Optimal Transport-Driven Machine Learning: Techniques and Applications

Hongteng Xu[1], Dixin Luo[2], Minjie Cheng[1]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Computer Science and Technology, Beijing Institute of Technology

Jan. 21, 2026

Part 1 **Computational Optimal Transport (Hongteng Xu)**

- Preliminaries and basic concepts
- Typical computation methods

Part 2 **Representation Learning Driven by OT (Dixin Luo)**

- OT-based multi-modal learning
- Monge gap and its Gromovization for information bottleneck

Part 3 **Neural Network Design Driven by OT (Minjie Cheng)**

- OT-based Transformer
- OT-based graph neural network

Part 4 **Recent Progress in Generative Modeling (Hongteng Xu)**

- OT-based flow matching
- Applications of optimal acceleration transport

Part 1 **Computational Optimal Transport (Hongteng Xu)**

- Preliminaries and basic concepts
- Typical computation methods

Part 2 **Representation Learning Driven by OT (Dixin Luo)**

- OT-based multi-modal learning
- Monge gap and its Gromovization for information bottleneck

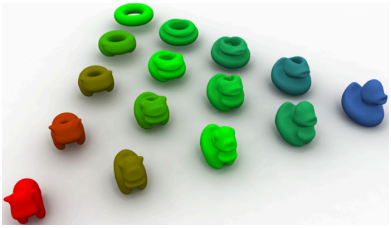
Part 3 **Neural Network Design Driven by OT (Minjie Cheng)**

- OT-based Transformer
- OT-based graph neural network

Part 4 **Recent Progress in Generative Modeling (Hongteng Xu)**

- OT-based flow matching
- Applications of optimal acceleration transport

Distribution Comparison and Transformation: Key Learning Tasks



Shape Matching and Interpolation

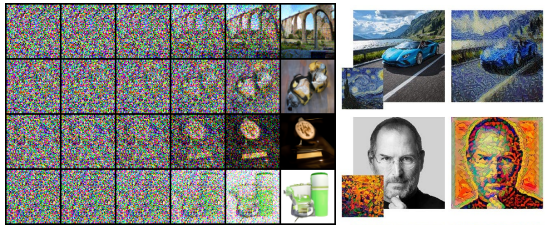


Image (Conditional) Generation

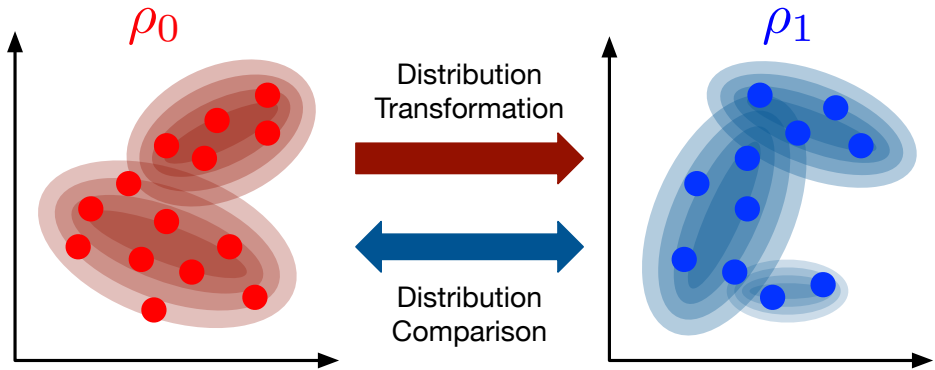
I want to have a pet, but I don't know whether to get a cat or a dog. Do you have any suggestions?

It	really	depends	on	your	preferences	,	lifestyle	,	and	the	type	of	pet	you	want									
.	If	you	want	an	independent	pet	and	have	a	busy	schedule	,	a	cat	is	a	good							
choice	.	If	you	prefer	a	companion	and	have	a	more	active	lifestyle	,	a	dog									
might	be	a	better	fit	.	Have	you	considered	your	options	?													

Textual Data Comparison and Generation

Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. ToG, 2015.
Flow Matching for Generative Modeling. ICLR, 2023.
Large language diffusion models. NeurIPS, 2025.

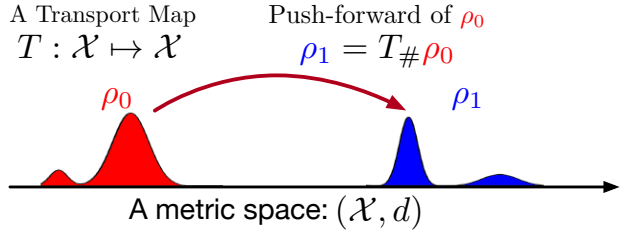
Distribution Comparison and Transformation: Key Learning Tasks



- Data Clustering, Domain Adaptation, Metric Learning, Representation Learning, Generative Modeling, ...
- Optimal transport theory provides solid and effective solutions to distribution comparison and transformation.

Origin: The Monge-form of The Optimal Transport Problem

Gaspard Monge (1746-1818)

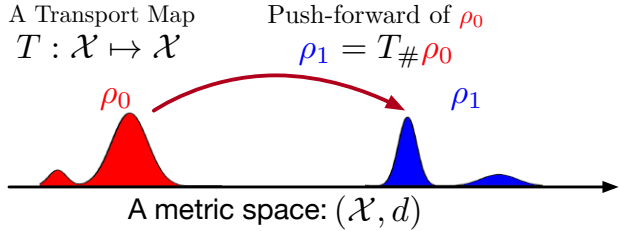


The Monge-form of OT problem proposed in 1781.

Key question: How to find a map that minimizes the transport cost?

Origin: The Monge-form of The Optimal Transport Problem

Gaspard Monge (1746-1818)



The Monge-form of OT problem proposed in 1781.

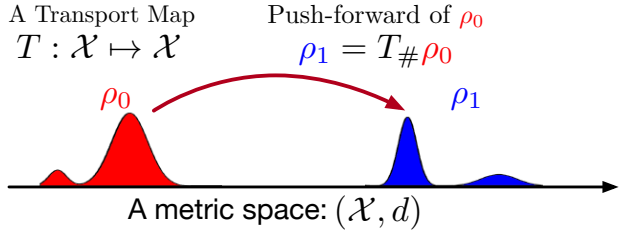
Key question: How to find a map that minimizes the transport cost?

- The p-order Monge problem:

$$\mathcal{M}_p(\rho_0, \rho_1) := \left(\inf_T \int_{x\in\mathcal{X}} \underbrace{d^p(x, T(x))}_{\text{cost per sample}} \mathrm{d}\mu(x)\right)^{1/p}, \quad s.t. \quad \underbrace{T_{\#}\rho_0 = \rho_1}_{\text{measure preserving}} \tag{1}$$

Origin: The Monge-form of The Optimal Transport Problem

Gaspard Monge (1746-1818)



The Monge-form of OT problem proposed in 1781.

Key question: How to find a map that minimizes the transport cost?

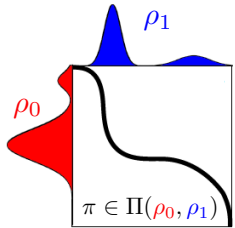
- The p-order Monge problem:

$$\mathcal{M}_p(\rho_0, \rho_1) := \Big(\inf_T \int_{x\in\mathcal{X}} \underbrace{d^p(x, T(x))}_{\text{cost per sample}} \mathrm{d}\mu(x)\Big)^{1/p}, \quad s.t. \quad \underbrace{T_{\#}\rho_0 = \rho_1}_{\text{measure preserving}} \tag{1}$$

- Notably, the minimizer of (1) may not exist, e.g., ρ_0 is a Dirac measure while ρ_1 is not.

From Transport Map to Transport Plan: The Kantorovich-form of OT

Leonid Kantorovich (1912-1986)

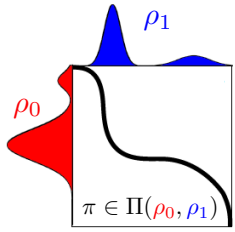


The Kantorovich-form of OT proposed in 1939

- Find a **transport plan/coupling** to minimize the expected cost.

From Transport Map to Transport Plan: The Kantorovich-form of OT

Leonid Kantorovich (1912-1986)



The Kantorovich-form of OT proposed in 1939

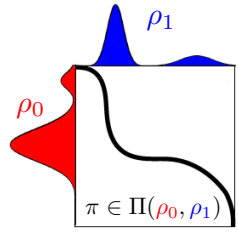
- Find a **transport plan/coupling** to minimize the expected cost.

$$\mathcal{W}_p(\rho_0, \rho_1) := \Big(\inf_{\pi} \underbrace{\int_{(x,y)\in\mathcal{X}^2} d^p(x,y)\pi(x,y)\mathrm{d}x\mathrm{d}y}_{\mathbb{E}_{x,y\sim\pi}[d^p(x,y)]}\Big)^{1/p} \tag{2}$$

$$s.t.\ \pi \in \Pi(\rho_0, \rho_1) = \Big\{\pi \geq 0 \Big| \int_{\mathcal{X}} \pi(x,\cdot)\mathrm{d}x = \rho_1, \int_{\mathcal{X}} \pi(\cdot,y)\mathrm{d}y = \rho_0\Big\}.$$

From Transport Map to Transport Plan: The Kantorovich-form of OT

Leonid Kantorovich (1912-1986)



The Kantorovich-form of OT proposed in 1939

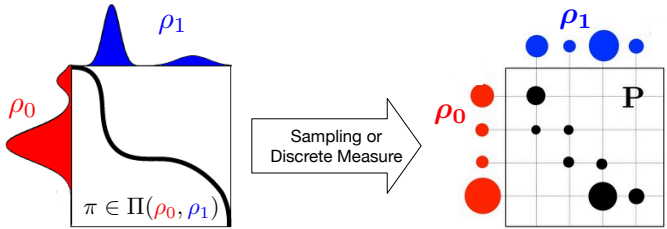
- Find a **transport plan/coupling** to minimize the expected cost.

$$\mathcal{W}_p(\rho_0, \rho_1) := \Big(\inf_{\pi} \underbrace{\int_{(x,y)\in\mathcal{X}^2} d^p(x,y)\pi(x,y)\mathrm{d}x\mathrm{d}y}_{\mathbb{E}_{x,y\sim\pi}[d^p(x,y)]}\Big)^{1/p} \tag{2}$$

$$s.t.\ \pi \in \Pi(\rho_0, \rho_1) = \Big\{\pi \geq 0 \Big| \int_{\mathcal{X}} \pi(x,\cdot)\mathrm{d}x = \rho_1, \int_{\mathcal{X}} \pi(\cdot,y)\mathrm{d}y = \rho_0\Big\}.$$

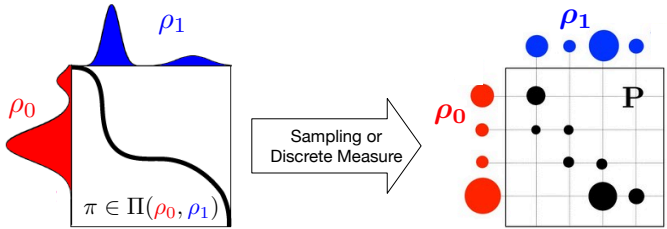
- When $d(x,y) = \|x-y\|_p$, $\mathcal{W}_p$ is p-**order Wasserstein distance**.

From Transport Map to Transport Plan: The Kantorovich-form of OT



Given $\boldsymbol{X} = \{x_m\}_{m=1}^{M}$, $\rho_0 = \sum_{m=1}^{M} \rho_{0,m} \delta_{x_m}$ and $\boldsymbol{Y} = \{y_n\}_{n=1}^{N}$, $\rho_1 = \sum_{n=1}^{N} \rho_{1,n} \delta_{y_n}$,

From Transport Map to Transport Plan: The Kantorovich-form of OT

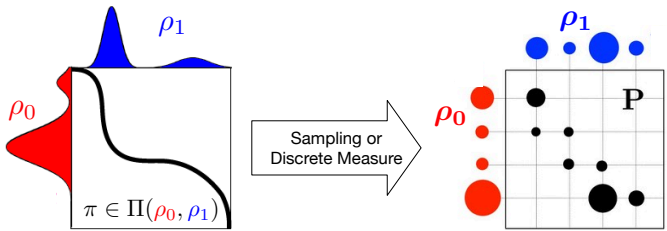


Given $\boldsymbol{X} = \{x_m\}_{m=1}^{M}$, $\rho_0 = \sum_{m=1}^{M} \rho_{0,m}\delta_{x_m}$ and $\boldsymbol{Y} = \{y_n\}_{n=1}^{N}$, $\rho_1 = \sum_{n=1}^{N} \rho_{1,n}\delta_{y_n}$,

$$\mathcal{W}_p(\boldsymbol{X}, \boldsymbol{Y}) := \Big(\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \sum_{m=1}^{M}\sum_{n=1}^{N} d^p(x_m, y_n)p_{mn}\Big)^{1/p} = \Big(\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle\Big)^{1/p}, \quad (3)$$

where $\boldsymbol{D} = [d^p(x_m, y_n)]$, $\boldsymbol{P} = [p_{mn}]$, $\Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \{\boldsymbol{P} > \mathbf{0} | \boldsymbol{P}\mathbf{1}_N = \boldsymbol{\rho}_0, \boldsymbol{P}^\top \mathbf{1}_M = \boldsymbol{\rho}_1\}$.

From Transport Map to Transport Plan: The Kantorovich-form of OT



Given $\boldsymbol{X} = \{x_m\}_{m=1}^M$, $\rho_0 = \sum_{m=1}^M \rho_{0,m}\delta_{x_m}$ and $\boldsymbol{Y} = \{y_n\}_{n=1}^N$, $\rho_1 = \sum_{n=1}^N \rho_{1,n}\delta_{y_n}$,

$$\mathcal{W}_p(\boldsymbol{X}, \boldsymbol{Y}) := \Big(\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \sum_{m=1}^M \sum_{n=1}^N d^p(x_m, y_n) p_{mn}\Big)^{1/p} = \Big(\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle\Big)^{1/p}, \quad (3)$$

where $\boldsymbol{D} = [d^p(x_m, y_n)]$, $\boldsymbol{P} = [p_{mn}]$, $\Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \{\boldsymbol{P} > \boldsymbol{0} | \boldsymbol{P}\mathbf{1}_N = \boldsymbol{\rho}_0, \boldsymbol{P}^\top \mathbf{1}_M = \boldsymbol{\rho}_1\}$.

- Applying the transport plan $\pi/\boldsymbol{P}$, we allow each sample $x \sim \rho_0$ to be split and mapped to multiple locations.
- If the optimal T^* exists, it determines an OT plan $\pi^*/\boldsymbol{P}^*$, so $\mathcal{W}_p \leq \mathcal{M}_p$.

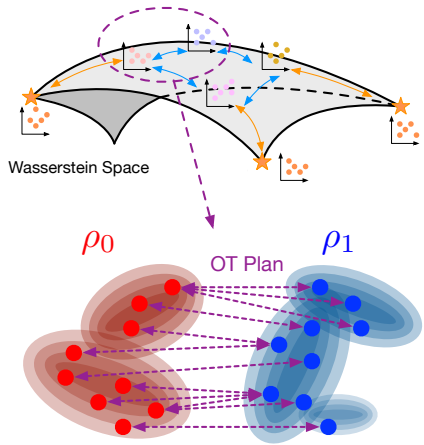
Advantages of Optimal Transport

A valid metric for probability measures

- $(\mathcal{P}(\mathcal{X}), \mathcal{W})$ is a metric space of probability measures defined in $\mathcal{X}$ (i.e., Wasserstein space).
- Apply to distribution comparison, fitting, and interpolation

OT plan indicates sample pairs

- Apply to point cloud/shape/graph matching



Computational Bottlenecks of Optimal Transport and Possible Solutions

- A constrained linear programming problem:

$$\mathcal{W}_p^p(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P} \rangle, \tag{4}$$

  Lead to $\mathcal{O}(N^3)$ complexity.

Computational Bottlenecks of Optimal Transport and Possible Solutions

- A constrained linear programming problem:

$$\mathcal{W}_p^p(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P} \rangle, \tag{4}$$

  Lead to $\mathcal{O}(N^3)$ complexity.

- Solution 1: Develop efficient optimization algorithms and acceleration methods
  - Sinkhorn-scaling
  - Proximal point
  - Bregman ADMM

Computational Bottlenecks of Optimal Transport and Possible Solutions

- A constrained linear programming problem:

$$\mathcal{W}_p^p(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P} \rangle, \tag{4}$$

  Lead to $\mathcal{O}(N^3)$ complexity.

- Solution 1: Develop efficient optimization algorithms and acceleration methods
  - Sinkhorn-scaling
  - Proximal point
  - Bregman ADMM
- Solution 2: Apply structured/stochastic OT plan
  - Stochastic optimization
  - Sinkhorn-scaling with importance sparsification

Computational Bottlenecks of Optimal Transport and Possible Solutions

- A constrained linear programming problem:

$$\mathcal{W}_p^p(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P} \rangle, \tag{4}$$

  Lead to $\mathcal{O}(N^3)$ complexity.

- Solution 1: Develop efficient optimization algorithms and acceleration methods
  - Sinkhorn-scaling
  - Proximal point
  - Bregman ADMM
- Solution 2: Apply structured/stochastic OT plan
  - Stochastic optimization
  - Sinkhorn-scaling with importance sparsification
- Solution 3: Explore efficient surrogates of OT distance
  - Sliced Wasserstein (SW) distance
  - Hilbert curve projection (HCP) distance

Sinkhorn-scaling Algorithm for Entropic OT

Sinkhorn Distance (Entropic OT): Improve the smoothness of OT problem

$$\mathcal{W}_{p,\epsilon}^{p} := \min_{\boldsymbol{T}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle - \epsilon \underbrace{\mathrm{H}(\boldsymbol{P})}_{\text{Entropy}}, \quad \text{where } \mathrm{H}(\boldsymbol{P}) = -\langle \log \boldsymbol{P} - \mathbf{1}_{M\times N}, \boldsymbol{P}\rangle. \tag{5}$$

Sinkhorn distances: Lightspeed computation of optimal transport. NeurIPS, 2013.

Sinkhorn-scaling Algorithm for Entropic OT

Sinkhorn Distance (Entropic OT): Improve the smoothness of OT problem

$$\mathcal{W}_{p,\epsilon}^{p} := \min_{\boldsymbol{T}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle - \epsilon \underbrace{\mathsf{H}(\boldsymbol{P})}_{\text{Entropy}}, \quad \text{where } \mathsf{H}(\boldsymbol{P}) = -\langle \log \boldsymbol{P} - \mathbf{1}_{M\times N}, \boldsymbol{P}\rangle. \tag{5}$$

Sinkhorn distances: Lightspeed computation of optimal transport. NeurIPS, 2013.

The Lagrangian form of EOT is

$$\max_{\boldsymbol{a}\in\mathbb{R}^M,\ \boldsymbol{b}\in\mathbb{R}^N} \min_{\boldsymbol{P}} \langle \boldsymbol{D}, \boldsymbol{P}\rangle - \epsilon H(\boldsymbol{P}) + \langle \boldsymbol{a}, \boldsymbol{P}\mathbf{1}_N - \boldsymbol{\rho}_0\rangle + \langle \boldsymbol{b}, \boldsymbol{P}^\top \mathbf{1}_M - \boldsymbol{\rho}_1\rangle. \tag{6}$$

Sinkhorn-scaling Algorithm for Entropic OT

Sinkhorn Distance (Entropic OT): Improve the smoothness of OT problem

$$\mathcal{W}_{p,\epsilon}^{p} := \min_{\boldsymbol{T}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle - \epsilon \underbrace{\mathsf{H}(\boldsymbol{P})}_{\text{Entropy}}, \quad \text{where } \mathsf{H}(\boldsymbol{P}) = -\langle \log \boldsymbol{P} - \mathbf{1}_{M\times N}, \boldsymbol{P}\rangle. \tag{5}$$

Sinkhorn distances: Lightspeed computation of optimal transport. NeurIPS, 2013.

The Lagrangian form of EOT is

$$\max_{\boldsymbol{a}\in\mathbb{R}^M,\ \boldsymbol{b}\in\mathbb{R}^N} \min_{\boldsymbol{P}} \langle \boldsymbol{D}, \boldsymbol{P}\rangle - \epsilon H(\boldsymbol{P}) + \langle \boldsymbol{a}, \boldsymbol{P}\mathbf{1}_N - \boldsymbol{\rho}_0\rangle + \langle \boldsymbol{b}, \boldsymbol{P}^\top \mathbf{1}_M - \boldsymbol{\rho}_1\rangle. \tag{6}$$

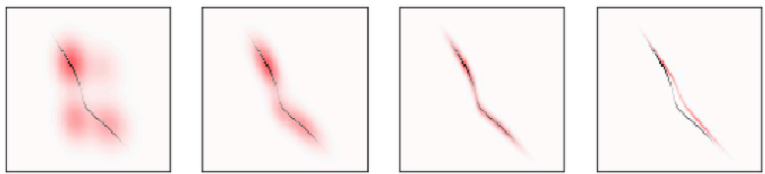
Sinkhorn-Knopp algorithm:

1. Set a kernel matrix $\boldsymbol{\Phi} = \exp(-\frac{\boldsymbol{D}}{\epsilon})$ and a dual variable $\boldsymbol{a} = \boldsymbol{\rho}_0$.
2. **Sinkhorn iteration:** Repeat $\boldsymbol{b} \leftarrow \frac{\boldsymbol{\rho}_1}{\boldsymbol{\Phi}^\top \boldsymbol{a}}$, then $\boldsymbol{a} \leftarrow \frac{\boldsymbol{\rho}_0}{\boldsymbol{\Phi}\boldsymbol{b}}$ until convergence.
3. $\boldsymbol{P}^* \leftarrow \boldsymbol{\Phi} \odot (\boldsymbol{a}\boldsymbol{b}^\top)$.

Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics, 1967.

Drawbacks of Sinkhorn-scaling in OT Problem

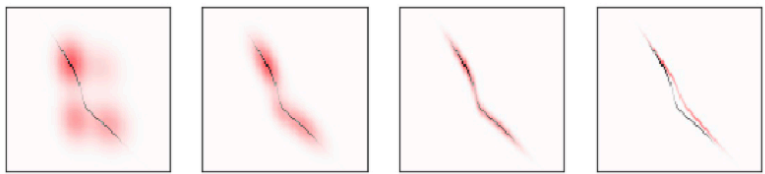
- EOT is sensitive to ϵ
  - A large ϵ leads to over-smoothed OT plan
  - A small ϵ causes numerical instability



Sinkhorn $\epsilon = 0.1$ | Sinkhorn $\epsilon = 0.01$ | Sinkhorn $\epsilon = 0.001$ | Sinkhorn $\epsilon = 0.0001$

Drawbacks of Sinkhorn-scaling in OT Problem

- EOT is sensitive to ϵ
  - A large ϵ leads to over-smoothed OT plan
  - A small ϵ causes numerical instability



Sinkhorn $\epsilon = 0.1$ | Sinkhorn $\epsilon = 0.01$ | Sinkhorn $\epsilon = 0.001$ | Sinkhorn $\epsilon = 0.0001$

- The (explicit) entropic regularizer might be unnecessary
- Solve the "exact" OT problem via a Sinkhorn-like algorithm.

Proximal Point Algorithm for "Exact" OT

1. Initialize $\boldsymbol{P}^{(0)} \leftarrow \boldsymbol{\rho}_0 \boldsymbol{\rho}_1^\top$
2. In the k-th iteration, consider the penalty between the optimal transport and its previous approximation

$$\min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P} \rangle + \beta \underbrace{\mathsf{KL}(\boldsymbol{P} \| \boldsymbol{P}^{(k)})}_{\text{Proximal term}} \Rightarrow \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \underbrace{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}_{:= \beta \log \boldsymbol{\Phi}^{(k)}}, \boldsymbol{P} \rangle - \epsilon \mathsf{H}(\boldsymbol{P}). \tag{7}$$

Proximal Point Algorithm for "Exact" OT

1. Initialize $\boldsymbol{P}^{(0)} \leftarrow \boldsymbol{\rho}_0 \boldsymbol{\rho}_1^\top$
2. In the k-th iteration, consider the penalty between the optimal transport and its previous approximation

$$\min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P} \rangle + \beta \underbrace{\mathsf{KL}(\boldsymbol{P} \| \boldsymbol{P}^{(k)})}_{\text{Proximal term}} \Rightarrow \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \underbrace{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}_{:= \beta \log \boldsymbol{\Phi}^{(k)}}, \boldsymbol{P} \rangle - \epsilon \mathsf{H}(\boldsymbol{P}). \tag{7}$$

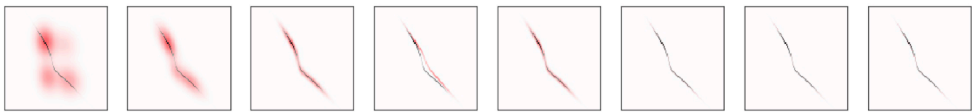
3. Apply the Sinkhorn iterations to obtain $\boldsymbol{P}^{(k+1)} = \boldsymbol{\Phi}^{(k)} \odot (\boldsymbol{a}^{(k)} (\boldsymbol{b}^{(k)})^\top)$.

Proximal Point Algorithm for "Exact" OT

1. Initialize $\boldsymbol{P}^{(0)} \leftarrow \boldsymbol{\rho}_0 \boldsymbol{\rho}_1^\top$
2. In the k-th iteration, consider the penalty between the optimal transport and its previous approximation

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle + \beta \underbrace{\mathsf{KL}(\boldsymbol{P}\|\boldsymbol{P}^{(k)})}_{\text{Proximal term}} \Rightarrow \min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \underbrace{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}_{:=\beta \log \boldsymbol{\Phi}^{(k)}}, \boldsymbol{P}\rangle - \epsilon \mathsf{H}(\boldsymbol{P}). \tag{7}$$

3. Apply the Sinkhorn iterations to obtain $\boldsymbol{P}^{(k+1)} = \boldsymbol{\Phi}^{(k)} \odot (\boldsymbol{a}^{(k)} (\boldsymbol{b}^{(k)})^\top)$.



A fast proximal point method for computing exact Wasserstein distance. UAI, 2020.

Proximal Point Algorithm = Adaptive Sinkhorn-scaling

- In the k-th iteration, denote $\boldsymbol{a}^{(k)}(\boldsymbol{b}^{(k)})^\top$ as $\boldsymbol{\Delta}^{(k)}$.
- According to the algorithm, we have $\boldsymbol{P}^{(k)} = \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)}$.

Proximal Point Algorithm = Adaptive Sinkhorn-scaling

- In the k-th iteration, denote $\boldsymbol{a}^{(k)}(\boldsymbol{b}^{(k)})^\top$ as $\boldsymbol{\Delta}^{(k)}$.
- According to the algorithm, we have $\boldsymbol{P}^{(k)} = \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)}$.

$$\boldsymbol{\Phi}^{(k)} = \exp\Big(-\frac{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}{\beta}\Big)$$

Proximal Point Algorithm = Adaptive Sinkhorn-scaling

- In the k-th iteration, denote $\boldsymbol{a}^{(k)}(\boldsymbol{b}^{(k)})^\top$ as $\boldsymbol{\Delta}^{(k)}$.
- According to the algorithm, we have $\boldsymbol{P}^{(k)} = \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)}$.

$$\boldsymbol{\Phi}^{(k)} = \exp\Big(-\frac{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}{\beta}\Big) = \exp\Big(-\frac{\boldsymbol{D}}{\beta}\Big) \odot \boldsymbol{P}^{(k)}$$

Proximal Point Algorithm = Adaptive Sinkhorn-scaling

- In the k-th iteration, denote $\boldsymbol{a}^{(k)}(\boldsymbol{b}^{(k)})^\top$ as $\boldsymbol{\Delta}^{(k)}$.
- According to the algorithm, we have $\boldsymbol{P}^{(k)} = \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)}$.

$$\begin{aligned}\boldsymbol{\Phi}^{(k)} &= \exp\Big(-\frac{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}{\beta}\Big) = \exp\Big(-\frac{\boldsymbol{D}}{\beta}\Big) \odot \boldsymbol{P}^{(k)} \\ &= \exp\Big(-\frac{\boldsymbol{D}}{\beta}\Big) \odot \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)}\end{aligned}$$

Proximal Point Algorithm = Adaptive Sinkhorn-scaling

- In the k-th iteration, denote $\boldsymbol{a}^{(k)}(\boldsymbol{b}^{(k)})^\top$ as $\boldsymbol{\Delta}^{(k)}$.
- According to the algorithm, we have $\boldsymbol{P}^{(k)} = \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)}$.

$$
\begin{aligned}
\boldsymbol{\Phi}^{(k)} &= \exp\Big(-\frac{\boldsymbol{D} - \beta \log \boldsymbol{P}^{(k)}}{\beta}\Big) = \exp\Big(-\frac{\boldsymbol{D}}{\beta}\Big) \odot \boldsymbol{P}^{(k)} \\
&= \exp\Big(-\frac{\boldsymbol{D}}{\beta}\Big) \odot \boldsymbol{\Phi}^{(k-1)} \odot \boldsymbol{\Delta}^{(k-1)} \\
&= \exp\Big(-\frac{k}{\beta}\boldsymbol{D}\Big) \odot \underbrace{(\odot_{i=0}^{k-1} \boldsymbol{\Delta}^{(i)})}_{\boldsymbol{\Delta}_k}.
\end{aligned} \tag{8}
$$

- $\boldsymbol{\Delta}_k$ determines the initial point while the problem corresponding to the iteration steps is convex.
- So proximal point algorithm implements the Sinkhorn-scaling with a decaying weight $\epsilon^{(k)} = \frac{\beta}{k}$.

Bregman ADMM: Solve OT without Sinkhorn

- The Sinkhorn-based algorithm often suffers from numerical instability issue.
- Only apply to the OT problems with entropy/KLD regularizers.

Bregman ADMM: Solve OT without Sinkhorn

- The Sinkhorn-based algorithm often suffers from numerical instability issue.
- Only apply to the OT problems with entropy/KLD regularizers.

Bregman ADMM: Simplifying the problem by decoupling the doubly-stochastic constraint to two one-side constraints.

- Introduce an auxiliary variable $\boldsymbol{S}$:

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle \quad \Leftrightarrow \quad \min_{\boldsymbol{P},\boldsymbol{S}} \langle \boldsymbol{D}, \boldsymbol{P}\rangle \;\; s.t.\; \boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \cdot),\; \boldsymbol{S} \in \Pi(\cdot, \boldsymbol{\rho}_1),\; \boldsymbol{P} = \boldsymbol{S}. \quad (9)$$

Bregman ADMM: Solve OT without Sinkhorn

- The Sinkhorn-based algorithm often suffers from numerical instability issue.
- Only apply to the OT problems with entropy/KLD regularizers.

Bregman ADMM: Simplifying the problem by decoupling the doubly-stochastic constraint to two one-side constraints.

- Introduce an auxiliary variable $\boldsymbol{S}$:

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \boldsymbol{P}\rangle \quad \Leftrightarrow \quad \min_{\boldsymbol{P},\boldsymbol{S}} \langle \boldsymbol{D}, \boldsymbol{P}\rangle \;\; s.t.\; \boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\cdot),\; \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\rho}_1),\; \boldsymbol{P}=\boldsymbol{S}. \quad (9)$$

- Introduce a dual variable $\boldsymbol{Z}$:

$$\min_{\boldsymbol{P},\boldsymbol{S}} \max_{\boldsymbol{Z}} \langle \boldsymbol{D}, \boldsymbol{P}\rangle + \underbrace{\langle \boldsymbol{Z}, \boldsymbol{T}-\boldsymbol{S}\rangle + \overbrace{\epsilon \mathrm{B}_\phi(\boldsymbol{T},\boldsymbol{S})}^{\text{Bregman Div.}}}_{\text{Augmented Lagrangian}} \;\; s.t.\; \boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\cdot),\; \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\rho}_1). \quad (10)$$

Bregman alternating direction method of multipliers. NeurIPS, 2014.

Bregman ADMM: Solve OT without Sinkhorn

Bregman Divergence: Given a differentiable and strictly convex function ϕ,

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle. \tag{11}$$

Bregman ADMM: Solve OT without Sinkhorn

Bregman Divergence: Given a differentiable and strictly convex function ϕ,

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle. \tag{11}$$

Commonly-used Bregman divergence:

- $\phi(x) = \frac{1}{2}x^2$: Euclidean distance $B_\phi(x, y) = \frac{1}{2}\|x - y\|^2$.
- $\phi(x) = x \log x - x$: KL-divergence $B_\phi(x, y) = \mathsf{KL}(x\|y) = x \log \frac{x}{y} - x + y$.

Bregman ADMM: Solve OT without Sinkhorn

Bregman Divergence: Given a differentiable and strictly convex function ϕ,

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle. \tag{11}$$

Commonly-used Bregman divergence:

- $\phi(x) = \frac{1}{2}x^2$: Euclidean distance $B_\phi(x, y) = \frac{1}{2}\|x - y\|^2$.
- $\phi(x) = x \log x - x$: KL-divergence $B_\phi(x, y) = \mathsf{KL}(x\|y) = x \log \frac{x}{y} - x + y$.

Naturally, the Bregman ADMM is also applicable for various regularized OT:

- Considering the above Bregman divergence leads to the OT problems with entropic or quadratic regularizers.

Bregman ADMM: Solve OT without Sinkhorn

Bregman Divergence: Given a differentiable and strictly convex function ϕ,

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle. \tag{11}$$

Commonly-used Bregman divergence:

- $\phi(x) = \frac{1}{2}x^2$: Euclidean distance $B_\phi(x, y) = \frac{1}{2}\|x - y\|^2$.
- $\phi(x) = x\log x - x$: KL-divergence $B_\phi(x, y) = \mathsf{KL}(x\|y) = x\log\frac{x}{y} - x + y$.

Naturally, the Bregman ADMM is also applicable for various regularized OT:

- Considering the above Bregman divergence leads to the OT problems with entropic or quadratic regularizers.

The Bregman ADMM algorithm solves the OT problems iteratively.

- Each step has a closed form.
- Sublinear convergence rate.

Besides improving smoothness, pursuing structured OT plans leads to efficient algorithms.

Structured OT Problems: Low-rank Optimal Transport

- The OT problem with a rank-r OT plan: $\boldsymbol{P} = \boldsymbol{Q}\text{diag}^{-1}(\boldsymbol{g})\boldsymbol{R}^\top \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)$

$$\begin{aligned} &\min_{\boldsymbol{Q},\, \boldsymbol{R},\, \boldsymbol{g}} \ \langle \boldsymbol{D},\ \boldsymbol{Q}\text{diag}^{-1}(\boldsymbol{g})\boldsymbol{R}^\top \rangle, \\ &\quad s.t.\ \boldsymbol{Q} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{g}),\ \boldsymbol{R} \in \Pi(\boldsymbol{\rho}_1, \boldsymbol{g}),\ \boldsymbol{g} \in \boldsymbol{\Delta}^{r-1}. \end{aligned} \tag{12}$$

Structured OT Problems: Low-rank Optimal Transport

- The OT problem with a rank-r OT plan: $\boldsymbol{P} = \boldsymbol{Q}\text{diag}^{-1}(\boldsymbol{g})\boldsymbol{R}^\top \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)$

$$\begin{aligned} &\min_{\boldsymbol{Q},\ \boldsymbol{R},\ \boldsymbol{g}} \ \langle \boldsymbol{D},\ \boldsymbol{Q}\text{diag}^{-1}(\boldsymbol{g})\boldsymbol{R}^\top \rangle, \\ &\quad s.t.\ \boldsymbol{Q} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{g}),\ \boldsymbol{R} \in \Pi(\boldsymbol{\rho}_1, \boldsymbol{g}),\ \boldsymbol{g} \in \boldsymbol{\Delta}^{r-1}. \end{aligned} \tag{12}$$

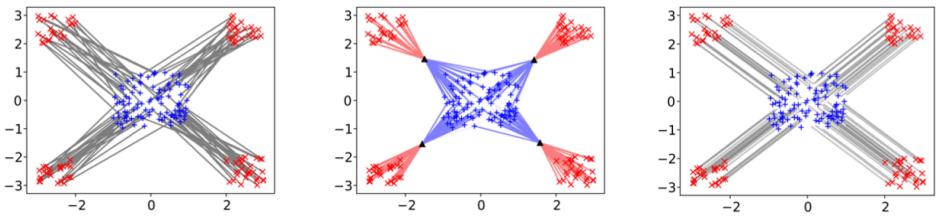
- A mirror descent scheme w.r.t. the KL-divergence, leading to proximal point algorithm in each step.
- Take the update of $\boldsymbol{Q}$ as an example:

$$\boldsymbol{Q}^{(k+1)} = \arg\min_{\boldsymbol{Q} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{g}^{(k)})} \langle \boldsymbol{Q}, \boldsymbol{D}\boldsymbol{R}^{(k)}\text{diag}^{-1}(\boldsymbol{g}^{(k)}) \rangle + \beta \text{KL}(\boldsymbol{Q} \| \boldsymbol{Q}^{(k)}). \tag{13}$$

Low-rank Sinkhorn factorization, ICML, 2021.

Structured OT Problems: Low-rank Optimal Transport

- Reduce the number of variables when r is small.
- Improve robustness to noise.



Statistical optimal transport via factored couplings. AISTATS, 2019.

Structured OT Problems: Sparse Optimal Transport

- **Replace the entropic regularizer to a quadratic regularizer**:

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D},\ \boldsymbol{P}\rangle + \frac{\epsilon}{2}\|\boldsymbol{P}\|_F^2. \tag{14}$$

Structured OT Problems: Sparse Optimal Transport

- **Replace the entropic regularizer to a quadratic regularizer**:

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D},\ \boldsymbol{P}\rangle + \frac{\epsilon}{2}\|\boldsymbol{P}\|_F^2. \tag{14}$$

- Applying the L-BFGS algorithm to solve the smoothed dual formulation of (14), the OT plan has a closed-form expression: for $\boldsymbol{P}^* = [p^*_{mn}]$,

$$p^*_{mn} = \frac{1}{\epsilon}[a^*_m + b^*_n - d_{mn}]_+. \tag{15}$$

Structured OT Problems: Sparse Optimal Transport

- **Replace the entropic regularizer to a quadratic regularizer**:

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{D},\ \boldsymbol{P}\rangle + \frac{\epsilon}{2}\|\boldsymbol{P}\|_F^2. \tag{14}$$

- Applying the L-BFGS algorithm to solve the smoothed dual formulation of (14), the OT plan has a closed-form expression: for $\boldsymbol{P}^* = [p^*_{mn}]$,

$$p^*_{mn} = \frac{1}{\epsilon}[a^*_m + b^*_n - d_{mn}]_+. \tag{15}$$

- This problem is highly correlated with LASSO, leading to a sparse OT plan.

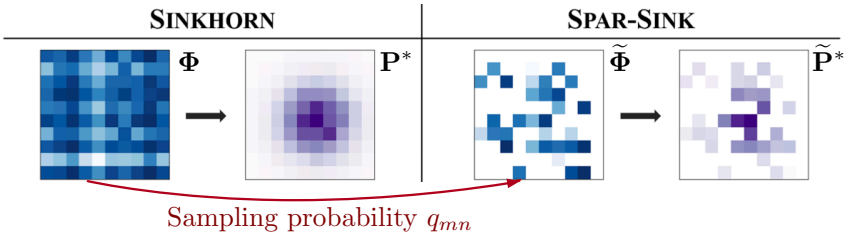
Smooth and sparse optimal transport. AISTATS, 2018.

When focusing on OT distance rather than OT plan, more efficient algorithms can be applied.

Approximated Sinkhorn Distance via Importance Sparsification

- **Sample the OT plan randomly via importance sparsification**: apply the principle of Poisson sampling to sketch the kernel matrix $\mathbf{\Phi} = [\phi_{mn}]$ to s nonzero elements:

$$\widetilde{\mathbf{\Phi}} = [\tilde{\phi}_{mn}], \quad \text{where } \tilde{\phi}_{mn} = \begin{cases} \frac{\phi_{mn}}{q^*_{mn}} & \text{with prob.} q^*_{mn} = \min\{1, q_{mn} s\} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$



Importance Sparsification for Sinkhorn Algorithm. JMLR, 2023.

Approximated Sinkhorn Distance via Importance Sparsification

- Intuitively, when $d_{mn}p^*_{mn}$ is large, we should sample it with a high probability. However, p^*_{mn} is unavailable.

Approximated Sinkhorn Distance via Importance Sparsification

- Intuitively, when $d_{mn}p^*_{mn}$ is large, we should sample it with a high probability. However, p^*_{mn} is unavailable.
- In practice, the sampling probability q_{mn} is set as the upper bound of $d_{mn}p^*_{mn}$:
  - Bounded distance/cost: $d_{mn} \leq c_0$
  - Bounded OT plan: $p^*_{mn} \leq \rho_{0,m},\ \rho_{1,n}$

$$d_{mn}p^*_{mn} \leq c_0\sqrt{\rho_{0,m}\rho_{1,n}} \quad \Rightarrow \quad q_{mn} = \frac{\sqrt{\rho_{0,m}\rho_{1,n}}}{\sum_{m,n}\sqrt{\rho_{0,m}\rho_{1,n}}} \tag{17}$$

Approximated Sinkhorn Distance via Importance Sparsification

- Intuitively, when $d_{mn}p^*_{mn}$ is large, we should sample it with a high probability. However, p^*_{mn} is unavailable.
- In practice, the sampling probability q_{mn} is set as the upper bound of $d_{mn}p^*_{mn}$:
  - Bounded distance/cost: $d_{mn} \leq c_0$
  - Bounded OT plan: $p^*_{mn} \leq \rho_{0,m},\ \rho_{1,n}$

$$d_{mn}p^*_{mn} \leq c_0\sqrt{\rho_{0,m}\rho_{1,n}} \quad \Rightarrow \quad q_{mn} = \frac{\sqrt{\rho_{0,m}\rho_{1,n}}}{\sum_{m,n}\sqrt{\rho_{0,m}\rho_{1,n}}} \tag{17}$$

- Reduce the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$ when $s \approx N\log N$.

