Optimal Transport in Large-Scale Machine Learning Applications

Nhat Ho

The University of Texas, Austin

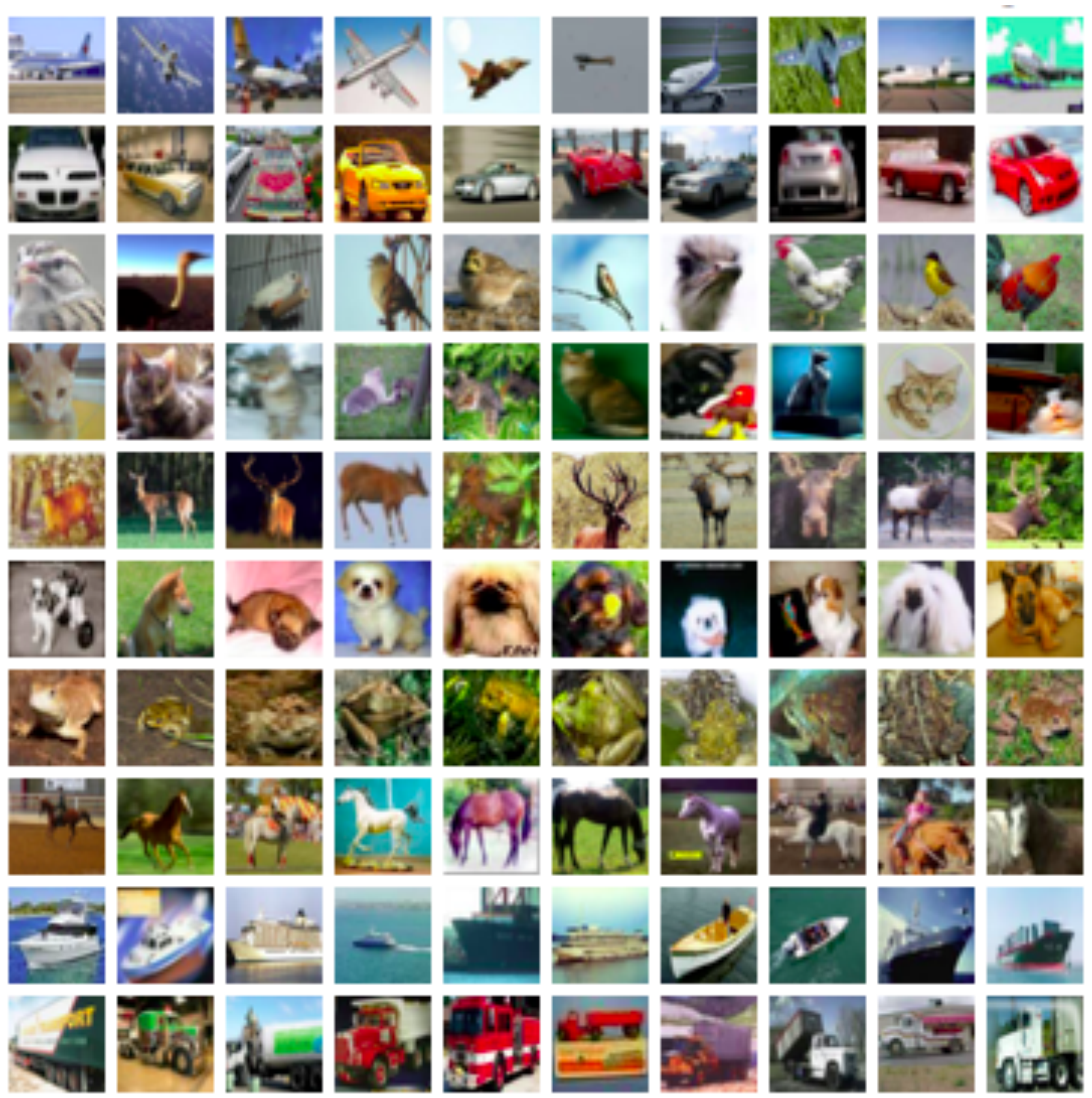
Talk Outline

- Applications/ Methods of Optimal Transport (OT): Brief Introduction
- Foundations of Optimal Transport
    - Monge's Optimal Transport Formulation
    - Kantorovich's Optimal Transport Formulation
    - Entropic Regularized Optimal Transport
- Application of Optimal Transport to Deep Generative Model
    - Wasserstein GAN
    - Issues of Wasserstein GAN and Solutions

Some Applications/ Methods of Optimal Transport (OT): Brief Introduction

OT's Method: Deep Generative Model

CIFAR 10



Speech

Goal: Given a set of data in high dimension (e.g., images, speeches, words, etc.), we would like to learn the underlying data distribution

OT's Method: Deep Generative Model

- OT is used as a loss between push-forward distribution from low-dimensional space and the empirical distribution from data

- Popular examples: Wasserstein GAN [1, 2], Wasserstein Autoencoder [3]

Image from Internet

OT's Method: Transfer Learning

Image from Internet

- **Domain Adaptation:** An important problem of designing autonomous vehicle is to make sure that the model we train in some particular weather/ environment/ time (source domains) will still perform well under other weathers/ environments/ time (target domains)

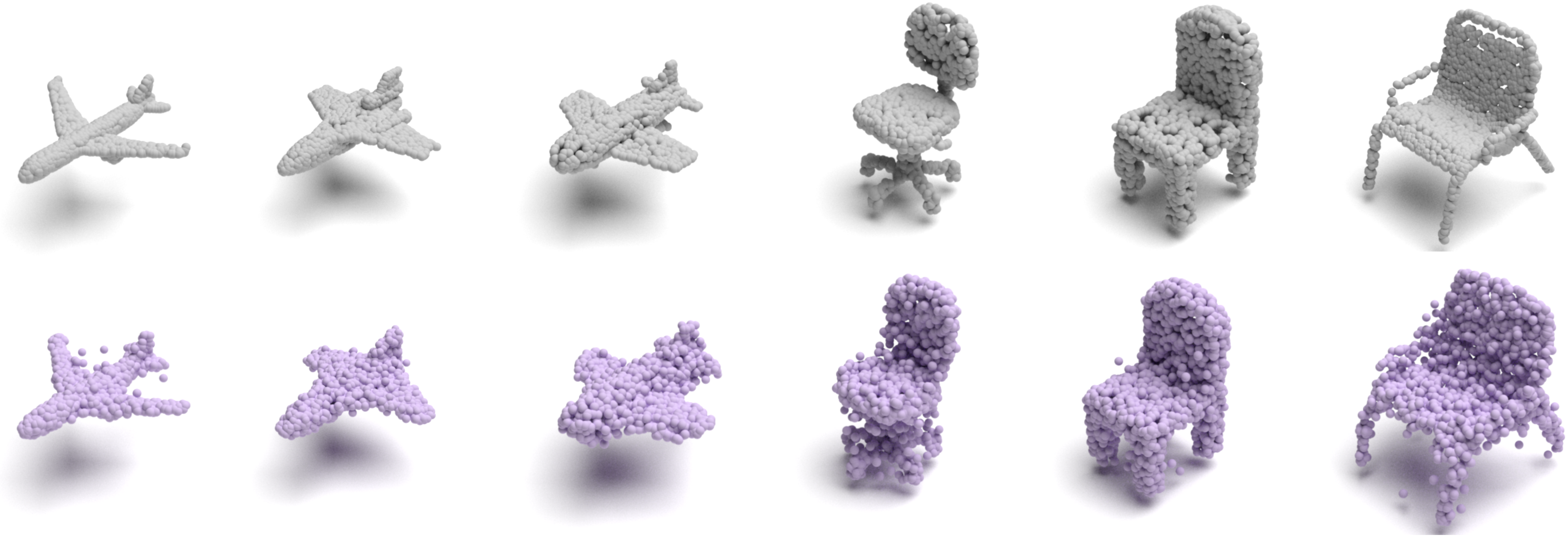
- Optimal transport is an efficient loss function capture the difference between these domains (e.g., [4] and [5])

OT’s Method: Transfer Learning

- **Domain Generalization:** An important example is that we would like to develop a face recognition system in new generation of Iphone (target domain) based on the previous Iphones (source domains) without the expensive cost of collecting new data for the new Iphone

- Optimal Transport also offers a great solution for this application

OT's method: 3D Objects' Representation



Above: Input 3D images

Below: Reconstruction of 3D images based on optimal transport [6]

[6] Trung Nguyen, Hieu Pham, Tam Le, Tung Pham, Nhat Ho, Son Hua. *Point-set distances for learning representations of 3D point clouds*. ICCV, 2021

OT's Method: (Multilevel) Clustering

- Each image contains several annotated regions, such as, those of animals, buildings, trees, etc.

- **Goal**: Based on the clustering behaviors of annotated regions from the images, we would like to learn the themes/ clusters of images

OT's Method: Multilevel Clustering

3 clusters of images based on using optimal transport (cf. [7], [8])

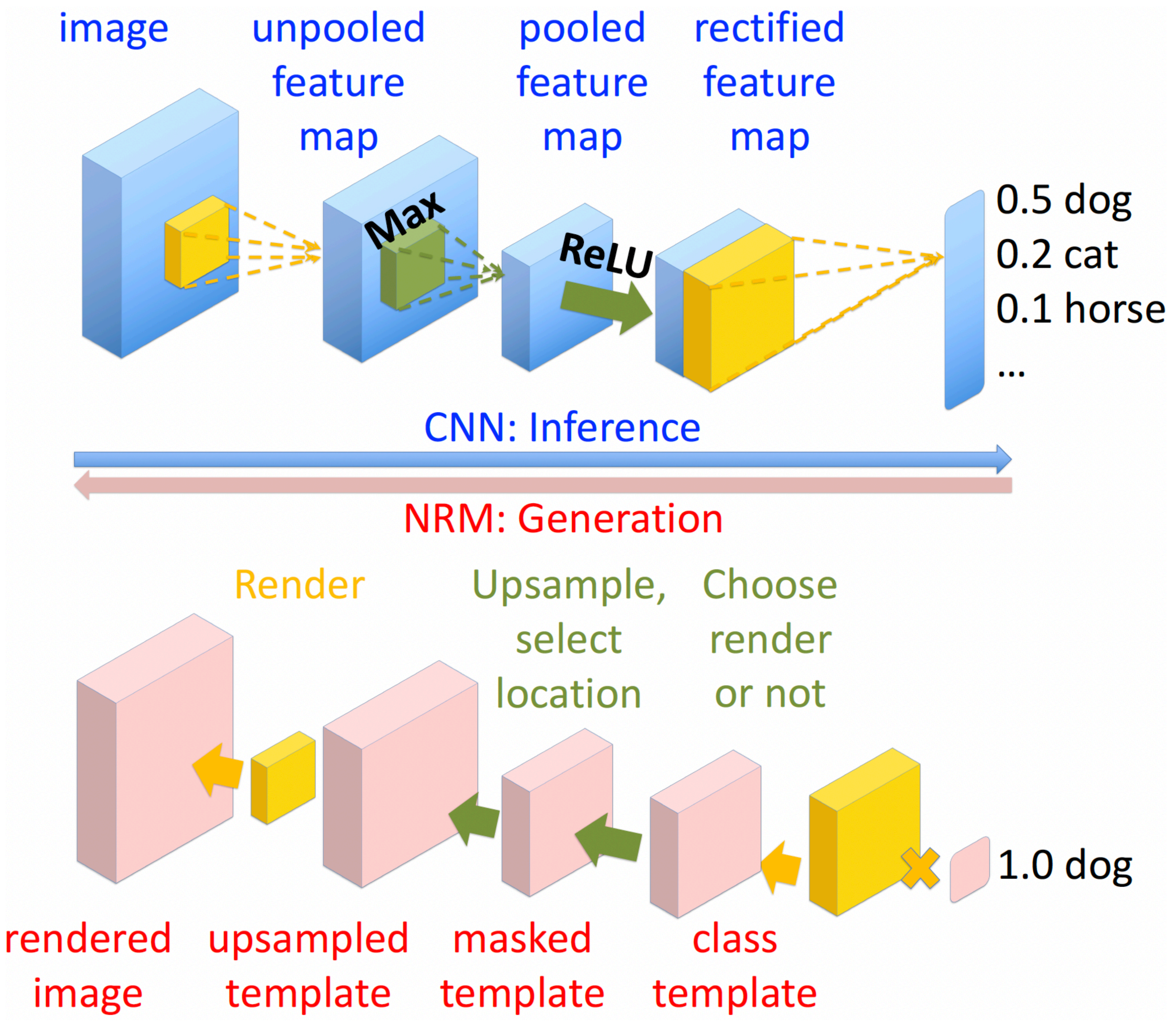
[7] Nhat Ho, Long Nguyen, Mikhail Yurochkin, Hung Bui, Viet Huynh, and Dinh Phung. *Multilevel clustering via Wasserstein means*. ICML, 2017

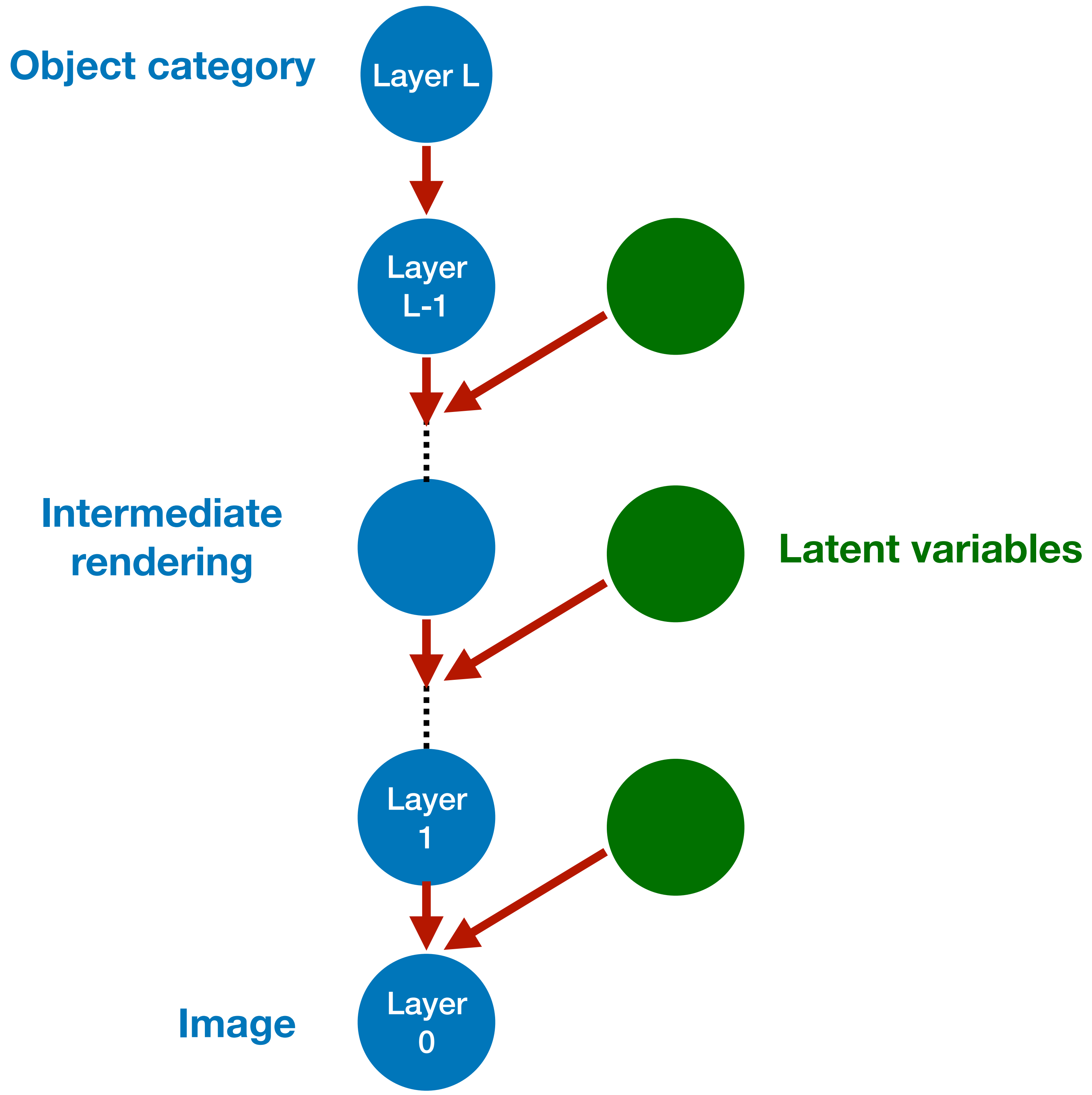
[8] Viet Huynh, Nhat Ho, Nhan Dam, Long Nguyen, Mikhail Yurochkin, Hung Bui, Dinh Phung. *On efficient multilevel clustering via Wasserstein distances*. Journal of Machine Learning Research (JMLR), 2021

OT's Method: Other Applications

- Optimal Transport is also a powerful tool for other important applications:
    - Forecasting Time Series (e.g., forecasting sales (Walmart), forecasting expenses (Amazon), etc.) [9]
    - Machine Translation [10]
    - Robust/ Reliable Machine Learning [11]
    - Fairness/ Responsible AI

OT is also useful as foundational theory tool





- Optimal transport can be used to understand the behaviors of latent variables associated with Relu, Maxpooling from Convolutional Neural Networks (CNNs) (cf. [12])

[12] Tan Nguyen, Nhat Ho, Ankit Patel, Anima Anandkumar, Michael I. Jordan, Richard Baraniuk. *A Bayesian Perspective of Convolutional Neural Networks through a Deconvolutional Generative Model.* Under Revision, Journal of Machine Learning Research (JMLR), 2022

OT is also useful as foundational theory tool

- A few other popular applications of OT for understanding machine learning methods and models include:
    - *Mixture models and hierarchical models*: Characterizing the convergence rates of estimating parameters, performing model selection, etc. (cf. [13], [14], [15])
    - *Distributional robust optimization*: Optimal Transport can be used to define a perturbed neighborhood of the true distribution (cf. [16], [17])
- **Some potential new research directions:** Optimal Transport can be useful to understand
    - (i) Self-training procedure in semi-supervised learning
    - (ii) Self-attention in Transformer
    - (iii) Contrastive Learning, Self-supervised Learning, etc.

