JKONet: Proximal Optimal Transport Modeling of Population Dynamics

Charlotte Bunne [1] **Laetitia Meng-Papaxanthos** [2] **Andreas Krause** [1] **Marco Cuturi** [2]

Abstract

Consider a heterogeneous population of points evolving with time. While the population evolves, both in size and nature, we can observe it periodically, through snapshots taken at different timestamps. Each of these snapshots is formed by sampling points from the population at that time, and then creating features to recover point clouds. While these snapshots describe the population's evolution on aggregate, they do not provide directly insights on individual trajectories. This scenario is encountered in several applications, notably single-cell genomics experiments, tracking of particles, or when studying crowd motion. In this paper, we propose to model that dynamic as resulting from the celebrated Jordan-Kinderlehrer-Otto (JKO) proximal scheme. The JKO scheme posits that the configuration taken by a population at time t is one that trades off a decrease w.r.t. an energy (the model we seek to learn) penalized by an optimal transport distance w.r.t. the previous configuration. To that end, we propose JKONet, a neural architecture that combines an energy model on measures, with (small) optimal displacements solved with input convex neural networks (ICNN). We demonstrate the applicability of our model to explain and predict population dynamics.

1. Introduction

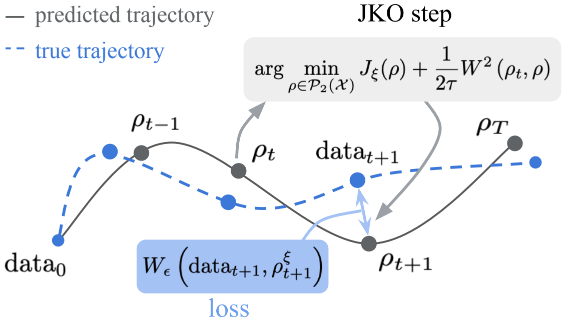

— predicted trajectory
-- true trajectory
JKO step
$\arg\min_{\rho\in\mathcal{P}_2(\mathcal{X})} J_\xi(\rho) + \frac{1}{2\tau} W^2(\rho_t, \rho)$
ρ_{t-1}
ρ_t
data_{t+1}
ρ_T
data_0
$W_\epsilon\left(\text{data}_{t+1}, \rho^\xi_{t+1}\right)$
ρ_{t+1}
loss

Figure 1. Given an observed trajectory of point clouds (blue), we seek a parameter ξ for the parameterized energy J_ξ such that the reconstructed JKO curve $\rho_0, \ldots, \rho_T$ (gray) is as close as possible to the blue trajectory. For this to happen, we minimize, as a function of ξ, the sum of (regularized) Wasserstein distances between the data observed at t and the output of the JKO module formed from ξ and ρ_{t-1}, obtained via an ICNN.

Population Dynamics Many fields in science draw insights by monitoring a complex system of interacting particles. Typically, this monitoring consists of sampling representative particles from that system at various timestamps to measure their characteristics. As a result, the observer has access to a collection of time-indexed discrete measures that describe the overall dynamic of the population; these measures provide a macroscopic view of the system (tracking the population's evolution) at different time points, but lack microscopic dynamic information (individuals cannot be tracked between timestamps). Such problems arise in many scientific studies, when for instance, observing a population of cells in biology to infer their developmental mechanisms (Schiebinger et al., 2019; Moon et al., 2019), or when monitoring brain activations in the cortex (Janati et al., 2020).

The JKO Scheme as a Model for Evolution The goal of this paper is to develop a new method to model such complex dynamics, by explaining the mechanism driving the population's time evolution. To this end, we exploit the Jordan-Kinderlehrer-Otto (JKO) flow (Jordan et al., 1998), widely regarded as one of the most influential mathematical papers in recent history. The JKO flow describes an iterative method that can solve partial differential equations (PDEs) such as the Fokker-Planck equations. Simply put, the JKO flow model states that the time evolution of a set of particles is controlled by an energy (a real-valued function defined on the space of measures): the particles take steps toward minimizing that energy, yet may not deviate too far from the previous configuration, as measured by the Wasserstein distance. A few approaches have been recently proposed to solve it in the literature (Burger et al., 2010; Carrillo et al., 2021; Peyré, 2015). In this work, our goal is to *differentiate* through the JKO flow, with the differentiating factor being the energy parameterization.

