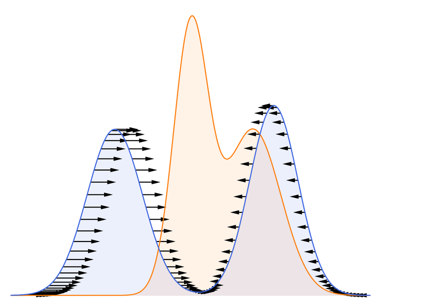
On Scalable and Efficient Computation of Large Scale Optimal Transport

Yujia Xie, Minshuo Chen, Haoming Jiang, Tuo Zhao, Hongyuan Zha

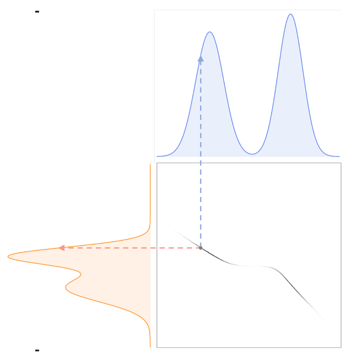
Georgia Institute of Technology

Introduction

- **Optimal Transport (OT)** in continuous setting:



The goal of optimal transport: move the mass from one distribution to another with minimum cost.



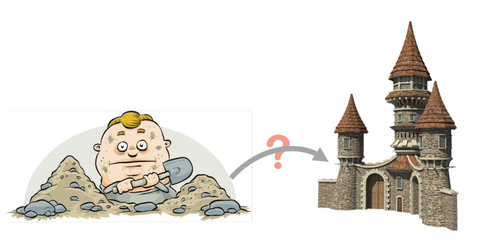
However, the direct mapping from one support to another is not always feasible. Therefore, people turn to compute the best joint distribution.

Mathematically, optimal transport seeks to solve

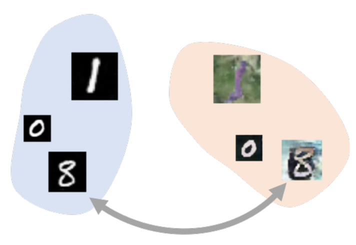
$$\gamma^* = \underset{\gamma\in\Pi(\mu,\nu)}{\operatorname{argmin}}\ \mathbb{E}_{(X,Y)\sim\gamma}[c(X,Y)], \qquad (1)$$

- μ,ν: two input distributions;
- $\Pi(\mu,\nu)$: requires the marginals of γ to be μ and ν;
- $c(\cdot,\cdot)$: the cost function;
- γ^*: the **optimal transport plan**, suggesting the way to transport between μ and ν with minimum cost.

- **Applications** of optimal transport:



Resource Allocation

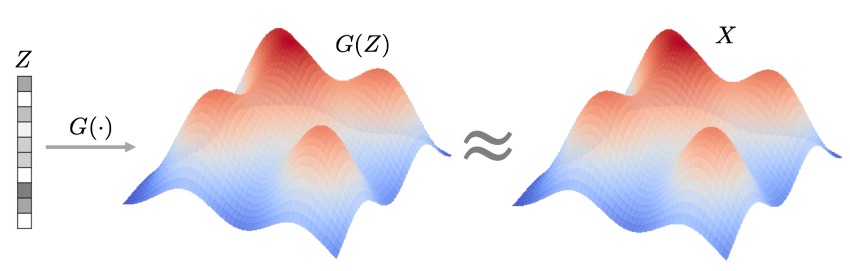


Domain Adaptation

- **The Difficulty** of solving optimal transport:
  - Infinite dimensional optimization problem;
  - If use discretization on the support, the number of grids needs to scale exponentially w.r.t. dimension.