Approximated Sinkhorn Distance via Importance Sparsification

- Intuitively, when $d_{mn}p^*_{mn}$ is large, we should sample it with a high probability. However, p^*_{mn} is unavailable.
- In practice, the sampling probability q_{mn} is set as the upper bound of $d_{mn}p^*_{mn}$:
  - Bounded distance/cost: $d_{mn} \leq c_0$
  - Bounded OT plan: $p^*_{mn} \leq \rho_{0,m},\ \rho_{1,n}$

$$d_{mn}p^*_{mn} \leq c_0\sqrt{\rho_{0,m}\rho_{1,n}} \quad \Rightarrow \quad q_{mn} = \frac{\sqrt{\rho_{0,m}\rho_{1,n}}}{\sum_{m,n}\sqrt{\rho_{0,m}\rho_{1,n}}} \tag{17}$$

- Reduce the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$ when $s \approx N\log N$.
- The approximation error between $\mathcal{W}_{p,\epsilon}$ and $\widetilde{\mathcal{W}}_{p,\epsilon}$ is bounded:

$$|\mathcal{W}_{p,\epsilon} - \widetilde{\mathcal{W}}_{p,\epsilon}| \leq c\epsilon\sqrt{\frac{N^{3-2\alpha}}{s}},\ \text{where } c > 0,\ \alpha \in (0.5, 1). \tag{18}$$

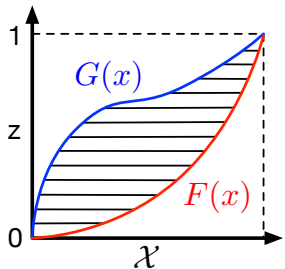
Importance Sparsification for Sinkhorn Algorithm. JMLR, 2023.

Sliced Wasserstein: A Surrogate of Wasserstein Distance

When $\dim(\mathcal{X}) = 1$, $\mathcal{W}_p$ has a closed form, related to **1D histogram transform and equalization**.

$$\mathcal{W}_p(\rho_0, \rho_1) = \left(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p \mathrm{d}z \right)^{1/p}, \quad (19)$$

where $F, G : \mathcal{X} \mapsto [0, 1]$ are CDF's of ρ_0 and ρ_1.

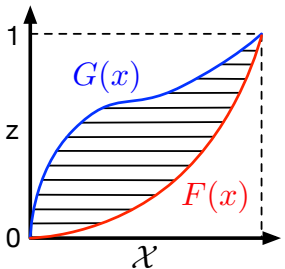


Sliced Wasserstein: A Surrogate of Wasserstein Distance

When $\dim(\mathcal{X}) = 1$, $\mathcal{W}_p$ has a closed form, related to **1D histogram transform and equalization**.

$$\mathcal{W}_p(\rho_0, \rho_1) = \left(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p \mathrm{d}z \right)^{1/p}, \quad (19)$$

where $F, G : \mathcal{X} \mapsto [0, 1]$ are CDF's of ρ_0 and ρ_1.



Theorem 1

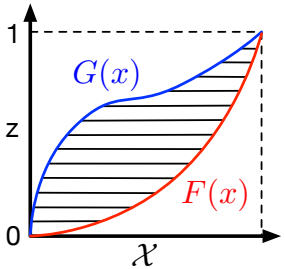
For 1D $x_1 \leq ... \leq x_N$ and $y_1 \leq ... \leq y_N$, identity permutation leads to the optimal transport between them.

Sliced Wasserstein: A Surrogate of Wasserstein Distance

When $\dim(\mathcal{X}) = 1$, $\mathcal{W}_p$ has a closed form, related to **1D histogram transform and equalization**.

$$\mathcal{W}_p(\rho_0, \rho_1) = \Big(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p \mathrm{d}z \Big)^{1/p}, \quad (19)$$

where $F, G : \mathcal{X} \mapsto [0, 1]$ are CDF's of ρ_0 and ρ_1.



Theorem 1

For 1D $x_1 \leq ... \leq x_N$ and $y_1 \leq ... \leq y_N$, identity permutation leads to the optimal transport between them.

- Given $\boldsymbol{x} = \{x_n\}_{n=1}^N \sim \rho_0$ and $\boldsymbol{y} = \{y_n\}_{n=1}^N \sim \rho_1$:

$$\mathcal{W}_p(\boldsymbol{x}, \boldsymbol{y}) = \Big(\sum_{n=1}^N |x_n - y_{\mathsf{sort}(n)}|^p \Big)^{1/p}, \quad (20)$$

Sliced and radon Wasserstein barycenters of measures. Journal of Mathematical Imaging and Vision, 2015.

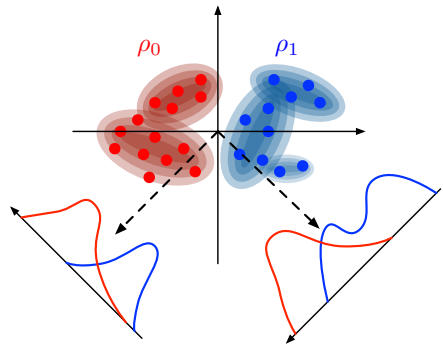
Sliced Wasserstein: A Surrogate of Wasserstein Distance

- Let $\theta \sim p_{\mathcal{S}^{D-1}}$ be random projection directions, and R_θ be the corresponding random projection, i.e., $R_\theta(x) = \langle x, \theta \rangle$ for $x \sim \rho_0, \rho_1$.
- R_θ pushes ρ_0, ρ_1 forward 1D distributions $R_{\theta\#}\rho_0, R_{\theta\#}\rho_1$.

Sliced Wasserstein: A Surrogate of Wasserstein Distance

- Let $\theta \sim p_{\mathcal{S}^{D-1}}$ be random projection directions, and R_θ be the corresponding random projection, i.e., $R_\theta(x) = \langle x, \theta \rangle$ for $x \sim \rho_0, \rho_1$.
- R_θ pushes ρ_0, ρ_1 forward 1D distributions $R_{\theta\#}\rho_0, R_{\theta\#}\rho_1$.
- **Sliced-Wasserstein distance**:

$$\begin{aligned} &\mathcal{SW}_p(\rho_0, \rho_1) \\ :=&\mathbb{E}_{\theta \sim p_{\mathcal{S}^{D-1}}}[\mathcal{W}_p(R_{\theta\#}\rho_0, R_{\theta\#}\rho_1)] \\ =&\int_{\theta \in \mathcal{S}^{D-1}} \mathcal{W}_p(R_{\theta\#}\rho_0, R_{\theta\#}\rho_1) p(\theta) \mathrm{d}\theta \end{aligned} \tag{21}$$



Sliced and radon wasserstein barycenters of measures. Journal of Mathematical Imaging and Vision, 2015.

Sliced Wasserstein: A Surrogate of Wasserstein Distance

- Practical implementation:
  - **Finite samples:** $\boldsymbol{X} = \{x_n\}_{n=1}^N$ and $\boldsymbol{Y} = \{y_n\}_{n=1}^N$
  - **Finite projections:** $\{\theta_l\}_{l=1}^L \sim p_{\mathcal{S}^{D-1}}$.

Sliced Wasserstein: A Surrogate of Wasserstein Distance

- Practical implementation:
  - **Finite samples:** $\boldsymbol{X} = \{x_n\}_{n=1}^N$ and $\boldsymbol{Y} = \{y_n\}_{n=1}^N$
  - **Finite projections:** $\{\theta_l\}_{l=1}^L \sim p_{\mathcal{S}^{D-1}}$.
- Sample-based sliced Wasserstein distance:

$$\widehat{\mathcal{SW}}_p(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{L} \sum_{l=1}^{L} \left(\min_{\boldsymbol{P} \in \Pi(\frac{1}{N}\mathbf{1}_N, \frac{1}{N}\mathbf{1}_N)} \sum_{m,n=1}^{N} |\theta_l^\top x_m - \theta_l^\top y_n|^p p_{mn} \right)^{1/p}$$

Sliced Wasserstein: A Surrogate of Wasserstein Distance

- Practical implementation:
  - **Finite samples:** $\boldsymbol{X} = \{x_n\}_{n=1}^N$ and $\boldsymbol{Y} = \{y_n\}_{n=1}^N$
  - **Finite projections:** $\{\theta_l\}_{l=1}^L \sim p_{\mathcal{S}^{D-1}}$.
- Sample-based sliced Wasserstein distance:

$$
\begin{aligned}
\widehat{\mathcal{SW}}_p(\boldsymbol{X}, \boldsymbol{Y}) &= \frac{1}{L}\sum_{l=1}^{L}\left(\min_{\boldsymbol{P}\in\Pi(\frac{1}{N}\mathbf{1}_N, \frac{1}{N}\mathbf{1}_N)} \sum_{m,n=1}^{N} |\theta_l^\top x_m - \theta_l^\top y_n|^p p_{mn}\right)^{1/p} \\
&= \frac{1}{L}\sum_{l=1}^{L}\left(\frac{1}{N}\min_{\sigma\in\mathcal{P}_N} \sum_{n=1}^{N} |\theta_l^\top x_m - \theta_l^\top y_{\sigma(n)}|^p\right)^{1/p} \\
&= \frac{1}{L}\sum_{l=1}^{L}\left(\frac{1}{N}\sum_{n=1}^{N} |\theta_l^\top x_{\mathsf{sort}(n)} - \theta_l^\top y_{\mathsf{sort}(n)}|^p\right)^{1/p}
\end{aligned}
\tag{22}
$$

Extensions of Sliced Wasserstein: Max-sliced Wasserstein

- **Max-sliced Wasserstein (MSW)**: Instead of randomly sampling projections, learn the optimal one in an adversarial way:

$$\mathcal{MSW}_p(\rho_0, \rho_1) := \max_{\theta \in \mathcal{S}^{D-1}} \mathcal{W}_p(R_{\theta\#}\rho_0, R_{\theta\#}\rho_1). \tag{23}$$

Extensions of Sliced Wasserstein: Max-sliced Wasserstein

- **Max-sliced Wasserstein (MSW)**: Instead of randomly sampling projections, learn the optimal one in an adversarial way:

$$\mathcal{MSW}_p(\rho_0, \rho_1) := \max_{\theta \in \mathcal{S}^{D-1}} \mathcal{W}_p(R_{\theta\#}\rho_0, R_{\theta\#}\rho_1). \tag{23}$$

- Given samples:

$$\widehat{\mathcal{MSW}}_p(\boldsymbol{X}, \boldsymbol{Y}) := \max_{\theta \in \mathcal{S}^{D-1}} \Big(\min_{\sigma \in \mathcal{P}_N} \sum_{n=1}^{N} |\theta^\top x_n - \theta^\top y_{\sigma(n)}|^p \Big)^{1/p}. \tag{24}$$

Extensions of Sliced Wasserstein: Max-sliced Wasserstein

- **Max-sliced Wasserstein (MSW)**: Instead of randomly sampling projections, learn the optimal one in an adversarial way:

$$\mathcal{MSW}_p(\rho_0, \rho_1) := \max_{\theta \in \mathcal{S}^{D-1}} \mathcal{W}_p(R_{\theta\#}\rho_0, R_{\theta\#}\rho_1). \tag{23}$$

- Given samples:

$$\widehat{\mathcal{MSW}}_p(\boldsymbol{X}, \boldsymbol{Y}) := \max_{\theta \in \mathcal{S}^{D-1}} \Big(\min_{\sigma \in \mathcal{P}_N} \sum_{n=1}^{N} |\theta^\top x_n - \theta^\top y_{\sigma(n)}|^p \Big)^{1/p}. \tag{24}$$

- $\mathcal{MSW}_p$ is strongly equivalence to $\mathcal{W}_p$: for $p = 1, 2$,

$$\exists 0 < c_1 < c_2,\ c_1 \mathcal{MSW}_p \le \mathcal{W}_p \le c_2 \mathcal{MSW}_p. \tag{25}$$

Max-sliced Wasserstein distance and its use for GANs. CVPR, 2019.
Subspace robust Wasserstein distances. ICML, 2019.

Strong equivalence between metrics of Wasserstein type. 2021.

Extensions of Sliced Wasserstein: Generalized Sliced Wasserstein

- **Generalized sliced Wasserstein (GSW)**: Replacing the linear projections to nonlinear ones (by generalized Radon transformation or a neural network)

$$\begin{aligned} \mathcal{GSW}_p(\rho_0, \rho_1) &:= \int_{F_\theta \in \Omega} \mathcal{W}_p(F_{\theta\#}\rho_0, F_{\theta\#}\rho_1) p(\theta) \mathrm{d}\theta, \\ \mathcal{MGSW}_p(\rho_0, \rho_1) &:= \max_{F_\theta \in \Omega} \mathcal{W}_p(F_{\theta\#}\rho_0, F_{\theta\#}\rho_1) \end{aligned} \tag{26}$$

  where $F_\theta \in \Omega$ is the generalized Radon transformation and θ is rotation angle.
- Alternating optimization is applied to compute these variants.

Generalized sliced wasserstein distances. NeurIPS, 2019.

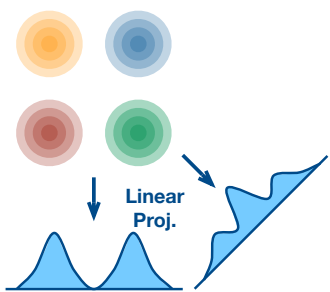
Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

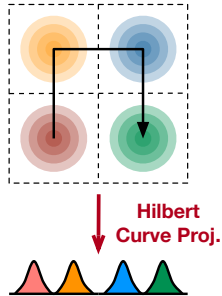
- Linear projections used in SW often break the **locality-preserving property**.
- Nonlinear projections used in GSW requires additional learning.

Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

- Linear projections used in SW often break the **locality-preserving property**.
- Nonlinear projections used in GSW requires additional learning.

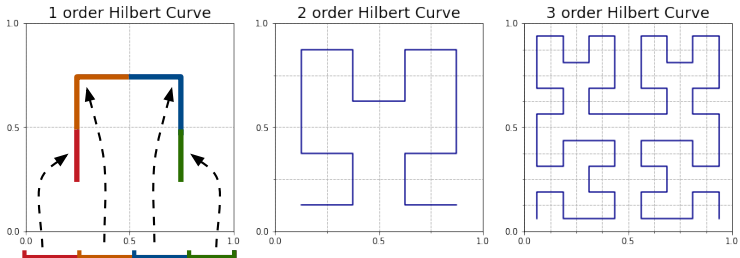
Hilbert Curve Projection Distance: apply Hilbert curve, a special kind of space-filling curve, to achieve projections with the locality-preserving property.





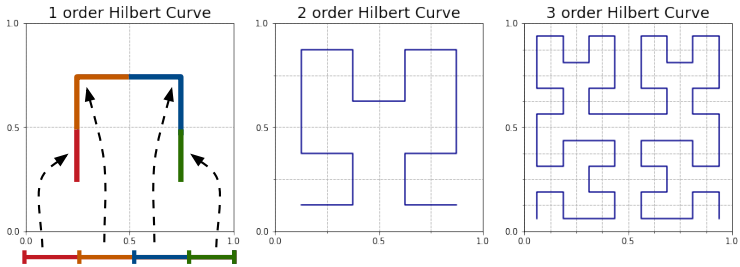
Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

- A K-order Hilbert curve H_K:
  - Partition the $[0,1]$ and D-dimensional unit hyper-cube $[0,1]^D$ into $(2^K)^D$ parts.
  - **Construct a bijection between them**.



Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

- A K-order Hilbert curve H_K:
  - Partition the $[0,1]$ and D-dimensional unit hyper-cube $[0,1]^D$ into $(2^K)^D$ parts.
  - **Construct a bijection between them**.



- Space-filing curve $H(x) = \lim_{K\to\infty} H_K(x)$ is a surjection $H : [0,1] \to [0,1]^d$.
- H covers the entire hyper-cube and enjoys the **locality-preserving property**:

$$\|H(x) - H(y)\|_2 \leq 2\sqrt{d+3}|x-y|^{1/d}, \ \forall x, y \in [0,1]. \tag{27}$$

Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

- Given a probability measure ρ defined on a hyper-cube Ω_ρ, denote its **Hilbert curve** as $H_\rho : [0, 1] \mapsto \Omega_\rho$.

Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

- Given a probability measure ρ defined on a hyper-cube Ω_ρ, denote its **Hilbert curve** as $H_\rho : [0,1] \mapsto \Omega_\rho$.
- **The CDF of ρ along** H_ρ is $g_\rho(t) = \inf_{s\in[0,t]} \rho\Big(\underbrace{H_\rho([0,s])}_{\text{A Borel set in } \Omega_\rho} \Big)$, for $t \in [0,1]$.

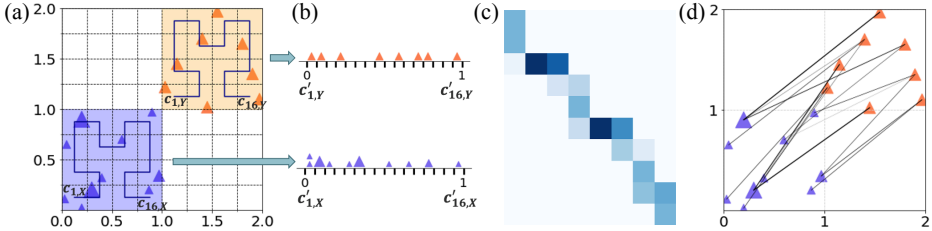
Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance

- Given a probability measure ρ defined on a hyper-cube Ω_ρ, denote its **Hilbert curve** as $H_\rho : [0,1] \mapsto \Omega_\rho$.
- **The CDF of ρ along** H_ρ is $g_\rho(t) = \inf_{s\in[0,t]} \rho\Big(\underbrace{H_\rho([0,s])}_{\text{A Borel set in } \Omega_\rho} \Big)$, for $t \in [0,1]$.
- **The Hilbert Curve Projection (HCP) distance determines OT plan via 1D Wasserstein along Hilbert curve:**

$$\mathcal{HCP}_p(\rho_0, \rho_1) := \Big(\int_0^1 \| \underbrace{H_{\rho_0} \circ g_{\rho_0}^{-1}}_{F^{-1}}(z)) - \underbrace{H_{\rho_1} \circ g_{\rho_1}^{-1}}_{G^{-1}}(z))\|_p^p \mathrm{d}z \Big)^{\frac{1}{p}} \tag{28}$$

Hilbert curve projection distance for distribution comparison. TPAMI, 2024.

Extensions of Sliced Wasserstein: Hilbert Curve Projection Distance



1. Project D-dimensional samples along their K-order Hilbert curves, and determine the OT plan accordingly. ($\mathcal{O}((N+M)DK)$)
2. Determine the OT plan via sorting the projected samples. ($\mathcal{O}(N \log N + M \log M)$)
3. Compute the HCP distance by the raw samples and the OT plan.

Hilbert curve projection distance for distribution comparison. TPAMI, 2024.

Summary

- $\mathcal{W}_p$ and its variants (e.g., $\mathcal{SW}_p$, $\mathcal{MSW}_p$ $\mathcal{HCP}_p$, and so on) provide valid metrics for probability measures.
  - $\mathcal{MSW}_p$ is strongly equivalent to $\mathcal{W}_p$
  - $\mathcal{SW}_p$ is weakly equivalent to $\mathcal{W}_p$
  - $\mathcal{HCP}_p$ is an upper bound of $\mathcal{W}_p$

Summary

- $\mathcal{W}_p$ and its variants (e.g., $\mathcal{SW}_p$, $\mathcal{MSW}_p$ $\mathcal{HCP}_p$, and so on) provide valid metrics for probability measures.
  - $\mathcal{MSW}_p$ is strongly equivalent to $\mathcal{W}_p$
  - $\mathcal{SW}_p$ is weakly equivalent to $\mathcal{W}_p$
  - $\mathcal{HCP}_p$ is an upper bound of $\mathcal{W}_p$
- Efficient approximation methods (Sinkhorn, Proximal Point, Bregman ADMM, etc.) are proposed with the help of various regularizers.
  - Sublinear convergence rate (i.e., $\mathcal{O}(1/\epsilon^2)$ steps to achieve ϵ-approximation)
  - Reduce the complexity to $\mathcal{O}(N^2)$

Summary

- $\mathcal{W}_p$ and its variants (e.g., $\mathcal{SW}_p$, $\mathcal{MSW}_p$ $\mathcal{HCP}_p$, and so on) provide valid metrics for probability measures.
  - $\mathcal{MSW}_p$ is strongly equivalent to $\mathcal{W}_p$
  - $\mathcal{SW}_p$ is weakly equivalent to $\mathcal{W}_p$
  - $\mathcal{HCP}_p$ is an upper bound of $\mathcal{W}_p$
- Efficient approximation methods (Sinkhorn, Proximal Point, Bregman ADMM, etc.) are proposed with the help of various regularizers.
  - Sublinear convergence rate (i.e., $\mathcal{O}(1/\epsilon^2)$ steps to achieve ϵ-approximation)
  - Reduce the complexity to $\mathcal{O}(N^2)$
- Structured OT plans (Low-rank and/or sparse OT plans) often lead to further accelerations.
  - The time complexity of low-rank OT is $\mathcal{O}(N^2 r)$ but it reduces memory cost and improves robustness.
  - Apply importance sparsification reduces the complexity to $\mathcal{O}(N \log N)$

Summary

- $\mathcal{W}_p$ and its variants (e.g., $\mathcal{SW}_p$, $\mathcal{MSW}_p$ $\mathcal{HCP}_p$, and so on) provide valid metrics for probability measures.
  - $\mathcal{MSW}_p$ is strongly equivalent to $\mathcal{W}_p$
  - $\mathcal{SW}_p$ is weakly equivalent to $\mathcal{W}_p$
  - $\mathcal{HCP}_p$ is an upper bound of $\mathcal{W}_p$
- Efficient approximation methods (Sinkhorn, Proximal Point, Bregman ADMM, etc.) are proposed with the help of various regularizers.
  - Sublinear convergence rate (i.e., $\mathcal{O}(1/\epsilon^2)$ steps to achieve ϵ-approximation)
  - Reduce the complexity to $\mathcal{O}(N^2)$
- Structured OT plans (Low-rank and/or sparse OT plans) often lead to further accelerations.
  - The time complexity of low-rank OT is $\mathcal{O}(N^2 r)$ but it reduces memory cost and improves robustness.
  - Apply importance sparsification reduces the complexity to $\mathcal{O}(N \log N)$
- Potential applications:
  - Distance-centric applications: design loss functions for representation and generative models.
  - OT plan-centric applications: solve matching problems and design models.