Foundations of Optimal Transport

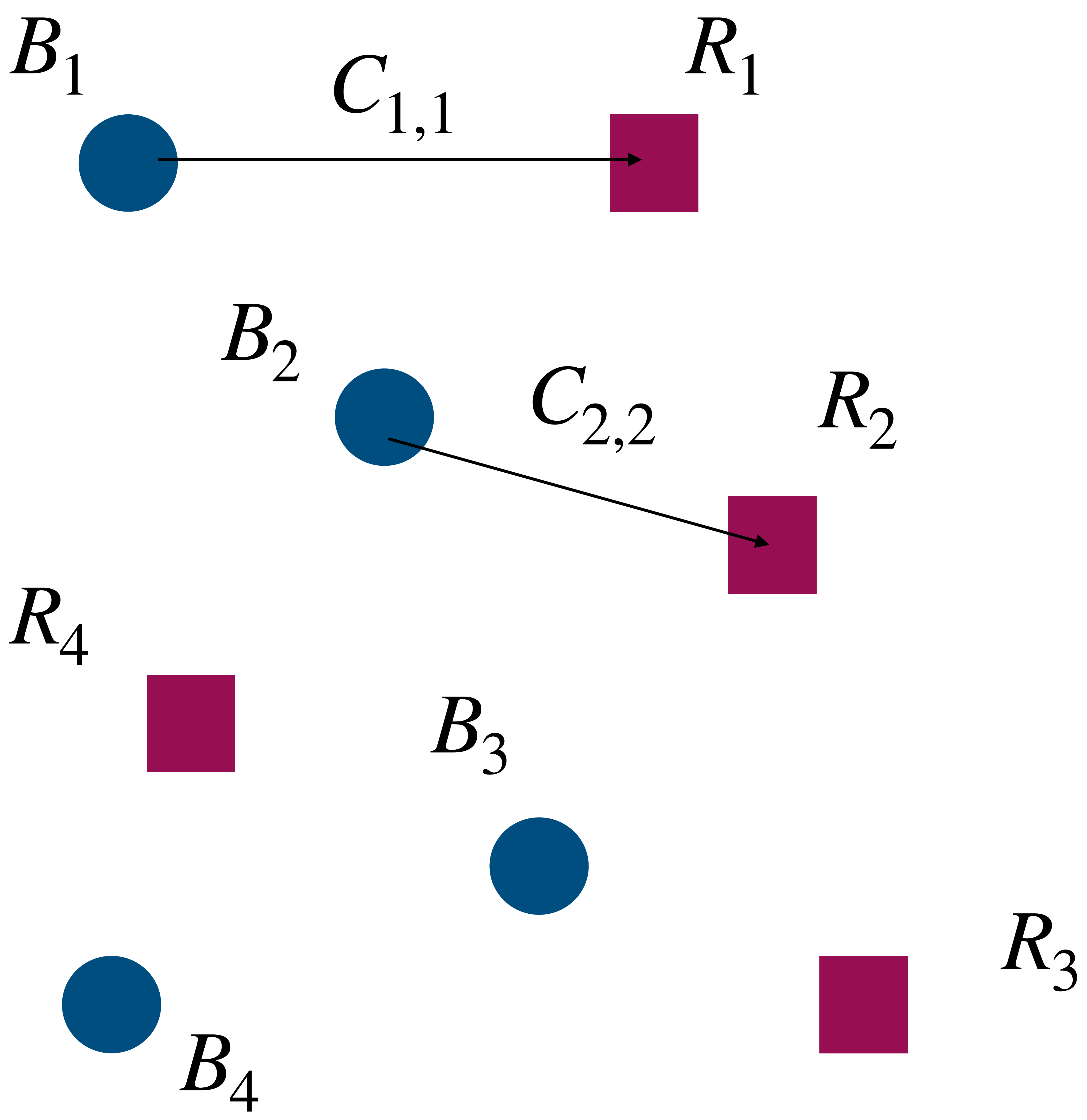
- Monge's Optimal Transport Formulation
- Kantorovich's Optimal Transport Formulation
- Entropic Regularized Optimal Transport

Monge’s OT Formulation: Motivation

- Optimal Transport was created by mathematician Gaspard Monge to find optimal ways to transport commodities and products under certain constraints

Image from Internet

Monge's OT Formulation: Motivation



- We start with a simple practical example of moving products from Bakeries (denoted by B) to Restaurants (denoted by R)
- Two bakeries will not transport the products to the same restaurant
- We denote by C_{ij} the distance between bakery B_i to restaurant R_j
- **Goal:** Find the shortest distance to move products from the bakeries to restaurants

Monge's OT Formulation

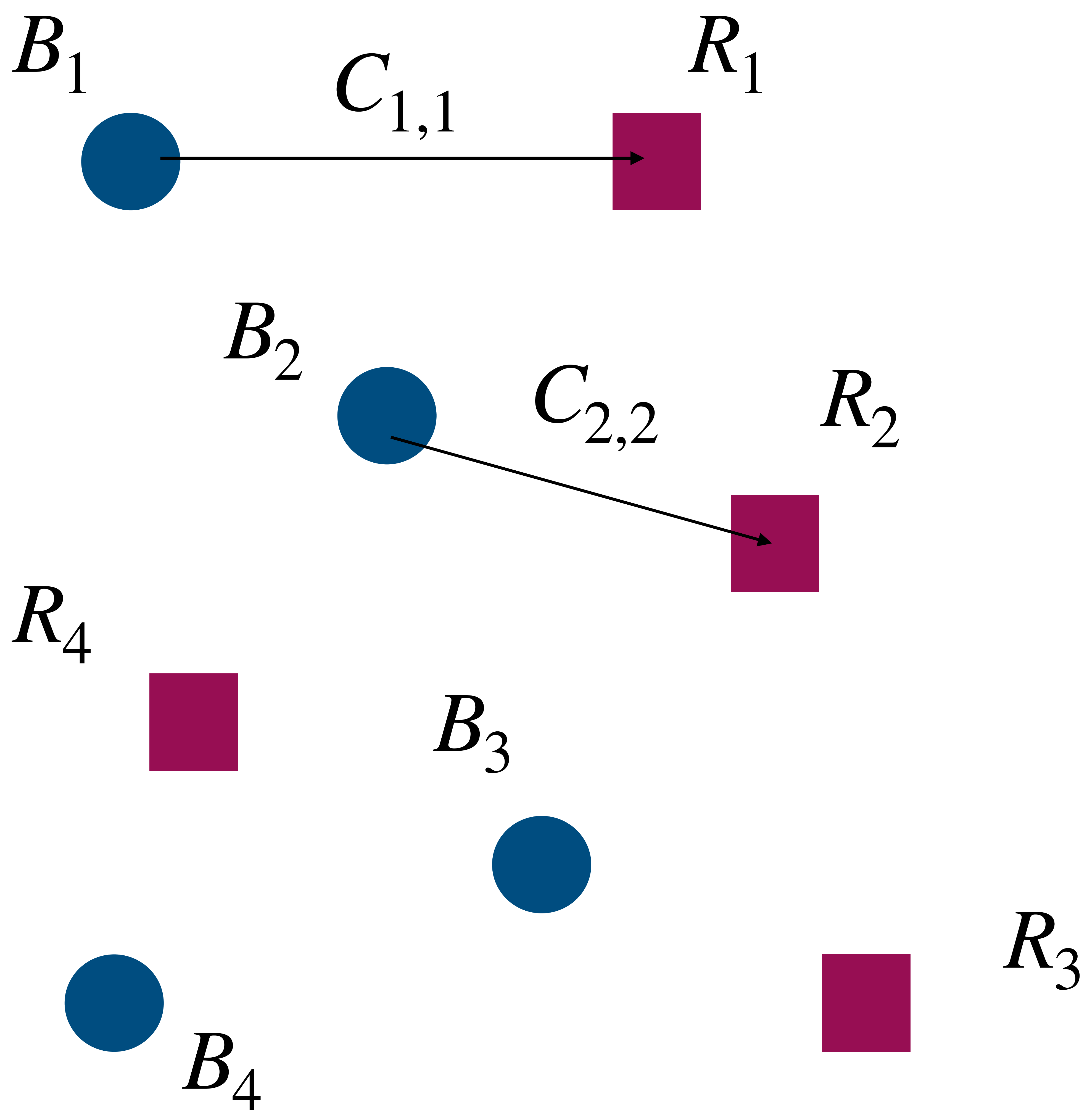
- *Monge's Optimal Transport* is:

$$\frac{1}{n}\min_{\sigma\in \mathrm{Per}_n}\sum_{i=1}^{n} C_{i,\sigma(i)}, \qquad \textbf{(1)}$$

where n: number of restaurants or bakeries

Per_n: the set of all permutations of $\{1,2,\ldots,n\}$

- Monge's formulation finds the optimal matching between the bakeries and restaurants



Monge's OT Formulation

- If we search for all the possible permutations in the optimization problem, the complexity of solving Monge's Optimal Transport is $\mathcal{O}(n!)$ (The total number of permutations of $\{1,2,\ldots,n\}$ is $n!$)
- By using Hungarian's algorithm for graph matching, we can obtain an improved complexity of $\mathcal{O}(n^3)$
- When we have $C_{ij} = |B_i - R_j|^2$, i.e., one dimensional setting, we can use quick sort algorithm to compute Monge's Optimal Transport in equation (1) with a complexity of $\mathcal{O}(n \log n)$

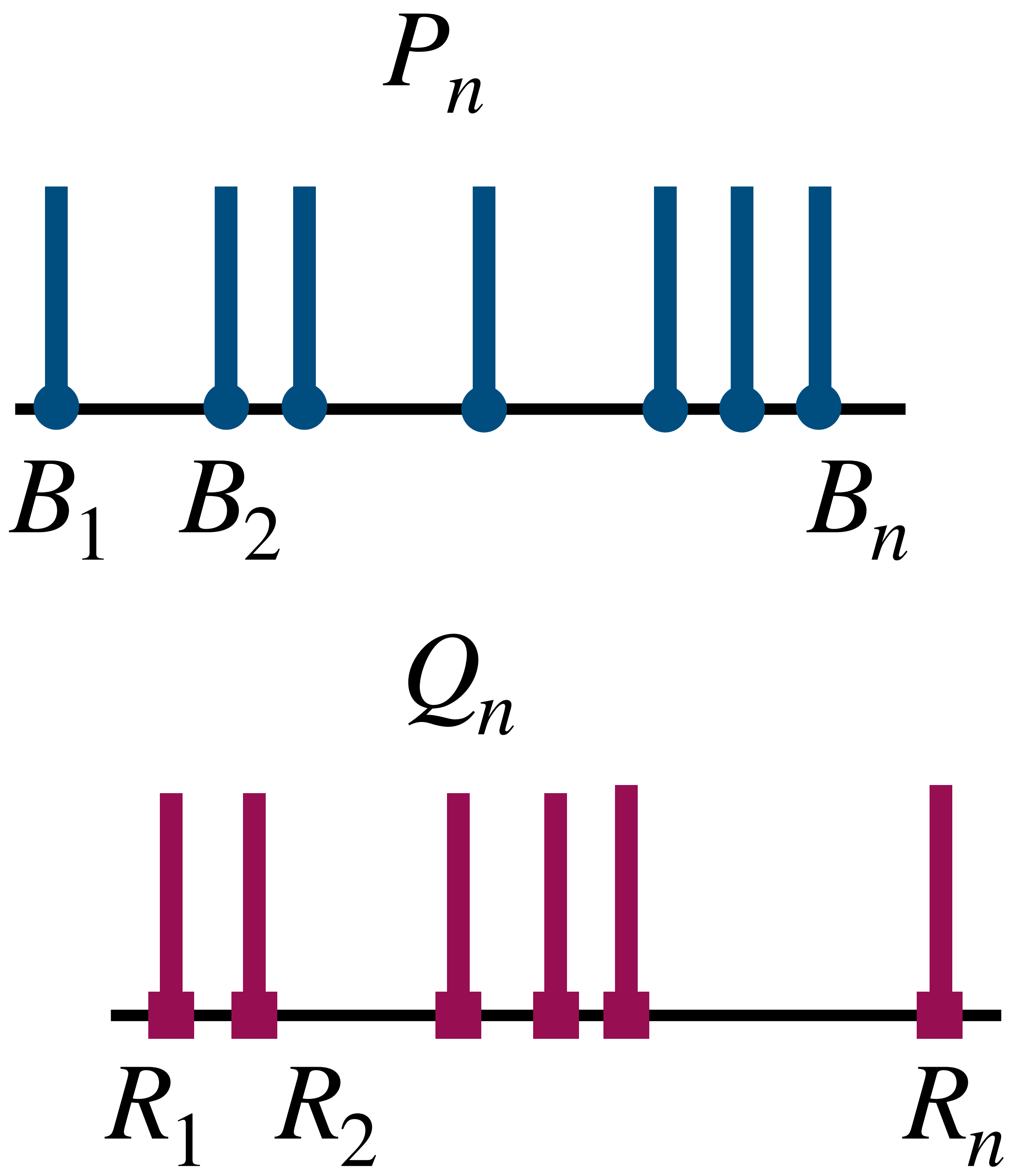
Monge's OT Formulation: Equivalent Form

- We define $P_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{B_i}$ and $Q_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{R_i}$ as corresponding empirical measures of bakeries and restaurants
- We denote $C_{ij} = \|B_i - R_j\|^2$ as the distance between B_i and R_j
- The Monge's formulation in equation (1) can be rewritten as

$$\inf_T \int \|x - T(x)\|^2 dP_n(x),$$

where the mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ in the infimum is such that $T\sharp P_n = Q_n$

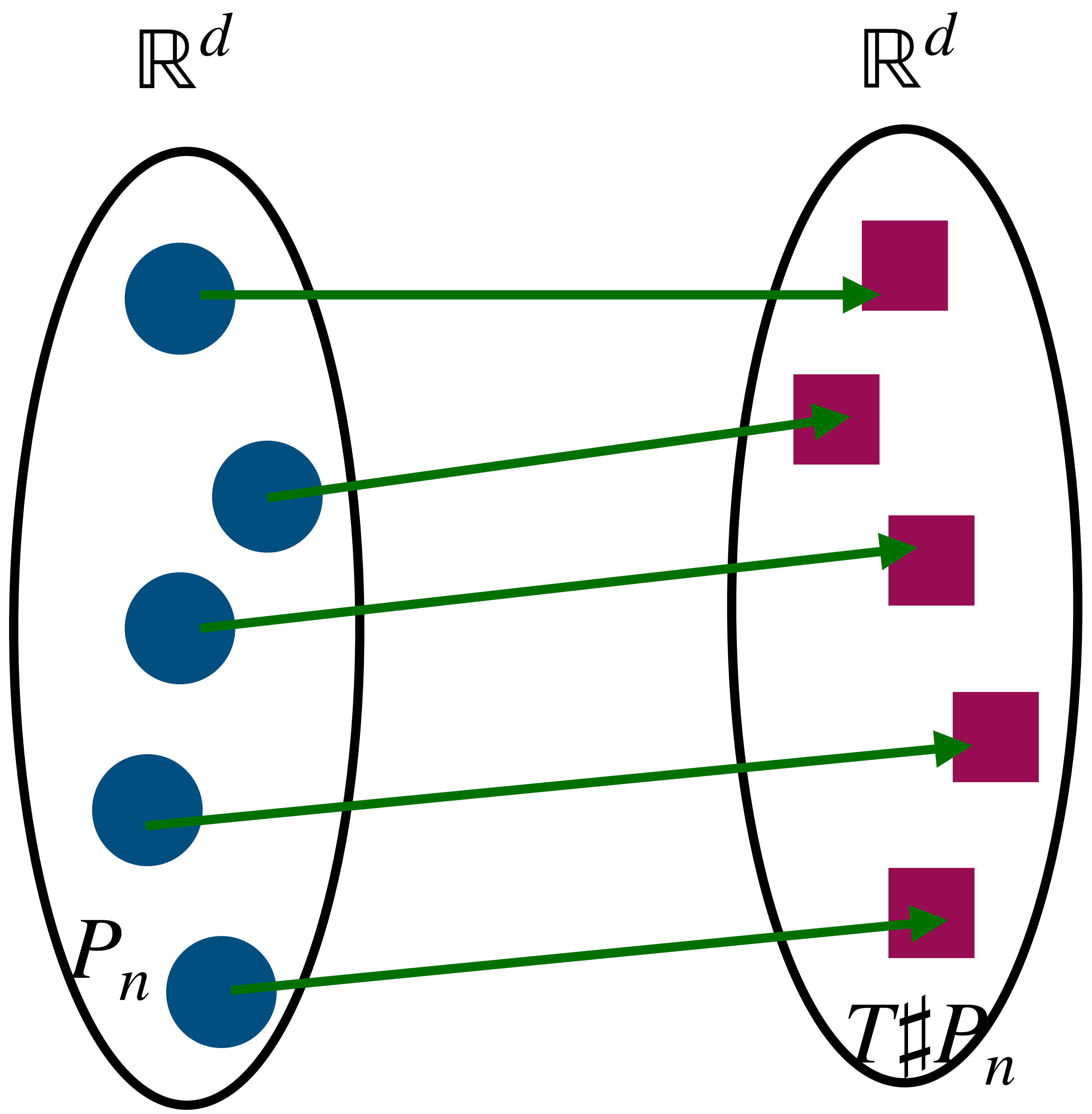
- Here, $T\sharp P_n$ denotes the *push-forward measure* of P_n via mapping T



Push-forward measure

- Recall that, $P_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{B_i}$ and $T : \mathbb{R}^d \to \mathbb{R}^d$
- Then, $T\sharp P_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{T(B_i)}$
- The equation $T\sharp P_n = Q_n$ implies that

$\{T(B_1), T(B_2), \ldots, T(B_n)\} \equiv \{R_1, R_2, \ldots, R_n\}$



General Monge's OT Formulation

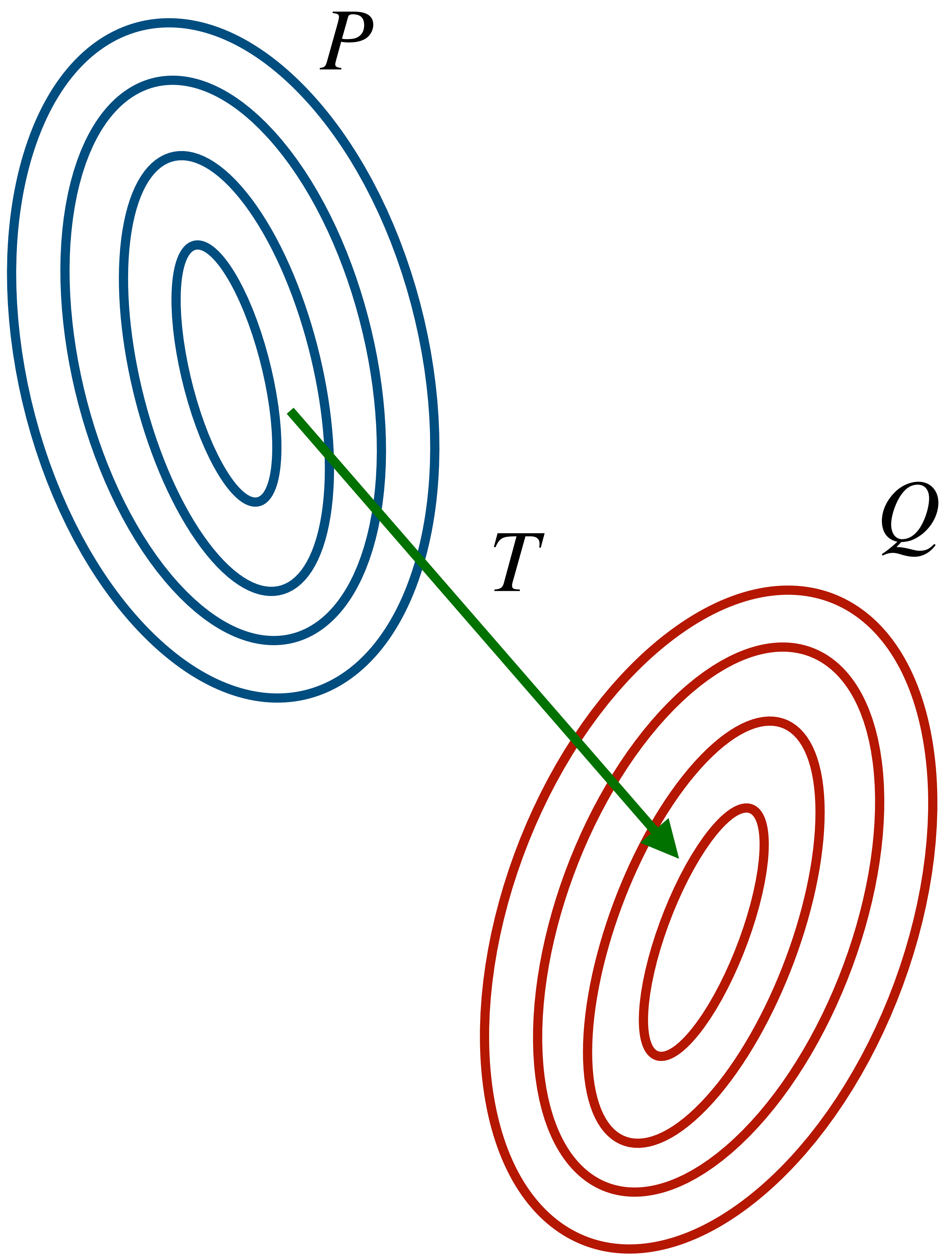
- In general, we can define the Monge's optimal transport beyond discrete probability distributions, such as Gaussian distributions

- For any two probability distributions P and Q, the Monge's Optimal Transport between P and Q can be defined as

$$\boxed{\inf_T \int \|x - T(x)\|^2 dP(x),} \qquad \textbf{(2)}$$

where the mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ in the infimum is such that $T\sharp P = Q$

- Note that, for continuous distributions, $T\sharp P = Q$ means that $P(T^{-1}(A)) = Q(A)$ for any measurable set A of $\mathbb{R}^d$



General Monge's OT Formulation: Challenges

- **Good settings**: When (i) P and Q admit density functions or (ii) P and Q are discrete with uniform weights, there exist optimal maps T that solve the Monge's OT in equation (2)