[1] Department of Computer Science, ETH Zurich [2] Google Research, Brain Team. Correspondence to: Charlotte Bunne <bunnec@ethz.ch>.

38^{th} *International Conference on Machine Learning* (ICML) 2021 Time-Series Workshop.

Contributions Our contributions are two-fold. First, we establish a novel way of solving JKO flows that builds on the recent proposal of input convex neural networks (ICNN) (Amos et al., 2017; Makkuva et al., 2020). We treat the JKO optimization as a single layer in our framework: given an energy and a configuration, the JKONET module outputs a new configuration by moving these particles along the gradient of an optimal ICNN. We then propose to differentiate a loss computed on the output of that module as a function of the energy itself, as illustrated in Figure 1. We demonstrate JKONET's range of applications by deploying it to potential- and trajectory-based population dynamics.

2. Background

Optimal Transport Let $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$ be two discrete probability measures in $\mathbb{R}^d$. Given $\varepsilon \geq 0$, the regularized optimal transport (OT) problem (Cuturi, 2013) reads

$$W_\varepsilon(\mu, \nu) := \min_{\mathbf{P} \in U(a,b)} \langle \mathbf{P}, [\|x_i - y_j\|^2]_{ij} \rangle - \varepsilon H(\mathbf{P}) \quad (1)$$

where $H(\mathbf{P}) := -\sum_{ij} \mathbf{P}_{ij}(\log \mathbf{P}_{ij} - 1)$, the polytope $U(a,b)$ is $\{\mathbf{P} \in \mathbb{R}_+^{n \times m}, \mathbf{P}\mathbf{1}_m = a, \mathbf{P}^\top \mathbf{1}_n = b\}$. Notice that the definition above reduces to that of the usual (squared) 2-Wasserstein distance when $\varepsilon = 0$. For computational reasons that involve parallelism, speed and, most importantly in what follows, differentiability of W_ε with respect to its inputs, we use in this work $\varepsilon > 0$. A minor drawback of this setting lies in the fact that $W_\varepsilon(\mu, \mu) \neq 0$ in general. To correct that bias, we use the *Sinkhorn divergence* (Ramdas et al., 2017; Genevay et al., 2019; Salimans et al., 2018; Feydy et al., 2019) to recover a nonnegative discrepancy,

$$\overline{W}_\varepsilon(\mu, \nu) := W_\varepsilon(\mu, \nu) - \frac{1}{2}\left(W_\varepsilon(\mu, \mu) + W_\varepsilon(\nu, \nu)\right). \quad (2)$$

Brenier's Theorem The Brenier theorem 1987 states that for any two probability measures μ and ν supported on $\mathbb{R}^d$, if at least one of the two input measures (denoted μ) has a density, the optimal transport map between μ and ν is unique and can be uniquely defined as the gradient of a convex function ψ:

$$\begin{aligned} W_2^2(\mu, \nu) &= \inf_{T: T_\# \mu = \nu} \int_{\mathcal{X}} ||x - T(x)||^2 d\mu(x) \\ &= \int_{\mathcal{X}} ||x - \nabla\psi(x)||^2 d\mu(x) \end{aligned} \quad (3)$$

where $T^\star(x) = \nabla\psi(x)$ and ψ is the unique convex function (up to an additive constant) such that $(\nabla\psi)_\# \mu = \nu$, establishing an equivalence between the Kantorovich formulation of OT (1), and the Monge formulation involving maps.