Thanks!
5-min break and QA

Part 1 **Computational Optimal Transport (Hongteng Xu)**

- Preliminaries and basic concepts
- Typical computation methods

Part 2 **Representation Learning Driven by OT (Dixin Luo)**

- OT-based multi-modal learning
- Monge gap and its Gromovization for information bottleneck

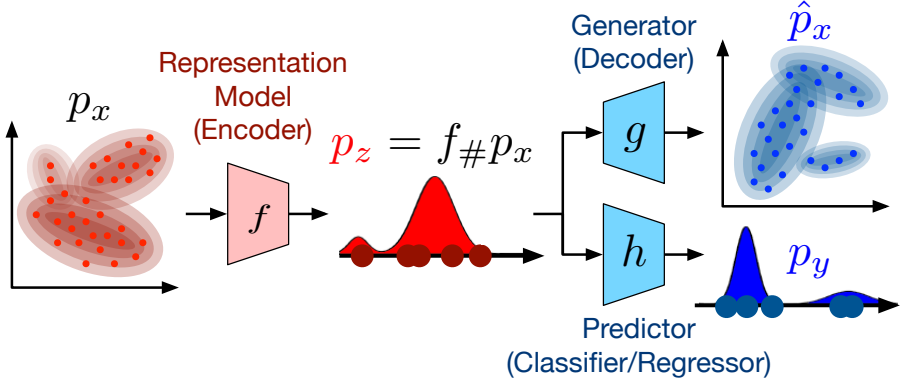
Part 3 **Neural Network Design Driven by OT (Minjie Cheng)**

- OT-based Transformer
- OT-based graph neural network

Part 4 **Recent Progress in Generative Modeling (Hongteng Xu)**

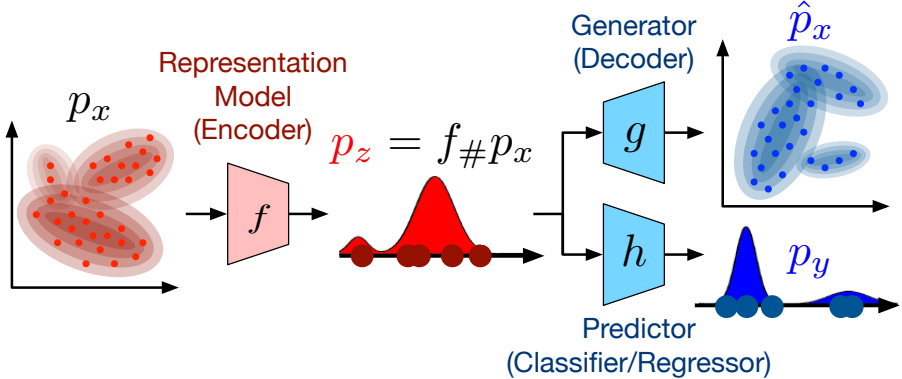
- OT-based flow matching
- Applications of optimal acceleration transport

Motivation: How to Improve Representation Learning?



- Representation learning aims to obtain informative and structured latent representation of data, supporting downstream discriminative and generative tasks.

Motivation: How to Improve Representation Learning?



- Representation learning aims to obtain informative and structured latent representation of data, supporting downstream discriminative and generative tasks.
- **Can we learn the encoder as an optimal transport map? What is its benefit?**

Outline

1. **Learning Encoders as an Optimal Transport Map**
   - Monge gap: a regularizer of neural optimal transport
   - Gromov-Wasserstein distance and Gromov-Monge gap
   - Gromov-Wasserstein Information Bottleneck

2. **Optimal Transport Driven Multi-modal Learning**
   - Gromov-Wasserstein barycenter for kernel fusion
   - Hierarchical optimal transport for multi-modal representation

Benefits and Challenges of OT Encoders

Let's start with a simple scenario: learning an OT encoder as a Monge map in $T : \mathcal{X} \mapsto \mathcal{X}$ under a cost function c, without dimension reduction, i.e.,

$$T^* = \arg\inf_T \int_{x \in \mathcal{X}} c(x, T(x)) \mathrm{d}x, \quad s.t.\ \rho_{\text{target}} = T_{\#}\rho_{\text{source}}. \tag{29}$$

Benefits and Challenges of OT Encoders

Let's start with a simple scenario: learning an OT encoder as a Monge map in $T : \mathcal{X} \mapsto \mathcal{X}$ under a cost function c, without dimension reduction, i.e.,

$$T^* = \arg\inf_{T} \int_{x \in \mathcal{X}} c(x, T(x)) \mathrm{d}x, \quad s.t.\ \rho_{\text{target}} = T_{\#}\rho_{\text{source}}. \tag{29}$$

A natural, physically meaningful way to suppress over-fitting and mode collapse

- Suppress the folding of latent space by minimizing the cost/geometric distortion:
- The norm-induced ($c(x, y) = \|x - y\|_p$) latent space tends to inherit the Wasserstein geometry:

$$\text{For } \rho_0, \rho_1 \in \mathcal{P}(\mathcal{X}), \quad \mathcal{W}_p(\rho_0, \rho_1) \approx \|T^*_{\#}\rho_0, T^*_{\#}\rho_1\|_p. \tag{30}$$

- Almost everywhere reversible.

Benefits and Challenges of OT Encoders

Let's start with a simple scenario: learning an OT encoder as a Monge map in $T : \mathcal{X} \mapsto \mathcal{X}$ under a cost function c, without dimension reduction, i.e.,

$$T^* = \arg\inf_T \int_{x\in\mathcal{X}} c(x, T(x))\mathrm{d}x, \quad s.t.\ \rho_{\text{target}} = T_{\#}\rho_{\text{source}}. \tag{29}$$

A natural, physically meaningful way to suppress over-fitting and mode collapse

- Suppress the folding of latent space by minimizing the cost/geometric distortion:
- The norm-induced ($c(x, y) = \|x - y\|_p$) latent space tends to inherit the Wasserstein geometry:

$$\text{For } \rho_0, \rho_1 \in \mathcal{P}(\mathcal{X}), \quad \mathcal{W}_p(\rho_0, \rho_1) \approx \|T^*_{\#}\rho_0, T^*_{\#}\rho_1\|_p. \tag{30}$$

- Almost everywhere reversible.

Challenges

- As aforementioned, Monge map may not exist.
- Even if it exists, it is hard to compute it exactly.
- Learning T as a neural network (i.e., neural transport) often suffers over-fitting.

Monge Gap: An Effective Regularizer of Neural Transport

- **Recall that if the optimal transport map T^* exists, it determines a transport plan π^***, so

$$\mathcal{W}_p(\rho_0, \rho_1) \leq \mathcal{M}_p(\rho_0, \rho_1). \tag{31}$$

Monge Gap: An Effective Regularizer of Neural Transport

- **Recall that if the optimal transport map T^* exists, it determines a transport plan π^***, so

$$\mathcal{W}_p(\rho_0, \rho_1) \leq \mathcal{M}_p(\rho_0, \rho_1). \tag{31}$$

- Therefore, given an encoder T, we can define and penalize a **Monge gap** to make it approach to an OT map:

$$\mathcal{MG}_\rho^c(T) = \underbrace{\mathbb{E}_{x\sim\rho}[c(x, T(x))]}_{\geq \mathcal{M}_p(\rho, T_\# \rho)} - \underbrace{\inf_{\pi\in\Pi(\rho, T_\#\rho)} \mathbb{E}_{x,y\sim\pi}[c(x,y)]}_{=\mathcal{W}(\rho, T_\#\rho)}. \tag{32}$$

Monge Gap: An Effective Regularizer of Neural Transport

- **Recall that if the optimal transport map T^* exists, it determines a transport plan π^***, so

$$\mathcal{W}_p(\rho_0, \rho_1) \leq \mathcal{M}_p(\rho_0, \rho_1). \tag{31}$$

- Therefore, given an encoder T, we can define and penalize a **Monge gap** to make it approach to an OT map:

$$\mathcal{MG}^c_\rho(T) = \underbrace{\mathbb{E}_{x\sim\rho}[c(x, T(x))]}_{\geq \mathcal{M}_p(\rho, T_\#\rho)} - \underbrace{\inf_{\pi\in\Pi(\rho, T_\#\rho)} \mathbb{E}_{x,y\sim\pi}[c(x, y)]}_{=\mathcal{W}(\rho, T_\#\rho)}. \tag{32}$$

  - Given $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N \sim \rho$, $T(\boldsymbol{x}_n) \sim T_\#\rho$:

$$\mathcal{MG}^c_\rho(T) = \frac{1}{N}\sum_{n=1}^N c(x_n, T(x_n)) - \min_{\boldsymbol{P}\in\Pi} \langle \boldsymbol{D}_T, \boldsymbol{P}\rangle^{1/2}, \tag{33}$$

where $\boldsymbol{D} = [c(x_n, T(x_m))] \in \mathbb{R}^{N\times N}$.

The Monge Gap: A Regularizer to Learn All Transport Maps. ICML, 2023

Monge Gap: An Effective Regularizer of Neural Transport

Useful properties:

- $\mathcal{MG}_\rho^c(T) \geq 0$, $\forall f$.
- T is an OT map between ρ and $T_\# \rho$ iff $\mathcal{MG}_\rho^c(T) = 0$.

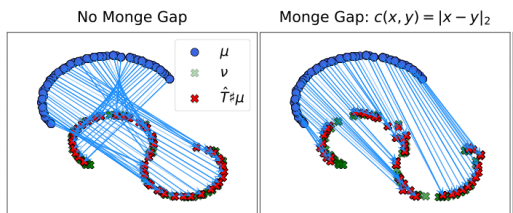
Monge Gap: An Effective Regularizer of Neural Transport

Useful properties:

- $\mathcal{MG}^c_\rho(T) \geq 0$, $\forall f$.
- T is an OT map between ρ and $T_\# \rho$ iff $\mathcal{MG}^c_\rho(T) = 0$.

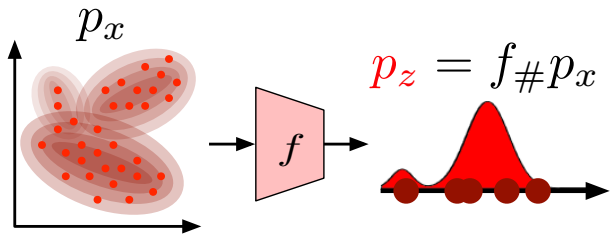
Regularizing representation learning by Monge Gap: given a source distribution μ and a target distribution ν in $(\mathcal{X}, c)$:

$$\min_{T:\mathcal{X}\mapsto\mathcal{X}} \underbrace{\text{Loss}(T_\#\mu, \nu)}_{\text{target fitting}} + \lambda \underbrace{\mathcal{MG}^c_\rho(T)}_{c\text{-optimality}}. \quad (34)$$



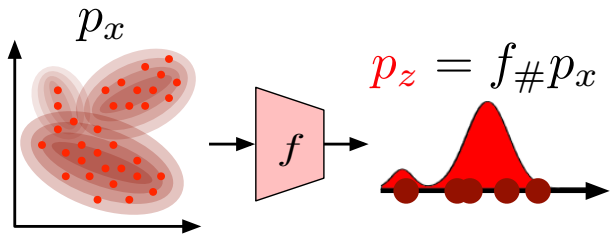
The Monge Gap: A Regularizer to Learn All Transport Maps. ICML, 2023

How to Apply Monge Gap across Incomparable Spaces?



- In general, p_x and p_z are in different spaces (representation learning achieves information compression).
- $\mathcal{MG}^c_{p_x}(f)$ becomes inapplicable because both $\mathcal{M}_p$ and $\mathcal{W}_p$ are undefined across incomparable spaces.

How to Apply Monge Gap across Incomparable Spaces?



- In general, p_x and p_z are in different spaces (representation learning achieves information compression).
- $\mathcal{MG}^c_{p_x}(f)$ becomes inapplicable because both $\mathcal{M}_p$ and $\mathcal{W}_p$ are undefined across incomparable spaces.
- **Solution: Gromovization of Monge gap by defining OT distance across different spaces.**

Gromovization of Classic OT Distances

Gaspard Monge

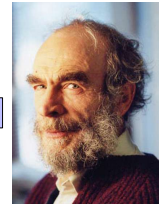
Leonid Kantorovich

18-20th Century
Compare distributions
Monge distance, Wasserstein Distance

Facundo Memoli

Karl-Theodor Sturm

2006-2011
Compare metric measure spaces
Gromov-Wasserstein Distance



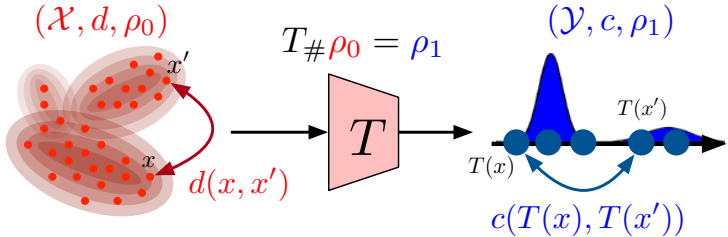
Mikhail Gromov

1980s
Compare metric spaces
Gromov-Hausdorff Distance

On the geometry of metric measure spaces. Acta Mathematica, 2006.

Gromov–Wasserstein distances and the metric approach to object matching. Foundations of computational mathematics, 2011.

Gromov-Monge Distance: Pursue OT Map across Spaces



Given two metric-measure (mm) spaces $(\mathcal{X}, d, \rho_0)$ and $(\mathcal{Y}, c, \rho_1)$, the p-**order Gromov-Monge distance** between them is

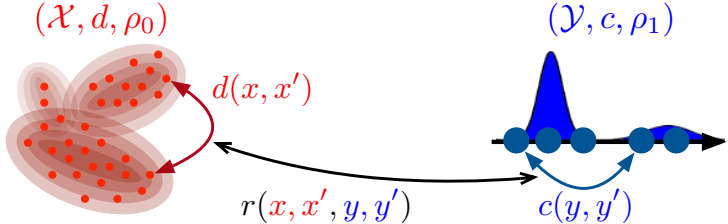
$$\mathcal{GM}_p(\mathcal{X}, \mathcal{Y}) := \Big(\inf_{T} \int_{\mathcal{X}\times\mathcal{X}} \underbrace{|d(x,x') - c(T(x),T(x'))|^p}_{:= r(x,x',T(x),T(x'))} \rho_0(x)\rho_0(x')\mathrm{d}x\mathrm{d}x' \Big)^{1/p}, \tag{35}$$

$$s.t.\ \rho_1 = T_{\#}\rho_0,$$

where r **is relational distance**.

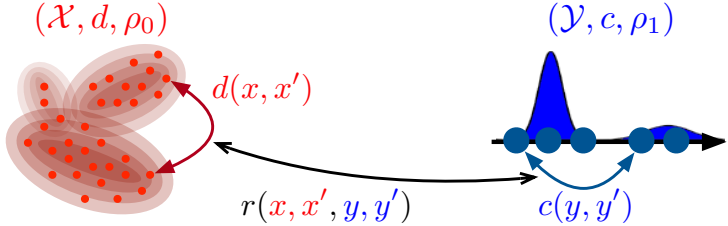
Distance distributions and inverse problems for metric measure spaces. Studies in Applied Mathematics, 2022.

Gromov-Wasserstein Distance: Pursue OT Plan across Spaces



Relational distance across spaces

Gromov-Wasserstein Distance: Pursue OT Plan across Spaces



Relational distance across spaces

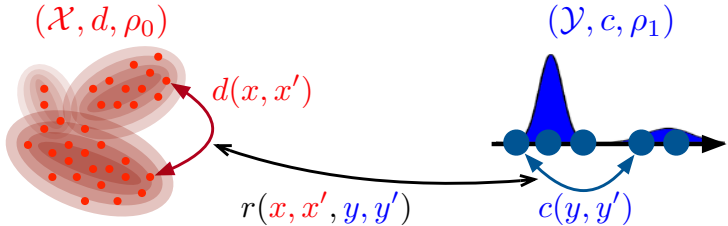
p-order Gromov-Wasserstein distance: Minimize expected relational distance $r(x, x', y, y') = |d(x, x') - c(y, y')|^p$, i.e.,

$$\mathcal{GW}_p(\mathcal{X}, \mathcal{Y}) := \Big(\inf_{\pi \in \Pi(\rho_0, \rho_1)} \underbrace{\int_{\mathcal{X}^2 \times \mathcal{Y}^2} \overbrace{r(x, x', y, y')}^{\text{Relational distance}} \, \pi(x, y)\pi(x', y') \mathrm{d}x \mathrm{d}x' \mathrm{d}y \mathrm{d}y'}_{\mathbb{E}_{(x,y),(x',y') \sim \pi \times \pi}[r(x, x', y, y')]} \Big)^{1/p}. \quad (36)$$

On the geometry of metric measure spaces. Acta Mathematica, 2006.

Gromov–Wasserstein distances and the metric approach to object matching. Foundations of computational

Gromov-Wasserstein Distance: Pursue OT across Spaces



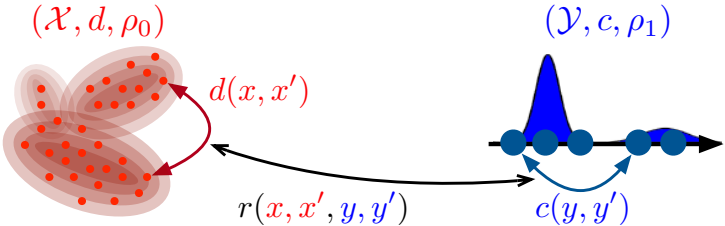
Relational distance across spaces

Given $\boldsymbol{X} = \{x_m\}_{m=1}^M$ with a probability measure $\boldsymbol{\rho}_0$, and $\boldsymbol{Y} = \{y_n\}_{n=1}^N$ with $\boldsymbol{\rho}_1$:

$$\mathcal{GW}_p(\boldsymbol{D}_X, \boldsymbol{D}_Y) = \Big(\min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \sum\nolimits_{m,m'=1}^{M} \sum\nolimits_{n,n'=1}^{N} r(x_m, x_{m'}, y_n, y_{n'}) p_{mn} p_{m'n'} \Big)^{1/p}, \quad (37)$$

where $\boldsymbol{D}_X = [d(x_n, x'_n)]$, $\boldsymbol{D}_Y = [c(y_n, y'_n)]$.

Gromov-Wasserstein Distance: Pursue OT across Spaces



Relational distance across spaces

Given $\boldsymbol{X} = \{x_m\}_{m=1}^M$ with a probability measure $\boldsymbol{\rho}_0$, and $\boldsymbol{Y} = \{y_n\}_{n=1}^N$ with $\boldsymbol{\rho}_1$:

$$\mathcal{GW}_p(\boldsymbol{D}_X, \boldsymbol{D}_Y) = \Big(\min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \sum\nolimits_{m,m'=1}^{M} \sum\nolimits_{n,n'=1}^{N} r(x_m, x_{m'}, y_n, y_{n'}) p_{mn} p_{m'n'} \Big)^{1/p}, \quad (37)$$

where $\boldsymbol{D}_X = [d(x_n, x'_n)]$, $\boldsymbol{D}_Y = [c(y_n, y'_n)]$.