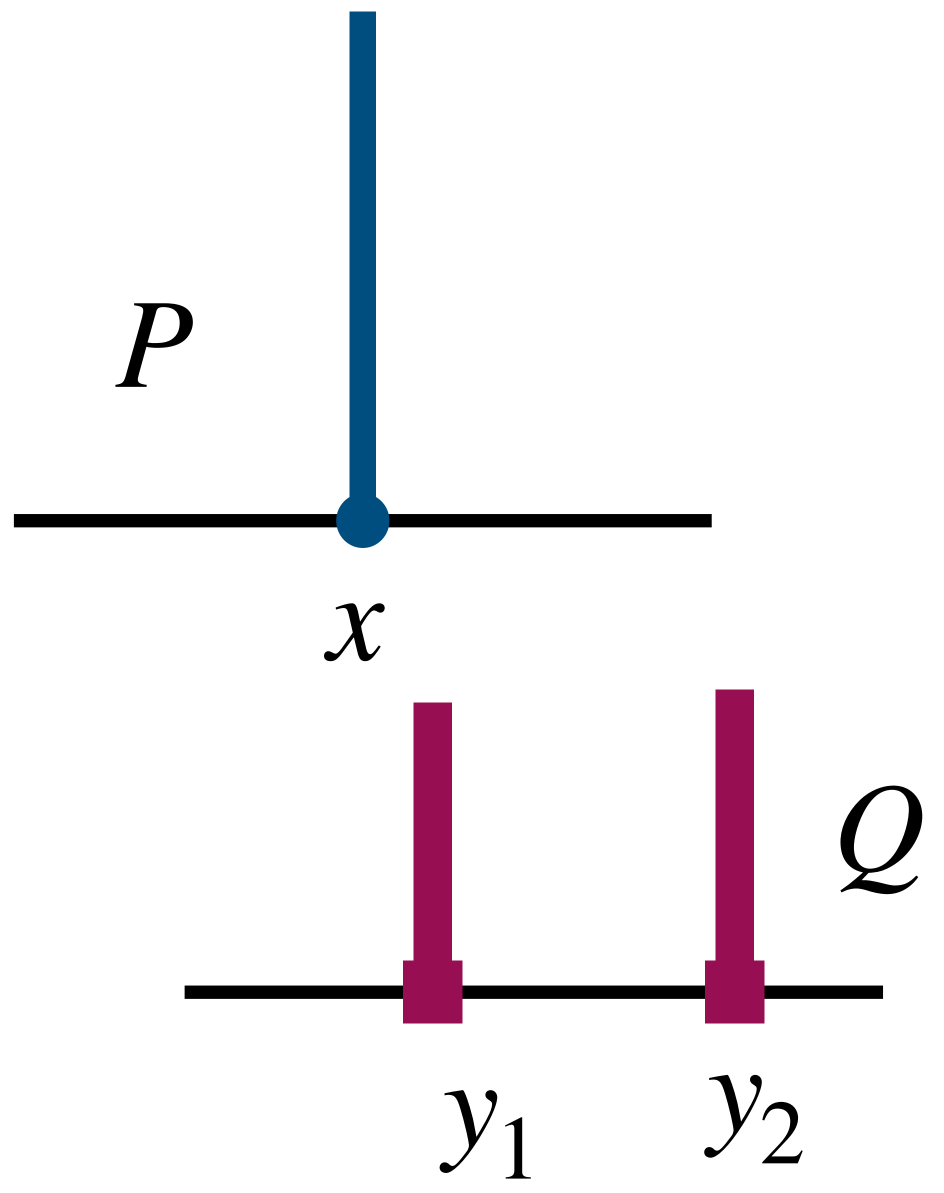
- **Pathological settings**:

  - In certain settings when P and Q are discrete, the existence of mapping T such that $T\sharp P = Q$ may not always be possible

  - Assume that $P = \delta_x$ and $Q = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$, the equation $T\sharp P = Q$ means that

$$P(T^{-1}(\{y_1\})) = Q(\{y_1\}) = \frac{1}{2}$$

  - However, it is not possible as $P(T^{-1}(\{y_1\})) \in \{0,1\}$ depending on whether $x \in T^{-1}(y_1)$



General Monge's OT Formulation: Challenges

- The non-existence of transport map T under pathological settings makes it challenging to use Monge's OT formulation when the probability distributions P and Q are discrete

- Furthermore, due to the non-linearity of the constraint $T \sharp P = Q$, it is non-trivial to solve for or approximate the optimal mapping T in equation (2)

- A relaxation and optimization friendly form of Monge's OT formulation is needed

Kantorovich’s Optimal Transport Formulation

Kantorovich's OT Formulation

- Given two probability distributions P and Q, the *Kantorovich's Optimal Transport* between P and Q can be defined as

$$\mathrm{OT}(P,Q) := \inf_{\pi \in \Pi(P,Q)} \int c(x,y) d\pi(x,y), \qquad \textbf{(3)}$$

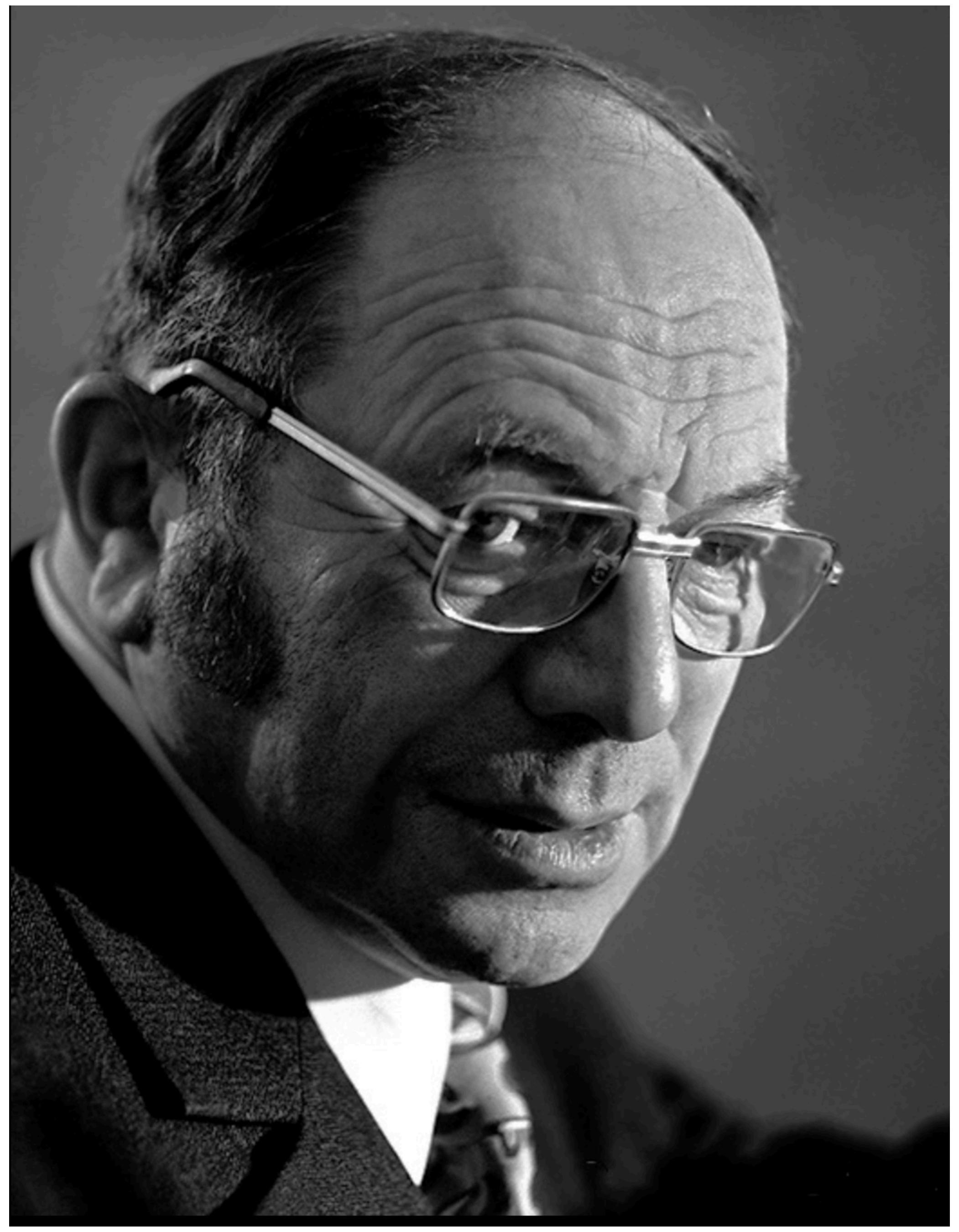


Image from Internet

where $\Pi(P,Q)$ is the set of all joint distributions between P and Q;

$c(\,.\,,\,.\,)$ is a given cost metric

- π is called *transportation plan*
- Under certain assumptions (see Section 4 in [18]), the Kantorovich's OT and Monge's OT are equivalent

Kantorovich's OT for Discrete Measures

- When $P = \delta_\eta$ and $Q = \sum_{i=1}^{m} q_i \delta_{\theta_i}$, then

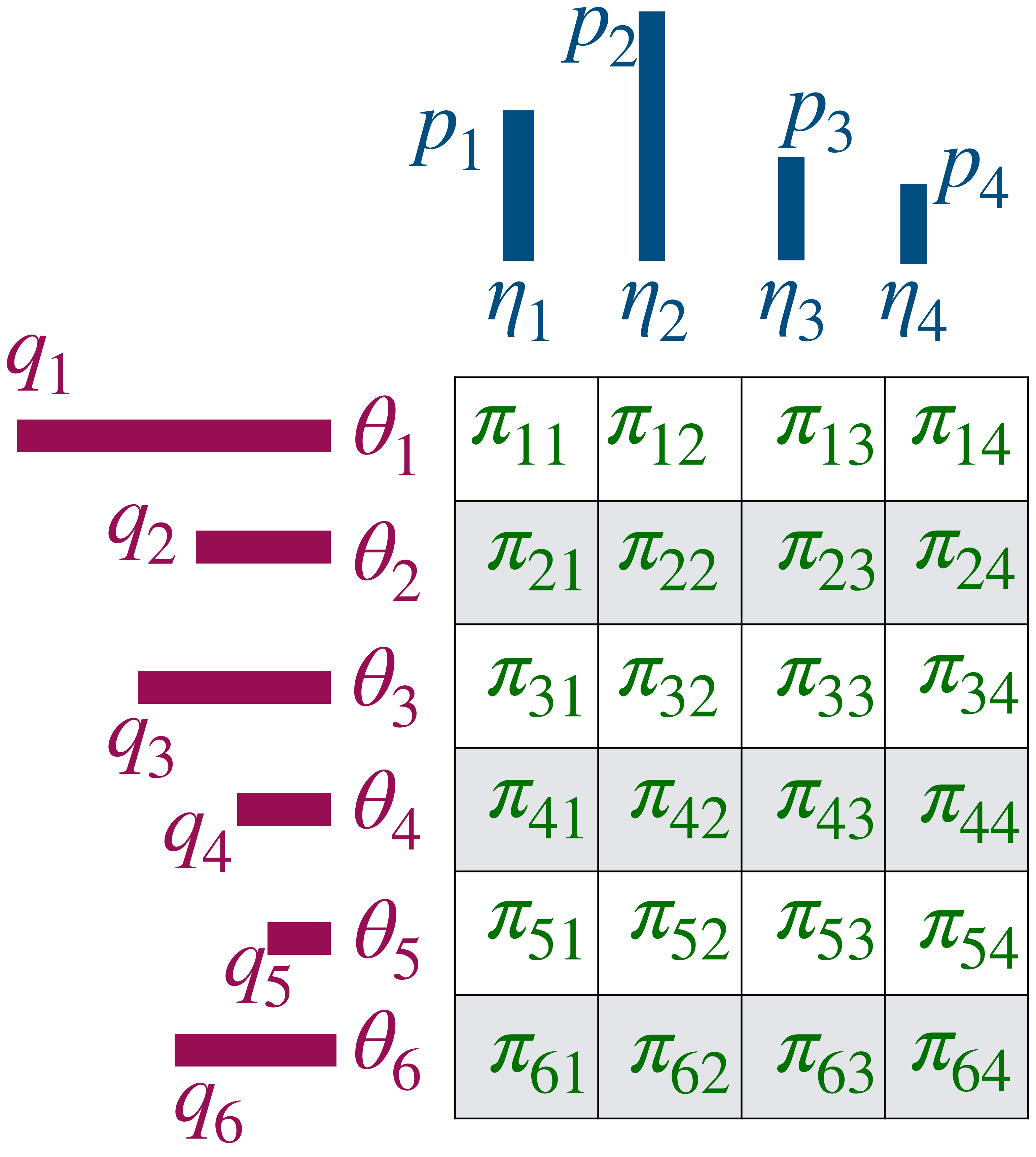
$$\text{OT}(P, Q) = \sum_{i=1}^{m} q_i \cdot c(\eta, \theta_i)$$

- When $P = \sum_{i=1}^{n} p_i \delta_{\eta_i}$ and $Q = \sum_{j=1}^{m} q_j \delta_{\theta_j}$, then

$$\text{OT}(P, Q) = \min_{\pi \geq 0} \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{ij} \cdot c(\eta_i, \theta_j), \quad \textbf{(4)}$$

$$\text{s.t.} \sum_{i=1}^{n} \pi_{ij} = q_j \text{ for all } 1 \leq j \leq m; \quad \sum_{j=1}^{m} \pi_{ij} = p_i \text{ for all}$$

$1 \leq i \leq n$



- These simple examples show that there always exists optimal transportation plan when P and Q are discrete, which is in contrast to the Monge's OT formulation

Kantorovich's OT for Discrete Measures

- We can rewrite the problem (4) as follows

$$\text{OT}(P, Q) = \min_{\pi \in \mathbb{R}^{n \times m}} \langle C, \pi \rangle \quad \textbf{(5)}$$

$$\text{s.t. } \pi \geq 0; \pi 1_m = \mathbf{p}; \pi^\top 1_n = \mathbf{q},$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_n); \mathbf{q} = (q_1, q_2, \ldots, q_m)$

- The problem (3) is a **linear programming** problem
- The set $\mathscr{P} = \{\pi \in \mathbb{R}^{n \times m}: \ \pi \geq 0, \ \pi 1_m = \mathbf{p}, \ \pi^\top 1_n = \mathbf{q}\}$ is called a transportation polytope, which is a *convex set*

Computational Complexity of Kantorovich's Formulation

- The below theorem yields the best computational complexity of the network simplex algorithm for solving the linear programming (5)

Theorem 1: The best computational complexity of the network simplex algorithm for solving the linear programming (5) is of the order of [19]

$$\mathcal{O}((n+m)nm\log(n+m)\log((n+m)\|C\|_{\infty}))$$

- When $n = m$, the complexity becomes $\mathcal{O}(n^3 \log n)$, which is practically very expensive when n is very large

- Therefore, the network simplex algorithm is not sufficiently scalable to use for large-scale machine learning and deep learning applications

Entropic (Regularized) Optimal Transport

Entropic (Regularized) Optimal Transport

- We now discuss an useful approach to obtain scalable approximation of optimal transport
- The idea is that we regularize the optimal transport (5) by the entropy of the transportation plan [20], named **entropic (regularized) optimal transport:**

$$\text{EOT}_\eta(P, Q) = \min_{\pi \in \mathscr{P}(\mathbf{p},\mathbf{q})} \langle C, \pi \rangle - \eta H(\pi), \qquad \textbf{(6)}$$

where $\eta > 0$ is the *regularized parameter*;

$$H(\pi) = -\sum_{i=1}^{n}\sum_{j=1}^{m} \pi_{ij} \log(\pi_{ij});$$

$$\mathscr{P}(\mathbf{p}, \mathbf{q}) = \{\pi \in \mathbb{R}^{n\times m} : \pi 1_m = \mathbf{p}, \pi^\top 1_n = \mathbf{q}\};$$

Here, we use a convention that $\log(x) = -\infty$ when $x \leq 0$

Properties of Entropic Optimal Transport

- For each regularized parameter $\eta > 0$, the objective function of the entropic regularized optimal transport is *$\eta-$strongly convex function*
  - It is because the function $-H(\,.\,)$ is 1-strongly convex function as long as $\pi_{ij} \leq 1$ for all (i, j)
- As the constrained set $\mathscr{P}(\mathbf{p}, \mathbf{q})$ is convex, it indicates that there exists *unique* optimal transportation plan, denoted by π_η^*, for solving the entropic regularized optimal transport

Properties of Entropic Optimal Transport

Theorem 2: (a) When $\eta \to 0$, we have

$$\text{EOT}_\eta(P, Q) \to \text{OT}(P, Q),$$
$$\pi^*_\eta \to \underset{\pi \in \mathcal{P}: \langle C, \pi \rangle = \text{OT}(P,Q)}{\arg\min} \{-H(\pi)\},$$

(b) When $\eta \to \infty$, we have

$$\text{EOT}_\eta(P, Q) \to \langle C, \mathbf{p} \otimes \mathbf{q} \rangle,$$
$$\pi^*_\eta \to \mathbf{p} \otimes \mathbf{q} = \mathbf{p}\mathbf{q}^\top$$

- The results of part (b) indicate that when the regularized parameter η is sufficiently large, we can treat the distributions P and Q as independent distributions

Sinkhorn Algorithm

- We now discuss a popular algorithm, named **Sinkhorn algorithm**, for solving the entropic regularized optimal transport (6)
- **Optimization challenges of primal form**: The primal form (6) is an constrained optimization problem with several constraints; therefore, it may be non-trivial to solve the primal form directly
- **Dual form of entropic optimal transport (6)**: We will demonstrate that solving the dual form of (9), which is an unconstrained optimization problem, is easier
- Solving the dual form is equivalent to solve

$$\min_{u \in \mathbb{R}^n, v \in \mathbb{R}^m} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \exp\left(u_i + v_j - \frac{C_{ij}}{\eta} \right) \right] - u^\top \mathbf{p} - v^\top \mathbf{q} \qquad \textbf{(7)}$$

Sinkhorn Algorithm: Detailed Description

- **Step 1**: Initialize $u^0 = \mathbf{0} \in \mathbb{R}^n$ and $v^0 = \mathbf{0} \in \mathbb{R}^m$
- **Step 2**: For any $t \geq 0$, we perform
  - If t is an even number, then for all (i, j)

$$u_i^{t+1} = \log(p_i) - \log\left(\sum_{j'=1}^{m} \exp\left(v_{j'}^t - \frac{C_{ij'}}{\eta}\right)\right), \qquad v_j^{t+1} = v_j^t$$

  - If t is an odd number, then for all (i, j)

$$v_j^{t+1} = \log(q_j) - \log\left(\sum_{i'=1}^{m} \exp\left(u_{i'}^t - \frac{C_{i'j}}{\eta}\right)\right), \qquad u_i^{t+1} = u_i^t$$

  - Increase $t \leftarrow t + 1$

Approximation of Optimal Transport via Sinkhorn algorithm

- Now, we discuss briefly the complexity of approximating the value of optimal transport via the Sinkhorn algorithm

- **Goal**: We would like to find a transportation plan $\bar{\pi} \in \mathscr{P}$ (see definition of $\mathscr{P}$ in Slide 28) such that

$$\langle C, \bar{\pi} \rangle \leq \min_{\pi \in \mathscr{P}} \langle C, \pi \rangle + \epsilon$$

- We call $\bar{\pi}$ the ϵ-approximation plan

Approximation of Optimal Transport via Sinkhorn algorithm

- Denote (u^t, v^t) as the updates of step t from the Sinkhorn algorithm (See Slide 35)

- The corresponding transportation plan is

$$\pi^t := \text{diag}(\exp(u^t)) \cdot K \cdot \text{diag}(\exp(v^t)),$$

where $\text{diag}(\exp(u^t))$ denotes the diagonal matrix with $\exp(u_1^t), \ldots, \exp(u_n^t)$ in its diagonal

- Unfortunately, $\pi^t \notin \mathscr{P}$, namely, we do not have either $\pi^t 1_m = \mathbf{p}$ or $(\pi^t)^\top 1_n = \mathbf{q}$

Approximation of Optimal Transport via Sinkhorn algorithm

- Therefore, we need to do an extra rounding step to transform π^t to $\bar{\pi}^t$ such that $\bar{\pi}^t 1_m = \mathbf{p}$ and $(\bar{\pi}^t)^\top 1_n = \mathbf{q}$

- Details of that rounding step are in Algorithm 2 in [21] (We skip this step in the lecture for the simplicity)

Theorem 3: Assume that $\eta = \frac{\epsilon}{4\log(\max\{n, m\})}$. Denote by (u^t, v^t) updates from the Sinkhorn algorithm for the entropic optimal transport with regularized parameter η and denote by $\bar{\pi}^t$ the rounding transportation plan we obtain from these updates. Then, we have

$$\langle C, \bar{\pi}^t \rangle \leq \min_{\pi \in \mathcal{P}} \langle C, \pi \rangle + \epsilon$$

as long as $t = \mathcal{O}(\frac{\|C\|_\infty^2 \log(\max\{n, m\})}{\epsilon^2})$.

