Entropic Neural Optimal Transport

Evgeny Burnaev
Prof., Head of Skoltech Applied AI Center
Head of AIRI Research Group

Entropic Neural Optimal Transport

https://arxiv.org/abs/2211.01156

Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry Vetrov, Evgeny Burnaev

We propose a novel neural algorithm for the fundamental problem of computing the entropic optimal transport (EOT) plan between probability distributions which are accessible by samples.

Overview

Introduction to optimal transport and Schrödinger Bridge problems

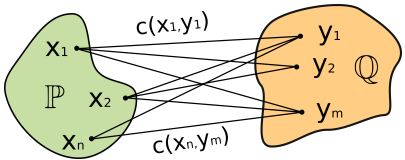
Entropic Neural Optimal Transport via Diffusion Processes

Experiments

Introduction to optimal transport and Schrödinger Bridge problems

Kantorovich's Formulation of Optimal Transport[1]

Let $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a cost function, e.g., $c(x, y) = \frac{\|x-y\|^2}{2}$.



The optimal transport **cost** between measures $\mathbb{P}$ and $\mathbb{Q}$ is

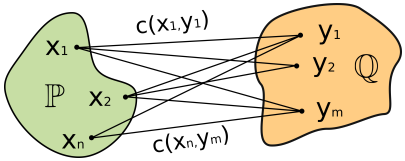
$$\text{Cost}(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\pi(x, y).$$

The plan π^* attaining the minimum is called
the optimal **transport plan** between $\mathbb{P}$ and $\mathbb{Q}$.

[1] Cédric Villani (2008). *Optimal transport: old and new.* Vol. 338. Springer Science & Business Media.

Entropy-regularized Optimal Transport

Let $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a cost function, e.g., $c(x, y) = \frac{\|x-y\|^2}{2}$.



The entropic optimal transport **cost** between measures $\mathbb{P}$ and $\mathbb{Q}$ is

$$\text{Cost}(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\pi(x, y) - \epsilon H(\pi),$$

where $H(\pi) = -\int_{\mathcal{X} \times \mathcal{Y}} \log \frac{d\pi(x,y)}{d[x,y]} d\pi(x, y)$ and ϵ is a regularization strength.

The plan π^* attaining the minimum is called
the optimal **transport plan** between $\mathbb{P}$ and $\mathbb{Q}$.

NOT for entropy-regularized Optimal Transport

Entropy-regularized OT is equivalent to the weak OT with the weak cost:

$$C(x, \pi(\cdot|x)) = \int_{\mathcal{Y}} c(x, y) d\pi(y|x) - H(\pi(y|x)).$$

NOT objective for the weak cost:

$$\sup_{\omega} \inf_{\theta} \mathcal{L}(\omega, \theta) = \sup_{\omega} \inf_{\theta} \left[\int_{\mathcal{Y}} f_\omega(y) d\mathbb{Q}(y) + \int_{\mathcal{X}} \left(C\big(x, T_\theta(x, \cdot)_{\#}\mathbb{S}\big) - \int_{\mathcal{Z}} f_\omega\big(T_\theta(x, z)\big) d\mathbb{S}(z) \right) d\mathbb{P}(x) \right].$$

But we can not optimize such objective because
there is no simple way to estimate $\pi(y|x)$ by samples.

Dynamic Schrödinger Bridge

We denote $\mathbb{P}_0 = \mathbb{P}$ and $\mathbb{P}_1 = \mathbb{Q}$. We consider stochastic processes T_f, which is described by the following SDE:

$$dX_t = f(X_t, t)dt + \sqrt{\epsilon}dW_t$$

Dynamic Schrödinger Bridge for this family of processes is formulated as follows:

$$\inf_{T_f \in \mathcal{D}(\mathbb{P}_0, \mathbb{P}_1)} \mathbb{E}_{T_f}[\int_0^1 \frac{1}{2\epsilon}||f(X_t, t)||^2 dt]$$

The process T_{f^*} attaining the minimum has joint distribution $\pi_{T_{f^*}}$ which is the solution to the Entropic OT with regularization parameter ϵ.[2]

[2] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon (2021). "Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrodinger bridge". In: *SIAM Review* 63.2, pp. 249–313.