- π^* or $\boldsymbol{P}^*$: the optimal transport plan between samples.
- $\pi^* \times \pi^*$ or $\boldsymbol{P}^* \otimes \boldsymbol{P}^*$: the optimal transport plan between sample pairs.
- **Useful properties: Translation-, rotation-, and permutation-invariance**

Typical Computation Methods of GW Distance

- When $p = 2$, $c(\cdot, \cdot)$ and $d(\cdot, \cdot)$ are Euclidean metrics, GW distance can be rewritten in a matrix format:

$$\min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P} \rangle. \tag{38}$$

- $\boldsymbol{D}_X = [\|x_n - x'_n\|_2^2]$, $\boldsymbol{D}_Y = [\|y_n - y'_n\|_2^2]$
- $\boldsymbol{C} = (\boldsymbol{X} \odot \boldsymbol{X})\mathbf{1}_{d_X \times N} + \mathbf{1}_{N \times d_Y}(\boldsymbol{Y} \odot \boldsymbol{Y})^\top$.

Gromov-Wasserstein averaging of kernel and distance matrices. ICML, 2016.

Typical Computation Methods of GW Distance

- When $p = 2$, $c(\cdot, \cdot)$ and $d(\cdot, \cdot)$ are Euclidean metrics, GW distance can be rewritten in a matrix format:

$$\min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P} \rangle. \tag{38}$$

- $\boldsymbol{D}_X = [\|x_n - x'_n\|_2^2]$, $\boldsymbol{D}_Y = [\|y_n - y'_n\|_2^2]$
- $\boldsymbol{C} = (\boldsymbol{X} \odot \boldsymbol{X})\mathbf{1}_{d_X \times N} + \mathbf{1}_{N \times d_Y}(\boldsymbol{Y} \odot \boldsymbol{Y})^\top$.

Gromov-Wasserstein averaging of kernel and distance matrices. ICML, 2016.

- Given N samples, **Conditional Gradient (CG) descent** leads to $\mathcal{O}(N^3)$ and **sparse OT plans**. In the k-th iteration:

$$\widetilde{\boldsymbol{P}} = \arg \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2 \underbrace{\boldsymbol{D}_X \boldsymbol{P}^{(k)} \boldsymbol{D}_Y^\top}_{\mathcal{O}(N^3)}, \boldsymbol{P} \rangle$$
$$\boldsymbol{P}^{(k+1)} = (1 - \tau^{(k)})\boldsymbol{P}^{(k)} + \tau^{(k)}\widetilde{\boldsymbol{P}}, \text{ where } \tau^{(k)} \text{ is determined by line-search.} \tag{39}$$

Optimal transport for structured data with application on graphs. ICML, 2019.

Typical Computation Methods of GW Distance

- Similar to Wasserstein distance, adding entropy and KL-divergence regularization improves the smoothness of the problem.

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P}\rangle + \epsilon H(\boldsymbol{P}). \tag{40}$$

Typical Computation Methods of GW Distance

- Similar to Wasserstein distance, adding entropy and KL-divergence regularization improves the smoothness of the problem.

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P}\rangle + \epsilon H(\boldsymbol{P}). \tag{40}$$

- **Iterative Sinkhorn-scaling** solves this problem, leading to faster convergence but **smooth OT plan**

$$\boldsymbol{P}^{(k+1)} = \arg\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P}^{(k)} \boldsymbol{D}_Y^\top, \boldsymbol{P}\rangle + \epsilon H(\boldsymbol{P}). \tag{41}$$

Gromov-Wasserstein averaging of kernel and distance matrices. ICML, 2016.

Typical Computation Methods of GW Distance

- Similar to Wasserstein distance, adding entropy and KL-divergence regularization improves the smoothness of the problem.

$$\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P}\rangle + \epsilon H(\boldsymbol{P}). \tag{40}$$

- **Iterative Sinkhorn-scaling** solves this problem, leading to faster convergence but **smooth OT plan**

$$\boldsymbol{P}^{(k+1)} = \arg\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P}^{(k)} \boldsymbol{D}_Y^\top, \boldsymbol{P}\rangle + \epsilon H(\boldsymbol{P}). \tag{41}$$

Gromov-Wasserstein averaging of kernel and distance matrices. ICML, 2016.

- **Iterative Proximal Gradient** is also applicable, which computes exact GW distance with adaptive Sinkhorn-scaling.

$$\boldsymbol{P}^{(k+1)} = \arg\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\boldsymbol{\rho}_1)} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P}^{(k)} \boldsymbol{D}_Y^\top, \boldsymbol{P}\rangle + \epsilon \mathsf{KL}(\boldsymbol{P}\|\boldsymbol{P}^{(k)}). \tag{42}$$

Scalable Gromov-Wasserstein learning for graph partitioning and matching. NeurIPS, 2019.

Typical Computation Methods of GW Distance

- **Bregman ADMM:** Decouple doubly stochastic constraint of $\boldsymbol{P}$. Alternating optimization is applied to solve the problem in augmented Lagrangian form:

$$\begin{aligned} &\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\cdot),\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\rho}_1),\boldsymbol{T}=\boldsymbol{S}} \langle \boldsymbol{C}-2\boldsymbol{D}_X\boldsymbol{S}\boldsymbol{D}_Y^\top,\boldsymbol{P}\rangle \\ \Rightarrow &\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\cdot),\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\rho}_1)} \max_{\boldsymbol{Z}} \langle \boldsymbol{C}-2\boldsymbol{D}_X\boldsymbol{S}\boldsymbol{D}_Y^\top,\boldsymbol{P}\rangle + \langle \boldsymbol{Z},\boldsymbol{T}-\boldsymbol{S}\rangle + \epsilon B_\phi(\boldsymbol{T},\boldsymbol{S}). \end{aligned} \tag{43}$$

  Each step has a closed-form solution when $B_\phi = \mathrm{KL}$.

Gromov-Wasserstein factorization models for graph clustering. AAAI, 2020.

Representing graphs via Gromov-Wasserstein factorization. TPAMI, 2022.

Typical Computation Methods of GW Distance

- **Bregman ADMM:** Decouple doubly stochastic constraint of $\boldsymbol{P}$. Alternating optimization is applied to solve the problem in augmented Lagrangian form:

$$\begin{aligned} &\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\cdot),\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\rho}_1),\boldsymbol{T}=\boldsymbol{S}} \langle \boldsymbol{C}-2\boldsymbol{D}_X\boldsymbol{S}\boldsymbol{D}_Y^\top,\boldsymbol{P}\rangle \\ \Rightarrow &\min_{\boldsymbol{P}\in\Pi(\boldsymbol{\rho}_0,\cdot),\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\rho}_1)} \max_{\boldsymbol{Z}} \langle \boldsymbol{C}-2\boldsymbol{D}_X\boldsymbol{S}\boldsymbol{D}_Y^\top,\boldsymbol{P}\rangle + \langle \boldsymbol{Z},\boldsymbol{T}-\boldsymbol{S}\rangle + \epsilon B_\phi(\boldsymbol{T},\boldsymbol{S}). \end{aligned} \tag{43}$$

  Each step has a closed-form solution when $B_\phi = \mathrm{KL}$.

Gromov-Wasserstein factorization models for graph clustering. AAAI, 2020.

Representing graphs via Gromov-Wasserstein factorization. TPAMI, 2022.

- **Sliced GW** is a theoretically incorrect but practically useful surrogate of GW.

Sliced Gromov-Wasserstein. NeurIPS, 2019.

On assignment problems related to Gromov–Wasserstein distances on the real line. SIAM Journal on Imaging Sciences, 2023.

Gromovized Monge Gap

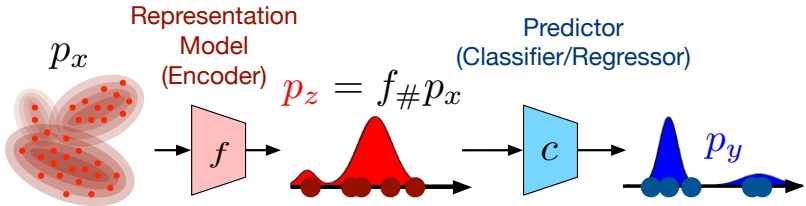
Accordingly, given $\boldsymbol{X} = \{x_n\}_{n=1}^N$ defined in the mm-space $(\mathcal{X}, d, \rho)$, the **Gromovized Monge Gap** of T is defined as

$$
\begin{aligned}
&\mathcal{GMG}_\rho^{d,r}(T) \\
:=&\underbrace{\mathbb{E}_{x,x'\sim\rho\times\rho}[r(x,x',T(x),T(x'))]}_{\geq \mathcal{GM}(\rho,T_\#\rho)} - \underbrace{\inf_{\pi\in\Pi(\rho,T_\#\rho)} \mathbb{E}_{(x,y),(x',y')\pi\times\pi}[r(x,x',y,y')]}_{=\mathcal{GW}(\rho,T_\#\rho)} \\
=&\frac{1}{N^2}\sum_{n,n'=1}^N r(x_n,x_n',T(x_n),T(x_n')) - \min_{\boldsymbol{P}\in\Pi}\sum_{n,n'=1}^N\sum_{m,m'=1}^N r(x_n,x_n',y_m,y_m'),
\end{aligned}
\tag{44}
$$

Revisiting Counterfactual Regression through the Lens of Gromov-Wasserstein Information Bottleneck. Arxiv, 2024.

Disentangled Representation Learning with the Gromov-Monge Gap. ICLR, 2025.

Representation Learning with Gromovized Monge Gap



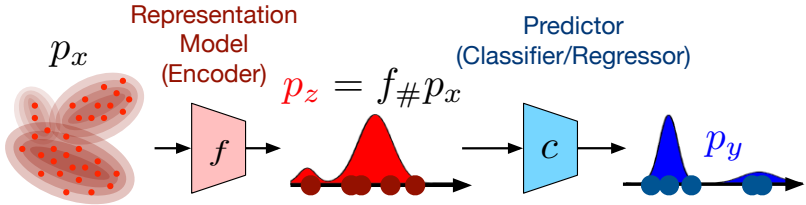
Suppose that $x \sim p_x$. Given a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$, we have

$$\min_{f,c} \mathbb{E}_{(x,y)\sim\mathcal{D}} \text{Loss}(c \circ f(x), y) + \lambda \mathcal{GMG}^{d,r}_{p_x}(f)$$

$$= \min_{f,c} \frac{1}{N} \sum_{n=1}^{N} \text{Loss}(c \circ f(x_n), y_n) + \lambda \Big(\frac{\|\boldsymbol{D}_X - \boldsymbol{D}_Z\|_F^2}{N^2} - \min_{\boldsymbol{P}\in\Pi} \langle \boldsymbol{C} - 2\boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Z^\top, \boldsymbol{P} \rangle \Big). \quad (45)$$

- $\boldsymbol{D}_X = [\|x_n - x'_n\|_2^2]$, $\boldsymbol{D}_Z = [\|f(x_n) - f(x'_n)\|_2^2]$
- $\boldsymbol{C} = (\boldsymbol{X} \odot \boldsymbol{X})\mathbf{1}_{D\times N} + \mathbf{1}_{N\times D}(\boldsymbol{X} \odot \boldsymbol{X})^\top$.

The Connection to Information Bottleneck



The rationality of Gromovized Monge gap can be explained in the information bottleneck framework:

$$\min_f \underbrace{-I(Z,\ Y)}_{\text{Fitting Acc.}} + \underbrace{\lambda I(X,\ Z)}_{\text{Complexity Reg.}}, \quad \text{where } I(X,Z) = \mathsf{KL}(p(X,Z)\|p(X)p(Z)). \tag{46}$$

- Penalizing $-I(Z,Y)$ corresponds to fitting data, which is often implemented as negative log-likelihood.
- Penalizing $I(X,Z)$ regularizes the complexity of representation model.

The information bottleneck method. Allerton Conference on Communication, Control, and Computing, 1999.

The Connection to Information Bottleneck

Given $\{x_n\}_{n=1}^N$ and f, the RBF kernel density estimations of the distributions are

$$p(X) = \frac{1}{N}\sum_{n=1}^{N}\kappa(X, x_n),\ p(Z) = \frac{1}{N}\sum_{n=1}^{N}\kappa(Z, z_n),\ p(X, Z) = \frac{1}{N}\sum_{n=1}^{N}\kappa(X, x_n)\kappa(Z, z_n).$$

The Connection to Information Bottleneck

Given $\{x_n\}_{n=1}^N$ and f, the RBF kernel density estimations of the distributions are

$$p(X) = \frac{1}{N}\sum_{n=1}^{N}\kappa(X, x_n),\ p(Z) = \frac{1}{N}\sum_{n=1}^{N}\kappa(Z, z_n),\ p(X, Z) = \frac{1}{N}\sum_{n=1}^{N}\kappa(X, x_n)\kappa(Z, z_n).$$

Based on N samples, the empirical mutual information between X and Z as

$$\hat{I}_N(Z, X; f) = \frac{1}{N}\sum_{n}\log\frac{p(x_n, z_n)}{p(x_n)p(z_n)} = \frac{1}{N}\sum_{n}\log\frac{N\sum_m \kappa(x_n, x_m)\kappa(z_n, z_m)}{\sum_m \kappa(x_n, x_m)\sum_m \kappa(z_n, z_m)}.$$

The Connection to Information Bottleneck

Given $\{x_n\}_{n=1}^N$ and f, the RBF kernel density estimations of the distributions are

$$p(X) = \frac{1}{N}\sum_{n=1}^{N}\kappa(X, x_n),\ p(Z) = \frac{1}{N}\sum_{n=1}^{N}\kappa(Z, z_n),\ p(X, Z) = \frac{1}{N}\sum_{n=1}^{N}\kappa(X, x_n)\kappa(Z, z_n).$$

Based on N samples, the empirical mutual information between X and Z as

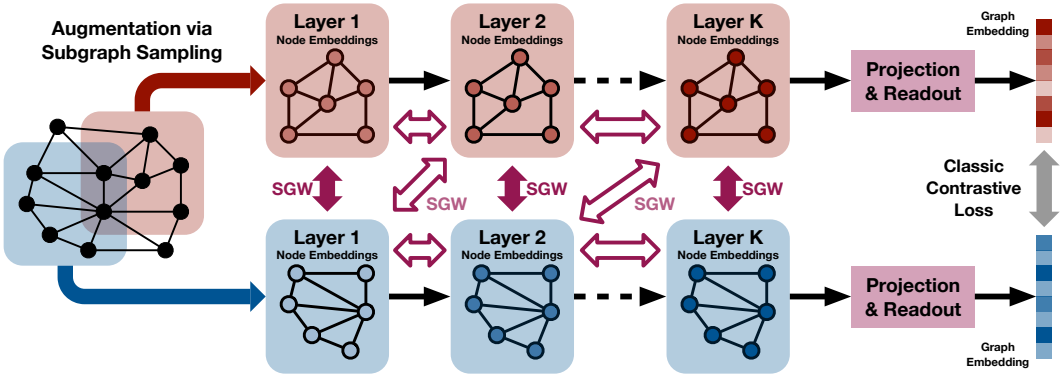
$$\hat{I}_N(Z, X; f) = \frac{1}{N}\sum_{n}\log\frac{p(x_n, z_n)}{p(x_n)p(z_n)} = \frac{1}{N}\sum_{n}\log\frac{N\sum_m \kappa(x_n, x_m)\kappa(z_n, z_m)}{\sum_m \kappa(x_n, x_m)\sum_m \kappa(z_n, z_m)}.$$

When using RBF kernel with bandwidth τ, we have

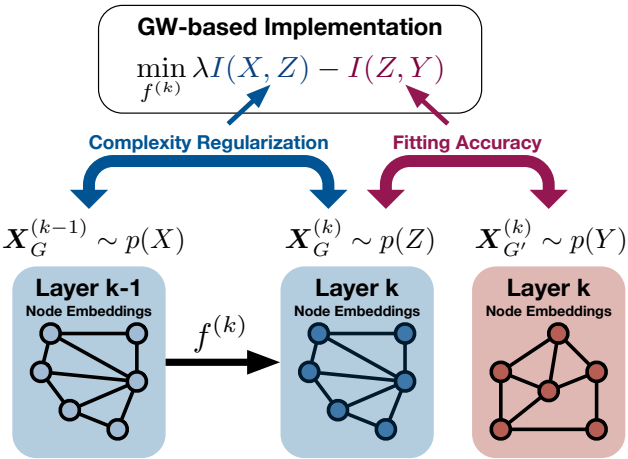
$$\hat{I}_N(Z, X) \leq \frac{1}{2\tau^2}\Big(\frac{1}{N^2}\|\boldsymbol{D}_X - \boldsymbol{D}_Z\|_F^2 - \mathcal{GW}_2^2(\boldsymbol{D}_X, \boldsymbol{D}_Z)\Big) + C_N = \frac{\mathcal{GMG}_\rho^{d,r}(f)}{2\tau^2} + C_N. \quad (47)$$

Revisiting Counterfactual Regression through the Lens of Gromov-Wasserstein Information Bottleneck. Arxiv, 2024.

Application in Graph Contrastive Learning



Application in Graph Contrastive Learning



- Complexity regularization:

$$I(X,Z) \leftarrow \mathcal{GMG}(f^{(k)}) \quad (48)$$

- Fitting Accuracy:

$$I(Z,Y) \leftarrow -\mathcal{W}(Z,Y) \quad (49)$$

- We can apply sliced Wasserstein and sliced GW to reduce computational costs.

Graph-level Classification

Datasets	Biochemical Molecular Graphs				Social Networks			
	MUTAG	DD	PROTEINS	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
DGK	87.44$_{\pm 2.72}$	-	73.30$_{\pm 0.82}$	80.31$_{\pm 0.46}$	-	66.96$_{\pm 0.56}$	78.04$_{\pm 0.39}$	41.27$_{\pm 0.18}$
WL	80.72$_{\pm 3.00}$	-	72.92$_{\pm 0.56}$	80.01$_{\pm 0.50}$	-	72.30$_{\pm 3.44}$	68.82$_{\pm 0.41}$	46.06$_{\pm 0.21}$
Graph2Vec	83.15$_{\pm 9.25}$	-	73.30$_{\pm 2.05}$	73.22$_{\pm 1.81}$	-	71.10$_{\pm 0.54}$	75.78$_{\pm 1.03}$	47.86$_{\pm 0.26}$
InfoGraph	89.01$_{\pm 1.13}$	72.85$_{\pm 1.78}$	74.44$_{\pm 0.31}$	76.20$_{\pm 1.06}$	70.65$_{\pm 1.13}$	73.03$_{\pm 0.87}$	82.50$_{\pm 1.42}$	53.46$_{\pm 1.03}$
JOAOv2	87.67$_{\pm 0.79}$	77.40$_{\pm 1.15}$	74.07$_{\pm 1.10}$	78.36$_{\pm 0.53}$	69.33$_{\pm 0.34}$	70.83$_{\pm 0.25}$	86.42$_{\pm 1.45}$	56.03$_{\pm 0.27}$
InfoGraph	89.01$_{\pm 1.13}$	72.85$_{\pm 1.78}$	74.44$_{\pm 0.31}$	76.20$_{\pm 1.06}$	70.65$_{\pm 1.13}$	73.03$_{\pm 0.87}$	82.50$_{\pm 1.42}$	53.46$_{\pm 1.03}$
AD-GCL	88.74$_{\pm 1.85}$	75.79$_{\pm 0.87}$	73.28$_{\pm 0.47}$	73.91$_{\pm 0.77}$	72.02$_{\pm 0.56}$	70.21$_{\pm 0.68}$	90.07$_{\pm 0.85}$	54.33$_{\pm 0.32}$
GraphACL	89.88$_{\pm 1.07}$	79.05$_{\pm 0.51}$	75.29$_{\pm 0.46}$	-	74.26$_{\pm 0.48}$	74.53$_{\pm 0.39}$	-	-
AutoGCL	85.15$_{\pm 1.10}$	75.75$_{\pm 0.60}$	69.73$_{\pm 0.40}$	78.32$_{\pm 0.50}$	71.40$_{\pm 0.70}$	72.00$_{\pm 0.40}$	86.60$_{\pm 1.50}$	55.71$_{\pm 0.20}$
HGCL	90.10$_{\pm 0.80}$	79.20$_{\pm 0.60}$	75.50$_{\pm 0.50}$	-	75.80$_{\pm 0.40}$	73.90$_{\pm 0.70}$	-	-
GCL-SPAN	85.00$_{\pm 0.80}$	78.78$_{\pm 0.50}$	75.78$_{\pm 0.40}$	75.43$_{\pm 0.40}$	71.40$_{\pm 0.50}$	66.00$_{\pm 0.70}$	86.50$_{\pm 0.10}$	54.10$_{\pm 0.50}$
GCS	88.19$_{\pm 0.90}$	76.28$_{\pm 0.30}$	74.04$_{\pm 0.40}$	77.18$_{\pm 0.30}$	74.00$_{\pm 0.40}$	72.90$_{\pm 0.50}$	86.50$_{\pm 0.30}$	56.30$_{\pm 0.30}$
SEGA	90.21$_{\pm 0.66}$	78.76$_{\pm 0.57}$	76.01$_{\pm 0.42}$	79.00$_{\pm 0.72}$	74.12$_{\pm 0.47}$	73.58$_{\pm 0.44}$	90.21$_{\pm 0.65}$	56.13$_{\pm 0.49}$
GraphCL	86.80$_{\pm 1.34}$	78.62$_{\pm 0.40}$	74.39$_{\pm 0.45}$	77.87$_{\pm 0.41}$	71.36$_{\pm 1.15}$	71.14$_{\pm 0.44}$	89.53$_{\pm 0.84}$	55.99$_{\pm 0.48}$
w. AIOTB	**91.30**$_{\pm 0.86}$	**79.30**$_{\pm 0.31}$	**75.85**$_{\pm 0.27}$	**79.57**$_{\pm 0.31}$	**74.10**$_{\pm 1.02}$	**73.65**$_{\pm 0.68}$	**90.57**$_{\pm 0.86}$	**56.62**$_{\pm 0.57}$
SimGRACE	89.01$_{\pm 1.31}$	77.44$_{\pm 1.11}$	75.35$_{\pm 0.09}$	79.12$_{\pm 0.44}$	71.72$_{\pm 0.82}$	71.30$_{\pm 0.77}$	89.51$_{\pm 0.89}$	55.91$_{\pm 0.34}$
w. AIOTB	**91.87**$_{\pm 0.80}$	**79.26**$_{\pm 0.62}$	**76.33**$_{\pm 0.36}$	**80.45**$_{\pm 0.62}$	**74.33**$_{\pm 0.80}$	**74.01**$_{\pm 0.70}$	**91.43**$_{\pm 0.80}$	**57.06**$_{\pm 0.44}$
RGCL	87.66$_{\pm 1.01}$	78.86$_{\pm 0.48}$	75.03$_{\pm 0.43}$	78.14$_{\pm 1.08}$	70.92$_{\pm 0.65}$	71.85$_{\pm 0.84}$	90.34$_{\pm 0.58}$	56.38$_{\pm 0.40}$
w. AIOTB	**91.44**$_{\pm 0.91}$	**79.77**$_{\pm 0.40}$	**76.35**$_{\pm 0.39}$	**79.87**$_{\pm 0.52}$	**73.98**$_{\pm 0.74}$	**74.45**$_{\pm 0.86}$	**91.81**$_{\pm 0.45}$	**57.20**$_{\pm 0.58}$

