dark pixel with size of $[0, 10/3^4] \times [0, 10/3^4]$. We transport 4096 initial samples from $N(0, 3^2 I)$ using $(f_{\mathrm{KL}}, \Gamma_1)$-GPA. Figures 7b and 7c indicate that $(f_{\mathrm{KL}}, \Gamma_1)$-GPA approximates the target distribution and stops in a reasonable time $T = 1000$



(a) Target distribution

(b) Kinetic energy of particles (3.3) for $(f_{\mathrm{KL}}, \Gamma_1)$-GPA

(c) Output of $(f_{\mathrm{KL}}, \Gamma_1)$-GPA

(d) Output of WGAN [4]

(e) Output of $(f_{\mathrm{KL}}, \Gamma_1)$-GAN [5]

(f) Output of SGM [58]

Figure 7. *(Sierpinski carpet of level* 4*) GPA for multiscale distributions. GPA demonstrates superior performance over two widely employed generative models in approximating multiscale distributions.* (a) *The problem is to approximate a target distribution with four different scales using* 4096 *samples.* (b)–(c) *The* $(f_{\mathrm{KL}}, \Gamma_1)$*-GPA successfully transports* 4096 *Gaussian samples to capture the three largest scales of the target distribution.* (d)–(e) *GANs exhibit notably inferior performance compared to GPA, even when sharing the same discriminator structure and loss function, as evidenced in* (e)*. See also supplementary material section* SM4*.* (f) *SGM is unable to capture finer scales, even with prolonged training.*

with time steps $n = 5000$. We also refer to the related three-dimensional (3D) result in Figure 1, where particles in three dimensions find a multiscale structure in the 2D plane. On the other hand, training the generator for a multiscale distribution with the given dataset size posed a significant challenge for both Wasserstein GAN [4], $(f_{\mathrm{KL}}, \Gamma_1)$-GAN [5] and score-based generative models (SGM) [58], as evident in Figures 7d to 7f.

9. Latent-space GPA for high-dimensional dataset integration.

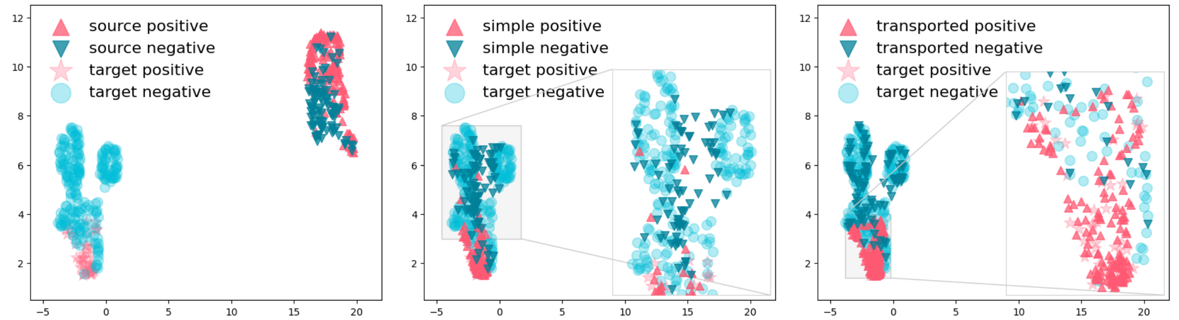
The integration of two or more datasets that essentially contain the same information, yet whose statistical properties are different due to, e.g., distributional shifts is crucial for the successful training and deployment of statistical and machine learning models [32, 26, 53]. Taking bioinformatics as an example, datasets, even when they study the same disease, have been created from different labs around the globe resulting in statistical differences which are also known as batch effects [59]. Furthermore, those datasets often have low sample size due to budget constraints or limited availability of patients (e.g., rare diseases). GPA offers an elegant solution for dataset integration by transporting samples from one dataset to another. Unlike the standard generation process, where the source distribution typically needs to be simple and explicit (e.g., isotropic Gaussian), GPA imposes no assumptions on the source and target distributions. It can also produce stable and accurate results even with very small sample sizes, as demonstrated in section 8. However, applying GPA becomes challenging when the dimensionality of the data rests in the order of tens of thousands. Therefore, we first substantially reduce the dimensionality of the data before employing GPA. After the dimensionality reduction, we apply GPA in the latent space and, when necessary, reconstruct the transported data back to its original high-dimensional space. *This three-step approach efficiently transports samples from the source dataset to the target dataset.* Additionally, it is worth noting that the error resulting from the projection to a lower-dimensional latent space is handled via Theorem 5.1. This theorem states that when the target distribution is supported on a lower-dimensional manifold, it is theoretically guaranteed through the new data processing inequality that the error in the original space can be bounded by the error occurred in the latent space.