JKO Flows In their seminal paper, Jordan et al. (1998) study diffusion processes under the lens of the optimal transport metric (see also Ambrosio et al., 2006) and introduce a scheme that is now known as the JKO flow (following the name of the authors): starting with an initial configuration ρ_0 they define iteratively for $t \geq 0$:

$$\rho_{t+1} = \arg\min_{\rho} J(\rho) + \frac{1}{2\tau} W^2(\rho_t, \rho). \quad (4)$$

These successive minimization problems defined on the set of probability measures $\mathcal{P}_2(\mathbb{R}^d)$ describe the evolution of a measure in the Wasserstein space. The JKO flow can thus be seen as the analogy of the usual proximal descent scheme, tailored for probability measures (Santambrogio, 2017, p.285). Jordan et al. (1998) show that in the limit where step size $\tau \to 0$, the measures describing the JKO flow can be interpreted as solutions to a very wide family of PDEs, chiefly among them the Fokker-Planck equations.

Convex Neural Architectures Convex neural network architectures are neural networks $f(x; \theta)$ with specific constraints on the architecture and parameters θ, such that the output is a convex function of some elements of the input x (Amos et al., 2017). We consider in this work *fully* input convex neural networks (ICNNs), such that the output is a convex function of the entire input x. A typical ICNN architecture is a k-layer, fully connected network such that, for $i = \{0, \cdots, k-1\}$:

$$h_{i+1} = a_i(W_i^x x + W_i^h h_i + b_i) \text{ and } f(x; \theta) = h_k, \quad (5)$$

where by convention, h_0 and W_0^h are 0, a_i are convex non-decreasing (non-linear) activation functions, $\theta = \{b_i, W_i^h, W_i^x\}_{i=0}^{k-1}$ are the weights and biases of the neural network, among which W_i^h are non-negative weights. Since Amos et al. (2017)'s work, convex neural architectures have been further extended and shown to capture relevant models despite these constraints (Amos et al., 2017; Chen et al., 2019; Makkuva et al., 2020; Huang et al., 2021).

3. Proximal Optimal Transport Model

Given snapshot observations $\text{data}_0, \ldots, \text{data}_T$ of a population, we posit that such an evolution follows a JKO flow for the free energy functional J_ξ, and our goal is to learn ξ. Our approach relies on finding a differentiable formulation, amenable for the resolution of the JKO step. This is needed because the JKO step is an implicit minimization (4), yet our goal is to build a loss that uses directly its solution (see Fig. 1). To that end, we introduce a novel approach to numerically solve JKO flows using ICNNs (§ 3.1), which allows to form a bilevel optimization problem that targets next the energy J_ξ (§ 3.2).

3.1. Reformulation of JKO Flows via ICNNs

Given an initial condition data_t and energy functional J_ξ, the JKO step consists in finding a measure ρ_{t+1} defined as the minimizer of Equation 4. That minimizer is then used to form a prediction for the population's configuration at time $t+1$. A direct approach aiming at solving ρ_{t+1} involves substantial computational costs: Different numerical schemes have been developed, e.g., based notably on Eulerian discretization of measures (Carrillo et al., 2021;

Benamou et al., 2016), and/or entropy-regularized optimal transport (Peyré, 2015). However, these methods are limited to small dimensions since the discretization of space grows exponentially. Apart from the Eulerian approach (Peyré, 2015), their direct differentiation is challenging.

We build upon Brenier's (1987) theorem to propose an explicit parameterization of the OT map when solving for the JKO scheme. As a result, we can bypass the computation of OT distances, relying instead on that alternative parameterization, which uses the gradient of an ICNN to define pushforward operations. This alternative parameterization allows to solve a variant of (4) parameterized using a family of ICNNs $\{\psi_\theta\}_\theta$, to set

$$\rho_{t+1}^{\xi} := \nabla(\psi_{\theta^\star})_{\#}\rho_t \tag{6}$$

given that $\theta^\star$ is implicitly defined through ξ and ρ_t as

$$\theta^\star := \arg\min_\theta \Big[J_\xi(\nabla\psi_{\theta\#}\rho_t) + \frac{1}{2\tau}\int \|x - \nabla\psi_\theta(x)\|^2 d\rho_t(x) \Big]. \tag{7}$$

3.2. Learning the Free Energy Functional

The energy function $J_\xi : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ can be parameterized using neural networks taking as inputs measures of variable size. Our model assumes that the entire observed dynamics is parameterized by that energy; the evolution between two steps being (locally) an optimal transport, governed by J_ξ, acting as a rudder. The more complex this dynamics, the more complex the energy J_ξ should be chosen from. In this first work we have limited ourselves to linear functions *in the space of measures*, that is expectations over ρ of a vector-input neural network E_ξ

$$J_\xi(\rho) := \int E_\xi(x) d\rho(x), \tag{8}$$

where $E_\xi : \mathbb{R}^d \to \mathbb{R}$ is a multi-layer perceptron (MLP). Future work will focus on inferring nonlinear energies on the space of probability measures that can account for population growth and decline, as well as interactions.