Approximation of Optimal Transport via Sinkhorn algorithm

- The proof of Theorem 3 can be found in Theorem 2 of [22]
- Each iteration of the Sinkhorn algorithm requires $\max\{n, m\}^2$ arithmetic operations
- The result of Theorem 6 indicates that the total computational complexity of approximating the optimal transport via the Sinkhorn algorithm is

$$\mathcal{O}(\max\{n, m\}^2 \frac{\|C\|_\infty^2 \log(\max\{n, m\})}{\epsilon^2})$$

- It is much cheaper than the complexity of the network simplex algorithm in Theorem 2, which is of the order $\mathcal{O}(\max\{n, m\}^3)$

Other Approximations of Optimal Transport

- There are other optimization algorithms that outperform Sinkhorn:
    - Greedy version of Sinkhorn (Greenkhorn) [23]
    - Accelerated Sinkhorn [24]
- The scalable approximations of optimal transport via these optimization algorithms have lead to several interesting methodological developments in machine learning

[23] Tianyi Lin, Nhat Ho, Michael I. Jordan.On efficient optimal transport: an analysis of greedy and accelerated mirror descent algorithms. ICML, 2019

[24] Tianyi Lin, Nhat Ho, Michael I. Jordan. On the efficiency of entropic regularized algorithms for optimal transport. Journal of Machine Learning Research (JMLR), 2022

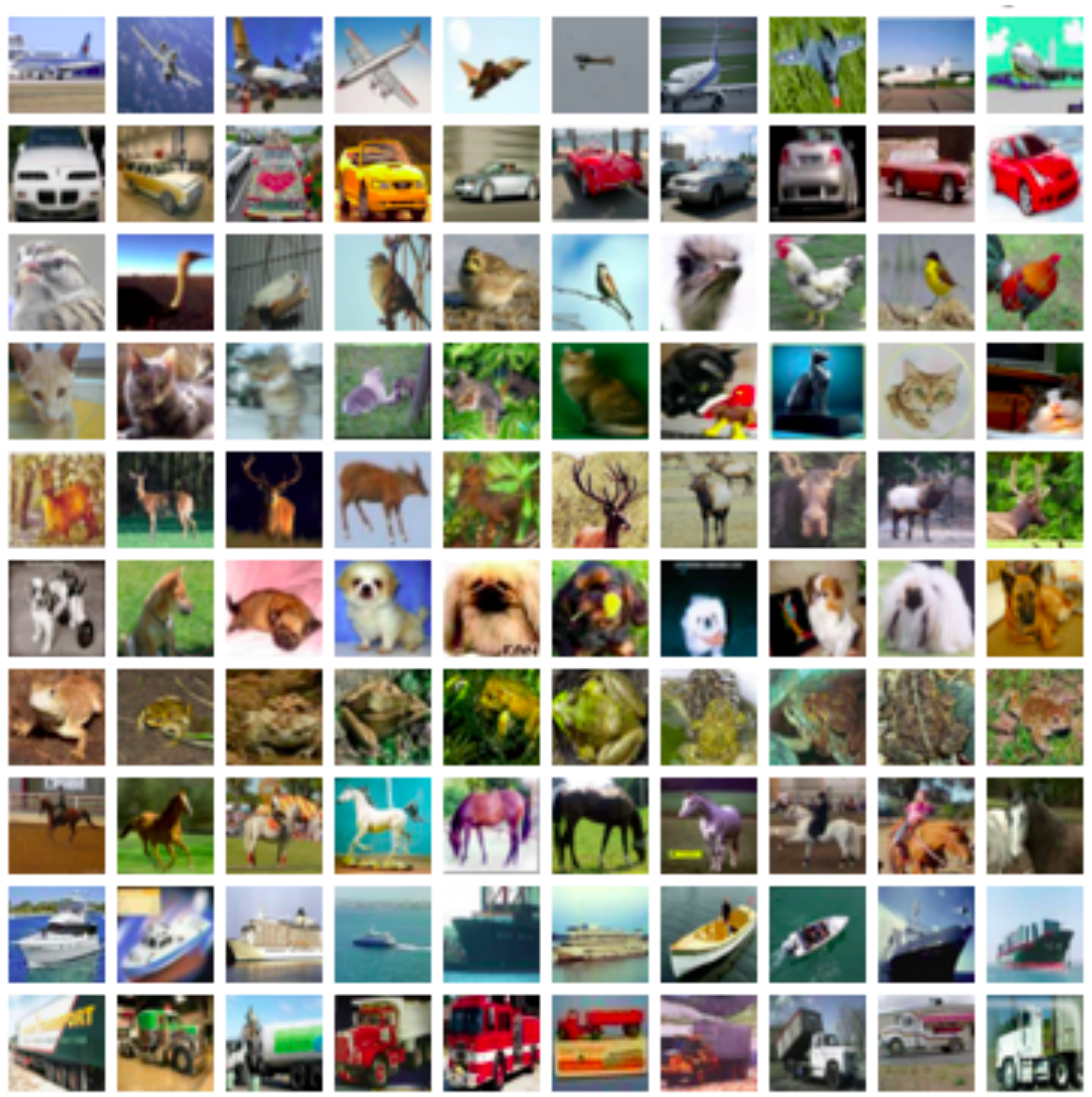
Deep Generative Model via Optimal Transport

- Wasserstein GAN
- Issues of Wasserstein GAN:
    - Misspecified Matchings of Minibatch Schemes
    - Curse of Dimensionality

Generative Model

- We now discuss an important application of optimal transport in generative modeling task

CIFAR 10



Imagenet

- **Goal**: Given a collection of very high dimensional data, we would like to learn the underlying data distribution P effectively

Generative Model

- There are several approaches:
  - Nonparametric approaches:
    - Frequentist density estimator
    - Bayesian nonparametric models
  - Parametric approaches via latent variable assumption:
    - Bayesian hierarchical models
    - Deep learning models, i.e., Variational Auto-Encoder (VAE) [25], Generative Adversarial Networks (GANs) [26], etc.

Generative Adversarial Networks (GANs)

- Generative Adversarial Networks is an instance of **implicit methods**, i.e., we do not need explicit density estimation
    - May allow a smooth interpolation across images
    - May be able to capture the underlying variation of the data (images with unseen patterns, etc.)
- It is different from Variational Auto-Encoder, which is an instance of **explicit methods**

Generative Adversarial Networks (GANs)

General recipe of implicit methods:

- We generate z from some distribution $p_Z(\,.\,)$ (e.g., Gaussian distribution)
- We consider a "fake" data generating distribution $T_\phi(z)$ where T_ϕ is some vector-value function parametrized by ϕ
- We need to make sure that $T_\phi(\,.\,)$ is as close as possible to the true distribution P of the data (Here, we do not make any parametric assumption on the true distribution)

➡ Some divergences between $T_\phi(\,.\,)$ and P are needed

Generative Adversarial Networks (GANs)

- For GANs [26], the choice of that divergence is the Jensen-Shannon divergence (JS):

$$\min_{\phi} \text{JS}(T_\phi(z), P), \qquad \textbf{(8)}$$

where $\text{JS}(T_\phi(z), P) := \text{KL}\left(T_\phi(z), \frac{P + T_\phi(z)}{2}\right) + \text{KL}\left(P, \frac{P + T_\phi(z)}{2}\right)$

- If we denote $G = T_\phi$, it is equivalent to the following minimax game:

$$\min_{G} \max_{D} \mathbb{E}_{x \sim P}[\log(D(x))] + \mathbb{E}_{z \sim p_Z}[\log(1 - D(G(z)))] \,,$$

where G : generator, D : discriminator

- This is an instance of **non-convex non-concave minimax optimization** problem

Continuity Issue of GANs

- The JS divergence being used in GANs is **problematic** [27] when $T_\phi(z)$ and P fall into the following cases:
    - Disjoint supports
    - One is continuous distribution and another one is discrete distribution
- **Example**: To see that, we will consider the following simple example: $T_\phi(z) = (\phi, z)$ where $z \sim U(0,1)$ and $P = (0, U(0,1))$
- Direct calculation shows that

$$JS(T_\phi(z), P) = \log(2) \text{ if } \phi \neq 0 \text{ and 0 otherwise}$$

- Therefore, the JS divergence is **discontinuous** at the true parameter $\phi = 0$ and takes constant value when $\phi \neq 0$ (Gradient descent method cannot be used!)

Wasserstein GANs

- One solution to the continuity issue of JS divergence is by using weaker metric, such as optimal transport

- The paper [27] suggests that we can use the **first order Wasserstein metric**

- For any two distributions P and Q, the first order Wasserstein metric between P and Q is defined as follows:

$$W_1(P, Q) = \inf_{\pi \in \Pi(P,Q)} \int \|x - y\| d\pi(x, y),$$

where $\Pi(P, Q)$ denotes the set of joint probability measures between P and Q

Wasserstein GANs

- The objective of **Wasserstein GANs** is then given by:

$$\min_{\phi} W_1(T_\phi(z), P) \quad \textbf{(9)}$$

- The first order Wasserstein metric is meaningful even when the two distributions
    - Have disjoint supports
    - One distribution is discrete and another distribution is continuous
- To see that, we reconsider the example in Slide 46

Wasserstein GANs

- Under this case, we can verify that $W_1(T_\phi(z), P) = |\phi|$ for all $\phi \in \mathbb{R}$
- It is clear that this function is continuous for all ϕ and we can use optimization method to solve $\min_{\phi} |\phi|$
- In general, if $T_\phi(\,.\,)$ is continuous in ϕ, the first order Wasserstein metric $W_1(T_\phi(z), P)$ is also continuous in ϕ
- If $T_\phi(\,.\,)$ is locally Lipschitz and satisfies some regularity conditions, then $W_1(T_\phi(z), P)$ is differentiable almost everywhere (See Theorem 1 in [27])

Wasserstein GANs

- These observations indicate that the first order Wasserstein metric is a valid choice for GANs

- From the definition of first order Wasserstein metric, we can rewrite equation (16) as follows:

$$\min_{\phi} W_1(T_\phi(z), P) = \min_{\phi} \min_{\pi \in \Pi(T_\phi(z), P)} \int \|x - y\| d\pi(x, y) \qquad \textbf{(10)}$$

- Directly optimizing the objective function in equation (10) is not feasible in general

- We will discuss a dual function approach for dealing with that optimization problem

Wasserstein GANs: Dual Function Approach

- **Dual Function Approach**: For any two probability distributions P and Q, the dual form of the first order Wasserstein metric between P and Q has the following form:

$$W_1(P, Q) = \sup_{f \in \mathscr{L}_1} \mathbb{E}_{x \sim P}[f(x)] - \mathbb{E}_{x \sim Q}[f(x)], \qquad \textbf{(11)}$$

where $\mathscr{L}_1$ is the set of 1-Lipschitz function f, i.e., $|f(x) - f(y)| \leq \|x - y\|$ for all $x, y \in \mathbb{R}^d$

- Please refer to Section 5 in [27] about how to derive the dual form (11)

Wasserstein GANs: Dual Function Approach

- Given the dual form of the first order Wasserstein metric in equation (18), we can rewrite Wasserstein GANs as follows:

$$\min_{\phi} W_1(T_\phi(z), P) = \min_{\phi} \max_{f \in \mathscr{L}_1} \mathbb{E}_{x \sim T_\phi(z)}[f(x)] - \mathbb{E}_{x \sim P}[f(x)]$$

$$= \min_{\phi} \max_{f \in \mathscr{L}_1} \mathscr{T}(\phi, f) \qquad \textbf{(12)}$$

- To update the function f in Wasserstein GANs, it is non-trivial as it is a maximization problem over the functional space

- We consider approximating the $\mathscr{L}_1$ space using deep neural networks where we parametrize it as $\{f_\omega\}$ and ω are the weights of neural networks

Wasserstein GANs: Dual Function Approach

- Therefore, we approximate the Wasserstein GANs (19) as

$$\min_{\phi} \max_{\omega} \mathbb{E}_{z \sim p_Z}[f_\omega(T_\phi(z))] - \mathbb{E}_{x \sim P}[f_\omega(x)] \qquad \textbf{(13)}$$

- We can solve both ϕ and ω via (stochastic) gradient descent methods

- The detailed optimization algorithm for solving the approximated Wasserstein GANs (20) is in Algorithm 1 in [27]

Limitations of Dual Function Approach

- **Limitations of dual function approach**:
  - It relies on the choice of first order Wasserstein metric and Euclidean distance to have a nice dual form
  - The Euclidean distance assumption can be very strong in practice as it is not good to capture the difference of high dimensional data
- In general, we would like to have a more general form of Wasserstein GANs, named **optimal transport GANs (OT-GANs)**:

$$\min_{\phi} \text{OT}(T_{\phi}(z), P), \quad \textbf{(14)}$$

where $\text{OT}(T_{\phi}(z), P) = \inf_{\pi \in \Pi(T_{\phi}(z), P)} \int c(x, y) d\pi(x, y)$ and $c(\,.\,,\,.\,)$ is some metric

Optimal Transport GANs (OT-GANs)

- For general cost matrix $c(\,.\,,\,.\,)$, the dual form of OT-GANs (21) can be non-trivial to use
- Therefore, people also advocate the direct optimization of OT-GANs
- **Challenge**: Since both $T_\phi(z)$ and P are continuous, we generally cannot compute directly OT$(T_\phi(z), P)$
- **Solution**: We can use the sample versions of $T_\phi(z)$ and P to approximate OT$(T_\phi(z), P)$

Optimal Transport GANs (OT-GANs)

- For the distribution P, we can use $P_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{X_i}$ where $X_1, X_2, \ldots, X_n$ are the data
- For $T_\phi(z)$, we can use $\frac{1}{M}\sum_{i=1}^{M}\delta_{T_\phi(z_i)}$ where $z_1, z_2, \ldots, z_M$ are i.i.d. samples from $p_Z(\,.\,)$
- It suggests the following approximation of OT-GANs (14)

$$\inf_{\phi} \mathrm{OT}\left(\frac{1}{M}\sum_{i=1}^{M}\delta_{T_\phi(z_i)}, \frac{1}{n}\sum_{i=1}^{n}\delta_{X_i}\right) \qquad \textbf{(15)}$$

Computational Challenge of OT-GANs

Computational Challenge of OT-GANs

- **Computational Challenge**:
  - The computational complexity of approximating the optimal transport between $\frac{1}{M}\sum_{i=1}^{M}\delta_{T_\phi(z_i)}$ and $\frac{1}{n}\sum_{i=1}^{n}\delta_{X_i}$ is $\mathcal{O}(\max\{M,n\}^2)$
  - In practice, n can be very large (as large as a few millions) and M need to be chosen to be quite large (scale with the dimension) to guarantee good approximation of $T_\phi(z)$ via the empirical distribution $\frac{1}{M}\sum_{i=1}^{M}\delta_{T_\phi(z_i)}$
  - Unfortunately, it is unavoidable memory issue of optimal transport
- **Practical Solution**: A popular approach for doing that is to consider minibatches of the entire data, which we refer to as *minibatch optimal transport GANs*

Minibatch Optimal Transport

Minibatch Optimal Transport GANs (mOT-GANs)

- To set up the stage, we need the following notations:
  - We denote by m the minibatch size where $m \leq \min\{M, n\}$
  - We denote $\binom{X^n}{m}$ and $\binom{z^M}{m}$ the sets of all m elements of $\{X_1, \ldots, X_n\}$ and $\{z_1, \ldots, z_M\}$ respectively
  - For any $X^m \in \binom{X^n}{m}$ and $z^m \in \binom{z^M}{m}$, we respectively denote by $P_{X^m} = \frac{1}{m} \sum_{x \in X^m} \delta_x$ and $P_{z^m} = \frac{1}{m} \sum_{z' \in z^m} \delta_{z'}$ the empirical measures of X^m and z^m

Minibatch Optimal Transport GANs (mOT-GANs)

Minibatch Optimal Transport GANs (mOT-GANs): For any batch size $1 \leq m \leq \min\{M, n\}$ and number of minibatches k, we draw $X_1^m, \ldots, X_k^m$ and $z_1^m, \ldots, z_k^m$ uniformly from $\binom{X^n}{m}$ and $\binom{z^M}{m}$. The minibatch optimal transport GANs is given by:

$$\min_{\phi} \frac{1}{k} \sum_{i=1}^{k} \text{OT}(T_\phi(P_{z_i^m}), P_{X_i^m}) \qquad \textbf{(16)}$$

- The common choice that people use in practice is $k = 1$ and m is chosen based on the memory of GPU
- Note that, the choice that $k = 1$ can lead to sub-optimal result in practice

Minibatch Optimal Transport GANs (mOT-GANs)

- **Computational Complexity of mOT-GANs**:
  - When ϕ is given, the complexity of computing OT$(T_\phi(P_{z_i^m}), P_{X_i^m})$ exactly is at the order of $\mathcal{O}(m^3)$ if we use exact-solver to solve the linear programming
  - We can improve the complexity to $\mathcal{O}(m^2)$ via using entropic regularized optimal transport to approximate OT$(T_\phi(P_{z_i^m}), P_{X_i^m})$
  - Therefore, the best complexity of approximating $\sum_{i=1}^{k}$ OT$(T_\phi(P_{z_i^m}), P_{X_i^m})$ is $\mathcal{O}(km^2)$

OT GANs: Minibatch Approach

- For the approximation of OT-GANs in equation (15), the complexity is $\mathcal{O}(\max\{M, n\}^2)$

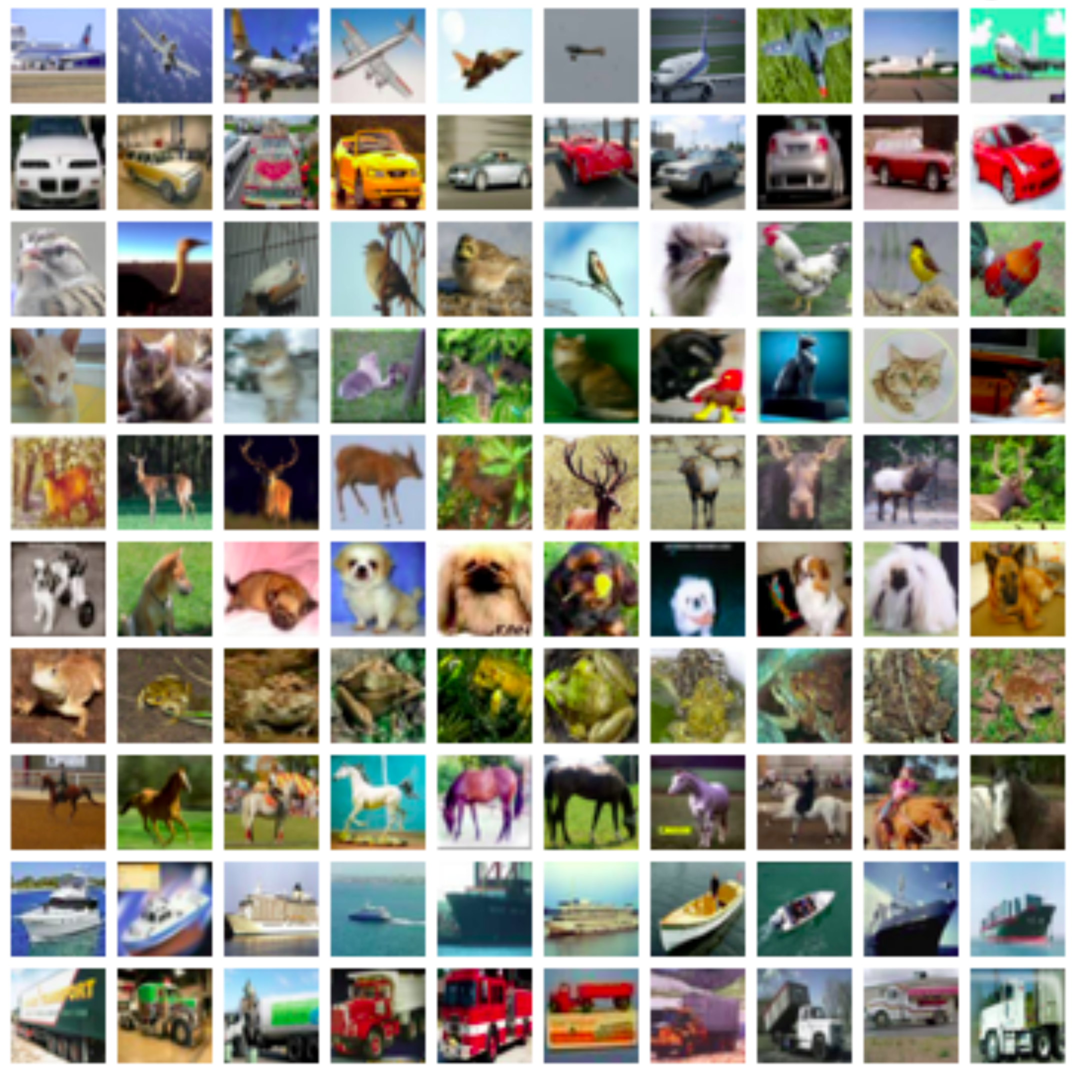
- As long as $km^2 \lll \max\{M, n\}^2$, the complexity of mOT-GANs is much cheaper than that of OT-GANs for each parameter ϕ