Gene expression datasets. We consider the integration of two gene expression datasets which are publicly available at https://www.ncbi.nlm.nih.gov/geo/ with accession codes GSE76275 and GSE26639. These datasets have been measured using the GLP570 platform which creates samples with $d = 54,675$ dimensions. Each dataset consists of a low number of data while each individual sample corresponds to the gene expression levels of a patient. Moreover, each sample is labeled by a clinical indicator which informs if the patient was positive or negative to ER (estrogen receptor); see Table 2. The dataset with accession code GSE26639 was selected as the source dataset, while GSE76275 was chosen as the target. In this example, we chose GSE76275 as the target due to its more distinguishable geometric structure compared to the source, as illustrated in Figure 8a. This choice is aimed at showcasing the transportation capabilities of GPA. However, in reality, the decision of selecting the source and target datasets depends on the user and the application context. Despite measuring the same quantities, a direct concatenation of the two datasets will result in erroneous statistics as is evident in Figure 8a where a 2D visualization reveals that the two datasets are completely separated.

Dimensionality reduction using PCA. Applying GPA, along with most machine learning models that do not utilize transfer learning, in the original high-dimensional space is especially challenging when dealing with a low sample size regime. Hence, we first perform dimensionality reduction constructing a latent space and subsequently perform GPA within the latent space. Specifically, we use invertible dimensionality reduction methods by deploying autoencoders suitable for the data. An autoencoder comprises of two functions: the encoder, denoted as $\mathcal{E}(\cdot)$, compresses information from a high-dimensional space to a lower-dimensional latent space, while the decoder, represented as $\mathcal{D}(\cdot)$, decompresses latent features back to the original

Table 2
Sample sizes of the studied gene expression datasets.

	Positive	Negative	Total
GSE26639 (source)	138	88	226
GSE76275 (target)	49	216	265



(a) Two gene expression data sets without any transformation.

(b) Dataset integration using mean and std adjustment.

(c) Dataset integration using latent $(f_{\mathrm{KL}}, \Gamma_1)$-GPA with $d' = 50$.

Figure 8. *Gene expression dataset integration by GPA. We integrate two high-dimensional gene expression datasets via GPA transportation.* (a) *A direct concatenation of the two datasets results in incorrect integration as visualized in the* 2*D plane using UMAP algorithm* [42]. (b) *The baseline approach consists of a mean and std adjustment of each feature in the original space. In the inset, we notice that transformed negative samples do not evenly cover the support of the negative target samples.* (c) *The proposed latent GPA data transportation results in transported distributions close to the target ones.*