To address slow convergence and instabilities, we use teacher forcing (Williams and Zipser, 1989) to learn J_ξ through time. During training and validation, J_ξ uses the ground truth as input instead of predictions from the previous time step. At test time, we do not use teacher forcing.

3.3. Bilevel Formulation of JKONET

Learning the free energy functional J_ξ while solving each JKO step via an ICNN results in an advanced bilevel optimization problem. At each time step, we measure the accuracy of the predicted dynamics to the ground truth population dynamics data $[\text{data}_0, \text{data}_1, \ldots, \text{data}_T]$ via the Sinkhorn loss (2),

$$\min_\xi \sum_t \overline{W}_\varepsilon(\text{data}_{t+1}, \rho_{t+1}^{\xi}). \tag{9}$$

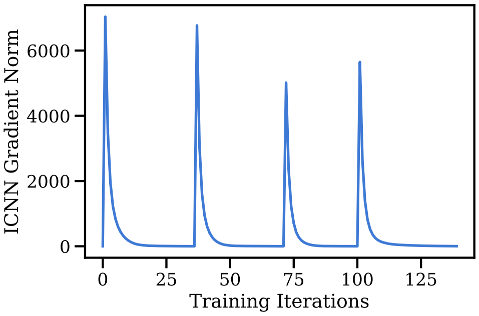


Figure 2. Optimization of the ICNN used in JKO steps. The bumps correspond to a change in the outer iteration, the smooth decrease in between correspond to a single minimization (7).

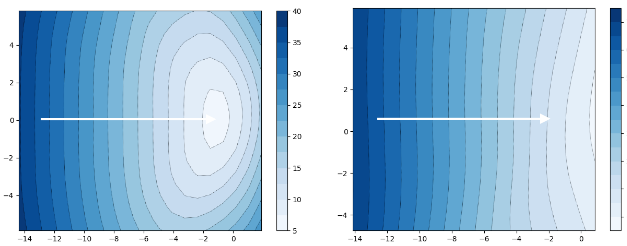
The dependence of the Sinkhorn divergence losses (9) on ξ only appears in the fact that the predictions ρ_{t+1}^{ξ} are themselves implicitly defined as solving a JKO step parameterized with the energy J_ξ. Learning J_ξ through the exclusive supervision of data observations requires therefore to differentiate the arg-minimum of a JKO problem, down therefore through to the lower-level optimization of the ICNN. We achieve this by implementing a differentiable double loop in `JAX`, differentiating first the Sinkhorn divergence using the `OTT`[1] package, and then backpropagating through the ICNN optimization by unrolling Adam steps (Kingma and Ba, 2014; Metz et al., 2017; Lorraine et al., 2020). The full procedure of JKONET is outlined in Algorithm 1.

Challenges A question that arises when defining ρ_{t+1}^{ξ} lies in the budget of gradient steps needed or allowed to optimize the parameters θ of the ICNN, before taking a new gradient step on ξ in the outer loss. A straightforward approach in `JAX` (Bradbury et al., 2018) would be to use a fixed number of iterations with a `for` loop (`jax.lax.scan`). We do observe, however, that the number of iterations needed to converge in relevant scenarios can vary significantly with the ICNN architecture and/or with the hardness of the underlying task. We propose to use instead a differentiable fixed-point loop to solve each JKO step up to a desired convergence threshold, using an adaptive number of iterations. We measure convergence of the optimization of the ICNN via the average norm of the gradient of the JKO objective w.r.t. the ICNN parameters θ, i.e., $\alpha = \frac{\sum_i \|\nabla_{\theta_i}\text{JKO}(\theta_i,\xi)\|_2}{\sum_i \text{count}(\theta_i)}$. We observe that this approach is robust across datasets and architectures of the ICNN. An exemplary training curve for the ICNNs updated successively along a time sequence is shown in Figure 2.