Entropic Neural Optimal Transport via Diffusion Processes

Reformulation of the Dynamic Schrödinger Bridge

Theorem (Minimax reformulation of the Dynamic Schrödinger Bridge)

$$\inf_{T_f \in \mathcal{D}(\mathbb{P}_0, \mathbb{P}_1)} \mathbb{E}_{T_f}\left[\int_0^1 \frac{1}{2\epsilon} ||f(X_t, t)||^2 dt\right] = \sup_{\beta} \inf_{T_f \in \mathcal{D}(\mathbb{P}_0)} \widetilde{\mathcal{L}}(\beta, T_f),$$

where the functional $\widetilde{\mathcal{L}}$ *is defined by*

$$\widetilde{\mathcal{L}}(\beta, T_f) := \mathbb{E}_{T_f}\left[\int_0^1 \frac{1}{2\epsilon} ||f(X_t, t)||^2 dt\right] + \int_{\mathcal{Y}} \beta(y) d\mathbb{P}_1(y) - \int_{\mathcal{Y}} \beta(y) d\mathbb{P}_1^{T_f}(y)$$

and $d\mathbb{P}_1^{T_f}(y)$ *is a marginal distribution of* T_f *at* $t = 1$.

The Algorithm: Preliminaries

$$\widetilde{\mathcal{L}}(\beta, T_f) := \mathbb{E}_{T_f}[\int_0^1 \frac{1}{2\epsilon}||f(X_t, t)||^2 dt] + \int_{\mathcal{Y}} \beta(y) d\mathbb{P}_1(y) - \int_{\mathcal{Y}} \beta(y) d\mathbb{P}_1^{T_f}(y)$$

We use ResNet[3] $\beta_\omega : \mathbb{R}^{3\times W\times H} \to \mathbb{R}$;

We use UNet $f_\theta(X, t) : \mathbb{R}^{3\times H\times W} \times [0, 1] \to \mathbb{R}^{3\times W\times H}$.

To condition on the time variable t we use positional encoding as in the transformer based models.[4]

For sampling from SDE we use Euler-Maruyama algorithm.

We solve the saddle point problem with the **stochastic gradient ascent-descent** by using random batches from $\mathbb{P}_0, \mathbb{P}_1$.

[3] Huidong Liu, Xianfeng Gu, and Dimitris Samaras (2019). "Wasserstein gan with quadratic transport cost". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4832–4841.

[4] Ashish Vaswani et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

The Algorithm: Entropic Neural Optimal Transport (ENOT)

Algorithm 1: Entropic Neural OT (ENOT)

repeat

Sample batches $X_0 \sim \mathbb{P}_0$, $Y \sim \mathbb{P}_1$;

$\{X_n\}_{n=1}^N, \{f_n\}_{n=1}^N \leftarrow \mathit{SDESolve}(X_0, f_\theta, N, \epsilon)$;

$\mathcal{L}_\beta \leftarrow \frac{1}{|X_N|} \sum_{x_N \in X_N} \beta_\phi(x_N) - \frac{1}{|Y|} \sum_{y \in Y} \beta_\phi(y)$;

Update ϕ by using $\frac{\partial L_\beta}{\partial \phi}$;

for $k = 1$ **to** K **do**

Sample batches $X_0 \sim \mathbb{P}_0$, $Y \sim \mathbb{P}_1$;

$\{X_n\}_{n=1}^N, \{f_n\}_{n=0}^{N-1} \leftarrow \mathit{SDESolve}(X_0, f_\theta, N, \epsilon)$;

$\mathcal{L}_f \leftarrow \frac{1}{N|X_0|} \sum_{t=0}^{N-1} \sum_{f_{t,k} \in f_t} ||f_{t,k}||^2 - \frac{1}{|X_N|} \sum_{x_N \in X_N} \beta_\phi(x_N)$;

Update θ by using $\frac{\partial L_f}{\partial \theta}$;

until *converged*;

The Algorithm: SDESolve function

Algorithm 2: Euler-Maruyama algorithm

Input : batch of initial states X_0 at time moment $t = 0$;
SDE drift network $f_\theta : {}^D \times [0, 1] \to {}^D$;
number of steps for the SDE solver $N \geq 1$;
noise variance $\epsilon \geq 0$.