- The mOT-GANs is convenient for large-scale settings of deep generative model

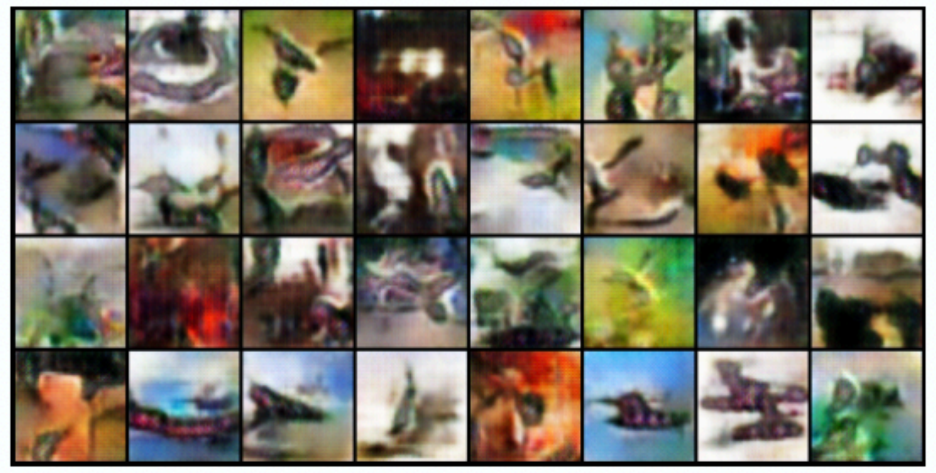
- Similar to OT-GANs, we can solve optimal parameter ϕ of mOT-GANs (16) via (stochastic) gradient descent methods

Wasserstein GANs: Minibatch Approach

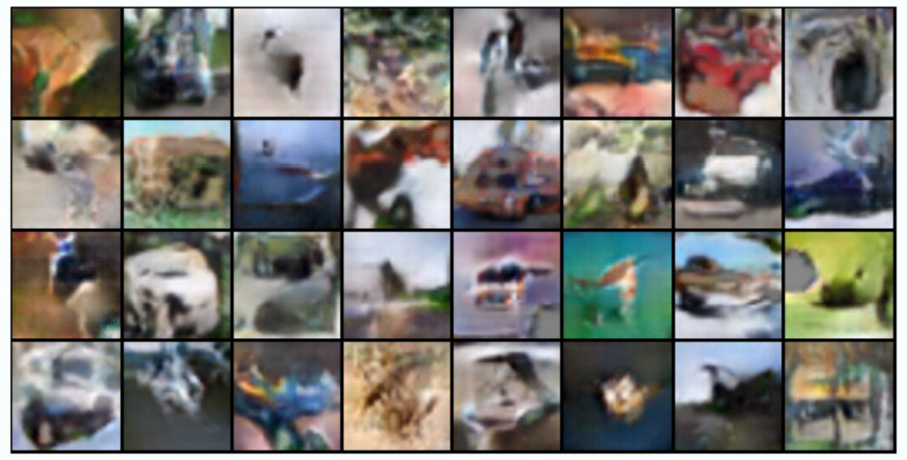
- Examples of CIFAR 10 generated images via mOT-GANs:



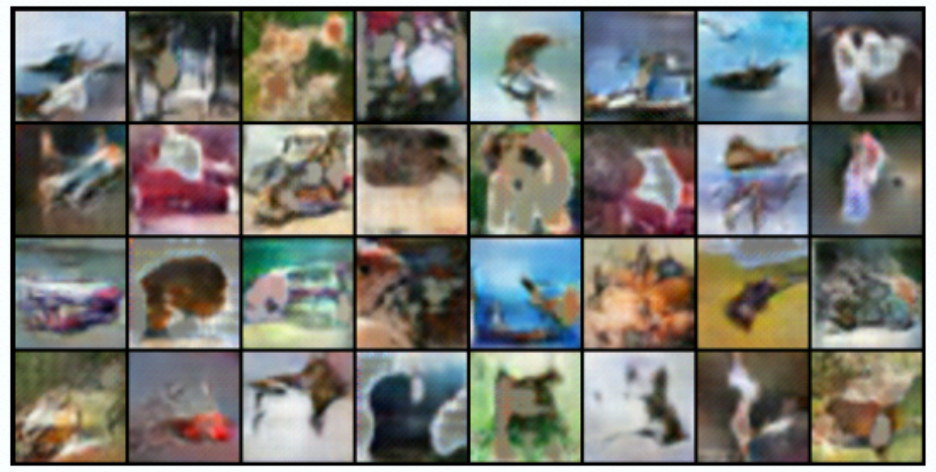
Data



Minibatch size: m= 200
Number of minibatches: k = 2



Minibatch size: m= 200
Number of minibatches: k = 4



Minibatch size: m= 200
Number of minibatches: k = 8

Generated data

Issues of mOT-GANs

- mOT-GANs suffer from misspecified matching issue, i.e., the optimal transport plan from the mOT-GANs contains wrong matchings that do not appear in the original optimal transport plan of OT-GANs

- The misspecified matchings lead to a decline in the performance of mOT-GANs

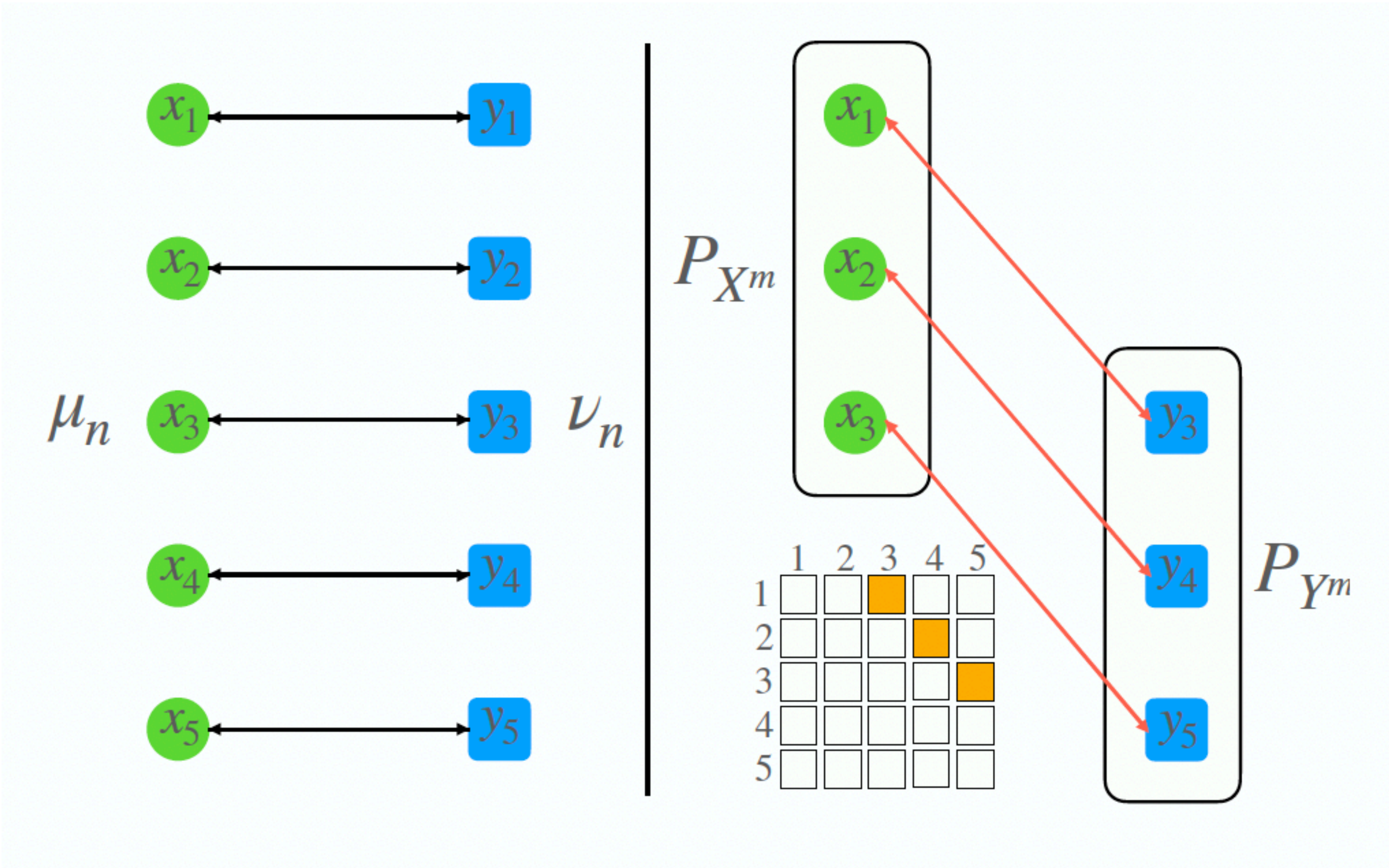
- There are a few recent proposals to solve the misspecified matching issue, includes using partial optimal transport [28], hierarchical optimal transport [29], unbalanced optimal transport [30]

Minibatch Partial Optimal Transport [28]

[28] Khai Nguyen, Dang Nguyen, Tung Pham, Nhat Ho. *Improving minibatch optimal transport via partial transportation*. ICML, 2022

Misspecified Matching Issue of MOT

- We consider a simple example where P_n, Q_n are two empirical distributions with 5 supports on 2D: $\{(0,1),(0,2),(0,3),(0,4),(0,5)\}$, $\{(1,1),(1,2),(1,3),(1,4),(1,5)\}$



LHS: Optimal matching (black color) between P_n, Q_n;

RHS: Wrong matchings (red color) induced by minibatches

Alleviating Misspecified Matching of M-OT via Partial Transportation

- We now demonstrate that we can alleviate the misspecified matching issue via partial optimal transport

- The *Partial Optimal Transport (POT)* between P_n and Q_n is defined as follow:

$$\text{POT}_s(P_n, Q_n) = \min_{\pi \in \Pi_s(\boldsymbol{u}_n, \boldsymbol{u}_n)} \langle C, \pi \rangle,$$

where C is the distance matrix; s : transportation fraction;
$\boldsymbol{u}_n$ is the uniform measures over n supports; and

$$\Pi_s(\boldsymbol{u}_n, \boldsymbol{u}_n) := \left\{ \pi \in \mathbb{R}_+^{n \times n} : \pi 1_n \leq \boldsymbol{u}_n, \pi^\top 1_n \leq \boldsymbol{u}_n, 1^\top \pi 1 = s \right\}$$

Minibatch Partial Optimal Transport

- The *Minibatch Partial Optimal Transport* (m-POT) [21] between P_n and Q_n with transportation fraction s is defined as

$$\text{m-POT}_s(P_n, Q_n) = \frac{1}{k} \sum_{i=1}^{k} \text{POT}_s(P_{X_i^m}, P_{Y_i^m}),$$

where $X_1^m, \ldots, X_k^m \in \binom{X^n}{m}$; $Y_1^m, \ldots, Y_k^m \in \binom{Y^n}{m}$;

$P_{X_i^m}, P_{Y_i^m}$ are empirical measures associated with X_i^m and Y_i^m

Computational Complexity of Minibatch Partial Optimal Transport

- We have an equivalent way to write m-POT in terms of m-OT as follows:

$$\text{m-POT}_s(P_n, Q_n) = \frac{1}{k}\sum_{i=1}^{k} \min_{\pi \in \Pi(\bar{\boldsymbol{\alpha}}_i, \bar{\boldsymbol{\alpha}}_i)} \langle \bar{C}_i, \pi \rangle,$$

where $\overline{C}_i = \begin{pmatrix} C_i & 0 \\ 0 & A_i \end{pmatrix} \in \mathbb{R}_+^{(m+1)\times(m+1)}$;

C_i is a cost matrix formed by the differences of elements of X_i^m and Y_i^m;

$A_i > 0$ for all $i = 1,2,\ldots,k$;

$\bar{\boldsymbol{\alpha}}_i = [\boldsymbol{u}_m, 1 - s]$ for all $i = 1,2,\ldots,k$

- By using entropic regularized approach, we can compute the m-POT with computational complexity $\mathcal{O}(k(m+1)^2)$, which is comparable to that of m-OT

Minibatch Partial Optimal Transport

- The corresponding transportation plan of minibatch partial optimal transport with transportation fraction s is given by:

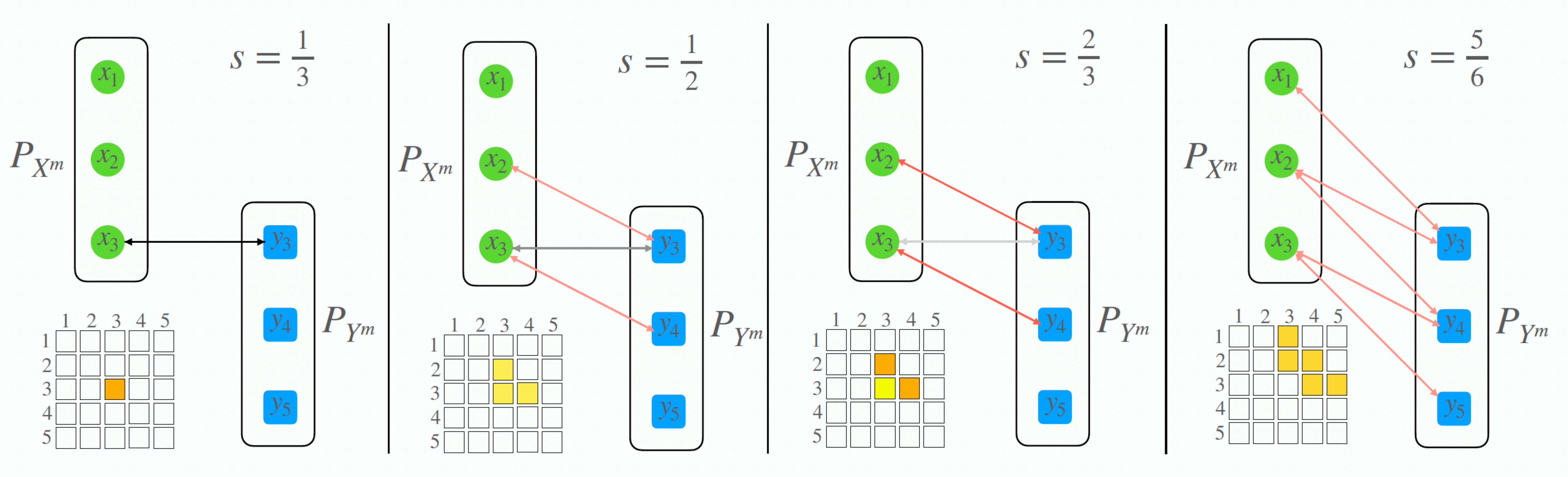
$$\pi^{\text{m-POT}_k^s} = \frac{1}{k}\sum_{i=1}^{k} \pi^{POT_s}_{P_{X_i^m}, P_{Y_i^m}},$$

where $\pi^{\text{POT}_s}_{P_{X_i^m}, P_{Y_i^m}}$ is a transportation matrix from solving $\text{POT}_s(P_{X_i^m}, P_{Y_i^m})$;

$\pi^{\text{POT}_s}_{P_{X_i^m}, P_{Y_i^m}}$ is expanded to a $n \times n$ matrix that has padded zero entries to indices which are different from those of X_i^m and Y_i^m

Minibatch Partial Optimal Transport

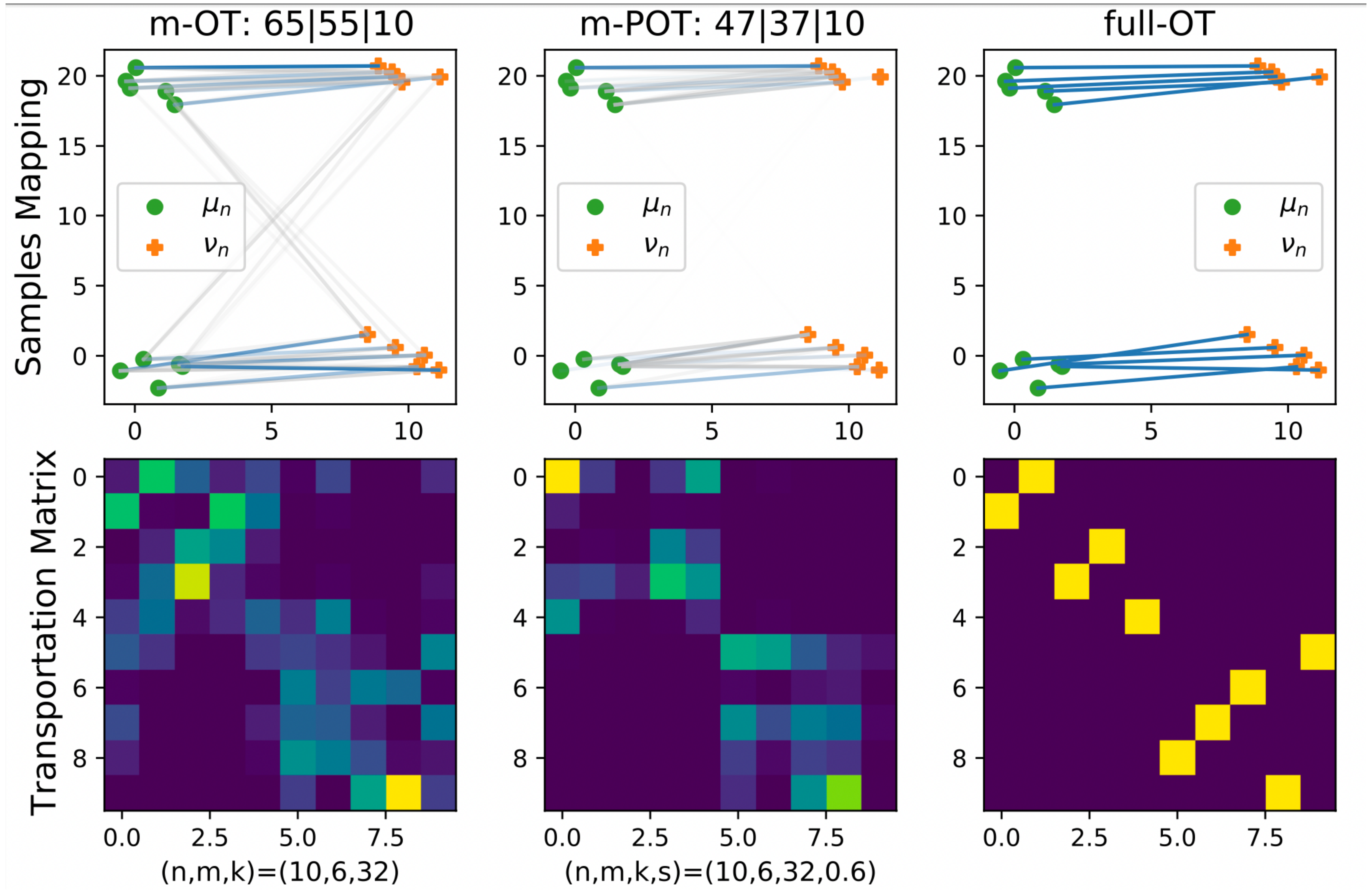
- The m-POT can alleviate misspecified matchings



P_n, Q_n are two empirical distributions with 5 supports on 2D:
$\{(0,1), (0,2), (0,3), (0,4), (0,5)\}, \{(1,1), (1,2), (1,3), (1,4), (1,5)\}$