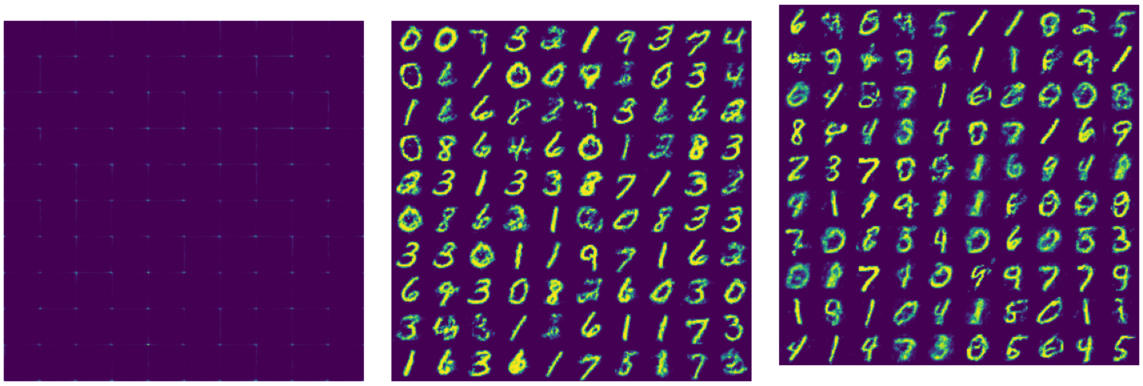
space. Given that training a nonlinear autoencoder based on NNs requires tens of thousands of samples, we choose PCA as a linear alternative [8, 29, 25]. Using PCA, we derive a d'-dimensional linear basis $\{\mathbf{v}_i\}_{i=1}^{d'}$ from the entire set of samples in both the source and the target datasets. Then each sample $\mathbf{x}$ is projected to a d'-dimensional space, defining the encoder as the corresponding projection: $\mathbf{z} = \mathcal{E}(\mathbf{x}) = \mathrm{Proj}_{\mathbf{v}_{1:d'}}(\mathbf{x})$. Subsequently, the GPA Algorithm 3.1 will be applied on the latent samples $\mathbf{z}$. The decoder $\mathbf{x} = \mathcal{D}(\mathbf{z})$ is also defined by PCA using a reconstruction on the entire d-dimensional space, e.g., [8, Chap. 12.1.2]. The decoder is 1-Lipschitz continuous since $\|\mathcal{D}(\mathbf{z}) - \mathcal{D}(\mathbf{z}')\|^2 = \|\sum_{i=1}^{d'}(z_i - z_i')\mathbf{v}_i\|^2 = \sum_{i=1}^{d'}|z_i - z_i'|^2\|\mathbf{v}_i\|^2 = \|\mathbf{z} - \mathbf{z}'\|^2$. Here we used that $\mathbf{v}_i$'s are orthonormal and that decoders $\mathcal{D}(\mathbf{z}), \mathcal{D}(\mathbf{z}')$ only differentiate on the d'-dimensional space in PCA [8, Chap. 12.1.2]. Here we chose $d' = 50$ to balance computational cost of Algorithm 3.1 and error between reconstructed and original datasets, aiming for a practically applicable approximation of an ideal encoder/decoder; see Figure SM7 in the supplementary material. In this context, Theorem 5.1 guarantees that the projection error remains controlled under encoding/decoding assuring that the performance of the transportation in the original space is dictated by the performance of the GPA in the latent space.

Results on dataset integration. We integrate gene expression datasets by applying the latent-space GPA, transporting samples from the positive-labeled source distribution to the corresponding positive-labeled target distribution and similarly for the negative-labeled data. The respective transportation maps $\mathcal{T}^{n,+}$ and $\mathcal{T}^{n,-}$ are composed of $(f_{\mathrm{KL}}, \Gamma_1)$-GPA transport maps as defined in (3.1), executed for $n = 5000$ time steps (and $\Delta t = 0.2$). Each of these separate transportation maps utilizes its own independent discriminator, each with its own unique parameters. The visualization of the dataset integration in Figure 8c shows that both positive and negative distributions have been efficiently transported via latent-space GPA.

As a comparison, we present a baseline data transformation for each class, denoted by $\mathcal{F}^+$ and $\mathcal{F}^-$, respectively, which performs mean and standard deviation (std) adjustment. As it is evident in Figure 8b, the baseline dataset integration only partially relocates the samples from the transformed distribution to the target distribution. The discrepancies are especially pronounced in the negative samples (see inset in Figure 8b).

We quantify the distributional differences between the transported and target distributions via the 2-Wasserstein distance [18, 20] in Table SM2 in the supplementary material, which is a metric not used in latent GPA and can also be efficiently computed with the Sinkhorn algorithm. In summary, the 2-Wasserstein distance between datasets in the original space ($d = 54,675$) is reduced by two orders of magnitude (1.4726% on positive datasets and 2.6104% on negative datasets), while GPA is twice as effective compared to the baseline mean and standard deviation adjustment transformation (3.9526% on positive datasets and 4.8718% on negative datasets). Finally, we remark that there are other metrics that can be used to assess the quality of the latent GPA-based dataset integration. For instance, the merged dataset can be tested on subsequent tasks such as phenotype classification or feature selection and evaluate the relative improvement resulting from the integration. We reserve this type of evaluation for future research since it is beyond the scope of this paper. Conducting such an analysis would require dedicated experiments and comparisons specific to the selected subsequent task.

Appendix. Here we provide Figures 9 and 10, discussed earlier in section 8.