Graph-level Regression

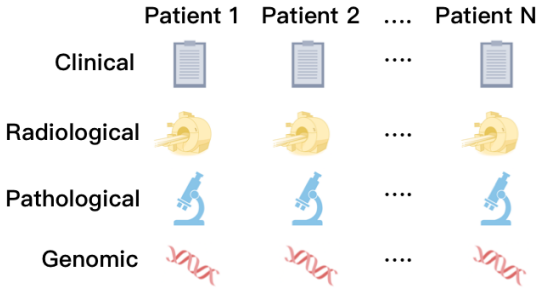
Task Types	Regression (RMSE↓)			Classification (ROC-AUC %↑)				
Datasets	molesol	mollipo	molfreesolv	molbace	molbbbp	molclintox	moltox21	molsider
#Graphs	1,128	4,200	642	1,513	2,039	1,477	7,831	1,427
Avg. #Nodes	13.3	27.0	8.7	34.1	24.1	26.2	18.6	33.6
Avg. Node Degree	13.7	29.5	8.4	36.9	26.0	27.9	19.3	35.4
InfoGraph	$1.34_{\pm 0.18}$	$1.01_{\pm 0.02}$	$10.01_{\pm 4.82}$	$74.74_{\pm 3.60}$	$66.33_{\pm 2.79}$	$64.50_{\pm 5.32}$	$69.74_{\pm 0.57}$	$60.54_{\pm 0.90}$
MVGRL	$1.43_{\pm 0.15}$	$0.96_{\pm 0.04}$	$9.02_{\pm 1.98}$	$74.20_{\pm 2.31}$	$67.24_{\pm 1.39}$	$73.84_{\pm 4.25}$	$70.48_{\pm 0.83}$	$61.94_{\pm 0.94}$
JOAO	$1.29_{\pm 0.12}$	$0.87_{\pm 0.03}$	$5.13_{\pm 0.72}$	$74.43_{\pm 1.94}$	$67.62_{\pm 1.29}$	$78.21_{\pm 4.12}$	$71.83_{\pm 0.92}$	$62.73_{\pm 0.92}$
GCL-SPAN	$1.22_{\pm 0.05}$	$0.80_{\pm 0.02}$	$4.53_{\pm 0.46}$	$76.74_{\pm 2.02}$	$69.59_{\pm 1.34}$	$80.28_{\pm 2.42}$	$72.83_{\pm 0.62}$	$64.87_{\pm 0.88}$
AD-GCL	$1.22_{\pm 0.09}$	$0.84_{\pm 0.03}$	$5.15_{\pm 0.62}$	$76.37_{\pm 2.03}$	$68.24_{\pm 1.47}$	$80.77_{\pm 3.92}$	$71.42_{\pm 0.73}$	$63.19_{\pm 0.95}$
GraphCL	$1.27_{\pm 0.09}$	$1.14_{\pm 0.02}$	$7.68_{\pm 2.75}$	$74.32_{\pm 2.70}$	$68.22_{\pm 1.89}$	$74.92_{\pm 4.42}$	$71.92_{\pm 1.01}$	$61.25_{\pm 1.11}$
w. AIOTB	**$1.20_{\pm 0.12}$**	**$1.06_{\pm 0.06}$**	**$5.13_{\pm 1.52}$**	**$76.87_{\pm 3.40}$**	**$69.44_{\pm 1.80}$**	**$77.30_{\pm 4.10}$**	**$72.63_{\pm 0.97}$**	**$62.80_{\pm 0.88}$**
SimGRACE	$1.30_{\pm 0.04}$	$1.03_{\pm 0.03}$	$5.12_{\pm 0.71}$	$76.44_{\pm 2.89}$	$69.08_{\pm 1.11}$	$81.03_{\pm 4.30}$	$72.55_{\pm 0.44}$	$62.64_{\pm 0.82}$
w. AIOTB	**$1.22_{\pm 0.05}$**	**$0.91_{\pm 0.03}$**	**$4.48_{\pm 0.68}$**	**$77.52_{\pm 3.01}$**	**$69.75_{\pm 0.92}$**	**$82.56_{\pm 4.10}$**	**$73.30_{\pm 0.62}$**	**$63.42_{\pm 0.80}$**
RGCL	$1.26_{\pm 0.09}$	$1.12_{\pm 0.04}$	$5.69_{\pm 0.67}$	$76.46_{\pm 0.67}$	$70.33_{\pm 1.08}$	$78.97_{\pm 4.65}$	$72.27_{\pm 0.84}$	$61.90_{\pm 1.05}$
w. AIOTB	**$1.20_{\pm 0.07}$**	**$1.01_{\pm 0.05}$**	**$4.62_{\pm 0.65}$**	**$77.20_{\pm 0.95}$**	**$71.21_{\pm 0.61}$**	**$80.05_{\pm 4.50}$**	**$72.94_{\pm 0.86}$**	**$63.20_{\pm 1.13}$**

Node-level Classification

Datasets	Cora	Citeseer	Pubmed	Squirrel	Chameleon	Texas
#Nodes	2,708	3,312	19,717	5,201	2,277	183
#Edges	5,429	4,732	44,338	198,423	31,371	279
#Features	1,433	3,703	500	2,089	2,325	1,703
GCN	87.14	79.60	86.19	34.80	58.82	74.59
w. AIOTB	**87.68**	**79.86**	**86.70**	**35.65**	**59.33**	**74.85**
GAT	88.03	80.52	85.20	35.89	58.73	76.39
w. AIOTB	**88.41**	**81.05**	**85.88**	**36.26**	**59.08**	**77.84**
ChebNet	85.94	79.15	87.95	36.81	56.37	83.61
w. AIOTB	**86.54**	**79.59**	**88.46**	**36.75**	**56.63**	**84.43**
BernNet	87.13	79.92	86.63	46.22	67.33	89.67
w. AIOTB	**88.12**	**80.63**	**87.32**	**47.07**	**67.96**	**90.98**

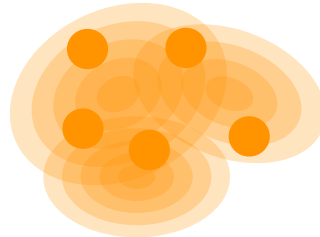
When representing multi-modal data, besides bridging sample and latent spaces, we also need to compare and align distributions across modalities.

Two (Questionable) Assumptions on Multi-modal Learning



Well-aligned multi-modal data

$$\mathbf{Z}_i \sim p_z \ \forall \ i$$

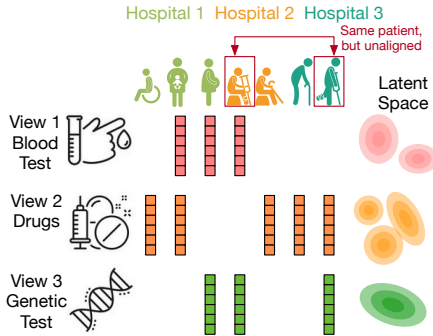


Shared latent distribution

Real-world Multi-modal Scenarios

Take healthcare data as an example:

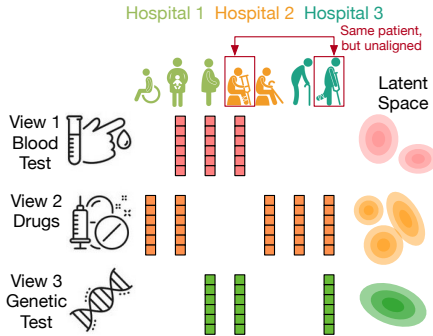
- For patients:
  - Only do some tests
  - Have admissions in different hospitals
- For hospitals:
  - Collect and store data independently from different hospitals
  - Complementary and heterogeneous modalities



Real-world Multi-modal Scenarios

Take healthcare data as an example:

- For patients:
  - Only do some tests
  - Have admissions in different hospitals
- For hospitals:
  - Collect and store data independently from different hospitals
  - Complementary and heterogeneous modalities



Unaligned and incomplete samples + Clustered modalities in (incomparable) latent spaces.

- Align samples across different modalities (Alignment)
- Cluster modalities and samples jointly (Co-clustering)

Optimal transport provide potential solutions.

Traditional Multi-modal Learning Paradigms

- Multi-modal data $[\boldsymbol{X}_1, ..., \boldsymbol{X}_S] \in \mathbb{R}^{N \times (D_1 + ... + D_S)}$.
- Learn latent representations implicitly or learn S encoders $\{f_s : \mathbb{R}^{D_s} \mapsto \mathcal{Z}\}_{s=1}^S$.

Multi-kernel Fusion (MKF): Learn the encoders implicitly

$$\max_{\boldsymbol{U}, \{\alpha_s\}_{s=1}^S} \text{tr}(\boldsymbol{U}^\top \bar{\boldsymbol{K}} \boldsymbol{U}), \quad s.t.\ \bar{\boldsymbol{K}} = \sum\nolimits_{s=1}^{S} \alpha_s \boldsymbol{K}_s. \tag{50}$$

Canonical Correlation Analysis (CCA):

$$\begin{aligned} &\min_{\{f_s, U_s\}_{s=1}^S} \sum\nolimits_{s \neq s'} \|U_s \circ f_s(\boldsymbol{X}_s) - U_{s'} \circ f_{s'}(\boldsymbol{X}_{s'})\|_F^2, \\ &s.t.\ (U_s \circ f_s(\boldsymbol{X}_s))^\top U_s \circ f_s(\boldsymbol{X}_s) = \boldsymbol{I},\ \forall s \end{aligned} \tag{51}$$

Traditional Multi-modal Learning Paradigms

- Multi-modal data $[\boldsymbol{X}_1, ..., \boldsymbol{X}_S] \in \mathbb{R}^{N \times (D_1 + ... + D_S)}$.
- Learn latent representations implicitly or learn S encoders $\{f_s : \mathbb{R}^{D_s} \mapsto \mathcal{Z}\}_{s=1}^S$.

Multi-kernel Fusion (MKF): Learn the encoders implicitly

$$\max_{\boldsymbol{U}, \{\alpha_s\}_{s=1}^S} \text{tr}(\boldsymbol{U}^\top \bar{\boldsymbol{K}} \boldsymbol{U}), \quad s.t.\ \bar{\boldsymbol{K}} = \sum\nolimits_{s=1}^{S} \alpha_s \boldsymbol{K}_s. \tag{50}$$

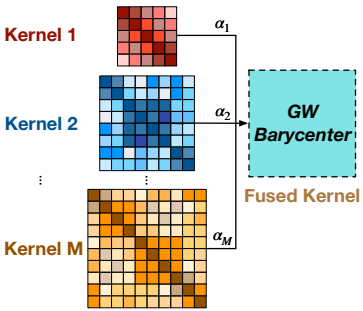
Canonical Correlation Analysis (CCA):

$$\begin{aligned} &\min_{\{f_s, U_s\}_{s=1}^S} \sum\nolimits_{s \neq s'} \|U_s \circ f_s(\boldsymbol{X}_s) - U_{s'} \circ f_{s'}(\boldsymbol{X}_{s'})\|_F^2, \\ &s.t.\ (U_s \circ f_s(\boldsymbol{X}_s))^\top U_s \circ f_s(\boldsymbol{X}_s) = \boldsymbol{I},\ \forall s \end{aligned} \tag{51}$$

- How to make them applicable for unaligned data?
- How to introduce modality-level clustering structure?

Extend MKF to Unaligned Data via GW Barycenters

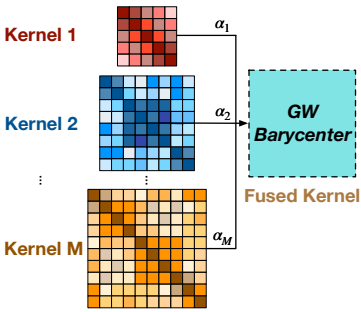
Fuse kernels by solving a GW barycenter problem:



$$\max_{\boldsymbol{U},\{\alpha_s\}_{s=1}^{S}} \operatorname{tr}(\boldsymbol{U}^\top \boldsymbol{K}\boldsymbol{U}),$$

$$s.t.\ \bar{\boldsymbol{K}} \in \underbrace{\min_{\boldsymbol{K}} \sum_{s=1}^{S} \alpha_s \mathcal{GW}_2^2(\boldsymbol{K}, \boldsymbol{K}_s)}_{\text{GW barycenter}}. \tag{52}$$

Extend MKF to Unaligned Data via GW Barycenters



Fuse kernels by solving a GW barycenter problem:

$$\max_{\boldsymbol{U},\{\alpha_s\}_{s=1}^S} \operatorname{tr}(\boldsymbol{U}^\top \boldsymbol{K}\boldsymbol{U}),$$
$$s.t.\ \bar{\boldsymbol{K}} \in \underbrace{\min_{\boldsymbol{K}} \sum\nolimits_{s=1}^{S} \alpha_s \mathcal{GW}_2^2(\boldsymbol{K}, \boldsymbol{K}_s)}_{\text{GW barycenter}}. \tag{52}$$

Nested optimization:

1. Compute the barycenter iteratively

$$\bar{\boldsymbol{K}} \leftarrow \frac{1}{S^2}\sum_{s=1}^{S} \alpha_s \boldsymbol{T}_s^* \boldsymbol{K}_s (\boldsymbol{T}_s^*)^\top,\ \boldsymbol{T}_s^* \leftarrow \mathcal{GW}_2(\bar{\boldsymbol{K}}, \boldsymbol{K}_s). \tag{53}$$

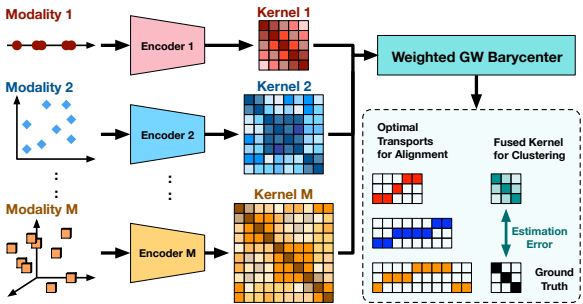
2. Plug the barycenter into the objective function:

$$\max_{\boldsymbol{U},\{\alpha_s\}_{s=1}^S} \operatorname{tr}\Big(\boldsymbol{U}^\top \Big(\sum\nolimits_{s=1}^{S} \alpha_s \boldsymbol{T}_s^* \boldsymbol{K}_s (\boldsymbol{T}_s^*)^\top\Big)\boldsymbol{U}\Big). \tag{54}$$

Extend MKF to Unaligned Data via GW Barycenters

When computing the kernels by latent codes, we obtain parametric kernels and the **Gromov-Wasserstein multi-modal alignment and clustering model:**

$$\underbrace{\max_{\boldsymbol{U}\in\Pi,\{f_s\}_{s=1}^{S}} \mathrm{tr}(\boldsymbol{U}^\top \boldsymbol{K}\boldsymbol{U})}_{\Leftrightarrow \min \mathcal{GW}_2^2(\bar{\boldsymbol{K}},\boldsymbol{I}_C)}, \quad s.t.\ \bar{\boldsymbol{K}} \in \min_{\boldsymbol{K}} \sum\nolimits_{s=1}^{S} \alpha_s \mathcal{GW}_2^2(\boldsymbol{K}, \underbrace{\boldsymbol{K}(f_s)}_{\text{param. kernel}}). \tag{55}$$



Gromov-Wasserstein multi-modal alignment and clustering. CIKM, 2022.

Multi-modal Clustering Performance

Data type	Datasets	HandWritten		Caltech 7		ORL		Movies		Prokaryotic	
	Algorithms	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Well-aligned ($\beta = 0$)	MCCA	0.8269	0.7775	**0.5313**	0.4716	0.3475	0.4992	0.0989	0.0722	0.5620	0.1204
	DCCAE	0.6537	0.6216	0.4110	0.3850	0.5625	0.7373	0.1572	0.1194	0.5070	0.1827
	AttnAE	0.7505	0.6912	0.4600	0.4575	0.4600	0.6603	0.1880	0.1918	0.5390	0.2625
	MVKSC	0.6749	0.6376	0.5196	0.2537	0.3013	0.5291	**0.2285**	**0.2098**	**0.6188**	**0.3191**
	MultiNMF	**0.8882**	**0.8279**	0.4525	**0.5120**	**0.6900**	**0.8100**	0.1726	0.1856	0.5771	0.2495
50% unaligned ($\beta = 0.5$)	CPM-GAN	0.7250	0.6069	0.3472	0.3151	0.1987	0.3703	0.1210	0.1753	0.3793	**0.3294**
	MVC-UM	-	-	**0.3958**	0.3838	**0.5863**	**0.7586**	0.1831	0.1950	0.3950	0.0807
	GWMAC	**0.8469**	**0.8156**	0.3541	**0.5010**	0.5322	0.7068	**0.1993**	**0.2195**	**0.5515**	0.3286
100% unaligned ($\beta = 1$)	MVC-UM	-	-	0.3112	0.2456	**0.5431**	**0.7452**	0.1841	0.1953	0.4451	0.0554
	GWMAC	**0.8144**	**0.7546**	**0.3568**	**0.4945**	0.5118	0.7026	**0.1928**	**0.2138**	**0.5479**	**0.3259**

Extend CCA to Unaligned Data via Sliced Wasserstein

Sliced Wasserstein Canonical Correlation Analysis (SW-CCA):

$$\begin{aligned} &\min_{\{f_s, U_s\}_{s=1}^S} \sum_{s \neq s'} \mathcal{SW}_2^2(U_s \circ f_s(\boldsymbol{X}_s), U_{s'} \circ f_{s'}(\boldsymbol{X}_{s'})), \\ &s.t.\ (U_s \circ f_s(\boldsymbol{X}_s))^\top U_s \circ f_s(\boldsymbol{X}_s) = \boldsymbol{I},\ \forall s \end{aligned} \tag{56}$$

- Using SW distance does not require aligned data.
- It is differentiable, just requiring random projections and sorting operations.

Extend CCA to Unaligned Data via Sliced Wasserstein

Sliced Wasserstein Canonical Correlation Analysis (SW-CCA):

$$\begin{aligned} &\min_{\{f_s,U_s\}_{s=1}^S} \sum\nolimits_{s\neq s'} \mathcal{SW}_2^2(U_s \circ f_s(\boldsymbol{X}_s), U_{s'} \circ f_{s'}(\boldsymbol{X}_{s'})), \\ &s.t.\ (U_s \circ f_s(\boldsymbol{X}_s))^\top U_s \circ f_s(\boldsymbol{X}_s) = \boldsymbol{I},\ \forall s \end{aligned} \tag{56}$$

- Using SW distance does not require aligned data.
- It is differentiable, just requiring random projections and sorting operations.