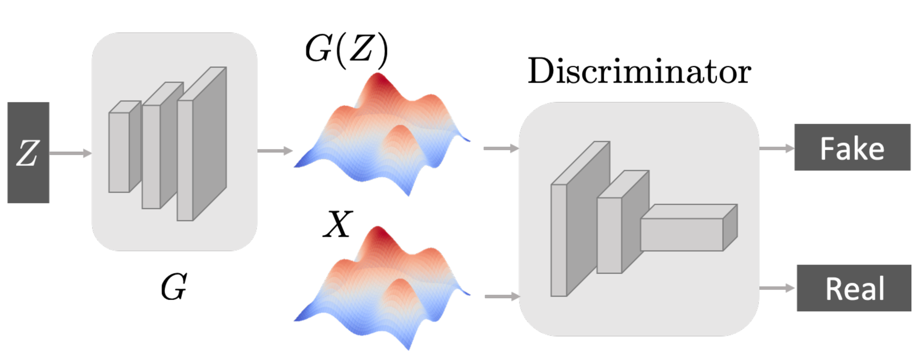
Background - Implicit Generative Learning

Implicit Generative Model: Given a latent variable Z, train a mapping $G(\cdot)$ so that $G(Z)$ and X, the random variable of interest, have the same distribution.



Several methods are of this kind:

- **Generative adversarial networks (GAN)**:



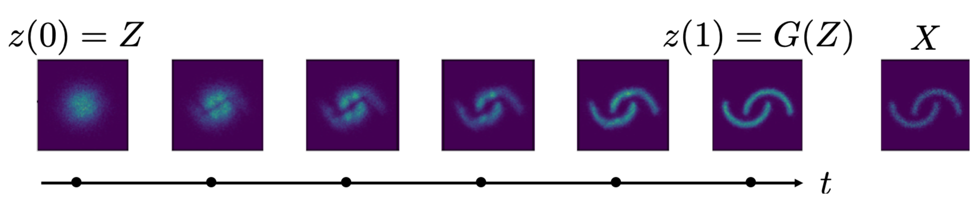
– Generator G wants to fool the discriminator;

– Discriminator wants to distinguish $G(Z)$ from the real data.

- **Neural ordinary differential equation (Neural ODE)** uses an ODE to characterize how the input latent variable Z evolves towards the output $G(Z)$ in continuous time,

$$dz/dt = \xi(z(t),t),$$

where ξ is a neural network (Chen et al., 2018).



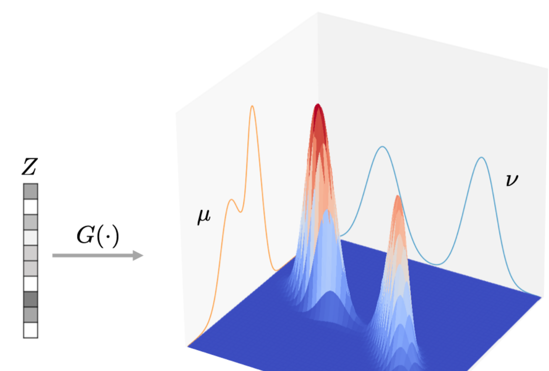
- Variational auto-encoder (VAE)
- Non-linear independent components estimation (NICE)
- ...

Scalable Pushforward based OT (SPOT)

- Approximate γ^* by an implicit generative model $G(Z)$, i.e., we seek to train

$$G(Z) = \begin{bmatrix} G_X(Z) \\ G_Y(Z)\end{bmatrix} \approx \begin{bmatrix} X \\ Y\end{bmatrix},$$