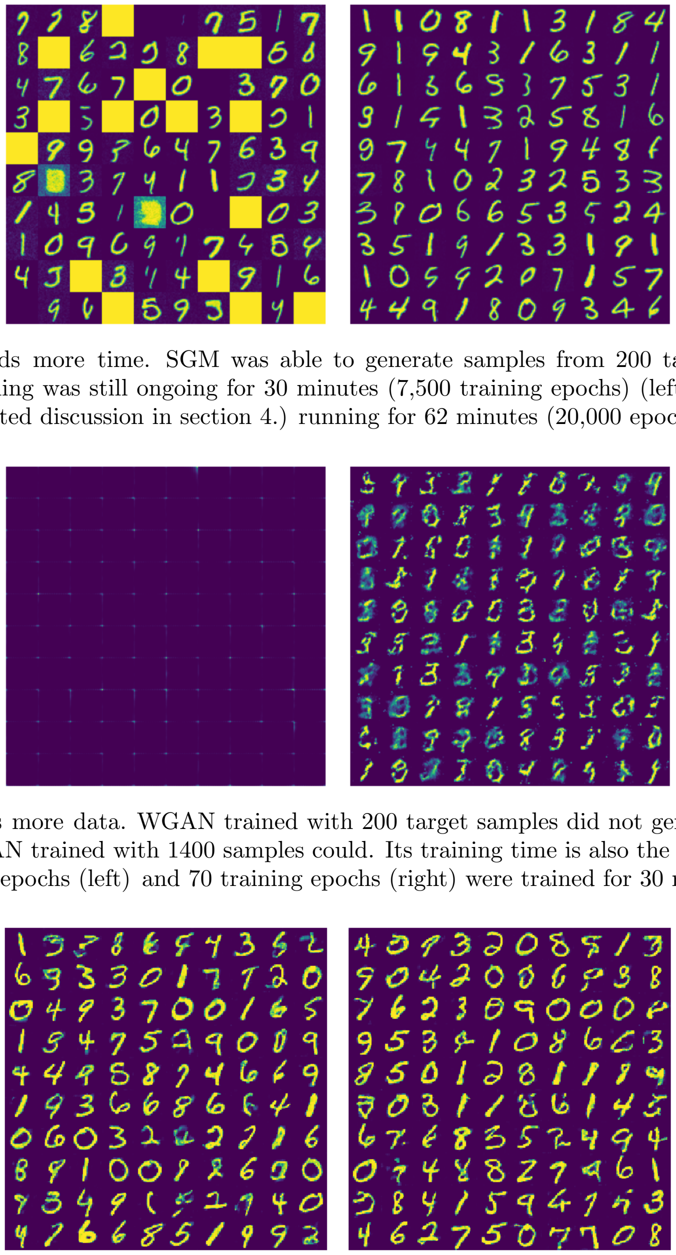


(a) WGAN [4] trained with 200 original data for 3000 training epochs

(b) WGAN [4] trained with original 1400 data for 500 training epochs

(c) WGAN [4] trained with 200 original data and 1200 GPA-augmented data for 500 training epochs

Figure 9. *(MNIST) Performance of data augmentation using GPA in a high-dimensional example.* (a) *WGAN was not able to learn from* 200 *original samples from the MNIST data base.* (b) *WGAN trained with* 1400 *original data can now generate samples but in a moderate quality.* (c) *We obtained* 600 *GPA-transported data in Figure* 5b *and* 600 *generated data in Figure* 5c *(see section* 4*) from the* 200 *original target samples and used them for augmenting data to train a WGAN with a mixture of* 1400 *real, transported, and generated samples in total. Such a GAN generated samples of similar quality compared to the GAN trained with* 1400 *original samples in* (b).



(a) SGM [58] needs more time. SGM was able to generate samples from 200 target samples. However, the training was still ongoing for 30 minutes (7,500 training epochs) (left), and eventually overfitted (see related discussion in section 4.) running for 62 minutes (20,000 epochs) (right).

(b) GAN [4] needs more data. WGAN trained with 200 target samples did not generate samples while the same GAN trained with 1400 samples could. Its training time is also the slowest among the three models: 350 epochs (left) and 70 training epochs (right) were trained for 30 minutes.

(c) GPA is "just right". $(f_{\mathrm{KL}}, \Gamma_5)$-GPA generated samples from $N = 200$ target samples in two different ways in section 4: (i) transporting $M = 600 \gg N$ samples (left), and (ii) generating additional 600 samples by transporting through the learned vector fields (right). Both settings in (i) and (ii) were able to produce samples. Lastly, 3160 training epochs were trained for 30 minutes.