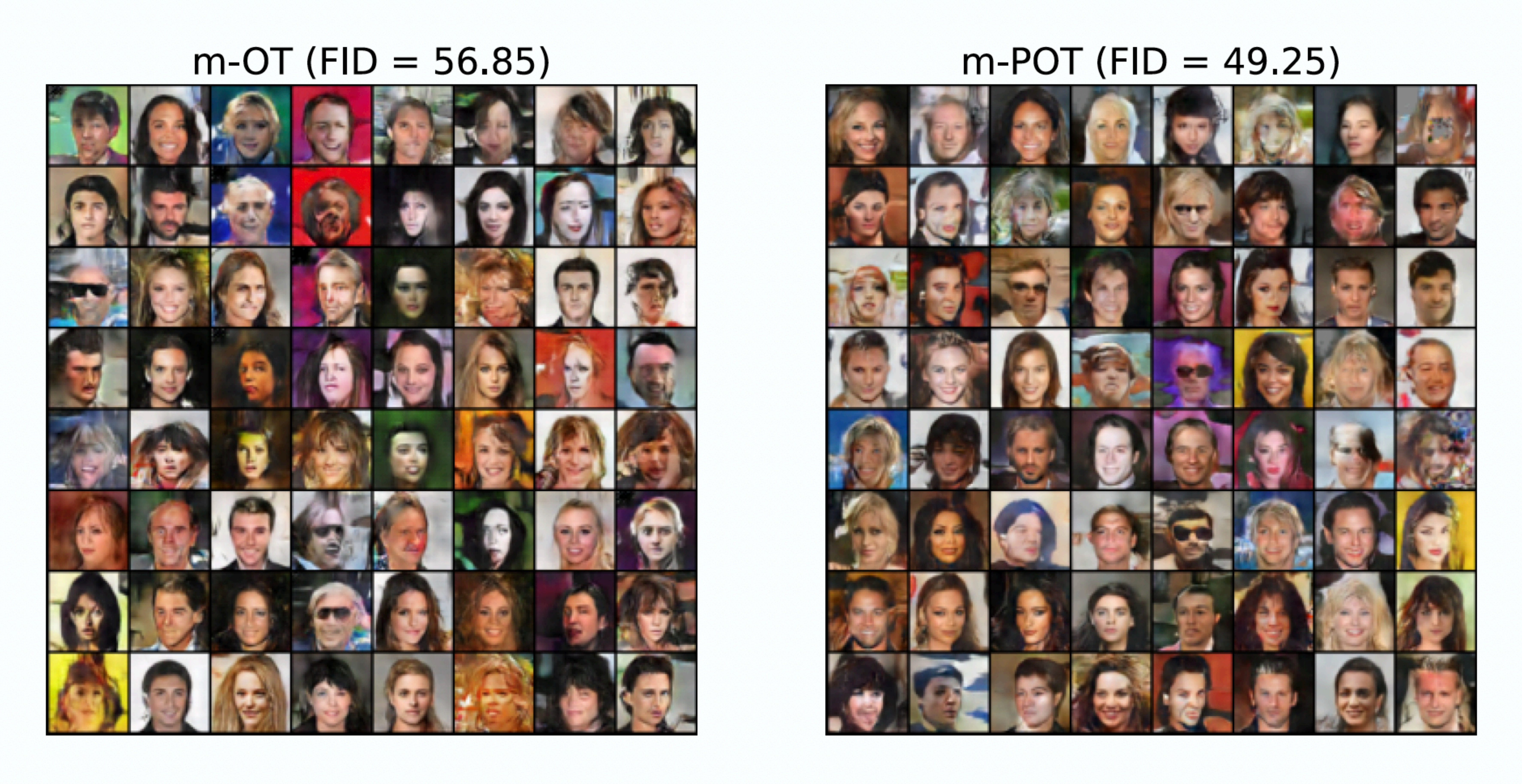
Minibatch Partial Optimal Transport

- The m-POT can alleviate misspecified matchings



The transportation between two empirical measures of 10 supports that are drawn from two mixture of Gaussians of two components.

Experiments: Deep Generative Model



CelebA is a large-scale face attributes dataset with more than 200000 celebrity images.

Batch of Minibatches Optimal Transport [29]

[29] Khai Nguyen, Dang Nguyen, Quoc Nguyen, Tung Pham, Dinh Phung, Hung Bui, Trung Le, Nhat Ho. *On transportation of mini-batches: A hierarchical approach*. ICML, 2022

Alleviating Misspecified Matching of m-OT via Hierarchical Approach

- The m-POT requires to choose good transportation fraction s, which can be non-trivial in practice

- We now describe another approach that can be used to alleviate the misspecified matching of m-OT without any tuning parameter

- The *Batch of Minibatches Optimal Transport* (BoMb-OT) between P_n and Q_n is defined as

$$\text{BoMb-OT}(P_n, Q_n) = \min_{\gamma \in \Pi(P_k^{\otimes m}, Q_k^{\otimes m})} \sum_{i=1}^{k} \sum_{j=1}^{k} \gamma_{ij} \text{OT}(P_{X_i^m}, P_{Y_j^m}),$$

where $X_1^m, \ldots, X_k^m \in \binom{X^n}{m}$; $Y_1^m, \ldots, Y_k^m \in \binom{Y^n}{m}$;

$$P_k^{\otimes m} = \frac{1}{k} \sum_{i=1}^{k} \delta_{X_i^m} \text{ and } Q_k^{\otimes m} = \frac{1}{k} \sum_{i=1}^{k} \delta_{Y_i^m};$$

$P_{X_i^m}, P_{Y_j^m}$ are empirical measures associated with X_i^m and Y_j^m

Batch of Minibatches Optimal Transport

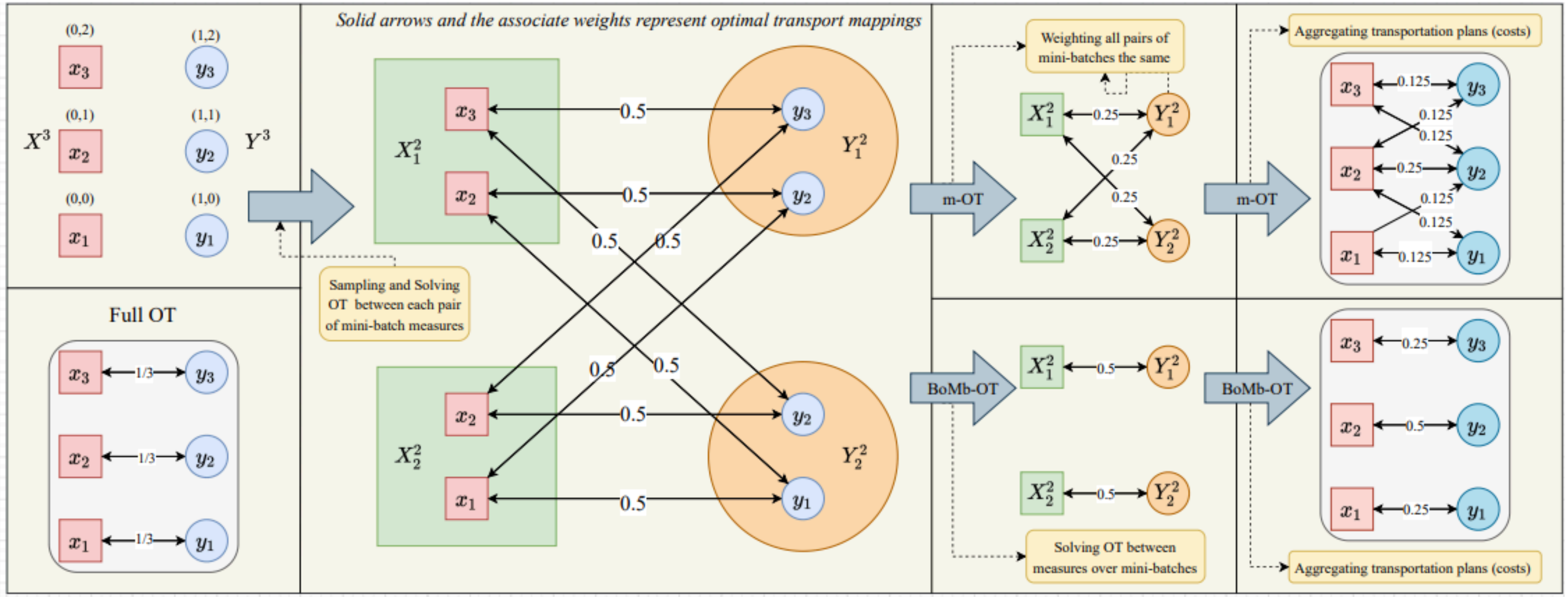


Figure 1: Visualization of the m-OT and the BoMb-OT in providing a mapping between samples.

Batch of Minibatches Optimal Transport

- The corresponding transportation plan of *Batch of minibatches optimal transport* (BoMb-OT) between P_n and Q_n is defined as

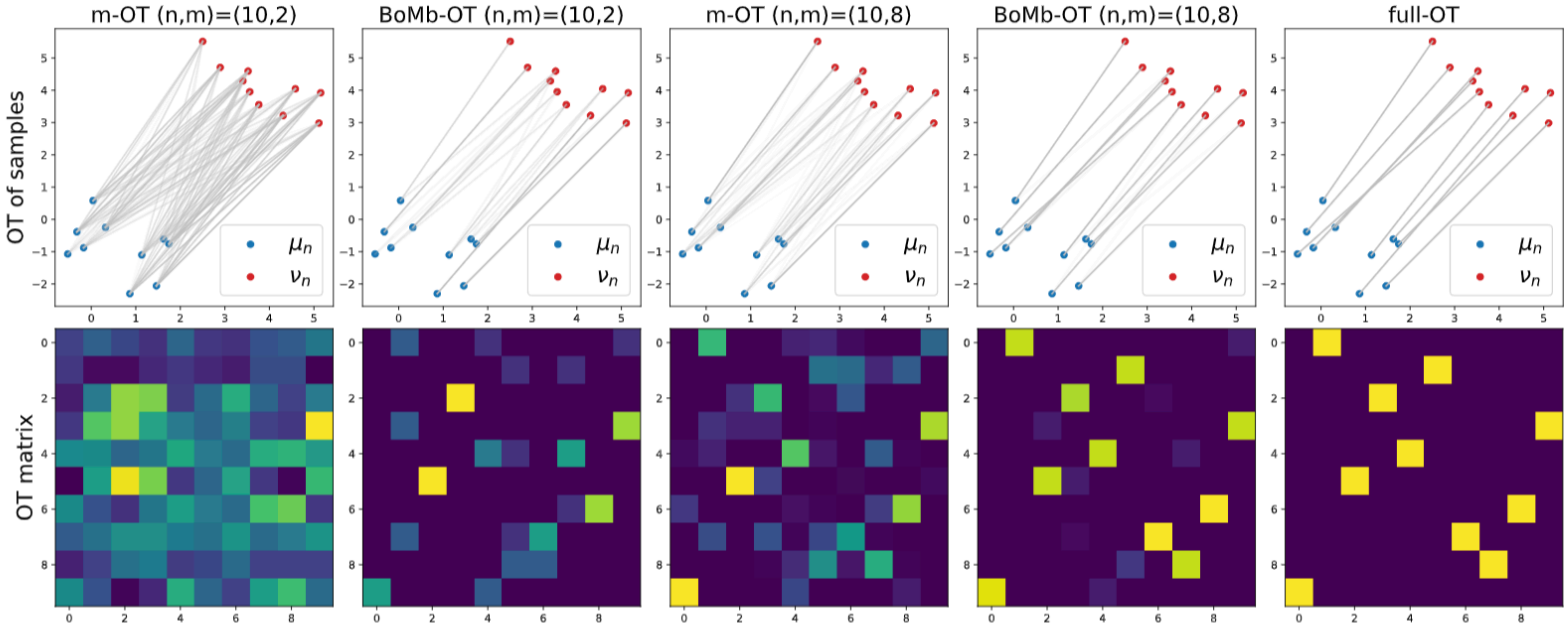
$$\pi^{\text{BoMb-OT}_k} = \sum_{i=1}^{k} \sum_{j=1}^{k} \gamma_{ij} \pi^{OT}_{P_{X_i^m}, P_{Y_j^m}},$$

where $\pi^{\text{OT}}_{P_{X_i^m}, P_{Y_j^m}}$ is a transportation matrix that is returned by solving $\text{OT}(P_{X_i^m}, P_{Y_j^m})$;

$\pi^{\text{OT}}_{P_{X_i^m}, P_{Y_j^m}}$ is expanded to a $n \times n$ matrix that has padded zero entries to indices which are different from those of X_i^m and Y_j^m;

γ is the transportation matrix between $P_k^{\otimes m}$ and $Q_k^{\otimes m}$

Batch of Minibatches Optimal Transport



The transportation between two empirical measures of 10 supports that are drawn from two Gaussians.

Experiments: Deep Generative Model

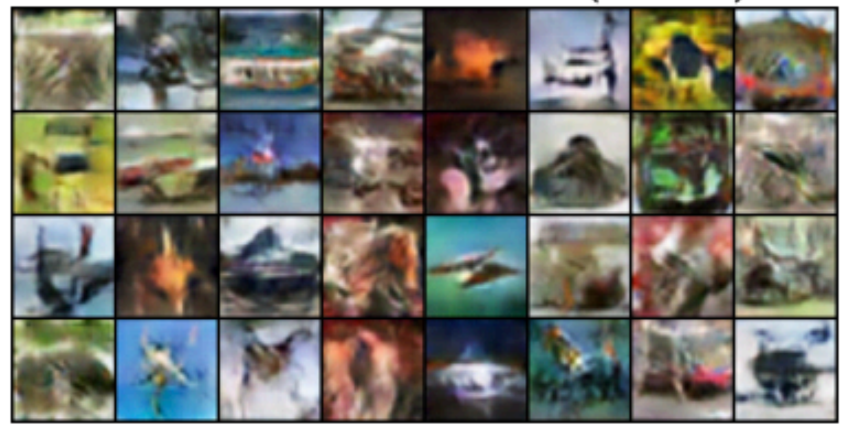
Dataset	k	m-OT(W_2^ϵ)	BoMb-OT(W_2^ϵ)
MNIST	1	28.12	28.12
	2	27.88	**27.53**
	4	27.60	**27.41**
	8	27.36	**27.10**
CIFAR10	1	78.34	78.34
	2	76.20	**74.25**
	4	76.01	**74.12**
	8	75.22	**73.33**
CelebA	1	54.16	54.16
	2	52.85	**51.53**
	4	52.56	**50.55**
	8	51.92	**49.63**

m-OT (W_2^ϵ)

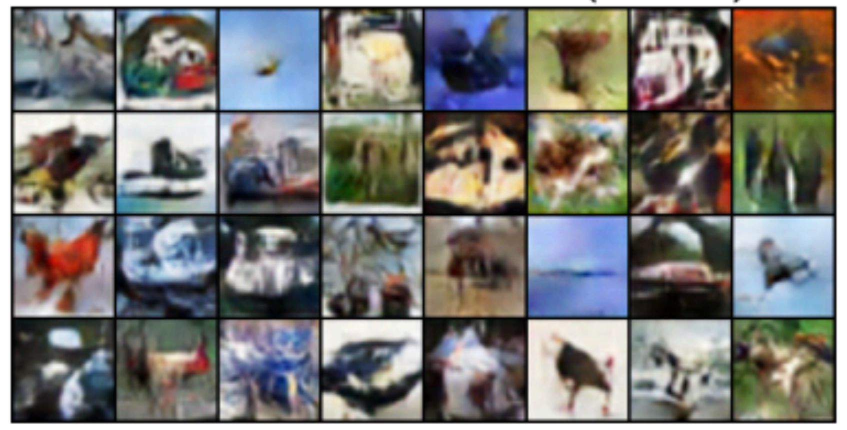
BoMb-OT $\lambda = 0$ (W_2^ϵ)

BoMb-OT $\lambda = 1$ (W_2^ϵ)

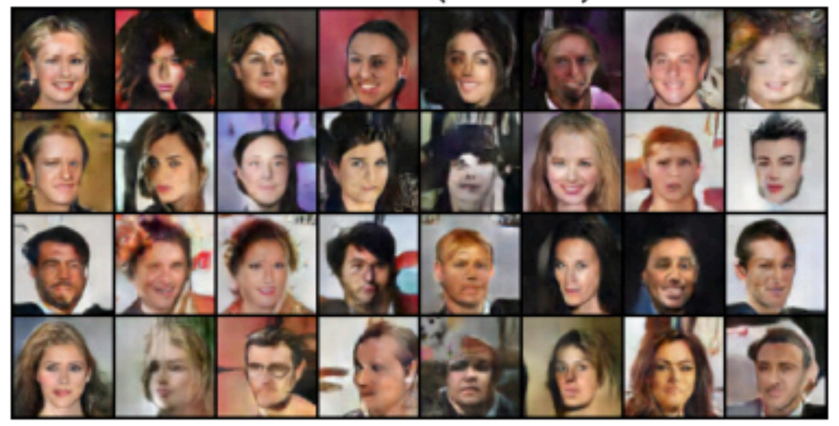
m-OT (W_2^ϵ)



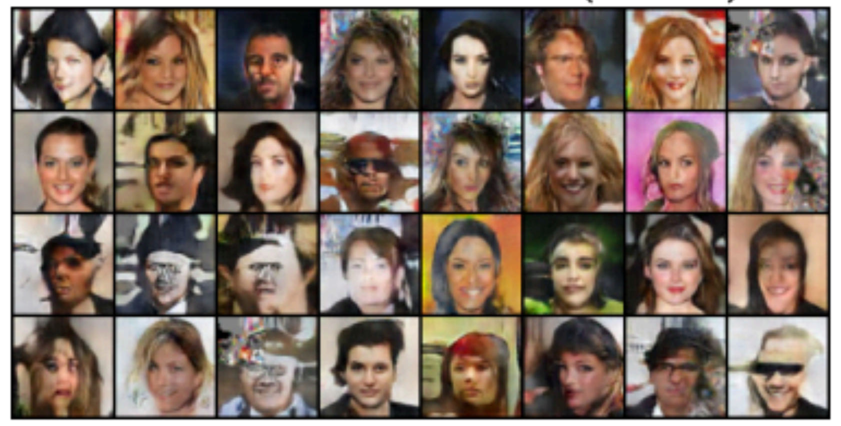
BoMb-OT $\lambda = 0$ (W_2^ϵ)



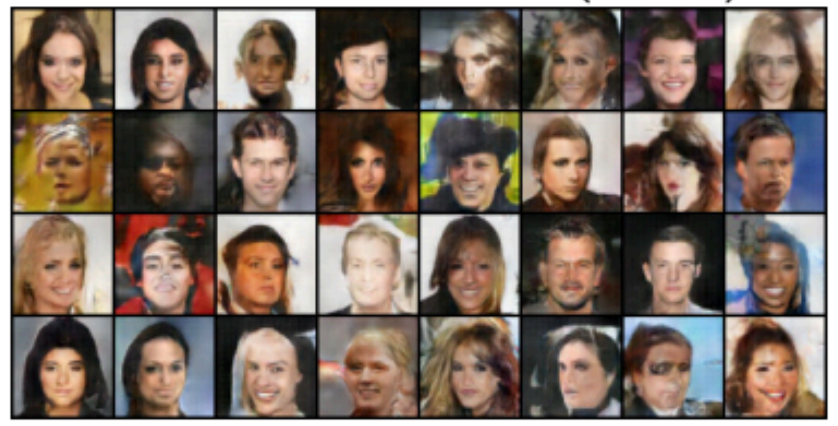
BoMb-OT $\lambda = 2$ (W_2^ϵ)



m-OT (W_2^ϵ)



BoMb-OT $\lambda = 0$ (W_2^ϵ)



BoMb-OT $\lambda = 1$ (W_2^ϵ)

Curse of Dimensionality of OT-GANs

Curse of Dimensionality of OT-GANs

- Another important issue of OT-GANs is curse of dimensionality
    - The required number of samples for OT-GANs to obtain good estimation of the underlying distribution of the data is exponential in the number of the dimension
    - Therefore, using OT-GANs for large-scale deep generative model can be expensive in terms of the sample size
- Solutions: We utilize sliced OT-GANs and their variants [31], [32], [33], [34]

[31] Khai Nguyen, Nhat Ho, Tung Pham, Hung Bui. *Distributional sliced-Wasserstein and applications to deep generative modeling*. ICLR, 2021

[32] Khai Nguyen, Nhat Ho, Tung Pham, Hung Bui. *Improving relational regularized autoencoders with spherical sliced fused Gromov Wasserstein*. ICLR, 2021