Output: batches $\{X_n\}_{n=1}^N$ of intermediate states at $t = \frac{n}{N}$ simulating the proccess $dX_t = f(X_t, t)dt + \sqrt{\epsilon}dW_t$;
batches $\{f_n\}_{n=0}^{N-1}$ of drift values $f(X_n, t_n)$ at $t = \frac{n-1}{N}$ simulating the process;

$\Delta t \leftarrow \frac{1}{N}$;

for $t = 1, 2, \ldots, N$ **do**

  for $i = 1, 2, \ldots, |X_0|$ **do**

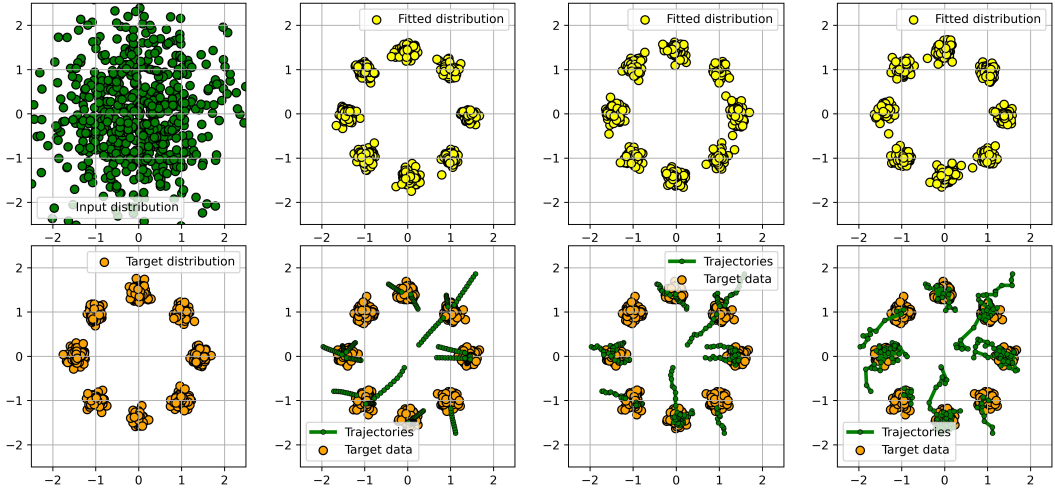
    Sample noise W from $\mathcal{N}(0, I)$;

    $f_{t-1,i} \leftarrow f(X_{t-1}, t-1)$;

    $X_{t,i} \leftarrow X_{t-1,i} + f_{t-1,i}\Delta t + \sqrt{\epsilon \Delta t}W$;

Experiments

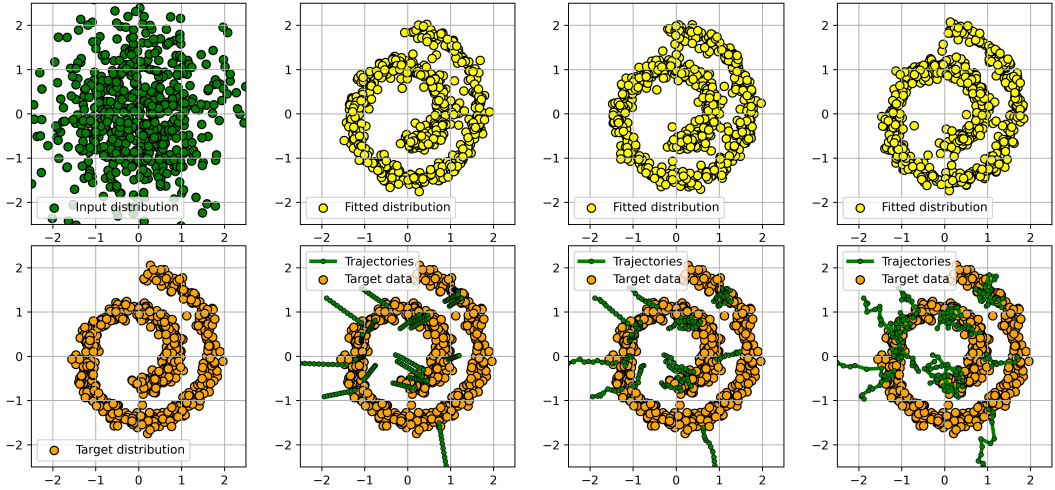
Toy example I



(a) $x \sim \mathbb{P}_0$, $y \sim \mathbb{P}_1$ **(b)** ENOT (ours), $\epsilon = 0$ **(c)** ENOT (ours), $\epsilon = 0.01$ **(d)** ENOT (ours), $\epsilon = 0.1$

Figure 1: *Gaussian* $\rightarrow$ *Mixture of 8 Gaussians.*, learned stochastic process with ENOT (ours).