Figure 10. *(MNIST) Comparison of image generation via GPA to SGM and GAN models. We demonstrate the efficiency of training GPA for image generation in terms of both training time and target sample sizes. The baseline setting restricts training time to* 30 *minutes and provides a fixed number of* 200 *target samples to each model. GPA learns to generate samples within this restricted setting, while SGM requires longer training time and WGAN requires more data to be trained.*

REFERENCES

[1] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, Lectures Math. ETH Zürich, Birkhäuser Verlag, Basel, 2005.

[2] L. Ambrosio and D. Trevisan, *Lecture notes on the DiPerna–Lions theory in abstract measure spaces*, Ann. Fac. Sci. Toulouse Math. (6), 26 (2017), pp. 729–766.

[3] M. Arbel, A. Korba, A. Salim, and A. Gretton, *Maximum mean discrepancy gradient flow*, in Advances in Neural Information Processing Systems, Vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., Curran Associates, 2019, pp. 6484–6494, https://proceedings.neurips.cc/paper/2019/file/944a5ae3483ed5c1e10bbccb7942a279-Paper.pdf.

[4] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein generative adversarial networks*, in Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research 70, PMLR, D. Precup and Y. W. Teh, eds., 2017, pp. 214–223, https://proceedings.mlr.press/v70/arjovsky17a.html.

[5] J. Birrell, P. Dupuis, M. A. Katsoulakis, Y. Pantazis, and L. Rey-Bellet, *$(f$-$\gamma)$-divergences: Interpolating between f-divergences and integral probability metrics*, J. Mach. Learn. Res., 23 (2022), 39.

[6] J. Birrell, M. A. Katsoulakis, and Y. Pantazis, *Optimizing variational representations of divergences and accelerating their statistical estimation*, IEEE Trans. Inform. Theory, 68 (2022), pp. 4553–4572, https://doi.org/10.1109/TIT.2022.3160659.

[7] J. Birrell, M. A. Katsoulakis, L. Rey-Bellet, and W. Zhu, *Structure-preserving GANs*, in Proceedings of the 39th International Conference on Machine Learning, ICML 2022, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, S. Niu, and S. Sabato, eds., Proceedings of Machine Learning Research 162, PMLR, Baltimore, MD, 2022, pp. 1982–2020, https://proceedings.mlr.press/v162/birrell22a.html.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.

[9] N. M. Boffi and E. Vanden-Eijnden, *Probability Flow Solution of the Fokker-Planck Equation*, arXiv e-prints, https://arxiv.org/abs/2206.04642, 2022.

[10] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, *Extracting training data from diffusion models*, in Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23), 2023, pp. 5253–5270.

[11] J. A. Carrillo, Y. Huang, F. S. Patacchini, and G. Wolansky, *Numerical study of a particle method for gradient flows*, Kinet. Relat. Models, 10 (2017), pp. 613–641.

[12] C. Chen, C. Li, L. Chen, W. Wang, Y. Pu, and L. C. Duke, *Continuous-time flows for efficient inference and density estimation*, in Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research 80, PMLR, J. Dy and A. Krause, eds., 2018, pp. 824–833, https://proceedings.mlr.press/v80/chen18d.html.

[13] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, *Neural ordinary differential equations*, in Advances in Neural Information Processing Systems, Vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, 2018, pp. 6572–6583, https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

[14] R. J. DiPerna and P.-L. Lions, *Ordinary differential equations, transport theory and Sobolev spaces*, Invent. Math., 98 (1989), pp. 511–547.

[15] J. Dolbeault, I. Gentil, A. Guillin, and F.-Y. Wang, *L^q-functional inequalities and weighted porous media equations*, Potential Anal., 28 (2008), pp. 35–59, https://doi.org/10.1007/s11118-007-9066-0.

[16] P. Dupuis and Y. Mao, *Formulation and properties of a divergence used to compare probability measures without absolute continuity*, ESAIM Control Optim. Calc. Var., 28 (2022), 10.

[17] A. Durmus and E. Moulines, *Nonasymptotic convergence analysis for the unadjusted Langevin algorithm*, Ann. Appl. Probab., 27 (2017), pp. 1551–1587, https://doi.org/10.1214/16-AAP1238.

[18] J. Feydy, *Geometric Data Analysis, Beyond Convolutions*, 2020, https://www.jeanfeydy.com/geometric_data_analysis.pdf (accessed 2020-07-02).