4. Evaluation

In the following, we evaluate our method on potential- and trajectory-based population dynamics, using convex (e.g., $J(x) = \|x\|_2^2$) and nonconvex potentials (e.g., Styblinski-

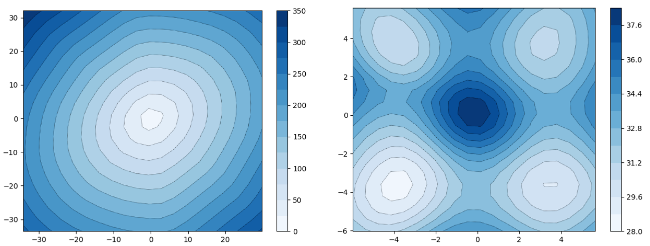
[1] https://github.com/google-research/ott



(a) Linear Potential Hashimoto et al. (2016)

(b) Linear Potential JKONET

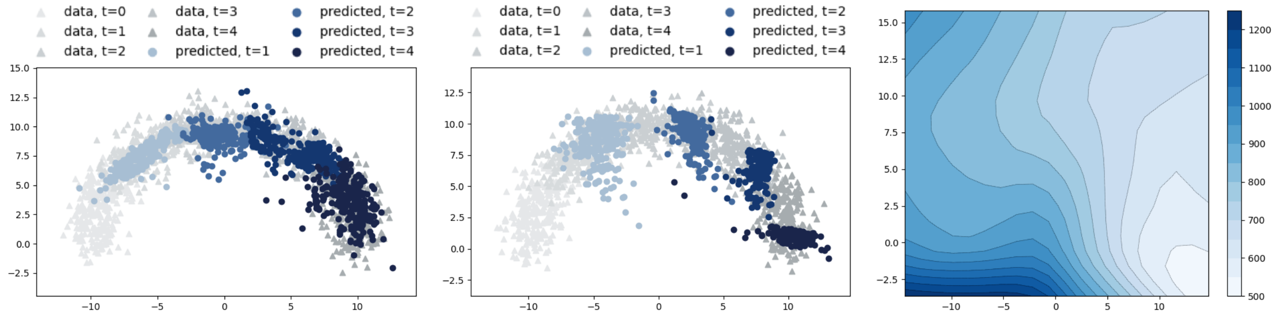
Figure 3. Comparison between learned energy functionals J_ξ based on *explicit* methods or JKONET.



(a) Quadratic Potential JKONET

(b) Styblinski-Tang Potential JKONET

Figure 4. Learned energy functionals J_ξ of JKONET on potential-based population dynamics.



(a) JKONET with Teacher Forcing

(b) JKONET without Teacher Forcing

(c) Learned energy functional J_ξ of JKONET

Figure 5. Trajectory-based population dynamics learned by JKONET.

Table 1. Comparison of JKONET to *explicit* methods for predicting and extrapolating linear translations (see Figure 3).

Method	**Sinkhorn Distance** ($\overline{W}_\varepsilon$) Validation	Test
Hashimoto et al. (2016)	**1.94 ± 0.06**	26.10 ± 1.76
JKONET	2.90 ± 0.37	**20.30 ± 0.65**

Tang flow). We generate the data using the Euler-Maruyama method (Kloeden and Platen, 1992). For details, see § C.

Comparison to *Explicit* Methods Instead of parameterizing the next iteration ρ_{t+1}^{ξ} as we do in the JKONET formulation 4, the *explicit* scheme simply states that ρ_{t+1} can be obtained as $(\nabla F_\xi)_{\#}\rho_t$, where F_ξ is any arbitrary neural network (Hashimoto et al., 2016; Salim et al., 2020). While this energy is still estimated by minimizing a Sinkhorn loss as in (9), this explicit approach can more easily get trapped in local minima. Figure 3 shows a simple experiment, in which we want to learn a translation. Due to the less constrained energy, the *explicit* method perfectly resembles the seen trajectory during training, but fails to extrapolate on shifted test data (see Table 1). For details, see § D.1.