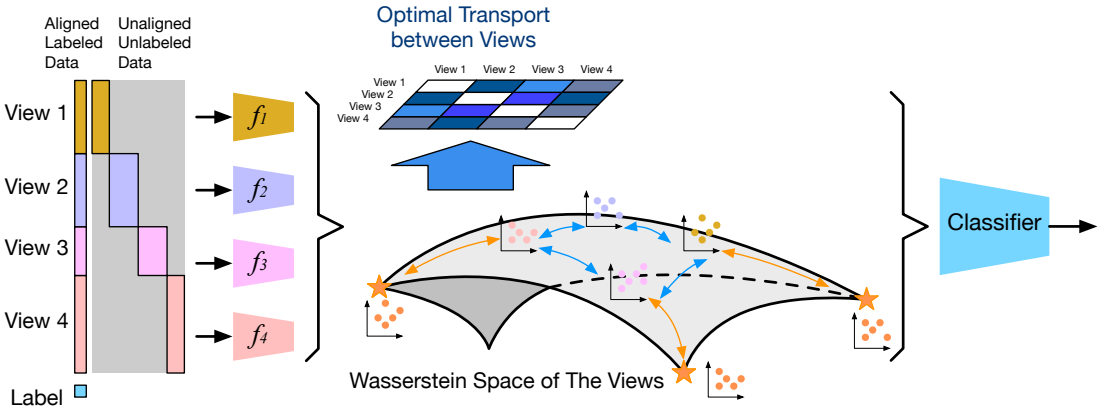
Max-Sliced Wasserstein Canonical Correlation Analysis (MSW-CCA):

$$\begin{aligned} &\min_{\{f_s\}_{s=1}^S} \sum\nolimits_{s\neq s'} \mathcal{MSW}_2^2(f_s(\boldsymbol{X}_s), f_{s'}(\boldsymbol{X}_{s'})), \\ &s.t.\ (U_s \circ f_s(\boldsymbol{X}_s))^\top U_s \circ f_s(\boldsymbol{X}_s) = \boldsymbol{I},\ \forall s \end{aligned} \tag{57}$$

- Treat U_s as a linear random projector, i.e., $U_s : \mathcal{Z} \mapsto \mathbb{R}$, and learn it in an adversarial way, we have

Differentiable Hierarchical Optimal Transport for Robust Multi-View Learning. TPAMI, 2022.

Hierarchical Optimal Transport for Modality Clustering



Principle:

- Further extend SW-CCA
- Capture the relations among the modalities by their OT distances.

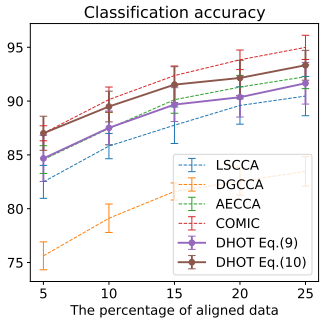
Hierarchical Optimal Transport for Modality Clustering

Extend SW-CCA: Learn the pairwise relations between different modalities.

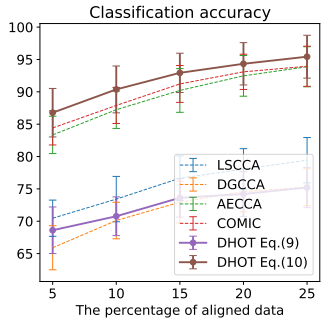
$$\min_{\substack{\{f_s,U_s\}_{s=1}^S,\\ \boldsymbol{W}\in\Pi(\frac{1}{S}1_S,\frac{1}{S}1_S)}} \underbrace{\sum\nolimits_{s\neq s'} w_{ss'}\mathcal{SW}_2^2(U_s\circ f_s(\boldsymbol{X}_s),U_{s'}\circ f_{s'}(\boldsymbol{X}_{s'}))}_{\text{Hierarchical OT}} + \alpha\underbrace{\Big\|\sum\nolimits_s (U_s\circ f_s(\boldsymbol{X}_s))^\top U_s\circ f_s(\boldsymbol{X}_s)-\boldsymbol{I}\Big\|_F^2}_{\text{CCA-Regularizer}} + \beta\underbrace{H(\boldsymbol{W})}_{\langle \boldsymbol{W},\log \boldsymbol{W}\rangle}. \tag{58}$$

- **Lower level:** the SW distance between different modalities' sample sets.
- **Upper level:** Take the SW distances as the cost matrix, compute the EOT between the group of modalities and itself. (Set $w_{ss}=0$ to avoid trivial solutions)
- $\boldsymbol{W}^*$ indicates the clustering structure implicitly by the pairwise similarity between different modalities.

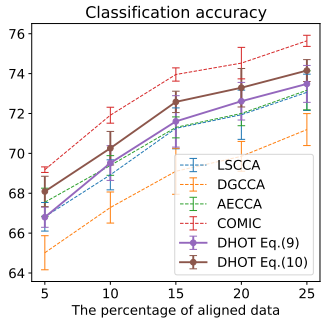
Performance on Multi-modal Classification



Caltech7



Handwritten



Cathgen

Differentiable Hierarchical Optimal Transport for Robust Multi-View Learning. TPAMI, 2022.

Summary

- Representation model can be effectively regularized to approach OT map
  - Monge gap and its Gromovization provides a potential solution
  - Achieve promising solution in graph representation learning

Summary

- Representation model can be effectively regularized to approach OT map
  - Monge gap and its Gromovization provides a potential solution
  - Achieve promising solution in graph representation learning

- Gromovize $\mathcal{W}_p$ leads to $\mathcal{GW}_p$.
  - The algorithms of $\mathcal{W}_p$ are applicable under slight modification
  - The problem becomes non-convex but the algorithms still lead to stationary points

Summary

- Representation model can be effectively regularized to approach OT map
  - Monge gap and its Gromovization provides a potential solution
  - Achieve promising solution in graph representation learning
- Gromovize $\mathcal{W}_p$ leads to $\mathcal{GW}_p$.
  - The algorithms of $\mathcal{W}_p$ are applicable under slight modification
  - The problem becomes non-convex but the algorithms still lead to stationary points
- In multi-modal learning, OT distances help align and cluster different modalities.
  - Robust to unaligned multi-modal samples
  - Hierarchical optimal transport leads to a joint framework for sample- and modality-level learning

Thanks!
5-min break and QA

Part 1 **Computational Optimal Transport (Hongteng Xu)**

- Preliminaries and basic concepts
- Typical computation methods

Part 2 **Representation Learning Driven by OT (Dixin Luo)**

- OT-based multi-modal learning
- Monge gap and its Gromovization for information bottleneck

Part 3 **Neural Network Design Driven by OT (Minjie Cheng)**

- OT-based Transformer
- OT-based graph neural network

Part 4 **Recent Progress in Generative Modeling (Hongteng Xu)**

- OT-based flow matching
- Applications of optimal acceleration transport

Neural Network Design: Engineering or Art?

The progress of AI is mainly attributed to the development of model architectures.

- Vision: AlexNet, VGG, ResNet, ViT, ...
- NLP: RNN, LSTM, BERT, GPT, ...
- Graph: Spatial and Spectral GNNs, Graph Transformer, ...

Essentially, the models serve to transform one data distribution to another.

Neural Network Design: Engineering or Art?

The progress of AI is mainly attributed to the development of model architectures.

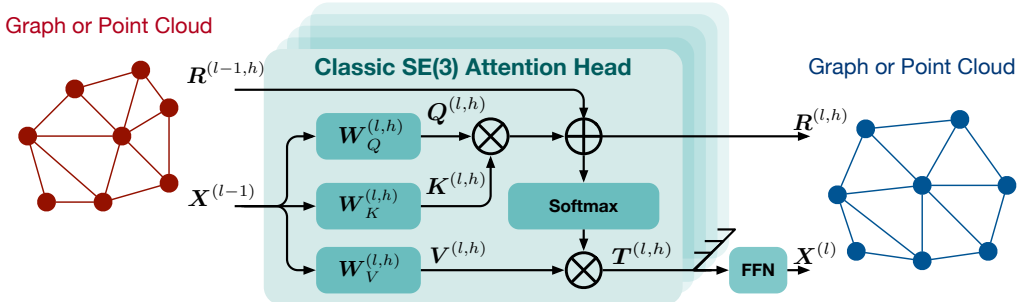
- Vision: AlexNet, VGG, ResNet, ViT, ...
- NLP: RNN, LSTM, BERT, GPT, ...
- Graph: Spatial and Spectral GNNs, Graph Transformer, ...

Essentially, the models serve to transform one data distribution to another.

However, till now, we only summarize very coarse and empirical design principle for neural networks.

- The deeper, the larger, the better (Scaling Laws).
- Tricks: Dropout, Batchnorm, Non-smooth activations, Residual Connection, ...

A Typical Example: SE(3)-Transformer



The Core of 3D Molecular Models (e.g., Uni-Mol)

- **Pros:** Large capacity, strong representation power, SE(3)-equivariance, ...
- **Cons:** High computational complexity, poor interpretability, ...

Uni-Mol: A Universal 3D Molecular Representation Learning Framework. ICLR, 2023.

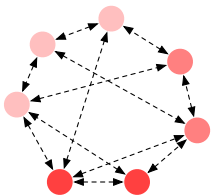
Motivation

Essentially, many existing NN layers work for information fusion

Motivation

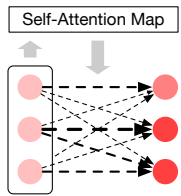
Essentially, many existing NN layers work for information fusion

Message-Passing



Fusion on graph

Self-Attention

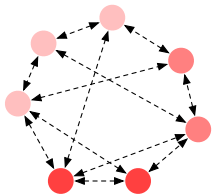


Fusion in a continuous space

Motivation

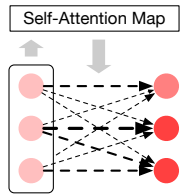
Essentially, many existing NN layers work for information fusion

Message-Passing



Fusion on graph

Self-Attention



Fusion in a continuous space

Develop OT-based surrogates for above layers, improving interpretability and boosting performance

- Explore the **alignment principle** of information fusion through the lens of OT
- Connect the alignment principle to **optimization**

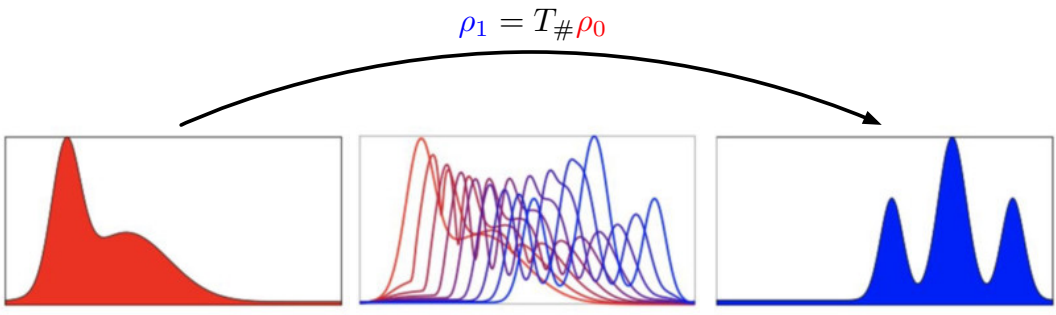
Outline

1 **Optimal Transport Driven Transformer**
- Improved Transformer Based on Wasserstein Gradient Flow
- Extension for deep geometric learning

2 **Optimal Transport Driven GNN**
- Optimal Transport on Graph: From continuous to discrete structured scenarios
- Label Flow and Its Amortization for GNNs

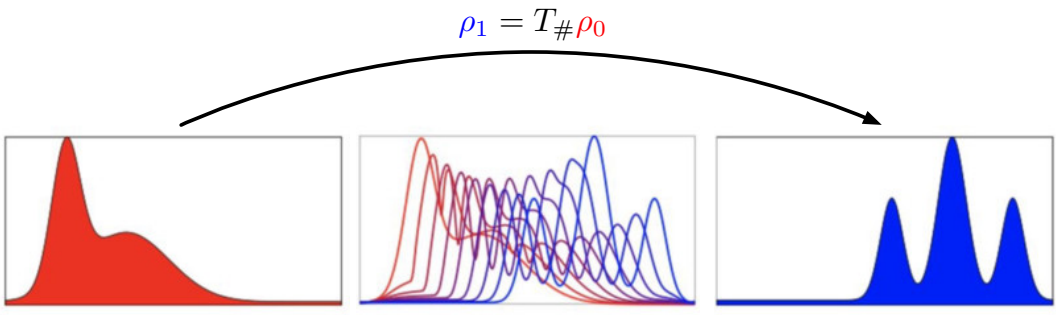
Revisit the Dynamic Definition of OT

The displacement interpolation determined by transport map T:



Revisit the Dynamic Definition of OT

The displacement interpolation determined by transport map T:



What is the relationship between optimal transport and displacement interpolation?

Revisit The Dynamic Definition of OT

Definition 2 (Dynamic Formulation of Optimal Transport)

Let $\mathcal{X} \subset \mathbb{R}^d$ be the Euclidean sample space. For $\rho_0, \rho_1 \in \mathbb{P}(\mathcal{X})$, $\mathcal{W}_2^2(\rho_0, \rho_1)$ corresponds to seeking a unique least–kinetic–energy **flow (velocity field)** v:

$$\mathcal{W}_2^2(\rho_0, \rho_1) = \inf_{v(x,t)} \underbrace{\int_0^1 \int_{\mathcal{X}} \frac{1}{2} \rho(x,t) \|v(x,t)\|_2^2 \mathrm{d}x\mathrm{d}t}_{\textbf{Kinetic Energy}}, \quad s.t. \ \underbrace{\partial_t \rho + \nabla_x \cdot (v\rho) = 0}_{\textbf{Continuity Equation}} \tag{59}$$

A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. Numerische Mathematik, 2000.

Revisit The Dynamic Definition of OT

Definition 2 (Dynamic Formulation of Optimal Transport)

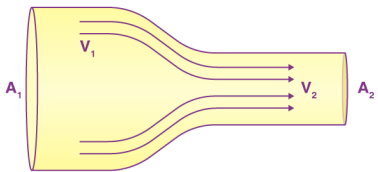
Let $\mathcal{X} \subset \mathbb{R}^d$ be the Euclidean sample space. For $\rho_0, \rho_1 \in \mathbb{P}(\mathcal{X})$, $\mathcal{W}_2^2(\rho_0, \rho_1)$ corresponds to seeking a unique least–kinetic–energy **flow (velocity field)** v:

$$\mathcal{W}_2^2(\rho_0, \rho_1) = \inf_{v(x,t)} \underbrace{\int_0^1 \int_{\mathcal{X}} \frac{1}{2} \rho(x,t) \|v(x,t)\|_2^2 \mathrm{d}x\mathrm{d}t}_{\textbf{Kinetic Energy}}, \quad s.t. \ \underbrace{\partial_t \rho + \nabla_x \cdot (v\rho) = 0}_{\textbf{Continuity Equation}} \tag{59}$$

A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. Numerische Mathematik, 2000.

- Solving the continuity equation with the **optimal flow** v^* leads to the **optimal displacement interpolation** between ρ_0 and ρ_1.

Continuity Equation

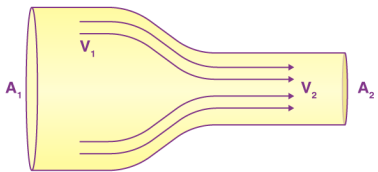


- Continuity equation describes the time rate of change of the fluid density $(\partial_t \rho(x,t))$ at a fixed point x in space.

$$\partial_t \rho + \nabla_x \cdot \big(v\rho\big) = 0 \tag{60}$$

- The rate equals to the rate of change of density by convection $(\nabla_x \cdot (v\rho))$.

Continuity Equation



- Continuity equation describes the time rate of change of the fluid density $(\partial_t \rho(x,t))$ at a fixed point x in space.

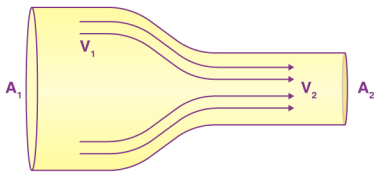
$$\partial_t \rho + \nabla_x \cdot \big(v\rho\big) = 0 \tag{60}$$

- The rate equals to the rate of change of density by convection $(\nabla_x \cdot (v\rho))$.

Given a sample at time t, i.e., $x_t \sim \rho_t$, we have

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v(x_t, t), \quad \underbrace{x_{t+\delta t} \approx x_t + \delta t \cdot v(x_t, t)}_{\text{Euler step}}. \tag{61}$$

Continuity Equation



- Continuity equation describes the time rate of change of the fluid density $(\partial_t \rho(x,t))$ at a fixed point x in space.

$$\partial_t \rho + \nabla_x \cdot \big(v\rho\big) = 0 \tag{60}$$

- The rate equals to the rate of change of density by convection $(\nabla_x \cdot (v\rho))$.

Given a sample at time t, i.e., $x_t \sim \rho_t$, we have

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v(x_t, t), \quad \underbrace{x_{t+\delta t} \approx x_t + \delta t \cdot v(x_t, t)}_{\text{Euler step}}. \tag{61}$$

Obviously, the keypoint is modeling the flow v. When the flow is a neural network, the Euler step corresponds to a ResNet.

Neural ordinary differential equations. NeurIPS, 2018.

Wasserstein Gradient Flow: Connect Flow to Energy Functional

- Let $E : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ be an energy functional with first variation $\frac{\delta E}{\delta \rho}$.
  - The first variation of Wasserstein space analogies to the gradient of Euclidean space.

Wasserstein Gradient Flow: Connect Flow to Energy Functional

- Let $E : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ be an energy functional with first variation $\frac{\delta E}{\delta \rho}$.
  - The first variation of Wasserstein space analogies to the gradient of Euclidean space.
- The Wasserstein gradient flow of E is defined by

$$\underbrace{\partial_t \rho = \nabla_x \cdot \Big(\rho \overbrace{\nabla_x \frac{\delta E}{\delta \rho}}^{-v} \Big)}_{\textbf{Continuity equation with } v = -\nabla_x \frac{\delta E}{\delta \rho}}$$

Wasserstein Gradient Flow: Connect Flow to Energy Functional

- Let $E : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ be an energy functional with first variation $\frac{\delta E}{\delta \rho}$.
  - The first variation of Wasserstein space analogies to the gradient of Euclidean space.
- The Wasserstein gradient flow of E is defined by

$$\underbrace{\partial_t \rho = \nabla_x \cdot \Big(\rho \overbrace{\nabla_x \frac{\delta E}{\delta \rho}}^{-v}\Big)}_{\textbf{Continuity equation with } v = -\nabla_x \frac{\delta E}{\delta \rho}} \quad \Leftrightarrow \quad \rho_{t+\delta_t} = \arg\min_{\rho \in \mathcal{P}(\mathcal{X})} E(\rho) + \frac{1}{2\delta_t} \mathcal{W}_2^2(\rho, \rho_t). \tag{62}$$

- Wasserstein gradient flow = Continuity equation with steepest descent velocity.

The variational formulation of the Fokker-Planck equation. SIAM Journal on Mathematical Analysis, 1998.

Wasserstein Gradient Flow: Connect Flow to Energy Functional

- Let $E : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ be an energy functional with first variation $\frac{\delta E}{\delta \rho}$.
  - The first variation of Wasserstein space analogies to the gradient of Euclidean space.
- The Wasserstein gradient flow of E is defined by

$$\underbrace{\partial_t \rho = \nabla_x \cdot \Big(\rho \overbrace{\nabla_x \frac{\delta E}{\delta \rho}}^{-v} \Big)}_{\textbf{Continuity equation with } v = -\nabla_x \frac{\delta E}{\delta \rho}} \quad \Leftrightarrow \quad \rho_{t+\delta_t} = \arg\min_{\rho \in \mathcal{P}(\mathcal{X})} E(\rho) + \frac{1}{2\delta_t} \mathcal{W}_2^2(\rho, \rho_t). \tag{62}$$

- Wasserstein gradient flow = Continuity equation with steepest descent velocity.

The variational formulation of the Fokker-Planck equation. SIAM Journal on Mathematical Analysis, 1998.

The design of v corresponds to the design of E.

Imitate Transformer by Wasserstein Gradient Flow

- A typical choice of energy functional: **the potential energy of particles**

$$E(\rho) = \frac{1}{2} \iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \rho(x)\rho(y)\mathrm{d}x\mathrm{d}y. \tag{63}$$

Imitate Transformer by Wasserstein Gradient Flow

- A typical choice of energy functional: **the potential energy of particles**

$$E(\rho) = \frac{1}{2} \iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \rho(x)\rho(y)\mathrm{d}x\mathrm{d}y. \tag{63}$$

- The flow v becomes

$$v = -\nabla_x \frac{\delta E}{\delta \rho} = -\nabla_x(\kappa * \rho)$$

Imitate Transformer by Wasserstein Gradient Flow

- A typical choice of energy functional: **the potential energy of particles**

$$E(\rho) = \frac{1}{2} \iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \rho(x)\rho(y)\mathrm{d}x\mathrm{d}y. \tag{63}$$

- The flow v becomes

$$v = -\nabla_x \frac{\delta E}{\delta \rho} = -\nabla_x(\kappa * \rho) \quad \Leftrightarrow \quad v(x) = -\frac{1}{2}\int_{\mathcal{X}} \nabla_x \kappa(x,y)\rho(y)\mathrm{d}y. \tag{64}$$

Imitate Transformer by Wasserstein Gradient Flow

- A typical choice of energy functional: **the potential energy of particles**

$$E(\rho) = \frac{1}{2} \iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \rho(x)\rho(y)\mathrm{d}x\mathrm{d}y. \tag{63}$$

- The flow v becomes

$$v = -\nabla_x \frac{\delta E}{\delta \rho} = -\nabla_x(\kappa * \rho) \quad \Leftrightarrow \quad v(x) = -\frac{1}{2}\int_{\mathcal{X}} \nabla_x \kappa(x,y)\rho(y)\mathrm{d}y. \tag{64}$$

- More specifically, when $\kappa(x,y) = \exp(x^\top W W^\top y)$, we have

$$v(x) = \int_{\mathcal{X}} \exp(x^\top \underbrace{W}_{:=W_Q^\top} \underbrace{W^\top}_{:=W_K} y)(\underbrace{-WW^\top}_{:=W_V}) y\rho(y)\mathrm{d}y. \tag{65}$$

Congratulations! Now, we have a continuous counterpart of an unnormalized attention layer with a structured QKV setting.