[19] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, *Stochastic optimization for large-scale optimal transport*, in Advances in Neural Information Processing Systems, Vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds., Curran Associates, 2016, pp. 3440–3448, https://proceedings.neurips.cc/paper/2016/file/2a27b8144ac02f67687f76782a3b5d8f-Paper.pdf.

[20] A. Genevay, G. Peyre, and M. Cuturi, *Learning generative models with Sinkhorn divergences*, in Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, A. Storkey and F. Perez-Cruz, eds., Proceedings of Machine Learning Research 84, PMLR, 2018, pp. 1608–1617, https://proceedings.mlr.press/v84/genevay18a.html.

[21] P. Glaser, M. Arbel, and A. Gretton, *KALE flow: A relaxed KL gradient flow for probabilities with disjoint support*, in Advances in Neural Information Processing Systems, Vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds., Curran Associates, 2021, pp. 8018–8031, https://proceedings.neurips.cc/paper/2021/file/433a6ea5429d6d75f0be9bf9da26e24c-Paper.pdf.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Adv. Neural Inf. Process. Syst., 27 (2014).

[23] X. Gu, C. Du, T. Pang, C. Li, M. Lin, and Y. Wang, *On Memorization in Diffusion Models*, preprint, https://arxiv.org/abs/2310.02664, 2023.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, *Improved training of Wasserstein GANs*, in Advances in Neural Information Processing Systems, Vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Curran Associates, 2017, pp. 5769–5779, https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer, New York, 2001.

[26] J. Hendler, *Data integration for heterogenous datasets*, Big Data, 2 (2014), pp. 205–215.

[27] J. Ho, A. Jain, and P. Abbeel, *Denoising Diffusion Probabilistic Models*, https://arxiv.org/abs/2006.11239, 2020.

[28] A. Hyvärinen, *Estimation of non-normalized statistical models by score matching*, J. Mach. Learn. Res., 6 (2005), pp. 695–709, http://dl.acm.org/citation.cfm?id=1046920.1088696.

[29] I. Jolliffe and J. Cadima, *Principal component analysis: A review and recent developments*, Philos. Trans. Roy. Soc. A, 374 (2016), 20150202, https://doi.org/10.1098/rsta.2015.0202.

[30] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the Fokker–Planck equation*, SIAM J. Math. Anal., 29 (1998), pp. 1–17, https://doi.org/10.1137/S0036141096303359.

[31] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, arXiv e-prints, https://arxiv.org/abs/1312.6114, 2013.

[32] P. Kirk, J. E. Griffin, R. S. Savage, Z. Ghahramani, and D. L. Wild, *Bayesian correlated clustering to integrate multiple datasets*, Bioinformatics, 28 (2012), pp. 3290–3297.

[33] J. Köhler, L. Klein, and F. Noe, *Equivariant flows: Exact likelihood generative learning for symmetric densities*, in Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research 119, PMLR, H. D. III and A. Singh, eds., 2020, pp. 5361–5370, https://proceedings.mlr.press/v119/kohler20a.html.

[34] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, *A tutorial on energy-based learning*, in Predicting Structured Data, G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, eds., MIT Press, Cambridge, 2006.

[35] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, Philadelphia, 2007.

[36] S. Li, S. Chen, and Q. Li, *A Good Score Does Not Lead to a Good Generative Model*, preprint, https://arxiv.org/abs/2401.04856, 2024.

[37] Q. Liu, *Stein variational gradient descent as gradient flow*, in Advances in Neural Information Processing Systems, Vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Curran Associates, 2017, https://proceedings.neurips.cc/paper/2017/file/17ed8abedc255908be746d245e50263a-Paper.pdf.

[38] Q. Liu and D. Wang, *Stein variational gradient descent: A general purpose Bayesian inference algorithm*, in Advances in Neural Information Processing Systems, Vol. 29, D. Lee, M. Sugiyama, U. Luxburg,

I. Guyon, and R. Garnett, eds., Curran Associates, 2016, pp. 2378–2386, https://proceedings.neurips.cc/paper/2016/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf.

[39] J. Lu, Y. Lu, and J. Nolen, *Scaling limit of the Stein variational gradient descent: The mean field regime*, SIAM J. Math. Anal., 51 (2019), pp. 648–671, https://doi.org/10.1137/18M1187611.