Prediction of Synthetic Population Dynamics For the experiments on synthetic potential- and trajectory-based population dynamics, we parameterize both energy J_ξ and ICNN$_\theta$ with linear layers ($\epsilon = 0.1$, $\tau = 1.0$, § D.2). More details on the architectures can be found in § B. Figure 4 demonstrates JKONET's ability to recover convex and non-convex potentials via energy J_ξ. As a sanity check, we further evaluate if JKONET can recover a potential from trajectories (Figure 5). As described in § 3.2, J_ξ is trained using teacher forcing and receives a ground truth population for each time step. We test the learning dynamics without teacher forcing, i.e., when only provided with the initial distribution. JKONET successfully learns energies J_ξ from which one can infer the entire trajectory.

5. Conclusion

In this paper, we present JKONET, a model to infer and predict the evolution of population dynamics using a proximal optimal transport scheme, the JKO flow. Besides proposing a novel numerical scheme for solving JKO flows using ICNNs, we establish a framework to learn the underlying temporal dynamics of time-resolved snapshot data via a fully differentiable bilevel optimization problem. This approach is validated here through simple experimental on potential- and trajectory-based dynamics. Using proximal optimal transport to model real complex population dynamics makes for an exciting avenue of future work, including trajectory inference for time-resolved single-cell genomics, which will certainly require pushing the limits of our scheme by inferring nonlinear energies on the space of probability measures.

6. Acknowledgments

This project received funding from the Swiss National Science Foundation under the National Center of Competence in Research (NCCR) Catalysis under grant agreement 51NF40 180544, and was supported by Google Cloud for Higher Education.

References

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Springer, 2006.

B. Amos, L. Xu, and J. Z. Kolter. Input Convex Neural Networks. In *International Conference on Machine Learning (ICML)*, volume 34, 2017.

J.-D. Benamou, G. Carlier, Q. Mérigot, and E. Oudet. Discretization of functionals involving the Monge–Ampére operator. *Numerische Mathematik*, 134(3), 2016.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Y. Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305, 1987.

M. Burger, J. A. Carrillo, and M.-T. Wolfram. A mixed finite element method for nonlinear diffusion equations. *Kinetic & Related Models*, 3(1), 2010.

J. A. Carrillo, K. Craig, L. Wang, and C. Wei. Primal Dual Methods for Wasserstein Gradient Flows. *Foundations of Computational Mathematics*, 2021.

Y. Chen, Y. Shi, and B. Zhang. Optimal Control Via Neural Networks: A Convex Approach. In *International Conference on Learning Representations (ICLR)*, 2019.

M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

J. Feydy, T. Séjourné, F.-X. Vialard, S.-I. Amari, A. Trouvé, and G. Peyré. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, 2019.

A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré. Sample Complexity of Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, 2019.

T. Hashimoto, D. Gifford, and T. Jaakkola. Learning Population-Level Diffusions with Generative Recurrent Networks. In *International Conference on Machine Learning (ICML)*, volume 33, 2016.

C.-W. Huang, R. T. Q. Chen, C. Tsirigotis, and A. Courville. Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

H. Janati, M. Cuturi, and A. Gramfort. Spatio-Temporal Alignments: Optimal transport through space and time. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 23, 2020.

R. Jordan, D. Kinderlehrer, and F. Otto. The Variational Formulation of the Fokker–Planck Equation. *SIAM Journal on Mathematical Analysis*, 29(1), 1998.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

P. E. Kloeden and E. Platen. Stochastic Differential Equations. In *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.

J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing Millions of Hyperparameters by Implicit Differentiation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 23, 2020.

A. Makkuva, A. Taghvaei, S. Oh, and J. Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning (ICML)*, volume 37, 2020.

L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. van den Elzen, M. J. Hirn, R. R. Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12), 2019.

G. Peyré. Entropic Approximation of Wasserstein Gradient Flows. *SIAM Journal on Imaging Sciences*, 8(4), 2015.

A. Ramdas, N. G. Trillos, and M. Cuturi. On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests. *Entropy*, 19(2):47, 2017.