where $\quad Z\sim\rho, X\sim\mu, Y\sim\nu.$



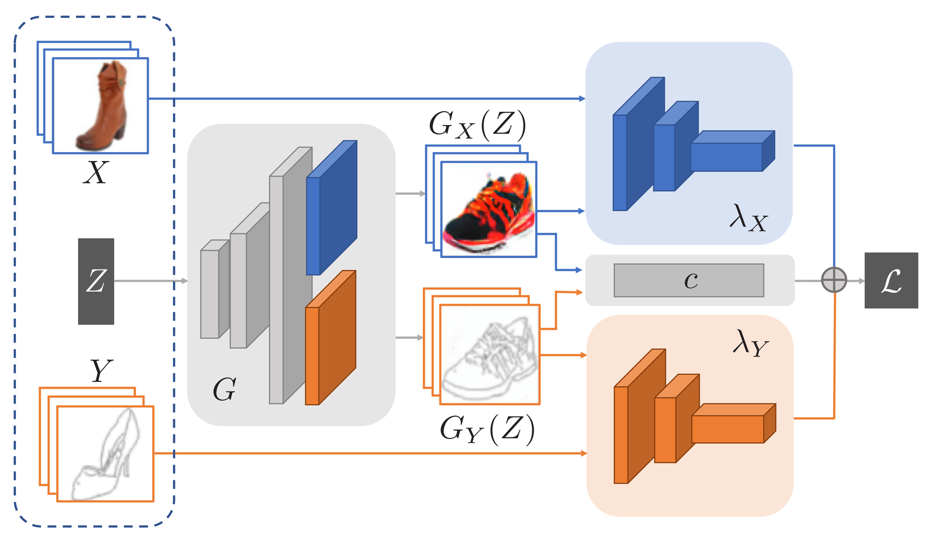
Substituting $\gamma = G(Z)$ into (1), we have

$$G^* = \underset{G}{\operatorname{argmin}} \quad \mathbb{E}_{Z\sim\rho}[c(G_X(Z),G_Y(Z))].$$
$$\text{subject to} \quad G_X(Z)\sim\mu,\ G_Y(Z)\sim\nu$$

Motivated by Wasserstein GAN, we cast the above formulation as a minimax problem:

$$\min_{G\in\mathcal{G}} \max_{\lambda_X\in\mathcal{F}_X^1,\lambda_Y\in\mathcal{F}_Y^1} \mathbb{E}_{Z\sim\rho}[c(G_X(Z),G_Y(Z))]$$
$$+\eta(\mathbb{E}_{Z\sim\rho}[\lambda_X(G_X(Z))]-\mathbb{E}_{X\sim\mu}[\lambda_X(X)]$$
$$+\mathbb{E}_{Z\sim\rho}[\lambda_Y(G_Y(Z))]-\mathbb{E}_{Y\sim\nu}[\lambda_Y(Y)]).$$

Here, λ_X and λ_Y are two discriminators (Arjovsky, M., 2017) encouraging $G_X(Z)\sim\mu$, $G_Y(Z)\sim\nu$.



An illustration of SPOT framework

- Our proposed framework has three major advantages:
  - Easily scales to very large OT problems by primal dual stochastic gradient-type algorithms;
  - Effectively adapts to data with intrinsic low dimensional structures;
  - Allows efficient sampling from the transport plans.

SPOT for Density Recovery

- Goal: Recover p_γ, the density of the transport plan.
- Method: Equip SPOT with Neural ODE.

Consider variable $z(t)$,

$$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t)\end{bmatrix} \quad \text{with} \quad \begin{matrix} z(0) = Z \\ z_1(1) = G_X(Z), z_2(1) = G_Y(Z)\end{matrix}$$

The dynamic of $z(t)$ is

$$dz_1/dt = \xi_1(z(t),t), \quad dz_2/dt = \xi_2(z(t),t).$$

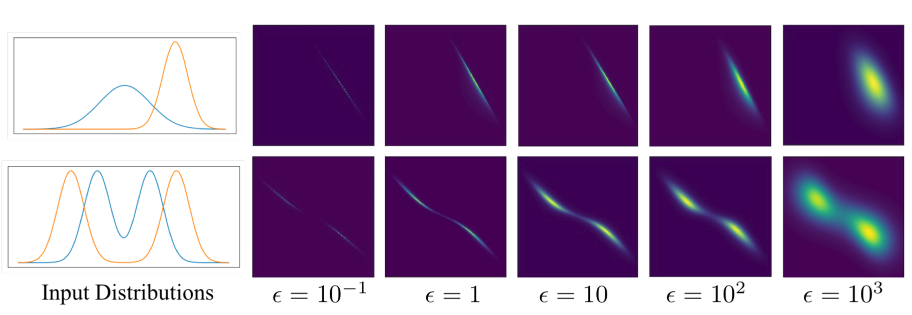
Proposition 1. *Under proper conditions, the log of joint density $p(t)$ satisfies the following ODE:*