[40] X. Mao, Q. Li, H. Xie, R. K. Lau, Z. Wang, and S. Smolley, *Least squares generative adversarial networks*, in Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE Computer Society, Los Alamitos, CA, 2017, pp. 2813–2821, https://doi.org/10.1109/ICCV.2017.304.

[41] D. Maoutsa, S. Reich, and M. Opper, *Interacting particle solutions of Fokker–Planck equations through gradient–log–density estimation*, Entropy, 22 (2020), 802, https://doi.org/10.3390/e22080802.

[42] L. McInnes, J. Healy, N. Saul, and L. Grossberger, *UMAP: Uniform manifold approximation and projection*, J. Open Source Software, 3 (2018), 861, https://doi.org/10.21105/joss.00861.

[43] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, *Spectral normalization for generative adversarial networks*, in 6th International Conference on Learning Representations, ICLR, Vancouver, BC, Canada, 2018.

[44] Y. Mroueh, T. Sercu, and A. Raj, *Sobolev descent*, in Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 2976–2985.

[45] S. Nowozin, B. Cseke, and R. Tomioka, *F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization*, in Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates, Red Hook, NY, 2016, pp. 271–279.

[46] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, *OT-Flow: Fast and Accurate Continuous Normalizing Flows via Optimal Transport*, preprint, https://arxiv.org/abs/2006.00104, 2021.

[47] F. Otto, *The geometry of dissipative evolution equations: The porous medium equation*, Comm. Partial Differential Equations, 26 (2001), pp. 101–174, https://doi.org/10.1081/PDE-100002243.

[48] F. Otto and C. Villani, *Generalization of an inequality by Talagrand and links with the logarithmic Sobolev inequality*, J. Funct. Anal., 173 (2000), pp. 361–400, https://doi.org/10.1006/jfan.1999.3557.

[49] J. Pidstrigach, *Score-Based Generative Models Detect Manifolds*, preprint, https://arxiv.org/abs/2206.01018, 2022.

[50] S. Reich and S. Weissmann, *Fokker–Planck particle systems for Bayesian inference: Computational approaches*, SIAM/ASA J. Uncertain. Quantif., 9 (2021), pp. 446–482, https://doi.org/10.1137/19M1303162.

[51] G. O. Roberts and R. L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2 (1996), pp. 341–363, https://doi.org/10.2307/3318418.

[52] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-Resolution Image Synthesis with Latent Diffusion Models*, CoRR, preprint, https://arxiv.org/abs/2112.10752, 2021.

[53] J. Samuelsen, W. Chen, and B. Wasson, *Integrating multiple data sources for learning analytics—Review of literature*, Res. Practice Technol. Enhanced Learn., 14 (2019), 11, https://doi.org/10.1186/s41039-019-0105-4.

[54] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, *Deep Unsupervised Learning Using Nonequilibrium Thermodynamics*, preprint, https://arxiv.org/abs/1503.03585, 2015.

[55] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. 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, 2023, pp. 6048–6058.

[56] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, *Understanding and Mitigating Copying in Diffusion Models*, preprint, https://arxiv.org/abs/2305.20086, 2023.

[57] Y. Song and S. Ermon, *Generative Modeling by Estimating Gradients of the Data Distribution*, preprint, https://arxiv.org/abs/1907.05600, 2020.

[58] Y. Song, J. N. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-Based Generative Modeling through Stochastic Differential Equations*, preprint, https://arxiv.org/abs/2011.13456, 2021.

[59] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, *A benchmark of batch-effect correction methods for single-cell RNA sequencing data*, Genome Biol., 21 (2020), 12.

[60] A. Vahdat, K. Kreis, and J. Kautz, *Score-based generative modeling in latent space*, in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing

Systems 2021, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds., NeurIPS, 2021, pp. 11287–11302.

[61] J. L. Vázquez, *Barenblatt solutions and asymptotic behaviour for a nonlinear fractional heat equation of porous medium type*, J. Eur. Math. Soc., 16 (2014), pp. 769–803.

[62] B. J. Zhang and M. A. Katsoulakis, *A Mean-Field Games Laboratory for Generative Modeling*, preprint, https://arxiv.org/abs/2304.13534, 2023.

[63] B. J. Zhang, S. Liu, W. Li, M. A. Katsoulakis, and S. J. Osher, *Wasserstein Proximal Operators Describe Score-Based Generative Models and Resolve Memorization*, https://arxiv.org/abs/2402.06162, 2024.