A. Salim, A. Korba, and G. Luise. The Wasserstein Proximal Gradient Algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

T. Salimans, H. Zhang, A. Radford, and D. Metaxas. Improving GANs Using Optimal Transport. In *International Conference on Learning Representations (ICLR)*, 2018.

F. Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1), 2017.

G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, et al. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4), 2019.

R. J. Williams and D. Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2), 1989.

Appendix

A. Proximal Optimal Transport Algorithm

JKONET provides a model to understand complex population dynamics, by inferring the mechanism driving the population's time evolution. This is achieved by solving a bilevel optimization problem which, given a potential function (the variable in the *upper level* problem) and an initial configuration of the data, outputs a new configuration by moving population particles along the JKO gradient flow, here approximated as the gradient of an ICNN that is recomputed precisely for that task (*lower level* objective). We describe the full framework in Algorithm 1. The Jacobian $\partial\rho_{t+1}^{\xi}/\partial\xi$ that appears when computing $\nabla_{\xi}\overline{W}_{\varepsilon}(\text{data}_{t+1}, \rho_{t+1}^{\xi})$ is computed by unrolling the iterations of the while loop above.

Algorithm 1 JKONET Algorithm.

Input: Dataset $\mathcal{D} = \{\{\text{data}_t^0\}_{t=0}^T, \{\text{data}_t^1\}_{t=0}^T, \ldots, \{\text{data}_t^N\}_{t=0}^T\}$ of time-resolved snapshot data, initial parameters ξ for free energy potential J_ξ, learning rates lr_θ and lr_ξ, JKO step size τ, Sinkhorn regularization parameter ε, and convergence threshold α. `TeacherForcing` is set to `True` during training.

Output: Optimal free energy functional $J_{\xi^\star}$ able to explain the underlying population dynamics of the snapshot data.

```
for data $\in \mathcal{D}$ do
    for $t \leftarrow 0$ to $T-1$ do
        initialize $\theta$
        if TeacherForcing then
            $\rho_t \leftarrow \text{data}_t$
        while $\frac{\sum_i \|\nabla_{\theta_i}\text{JKO}(\theta_i,\xi)\|_2}{\sum_i \text{count}(\theta_i)} \geq \alpha$ do
            $\text{JKO}(\theta,\xi) \leftarrow J_\xi(\nabla\psi_{\theta\#}\rho_t) + \frac{1}{2\tau}\int\|x - \nabla\psi_\theta(x)\|^2 d\,\rho_t(x)$
            $\theta \leftarrow \theta - \text{lr}_\theta \times \nabla_\theta\text{JKO}(\theta,\xi)$
        $\theta^\star \leftarrow \theta$
        $\rho_{t+1}^{\xi} \leftarrow \nabla(\psi_{\theta^\star})_{\#}\rho_t$
        $\xi \leftarrow \xi - \text{lr}_\xi \times \nabla_\xi\overline{W}_\varepsilon(\text{data}_{t+1}, \rho_{t+1}^{\xi})$
$\xi^\star \leftarrow \xi$
return $J_{\xi^\star}$
```

B. Network Architectures

In the following, we describe network architectures used in JKONET to parameterize the Brenier map ψ_θ (Section B.1) as well as the free energy functional J_ξ (Section B.2).

B.1. Parameterization of Brenier Map

In the following, we describe the architectural details of the ICNN, parametrizing the Brenier map ψ_θ. We set the hidden layer size of W_i^x and W_i^h (5) to 64 and use 3 hidden layers before the final output layer ($k = 4$ layers). Similar to (Makkuva et al., 2020), we use a squared leaky ReLU function with a small positive constant β as *convex* activation function for the first layer, i.e., $a_0(x) = \max(\beta x, x)^2$, and leaky ReLU $a_i(x) = \max(\beta x, x),\ i = 1, \ldots, k-1$ as *monotonically non-decreasing* and *convex* activation functions the remaining layers.

We tested the performance of the *vanilla* ICNN to advanced formulations such as input-augmented ICNNs (Huang et al., 2021), whereby no difference in performance is evident.