$$\frac{\partial\log p(t)}{\partial t} = -\left(\operatorname{tr}\left(\frac{\partial\xi_1}{\partial z_1}\right)+\operatorname{tr}\left(\frac{\partial\xi_2}{\partial z_2}\right)\right),$$

where $\partial\xi_1/\partial z_1$ and $\partial\xi_2/\partial z_2$ denote the Jacobian matrices of ξ_1 and ξ_2, respectively.

- Experimental result: Density with entropy regularizer

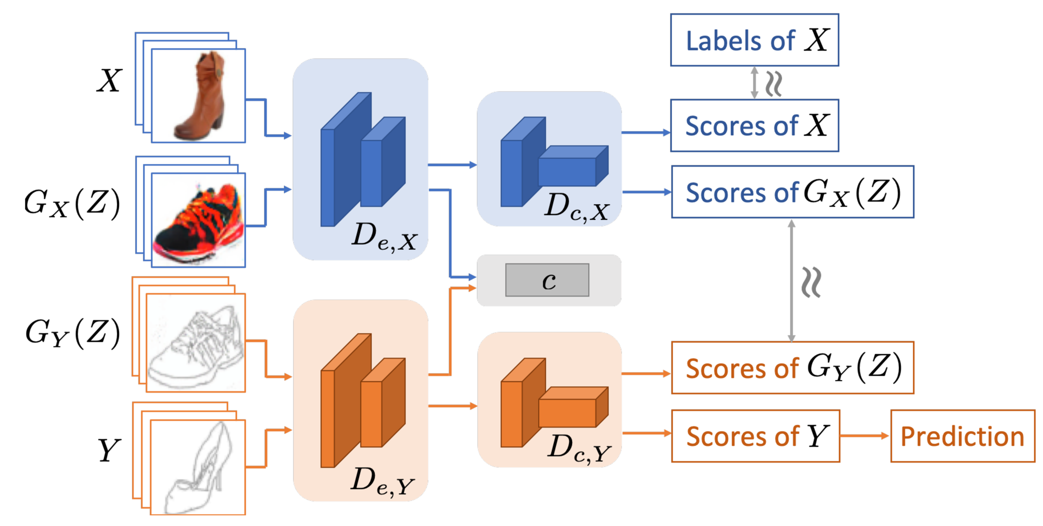
$$\epsilon\mathcal{H} = \epsilon\mathbb{E}_{G(Z)\sim\gamma}[\log p_\gamma(G(Z))].$$



SPOT for Domain Adaptation

- Setting: $\{x_i\}\sim\mu$ with known labels, $\{y_j\}\sim\nu$ with unknown labels. The goal is to predict the labels of $\{y_j\}$.

- Method: **DASPOT**



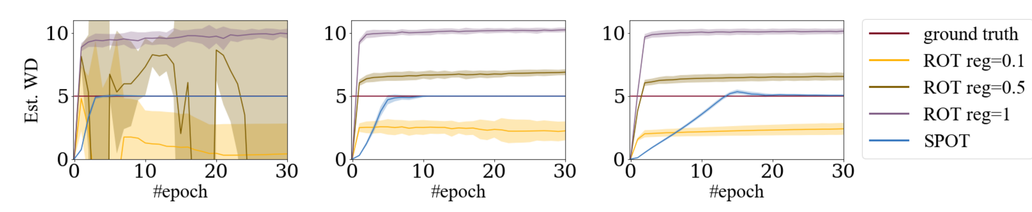
Neural networks $D_{e,X}$, $D_{c,X}$, $D_{e,Y}$, and $D_{c,Y}$ are jointly trained with G.