Toy example II

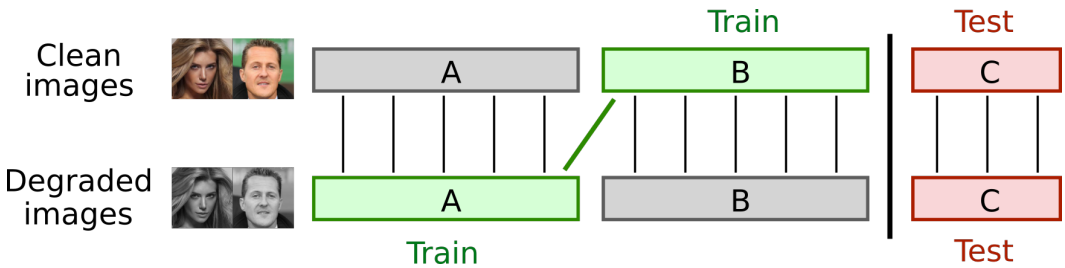


(a) $x \sim \mathbb{P}_0$, $y \sim \mathbb{P}_1$ **(b)** ENOT (ours), $\epsilon = 0$ **(c)** ENOT (ours), $\epsilon = 0.01$ **(d)** ENOT (ours), $\epsilon = 0.1$

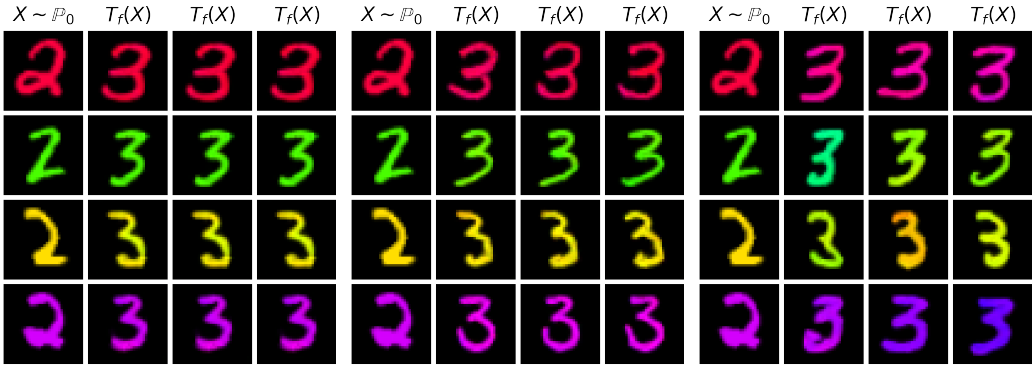
Figure 2: Gaussian $\rightarrow$ Swiss roll, learned stochastic process with ENOT (ours).

Unpaired image experiments setup

We do the *unpaired* train-test split as follows:



Colored MNIST I



(a) ENOT (ours) samples, $\epsilon = 0$. **(b)** ENOT (ours) samples, $\epsilon = 1$. **(c)** ENOT (ours) samples, $\epsilon = 10$.

Figure 3: Samples of colored MNIST obtained by ENOT (ours) for different ϵ.