B.2. Parameterization of Energy Functional

The free energy functional J_ξ can take various forms, accounting for diffusion as well as potentials of interaction. In this work, we concentrate on linear functions in the space of measures (8). We parametrize E_ξ as a MLP with 2 hidden layers of size 64 with softplus activation functions, followed by a one-dimensional output layer. Future work will involve an extension of the framework to energy functionals covering higher-level interactions and population growth and decline.

C. Datasets

To evaluate JKONET we use different data sources, which are either defined from a ground truth potential or directly computed using velocity fields.

C.1. Potential-Based Dynamics

In the following, we assume a random diffusion process evolving according to an Îto stochastic difference equation (SDE) across time

$$dX(t) = -\nabla\Phi(X(t))dt + \sqrt{2\sigma^2}dB(t),$$

where $B(t)$ is the unit Brownian motion (standard Wiener process with magnitude $\sigma > 0$) and the drift is defined via a potential function $\Phi(x) : \mathbb{R}^d \to \mathbb{R}$. The population-level inference problem on $X(t)$ at each t then satisfies the Fokker-Planck equation with fixed diffusion coefficient

$$\frac{\partial \rho_t}{\partial t} = \operatorname{div}\left(\nabla\Phi(x)\rho_t\right) + \sigma^{-1}\Delta\rho_t$$

with given initial condition $\rho_0 = \rho^0$. We generate the potential-based data by approximating trajectories X_t via the Euler-Maruyama method (Kloeden and Platen, 1992, § 9.2). In our experiments, we consider examples of convex, i.e., the quadratic potential $\Psi(x) = \|x\|_2^2$, and nonconvex potentials, i.e., Styblinski-Tang flow $\Psi(x) = \|3x^3 - 32x + 5\|_2^2$.

C.2. Trajectory-Based Dynamics

Besides population dynamics evolving according to a potential Ψ, we consider population dynamics following trajectories in space. To achieve this, we generate data by moving a 2-dimensional Gaussian distribution along a pre-defined trajectory. One example considered in the experiments above is a population moving along a semicircle ($T = 5$). Other trajectories and configurations are possible.

D. Experimental Details

D.1. Baselines

We compare JKONET with *explicit* integration schemes (forward methods) such as Hashimoto et al. (2016). In our proximal method, the prediction of the population ρ_t at the next time step $t + 1$ is parameterized via a separate function (ψ_θ (6)) and is thus decoupled from the free energy functional J_ξ driving the underlying dynamics. When learning *explicit* methods, however, the prediction is based on the gradient of an energy functional F_ξ. Given a distribution ρ_t at time t and energy F_ξ, the population particles at time $t + 1$ are thus predicted via

$$\rho_{t+1} := (\nabla F_\xi)_\# \rho_t.$$

While this energy is still estimated by minimizing a Sinkhorn loss as in (9), this explicit approach can fall more easily in local minima and overfit to seen population trajectories during the training phase.

To demonstrate this behavior, we design a simple experiment of a synthetic population, undergoing a translational shift. During training, the particles move in the interval $[-10, -2.5]$ ($T = 3$). During test time, however, we shift the interval by 5 units. In order to compare JKONET with *explicit* methods such as Hashimoto et al. (2016), we parameterize F_ξ and E_ξ (8) using an identical neural network architecture. We report the validation and test loss average over three independent runs (Table 1).

D.2. Hyperparameters and Training

For all experiments, we use a batch size of 250. For training the ICNN ψ_θ, we use the Adam optimizer (Kingma and Ba, 2014) with learning rate $\text{lr}_\theta = 0.01$ ($\beta_1 = 0.5$, $\beta_2 = 0.9$). The fixed-point loop runs for minimally 50 and maximally 100 iterations with $\alpha = 1$. We again use the Adam optimizer for learning the energy functional J_ξ with learning rate ranging from $\text{lr}_\xi = 0.001$ to 0.0001 ($\beta_1 = 0.5$, $\beta_2 = 0.9$). In our experiments, we use a constant JKO step size $\tau = 1.0$. For all experiments, we use $\varepsilon = 0.1$ for the Sinkhorn loss (9).