- Experimental results:

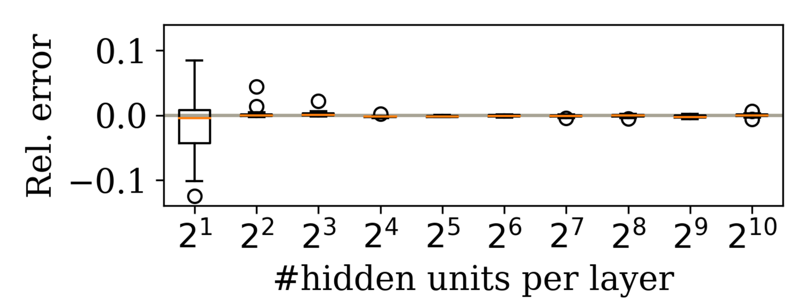
Source Target	MNIST USPS	USPS MNIST	SVHN MNIST	MNIST MNISTM
ROT	72.6%	60.5%	62.9%	–
StochJDOT	93.6%	90.5%	67.6%	66.7%
DeepJDOT	95.7%	96.4%	**96.7%**	92.4%
DASPOT	**97.5%**	**96.5%**	96.2%	**94.9%**

Experiment – Computing WD

- **Wasserstein Distance (WD)**: $\mathcal{W} = \mathbb{E}_{(X,Y)\sim\gamma^*}[c(X,Y)]$, i.e., the expected cost of optimal transport plan.

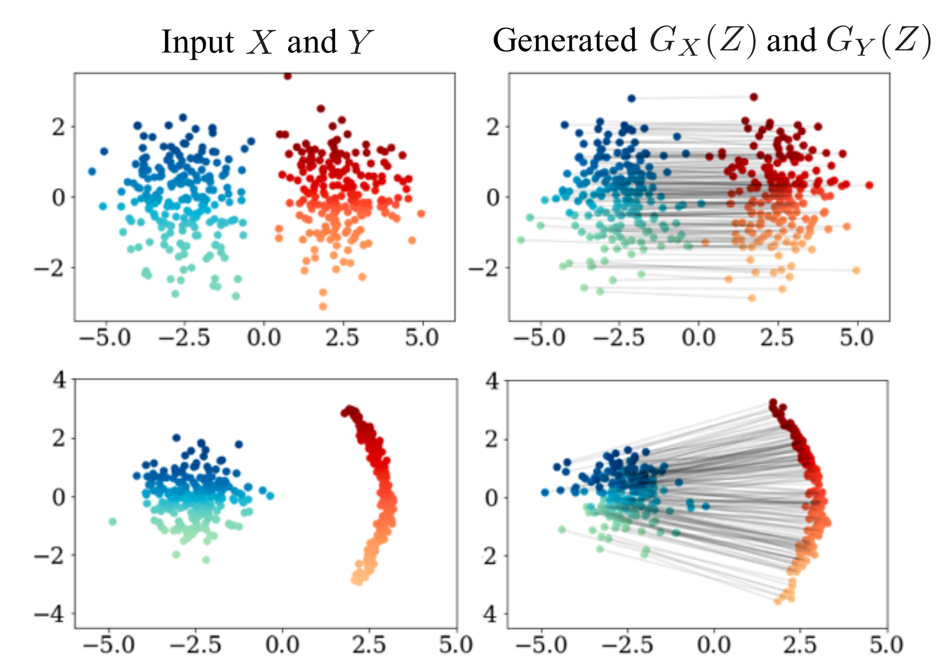


LR $=10^{-3}$ LR $=10^{-4}$ LR $=10^{-5}$



Experiment – Sample Generation

- Synthetic Datasets



- MNIST-MNISTM:

Real images ▷ Generated images ▽

CoGAN (Liu et al., 2016) SPOT

- Photos-Monet