Colored MNIST II

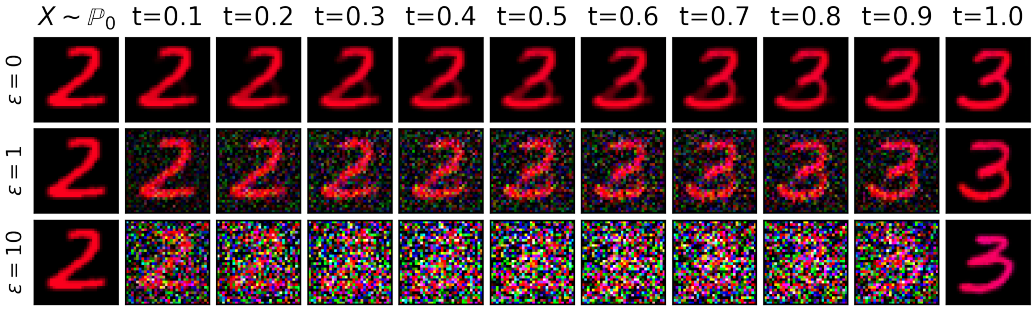
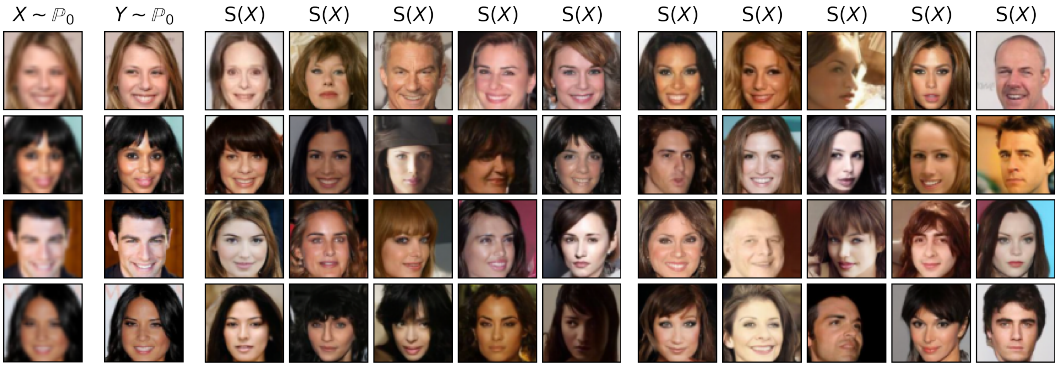


Figure 4: Trajectories from our learned ENOT for colored MNIST for different ϵ.

Celeba. Existing competitive algorithm - SCONES



(a) **(b)** **(c)** SCONES, $\epsilon = 100$ **(d)** SCONES, $\epsilon = 1250$

Figure 5: Faces produced by SCONEs for various ϵ.
Figure 5a shows test degraded images, 5b – their original high-resolution counterparts.

Competitive algorithm (SCONES) does not work for the small ϵ .

Celeba. Our algorithm I

(a) **(b)** **(c)** ENOT (ours), $\epsilon = 0$ **(d)** ENOT (ours), $\epsilon = 1$ **(e)** ENOT (ours), $\epsilon = 10$

Figure 6: Faces produced by ENOT (ours) for various ϵ.
Figure 6a shows test degraded images, 6b – their original high-resolution counterparts.

Our algorithm (ENOT) does work for small ϵ ϵ .

Celeba. Our algorithm II

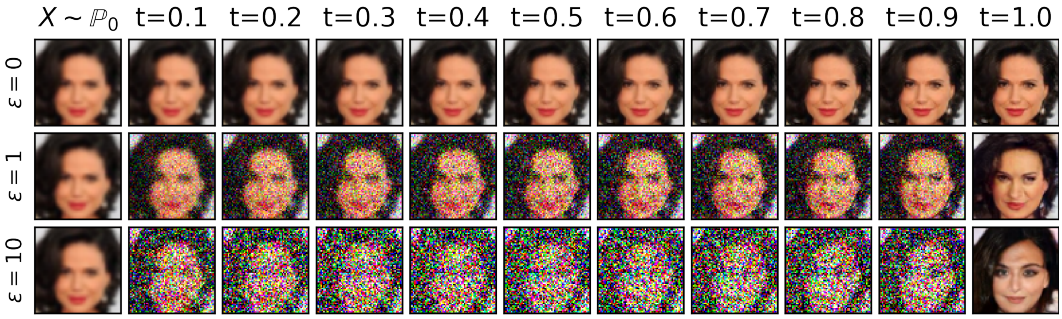


Figure 7: Trajectories of samples learned by our Algorithm 1 for Celeba with $\epsilon = 0, 1, 10$.

References i

- Chen, Yongxin, Tryphon T Georgiou, and Michele Pavon (2021). "Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrodinger bridge". In: *SIAM Review* 63.2, pp. 249–313.
- Liu, Huidong, Xianfeng Gu, and Dimitris Samaras (2019). "Wasserstein gan with quadratic transport cost". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4832–4841.
- Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.
- Villani, Cédric (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.