[33] Khai Nguyen, Nhat Ho. *Revisiting projected Wasserstein metric on images: from vectorization to convolution*. Arxiv Preprint, 2022

[34] Khai Nguyen, Nhat Ho. *Amortized projection optimization for sliced Wasserstein generative models*. Arxiv Preprint, 2022

Sliced Optimal Transport

- We first define sliced optimal transport, which is key to define sliced OT-GANs
- The sliced optimal transport (OT) between two probability distributions μ and ν is defined as follows:

$$\mathrm{SW}_p(\mu, \nu) := \left(\int_{\mathbb{S}^{d-1}} \mathrm{W}_p^p(\theta \sharp \mu, \theta \sharp \nu) d\theta \right)^{1/p},$$

where $\theta \sharp \mu$ is the push-forward probability measure of μ through the function $T_\theta : \mathbb{R}^d \to \mathbb{R}$ with $T_\theta(x) = \theta^\top x$;

$p \geq 1$ is the order of sliced optimal transport;

W_p is the p-th order Wasserstein metric

Properties of Sliced OT

There are three key properties of sliced optimal transport that make them appealing for large-scale applications:

- The sliced OT is a proper metric in the space of probability measures, namely, it satisfies the identity, symmetric, and triangle inequality properties
- The computational complexity of sliced OT between probability measures with at most n supports is $\mathcal{O}(n \log n)$, which is (much) faster than that of OT, which is $\mathcal{O}(n^2)$ (via entropic regularized approach)
- The sliced OT does not suffer from curse of dimensionality, namely, the required sample for the sliced OT to obtain good estimation of the underlying probability distribution does not scale exponentially with the dimension

Sliced-OT GANs

- Given the definition of sliced-OT, the sliced optimal transport GANs (Sliced-OT GANs) is:

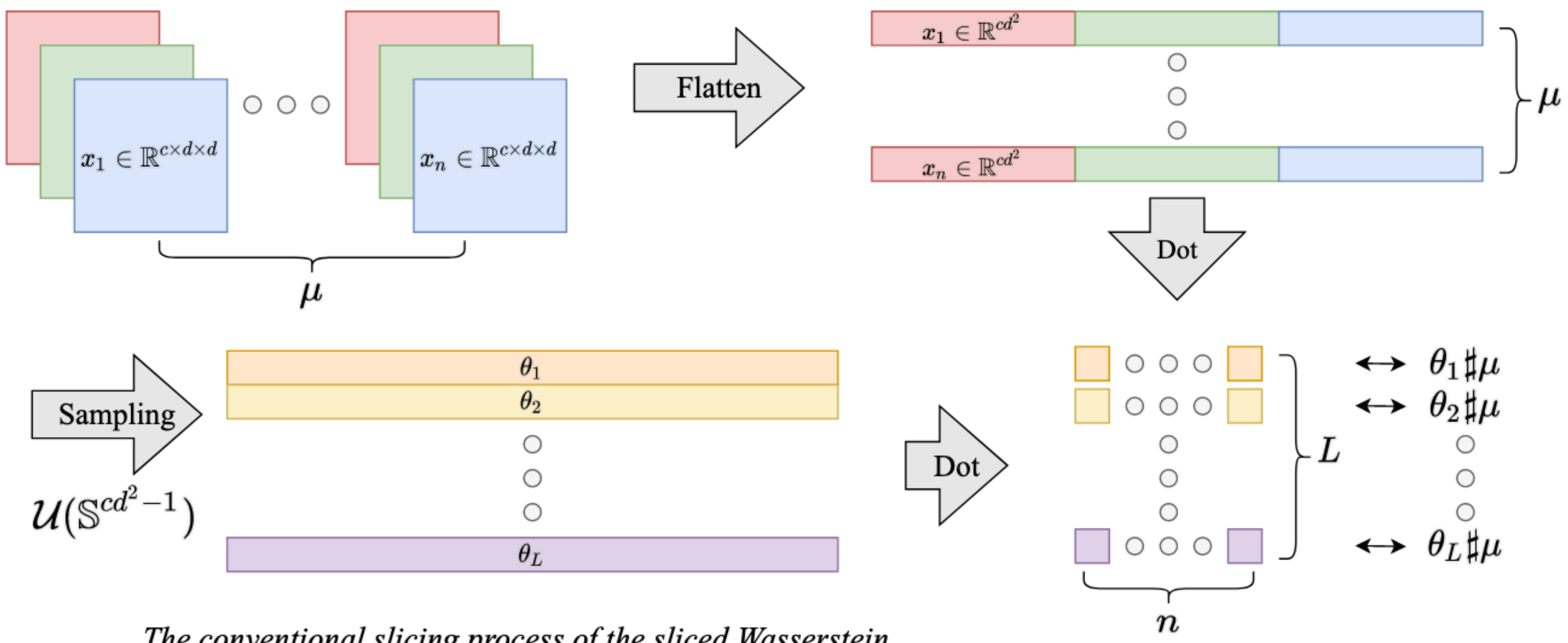
$$\min_{\phi} \mathsf{SW}_p(T_\phi(z), P),$$

where T_ϕ is some vector-value function parametrized by ϕ;

P is the true distribution of the data

- However, for generative models with images, that form of sliced-OT GANs means that we first vectorize images and then project them to one-dimensional space
    - The spatial structure of images is not captured efficiently by the vectorization step
    - Memory inefficiency since each slicing direction is a vector that has the same dimension as the images

Sliced-OT GANs



The conventional slicing process of the sliced Wasserstein

Figure 3: The conventional slicing process of sliced Wasserstein distance. The images $X_1, \ldots, X_n \in \mathbb{R}^{c\times d\times d}$ are first flattened into vectors in $\mathbb{R}^{cd^2}$ and then the Radon transform is applied to these vectors to lead to sliced Wasserstein (1) on images.

Convolution Sliced-OT GANs [33]

[33] Khai Nguyen, Nhat Ho. *Revisiting projected Wasserstein metric on images: from vectorization to convolution*. Arxiv Preprint, 2022

Convolution

- To efficiently capture the spatial structures and improve the memory efficiency of sliced OT, we utilize the convolution operators to the slicing process of sliced optimal transport

- The convolution operators had been demonstrated to be very efficient for images in Convolutional Neural Networks (CNNs)

Definition 1 *(Convolution) Given the number of channels* $c \geq 1$*, the dimension* $d \geq 1$*, the stride size* $s \geq 1$*, the dilation size* $b \geq 1$*, the size of kernel* $k \geq 1$*, the convolution of a tensor* $X \in \mathbb{R}^{c \times d \times d}$ *with a kernel size* $K \in \mathbb{R}^{c \times k \times k}$ *is* $X \overset{s,b}{*} K = Y$*,* $Y \in \mathbb{R}^{1 \times d' \times d'}$ *where* $d' = \frac{d - b(k-1) - 1}{s} + 1$*. For* $i = 1, \ldots, d'$ *and* $j = 1, \ldots, d'$*,* $Y_{1,i,j}$ *is defined as:* $Y_{1,i,j} = \sum_{h=1}^{c} \sum_{i'=0}^{k-1} \sum_{j'=0}^{k-1} X_{h, s(i-1)+bi'+1, s(j-1)+bj'+1} \cdot K_{h, i'+1, j'+1}$.

Convolution Slicer

Definition 2 *(Convolution Slicer) For* $N \geq 1$*, given a sequence of kernels* $K^{(1)} \in \mathbb{R}^{c^{(1)} \times d^{(1)} \times d^{(1)}}, \ldots, K^{(N)} \in \mathbb{R}^{c^{(N)} \times d^{(N)} \times d^{(N)}}$*, a* convolution slicer $\mathcal{S}(\cdot|K^{(1)}, \ldots, K^{(N)})$ *on* $\mathbb{R}^{c \times d \times d}$ *is a composition of* N *convolution functions with kernels* $K^{(1)}, \ldots, K^{(N)}$ *(with stride or dilation if needed) such that* $\mathcal{S}(X|K^{(1)}, \ldots, K^{(N)}) \in \mathbb{R} \quad \forall X \in \mathbb{R}^{c \times d \times d}$.

- There are three useful types of convolution slicers for images:
    - Convolution-base slicer: reduce the width and the height of the image by half after each convolution operator
    - Convolution-stride slicer: the size of its kernels does not depend on the width and the height of images as that of the convolution-base slicer
    - Convolution-dilation slicer: has bigger receptive field in each convolution operator than convolution-stride slicer

Convolution Sliced Optimal Transport

Definition 5 *For any* $p \geq 1$, *the* convolution sliced Wasserstein *(CSW) of order* $p > 0$ *between two given probability measures* $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^{c \times d \times d})$ *is given by:*

$$\text{CSW}_p(\mu, \nu) := \left(\mathbb{E} \left[W_p^p \left(\mathcal{S}(\cdot | K^{(1)}, \ldots, K^{(N)}) \sharp \mu, \mathcal{S}(\cdot | K^{(1)}, \ldots, K^{(N)}) \sharp \nu \right) \right] \right)^{\frac{1}{p}},$$

where the expectation is taken with respect to $K^{(1)} \sim \mathcal{U}(\mathcal{K}^{(1)}), \ldots, K^{(N)} \sim \mathcal{U}(\mathcal{K}^{(N)})$. *Here,* $\mathcal{S}(\cdot | K^{(1)}, \ldots, K^{(N)})$ *is a convolution slicer with* $K^{(l)} \in \mathbb{R}^{c^{(l)} \times k^{(l)} \times k^{(l)}}$ *for any* $l \in [N]$ *and* $\mathcal{U}(\mathcal{K}^{(l)})$ *is the uniform distribution with the realizations being in the set* $\mathcal{K}^{(l)}$ *which is defined as* $\mathcal{K}^{(l)} := \left\{ K^{(l)} \in \mathbb{R}^{c^{(l)} \times k^{(l)} \times k^{(l)}} \,|\, \sum_{h=1}^{c^{(l)}} \sum_{i'=1}^{k^{(l)}} \sum_{j'=1}^{k^{(l)}} K_{h,i',j'}^{(i)2} = 1 \right\}$, *namely, the set* $\mathcal{K}^{(l)}$ *consists of tensors* $K^{(l)}$ *whose squared* ℓ_2 *norm is 1.*

Convolution Sliced Optimal Transport

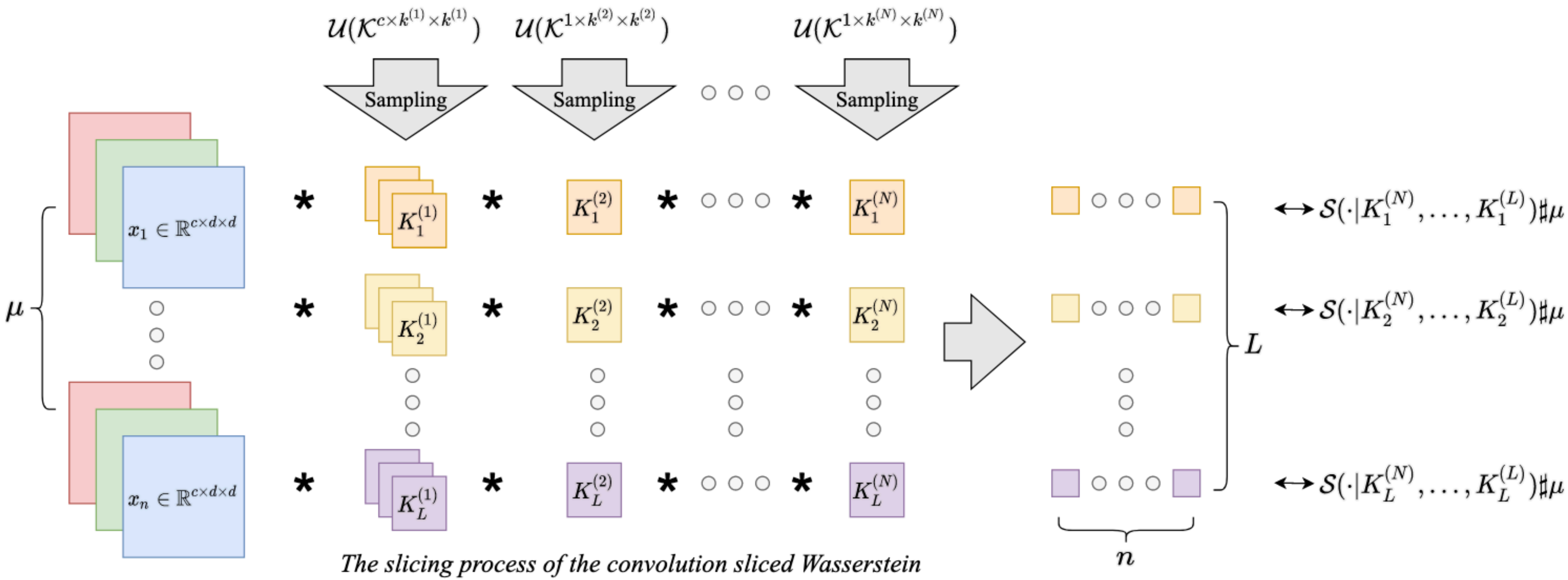


Figure 4: The convolution slicing process (using the convolution slicer). The images $X_1, \ldots, X_n \in \mathbb{R}^{c\times d\times d}$ are directly mapped to a scalar by a sequence of convolution functions which have kernels as random tensors. This slicing process leads to the convolution sliced Wasserstein on images.

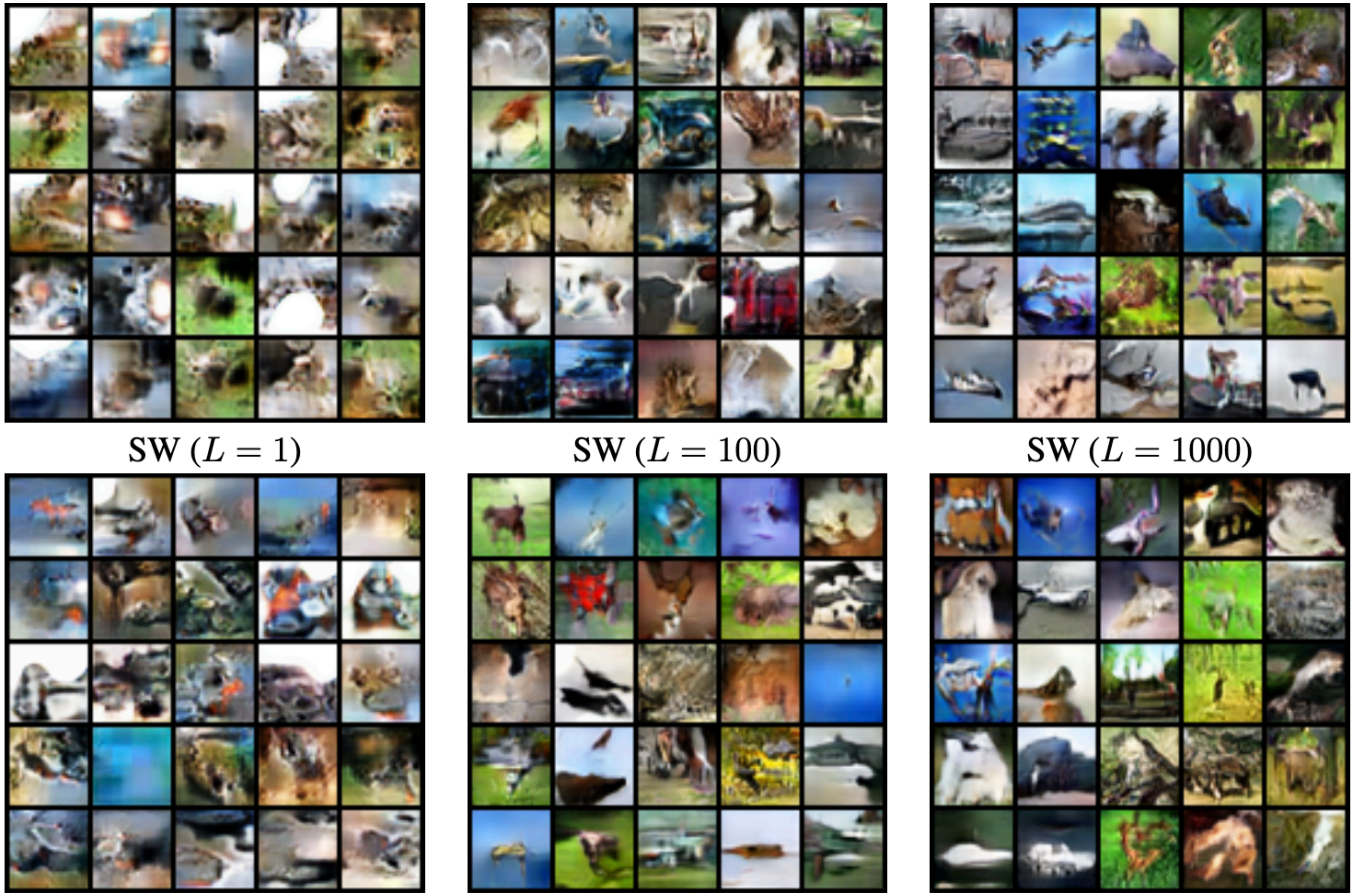
Experiments: Deep Generative Models

Table 1: Summary of FID and IS scores of methods on CIFAR10 (32x32), CelebA (64x64), STL10 (96x96), and CelebA-HQ (128x128).

Method	CIFAR10 (32x32)		CelebA (64x64)	STL10 (96x96)		CelebA-HQ (128x128)
	FID (↓)	IS (↑)	FID (↓)	FID (↓)	IS (↑)	FID (↓)
SW (L=1)	87.97	3.59	128.81	170.96	3.68	**275.44**
CSW-b (L=1)	84.38	4.28	85.83	173.33	**3.89**	315.91
CSW-s (L=1)	80.10	4.31	**66.52**	**168.93**	3.75	303.57
CSW-d (L=1)	**63.94**	**4.89**	89.37	212.61	2.48	321.06
SW (L=100)	53.67	5.74	20.08	100.35	8.14	51.80
CSW-b (L=100)	49.78	5.78	18.96	**91.75**	8.11	53.05
CSW-s (L=100)	**43.88**	**6.13**	**13.76**	97.08	**8.20**	**32.94**
CSW-d (L=100)	47.16	5.90	14.96	102.58	7.53	41.01
SW (L=1000)	43.11	6.09	14.92	84.78	9.06	28.19
CSW-b (L=1000)	43.17	6.07	14.75	86.98	9.11	29.69
CSW-s (L=1000)	**35.40**	**6.64**	**12.55**	**77.24**	9.31	**22.25**
CSW-d (L=1000)	41.34	6.33	13.24	83.36	**9.42**	25.93

L: the number of slices to approximate the integral (or equivalent expectation) in sliced and convolution sliced optimal transport;
b: base; s:slide; d: dilation.

Experiments: Deep Generative Models



SW ($L = 1$) | SW ($L = 100$) | SW ($L = 1000$)

CSW-s ($L = 1$) | CSW-s ($L = 100$) | CSW-s ($L = 1000$)

CIFAR10

Experiments: Deep Generative Models

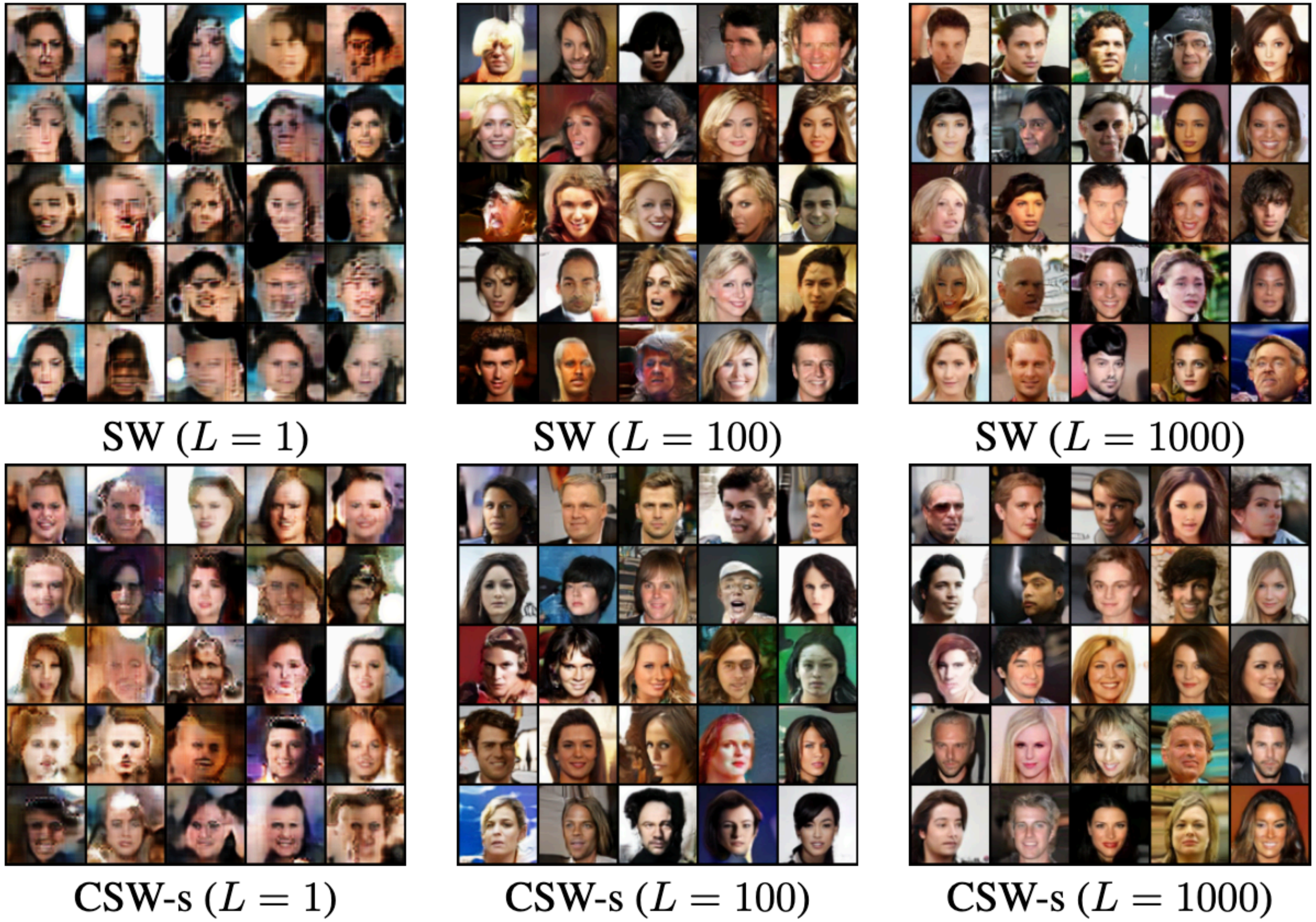
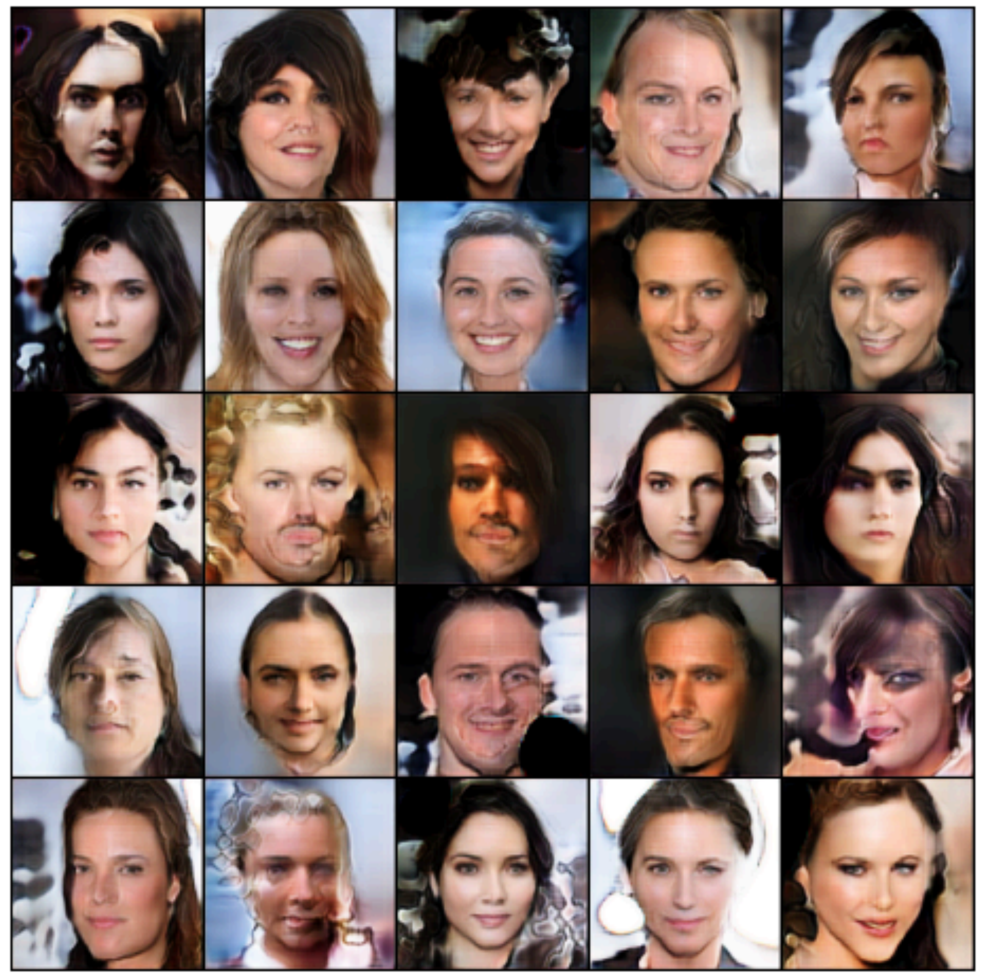


Figure 2: Random generated images of SW and CSW-s on CelebA.

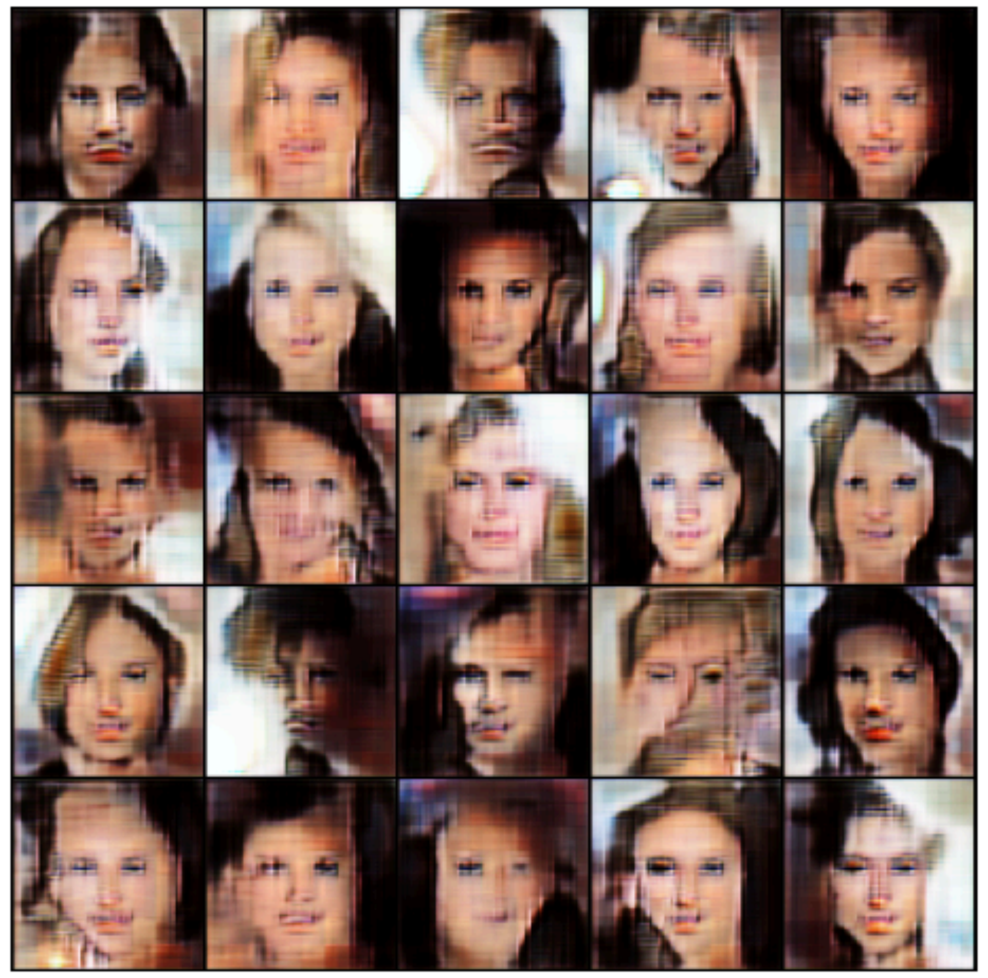
Experiments: Deep Generative Models

SW ($L = 1$)

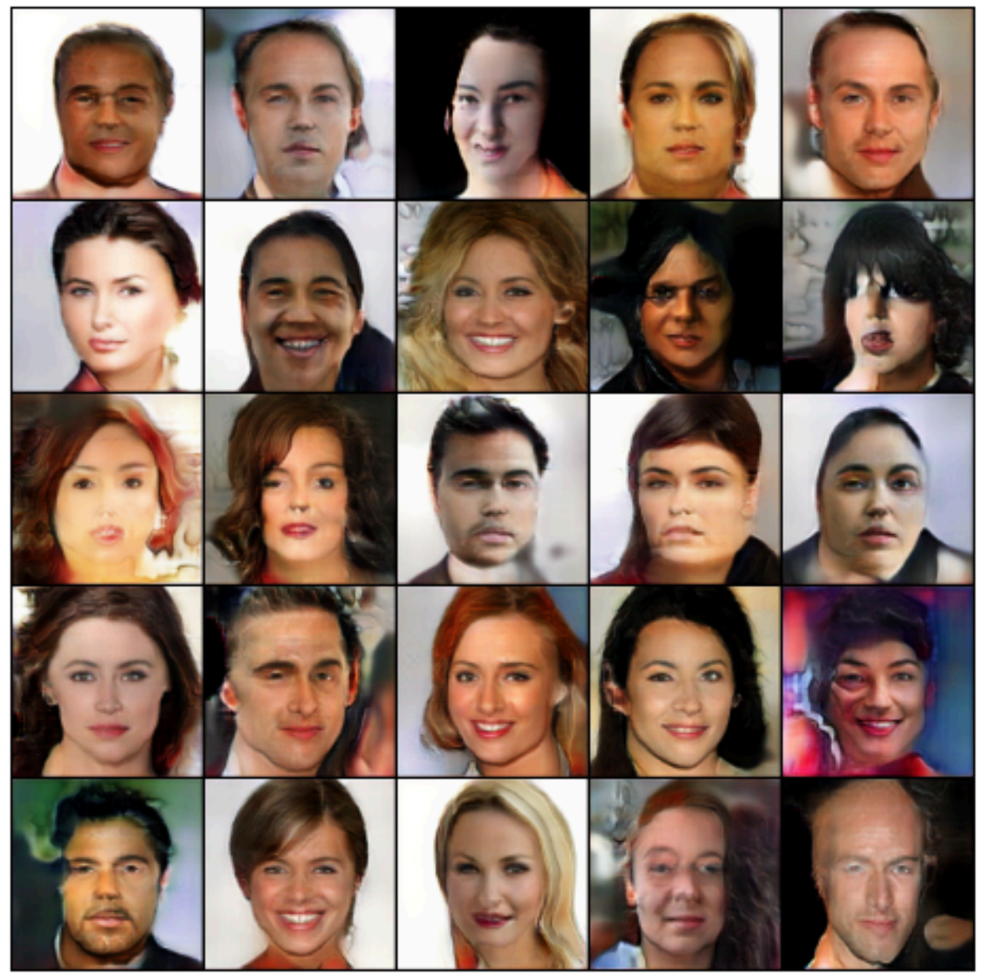


SW ($L = 100$)

SW ($L = 1000$)



CSW-s ($L = 1$)



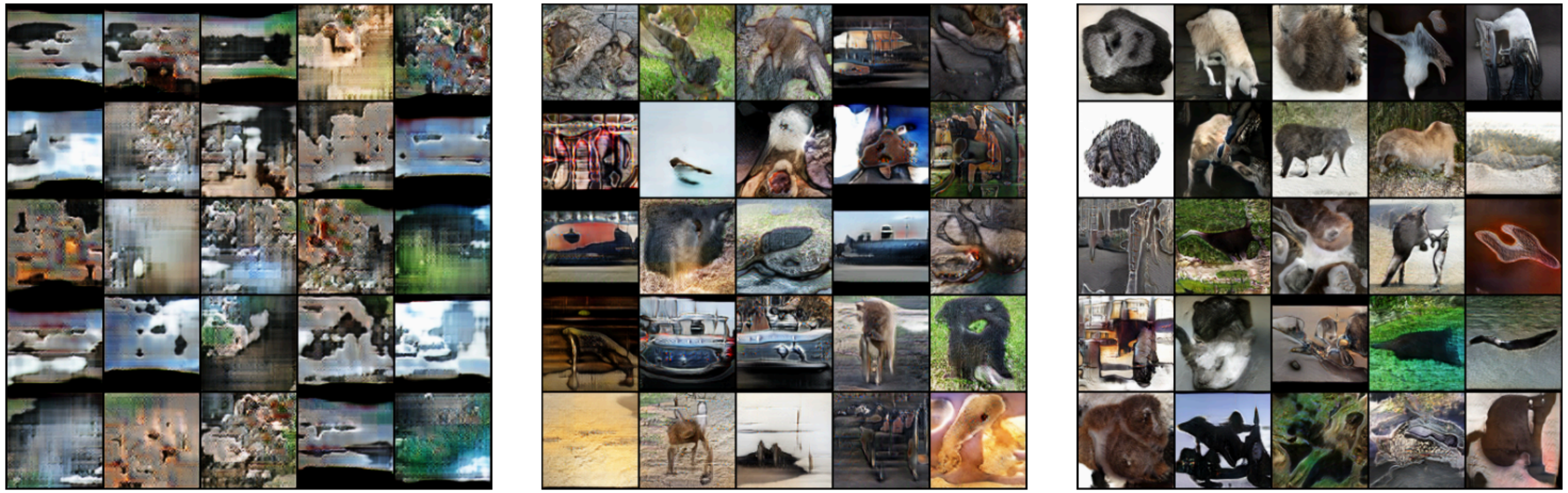
CSW-s ($L = 100$)

CSW-s ($L = 1000$)

CelebA-HQ.

Experiments: Deep Generative Models

SW ($L = 1$) SW ($L = 100$) SW ($L = 1000$)



CSW-s ($L = 1$) CSW-s ($L = 100$) CSW-s ($L = 1000$)

STL10.

Conclusion

- We have studied both the computational complexities of optimal transport as well as its applications to deep generative models

- There are several interesting open directions:

    - First direction: Improving further minibatch optimal transport in GANs and other deep learning applications

    - Second direction: Developing more efficient sliced optimal transport for other applications, such as language-models, etc.

    - Third direction: Exploring more computationally efficient ways to compute optimal transport

    - Fourth direction: Researching more important variants of optimal transport, such as unbalanced optimal transport, partial optimal transport, etc.

Thank You!

References

[1] Martin Arjovsky, Soumith Chintala, Léon Bottou. *Wasserstein Generative Adversarial Networks.* ICML, 2017

[2] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron C. Courville. *Improved Training of Wasserstein GANs.* NIPS, 2017

[3] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, Bernhard Scholkopf. *Wasserstein Auto-Encoders.* ICLR, 2018

[4] Nicolas Courty, Rémi Flamary, Devis Tuia, Alain Rakotomamonjy. *Optimal Transport for Domain Adaptation.* IEEE Transactions on Pattern Analysis and Artificial Intelligence (PAMI), 2017

[5] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, Nicolas Courty. *DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation*. ECCV, 2018

[6] Trung Nguyen, Hieu Pham, Tam Le, Tung Pham, Nhat Ho, Son Hua. *Point-set distances for learning representations of 3D point clouds*. ICCV, 2021

[7] Nhat Ho, Long Nguyen, Mikhail Yurochkin, Hung Bui, Viet Huynh, and Dinh Phung. *Multilevel clustering via Wasserstein means*. ICML, 2017

[8] Viet Huynh, Nhat Ho, Nhan Dam, Long Nguyen, Mikhail Yurochkin, Hung Bui, Dinh Phung. *On efficient multilevel clustering via Wasserstein distances*. Journal of Machine Learning Research, 2021

References

[9] Xing Han, Tongzheng Ren, Jing Hu, Joydeep Ghosh, Nhat Ho. *Efficient Forecasting of Large Scale Hierarchical Time Series via Multilevel Clustering.* Under review, NeurIPS, 2022

[10] Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, Lei Li. *Vocabulary Learning via Optimal Transport for Neural Machine Translation*. ACL, 2021

[11] Khang Le, Huy Nguyen, Quang Nguyen, Tung Pham, Hung Bui, Nhat Ho. *On robust optimal transport: Computational complexity and barycenter computation* . NeurIPS, 2021

[12] Nhat Ho, Tan Nguyen, Ankit Patel, Anima Anandkumar, Michael I. Jordan, Richard Baraniuk. *A Bayesian Perspective of Convolutional Neural Networks through a Deconvolutional Generative Model.* Under Revision, Journal of Machine Learning Research, 2021

[13] Long Nguyen. *Convergence of latent mixing measures in finite and infinite mixture models*. Annals of Statistics, 2013

[14] Nhat Ho, Long Nguyen. *Convergence rates of parameter estimation for some weakly identifiable finite mixtures*. Annals of Statistics, 2016

[15] Nhat Ho, Chiao-Yu Yang, Michael I. Jordan. *Convergence rates for Gaussian mixtures of experts*. Journal of Machine Learning Research, 2022 (Accepted Under Minor Revision)

[16] Rui Gao, Anton J Kleywegt. *Distributionally robust stochastic optimization with Wasserstein distance*. Arxiv preprint arXiv:1604.02199, 2016

[17] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, Soroosh Shafieezadeh-Abadeh. *Wasserstein distributionally robust optimization: Theory and applications in machine learning.* INFORMS Tutorials in Operations Research

References

[18] Matthew Thorpe. *Introduction to Optimal Transport* (https://www.math.cmu.edu/~mthorpe/OTNotes)

[19] Gabriel Peyré, Marco Cuturi. *Computational Optimal Transport: With Applications to Data Science*. Foundations and Trends® in Machine Learning, 2019

[20] Marco Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transport*. NIPS 2013

[21] Jason Altschuler, Jonathan Weed, Philippe Rigollet. *Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration.* NIPS, 2017

[22] Pavel Dvurechensky, Alexander Gasnikov, Alexey Kroshnin. *Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn's Algorithm.* ICML, 2018

[23] T. Lin, N. Ho, M. I. Jordan.On efficient optimal transport: an analysis of greedy and accelerated mirror descent algorithms. ICML, 2019

[24] T. Lin, N. Ho, M. I. Jordan. On the efficiency of entropic regularized algorithms for optimal transport. Journal of Machine Learning Research (JMLR), 2022

References

[25] Diederik P Kingma, Max Welling. *Auto-Encoding Variational Bayes*. ICLR, 2014

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. *Generative Adversarial Networks.* NIPS, 2014

[27] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein Generative Adversarial Networks. ICML, 2017

[28] Khai Nguyen, Dang Nguyen, Tung Pham, Nhat Ho. *Improving minibatch optimal transport via partial transportation*. ICML, 2022

[29] Khai Nguyen, Dang Nguyen, Quoc Nguyen, Tung Pham, Dinh Phung, Hung Bui, Trung Le, Nhat Ho. *On transportation of mini-batches: A hierarchical approach*. ICML, 2022

[30] Kilian Fatras, Thibault Sejourne, Rémi Flamary, and Nicolas Courty. *Unbalanced minibatch optimal transport; applications to domain adaptation.* ICML, 2021

References

[31] Khai Nguyen, Nhat Ho, Tung Pham, Hung Bui. *Distributional sliced-Wasserstein and applications to deep generative modeling*. ICLR, 2021

[32] Khai Nguyen, Nhat Ho, Tung Pham, Hung Bui. *Improving relational regularized autoencoders with spherical sliced fused Gromov Wasserstein*. ICLR, 2021

[33] Khai Nguyen, Nhat Ho. *Revisiting projected Wasserstein metric on images: from vectorization to convolution*. Arxiv Preprint, 2022

[34] Khai Nguyen, Nhat Ho. *Amortized projection optimization for sliced Wasserstein generative models*. Arxiv Preprint, 2022