Sinkformer: From Imitation to Improvement

- Revisit the potential energy: $E(\rho) = \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \underbrace{\rho(x)\rho(y)}_{\text{Assumed Independency}} \mathrm{d}x\mathrm{d}y.$

Sinkformer: From Imitation to Improvement

- Revisit the potential energy: $E(\rho) = \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \underbrace{\rho(x)\rho(y)}_{\text{Assumed Independency}} \mathrm{d}x\mathrm{d}y.$
- Define an entropic OT-based energy

$$E^{\infty}(\rho) = \inf_{\pi\in\Pi(\rho,\rho)} \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} -\log\kappa(x,y)\,\underbrace{\pi(x,y)}_{\text{Coupling}}\mathrm{d}x\mathrm{d}y - \underbrace{H(\pi)}_{\text{Entropy}}$$

Sinkformer: From Imitation to Improvement

- Revisit the potential energy: $E(\rho) = \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \underbrace{\rho(x)\rho(y)}_{\text{Assumed Independency}} \mathrm{d}x\mathrm{d}y.$
- Define an entropic OT-based energy

$$
\begin{aligned}
E^{\infty}(\rho) &= \inf_{\pi\in\Pi(\rho,\rho)} \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} -\log\kappa(x,y)\,\underbrace{\pi(x,y)}_{\text{Coupling}}\mathrm{d}x\mathrm{d}y - \underbrace{H(\pi)}_{\text{Entropy}} \\
&= \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} -\log\kappa(x,y)\,\underbrace{\pi^{\infty}(x,y)}_{\text{OT plan}}\mathrm{d}x\mathrm{d}y - H(\pi^{\infty}) \qquad (66) \\
&= \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} \pi^{\infty}(x,y)\log\frac{\pi^{\infty}(x,y)}{\kappa(x,y)}\,\mathrm{d}x\mathrm{d}y.
\end{aligned}
$$

Sinkformer: From Imitation to Improvement

- Revisit the potential energy: $E(\rho) = \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\kappa(x,y)}_{\text{Kernel}} \underbrace{\rho(x)\rho(y)}_{\text{Assumed Independency}} \mathrm{d}x\mathrm{d}y.$

- Define an entropic OT-based energy

$$
\begin{aligned}
E^{\infty}(\rho) &= \inf_{\pi\in\Pi(\rho,\rho)} \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} -\log\kappa(x,y)\,\underbrace{\pi(x,y)}_{\text{Coupling}}\mathrm{d}x\mathrm{d}y - \underbrace{H(\pi)}_{\text{Entropy}} \\
&= \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} -\log\kappa(x,y)\,\underbrace{\pi^{\infty}(x,y)}_{\text{OT plan}}\mathrm{d}x\mathrm{d}y - H(\pi^{\infty}) \qquad (66) \\
&= \frac{1}{2}\iint_{\mathcal{X}\times\mathcal{X}} \pi^{\infty}(x,y)\log\frac{\pi^{\infty}(x,y)}{\kappa(x,y)}\,\mathrm{d}x\mathrm{d}y.
\end{aligned}
$$

- Sinkhorn scaling, i.e., $\pi^{\infty} = \underbrace{N_c \circ N_r \circ \cdots \circ N_c \circ N_r}_{M \text{ steps, with } M\to\infty}(\kappa)$.

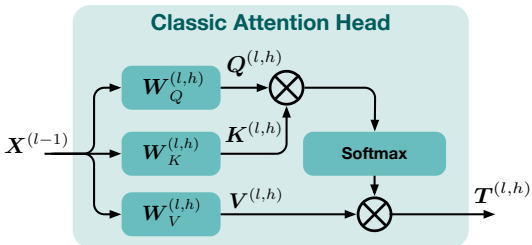
Sinkformer: From Imitation to Improvement

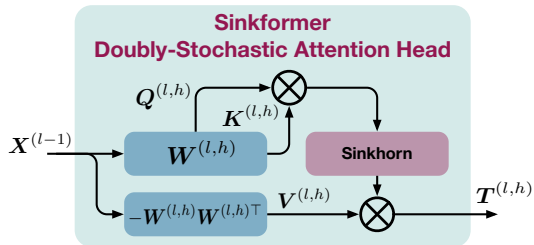
- When applying $E^{\infty}(\rho)$ and setting $\kappa(x,y) = \exp(x^{\top}WW^{\top}y)$, we have

$$v = -\nabla_x \frac{\delta E^{\infty}}{\delta \rho} = \int_{\mathcal{X}} \underbrace{\pi^{\infty}(x,y)}_{\text{Sinkhorn of } \kappa} (-WW^{\top}) y \rho(y) \mathrm{d}y. \tag{67}$$

- Given the input of the h-th head of the l-th layer, i.e., $\boldsymbol{X}^{(l-1)} \in \mathbb{R}^{N \times D}$, we have

$$\boldsymbol{T}^{(l,h)} = \mathsf{S}_M(\boldsymbol{X}^{(l-1)\top} \boldsymbol{W}^{(l,h)\top} \boldsymbol{W}^{(l,h)} \boldsymbol{X}^{(l-1)})(-\boldsymbol{W}^{(l,h)\top} \boldsymbol{W}^{(l,h)}) \boldsymbol{X}^{(l-1)}. \tag{68}$$

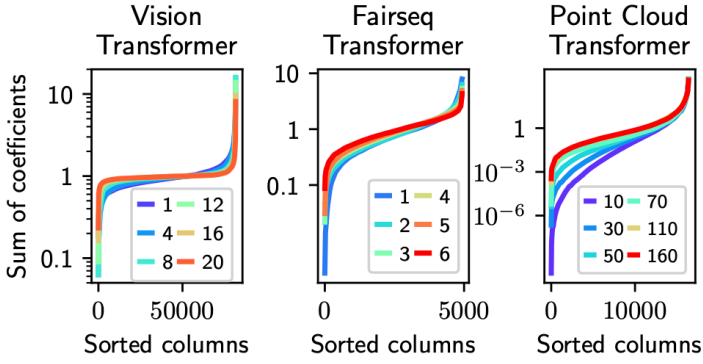




Sinkformers: Transformers with doubly stochastic attention. AISTATS, 2022.

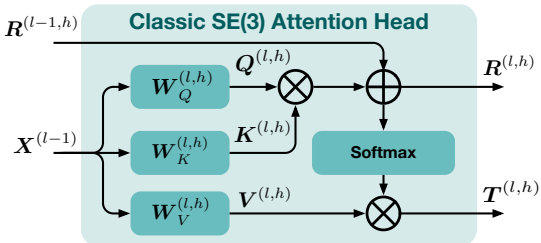
Sinkformer: From Imitation to Improvement

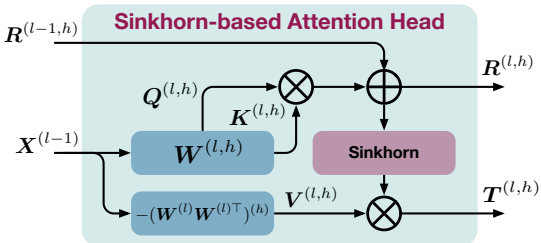
- Notably, apply doubly-stochastic attention map is reasonable — the classic Transformer tends to learn doubly-stochastic attention map during training.
- Sinkformer makes the tendency become a strict constraint.



Sinkformers: Transformers with doubly stochastic attention. AISTATS, 2022.

WGFormer: Extend and Improve Sinkformer to SE(3)-Transformer

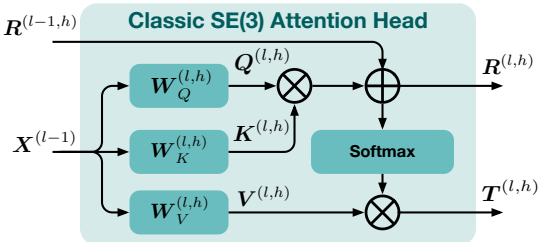


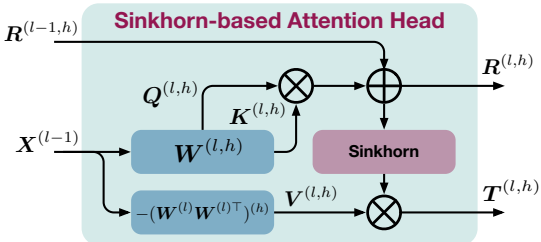


1) New "QKV" matrices:

$$\boldsymbol{Q}^{(l,h)} = \boldsymbol{K}^{(l,h)} = \boldsymbol{X}^{(l-1)}\boldsymbol{W}^{(l,h)}, \quad \boldsymbol{V}^{(l,h)} = -\boldsymbol{X}^{(l-1)}(\boldsymbol{W}^{(l)}\boldsymbol{W}^{(l)\top})^{(h)}. \tag{69}$$

WGFormer: Extend and Improve Sinkformer to SE(3)-Transformer





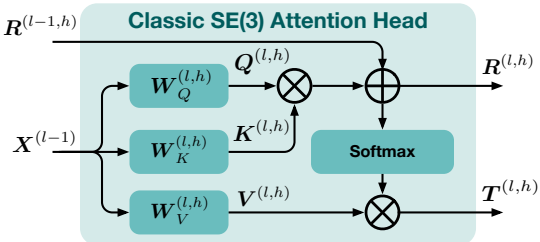
1) New "QKV" matrices:

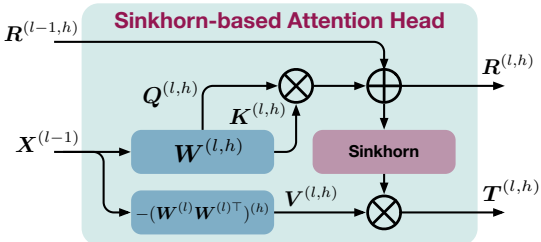
$$\boldsymbol{Q}^{(l,h)} = \boldsymbol{K}^{(l,h)} = \boldsymbol{X}^{(l-1)}\boldsymbol{W}^{(l,h)}, \quad \boldsymbol{V}^{(l,h)} = -\boldsymbol{X}^{(l-1)}(\boldsymbol{W}^{(l)}\boldsymbol{W}^{(l)\top})^{(h)}. \tag{69}$$

2) Sinkhorn-based attention map: $\kappa^{\infty}(\boldsymbol{R}) := N_c \circ N_r \cdots N_c \circ N_r(\exp(\boldsymbol{R}))$,

$$\boldsymbol{R}^{(l,h)} = \boldsymbol{R}^{(l-1,h)} + \frac{\boldsymbol{Q}^{(l,h)}(\boldsymbol{K}^{(l,h)})^{\top}}{\sqrt{D_a}}, \quad \boldsymbol{T}^{(l,h)} = \kappa^{\infty}(\boldsymbol{R}^{(l,h)})\boldsymbol{V}^{(l,h)}. \tag{70}$$

WGFormer: Extend and Improve Sinkformer to SE(3)-Transformer





1) New "QKV" matrices:

$$\boldsymbol{Q}^{(l,h)} = \boldsymbol{K}^{(l,h)} = \boldsymbol{X}^{(l-1)}\boldsymbol{W}^{(l,h)}, \quad \boldsymbol{V}^{(l,h)} = -\boldsymbol{X}^{(l-1)}(\boldsymbol{W}^{(l)}\boldsymbol{W}^{(l)\top})^{(h)}. \tag{69}$$

2) Sinkhorn-based attention map: $\kappa^{\infty}(\boldsymbol{R}) := N_c \circ N_r \cdots N_c \circ N_r(\exp(\boldsymbol{R}))$,

$$\boldsymbol{R}^{(l,h)} = \boldsymbol{R}^{(l-1,h)} + \frac{\boldsymbol{Q}^{(l,h)}(\boldsymbol{K}^{(l,h)})^{\top}}{\sqrt{D_a}}, \quad \boldsymbol{T}^{(l,h)} = \kappa^{\infty}(\boldsymbol{R}^{(l,h)})\boldsymbol{V}^{(l,h)}. \tag{70}$$

3) Concatenation:

$$\boldsymbol{X}^{(l)} = \boldsymbol{X}^{(l-1)} + \mathsf{Concat}(\{\boldsymbol{T}^{(l,h)}\}_{h=1}^{H}), \quad \boldsymbol{R}^{(l)} = \mathsf{Concat}(\{\boldsymbol{R}^{(l,h)}\}_{h=1}^{H}). \tag{71}$$

The Motivations Behind The Key Improvements

1. **Adjusting "QKV" matrices in a different manner:**
   - $\boldsymbol{W}_Q = \boldsymbol{W}_K = \boldsymbol{W}^{(l,h)}$: Resulting in a valid kernel for interpreting attention maps.
   - $(\boldsymbol{W}^{(l)}\boldsymbol{W}^{(l)\top})^{(h)} = (\sum_{h'=1}^{H} \boldsymbol{W}^{(l,h')}\boldsymbol{W}^{(l,h')\top})^{(h)}$ for the h-th head: Achieving feature-level fusion across the attention heads

The Motivations Behind The Key Improvements

1. **Adjusting "QKV" matrices in a different manner:**
   - $\boldsymbol{W}_Q = \boldsymbol{W}_K = \boldsymbol{W}^{(l,h)}$: Resulting in a valid kernel for interpreting attention maps.
   - $(\boldsymbol{W}^{(l)}\boldsymbol{W}^{(l)\top})^{(h)} = (\sum_{h'=1}^{H} \boldsymbol{W}^{(l,h')}\boldsymbol{W}^{(l,h')\top})^{(h)}$ for the h-th head: Achieving feature-level fusion across the attention heads

2. **Removing FFN module:**
   - The new "QKV" matrices has fused features across different attention heads.
   - Simplify the model architecture and reduce the computational cost

The Motivations Behind The Key Improvements

1. **Adjusting "QKV" matrices in a different manner:**
   - $\boldsymbol{W}_Q = \boldsymbol{W}_K = \boldsymbol{W}^{(l,h)}$: Resulting in a valid kernel for interpreting attention maps.
   - $(\boldsymbol{W}^{(l)}\boldsymbol{W}^{(l)\top})^{(h)} = (\sum_{h'=1}^{H} \boldsymbol{W}^{(l,h')}\boldsymbol{W}^{(l,h')\top})^{(h)}$ for the h-th head: Achieving feature-level fusion across the attention heads

2. **Removing FFN module:**
   - The new "QKV" matrices has fused features across different attention heads.
   - Simplify the model architecture and reduce the computational cost

3. **Sinkhorn-based attention maps:**
   - Achieving doubly-stochastic attention maps by few iterations
   - Increasing computation costs slightly, but enhancing the model interpretability (As shown in **Sinkformer**)

WGFormer: An SE (3)-Transformer Driven by Wasserstein Gradient Flows for Molecular Ground-State Conformation Prediction. ICML, 2025.

Rationality of Sinkformer and WGFormer

- **Feedforward computation = Wasserstein gradient flow minimizing the potential energy**

$$\textbf{Particle: } x_{t+\delta t} = x_t + \delta t \cdot v(x_t, t), \quad v(x_t, t) = \int_{\mathcal{X}} \pi^{\infty}(x_t, y)(-WW^{\top})y\rho(y)\mathrm{d}y.$$
$$\textbf{The distribution of particle: } \rho_{t+\delta t} = \arg\min_{\rho \in \mathcal{P}(\mathcal{X})} E(\rho) + \frac{1}{2\delta t}\mathcal{W}_2^2(\rho, \rho_t). \tag{72}$$

- The $\exp(\boldsymbol{R})$ used in each layer of WGFormer can be treated as the prior of κ from the previous layer.

Rationality of Sinkformer and WGFormer

- **Feedforward computation = Wasserstein gradient flow minimizing the potential energy**

$$\textbf{Particle: } x_{t+\delta t} = x_t + \delta t \cdot v(x_t, t), \quad v(x_t, t) = \int_{\mathcal{X}} \pi^{\infty}(x_t, y)(-WW^{\top}) y \rho(y) \mathrm{d}y.$$
$$\textbf{The distribution of particle: } \rho_{t+\delta t} = \arg\min_{\rho \in \mathcal{P}(\mathcal{X})} E(\rho) + \frac{1}{2\delta t} \mathcal{W}_2^2(\rho, \rho_t). \tag{72}$$

- The $\exp(\boldsymbol{R})$ used in each layer of WGFormer can be treated as the prior of κ from the previous layer.
- Given N particles (i.e., $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^{\top}$, $\rho = \sum_n \delta_{\boldsymbol{x}_n}$),

$$E^{\infty}(\rho) \Leftrightarrow \max_{\boldsymbol{P} \in \Pi_{\mathbf{1}}} \underbrace{\langle \boldsymbol{D}, \boldsymbol{P} \rangle}_{\text{expected distance}} \underbrace{- \langle \boldsymbol{P}, \log \boldsymbol{P} \rangle}_{\text{entropy}}, \quad \boldsymbol{D} = [\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{x}_j\|_2^2 + r_{ij}]. \tag{73}$$

Rationality of Sinkformer and WGFormer

- **Feedforward computation = Wasserstein gradient flow minimizing the potential energy**

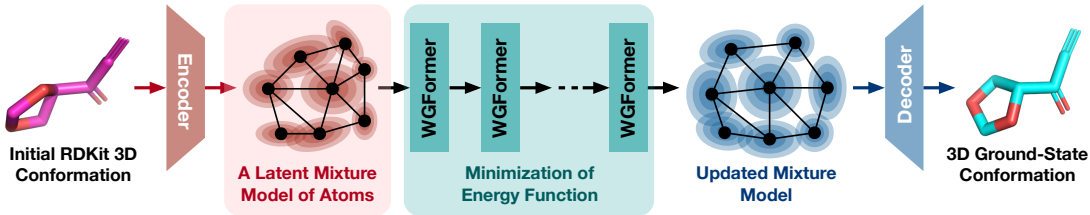
$$\textbf{Particle: } x_{t+\delta t} = x_t + \delta t \cdot v(x_t, t), \quad v(x_t, t) = \int_{\mathcal{X}} \pi^{\infty}(x_t, y)(-WW^{\top}) y \rho(y) \mathrm{d}y.$$
$$\textbf{The distribution of particle: } \rho_{t+\delta t} = \arg\min_{\rho \in \mathcal{P}(\mathcal{X})} E(\rho) + \frac{1}{2\delta t} \mathcal{W}_2^2(\rho, \rho_t). \tag{72}$$

- The $\exp(\boldsymbol{R})$ used in each layer of WGFormer can be treated as the prior of κ from the previous layer.
- Given N particles (i.e., $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^{\top}$, $\rho = \sum_n \delta_{\boldsymbol{x}_n}$),

$$E^{\infty}(\rho) \Leftrightarrow \max_{\boldsymbol{P} \in \Pi_1} \underbrace{\langle \boldsymbol{D}, \boldsymbol{P} \rangle}_{\text{expected distance}} \underbrace{- \langle \boldsymbol{P}, \log \boldsymbol{P} \rangle}_{\text{entropy}}, \quad \boldsymbol{D} = [\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{x}_j\|_2^2 + r_{ij}]. \tag{73}$$

- **Penalizing expected distance:** The particles should not aggregate together (avoid high potential energy).
- **Regularizing entropy:** The particles have dense interactions.

Applications of WGFormer: Ground-State Conformation Prediction



For a molecule: initial coordinates $\{\tilde{\boldsymbol{c}}_i\}_{i=1}^N$, distances $\{\tilde{d}_{ij}\}_{i,j=1}^N$, atom types $\{v_i\}_{i=1}^N$.

Encoder: Minimizing potential energy defined in the latent space

$$\boldsymbol{x}_i^{(0)} = f(v_i), \quad \boldsymbol{r}_{ij}^{(0)} = \mathcal{N}(\tilde{d}_{ij}\boldsymbol{u}_{v_iv_j} + \boldsymbol{v}_{v_iv_j}; \boldsymbol{\mu}, \boldsymbol{\sigma}), \quad \boldsymbol{X}^{(L)}, \boldsymbol{R}^{(L)} = \mathrm{WGFormer}_L(\boldsymbol{X}^{(0)}, \boldsymbol{R}^{(0)}),$$

Decoder: Predicting the translation of each atom

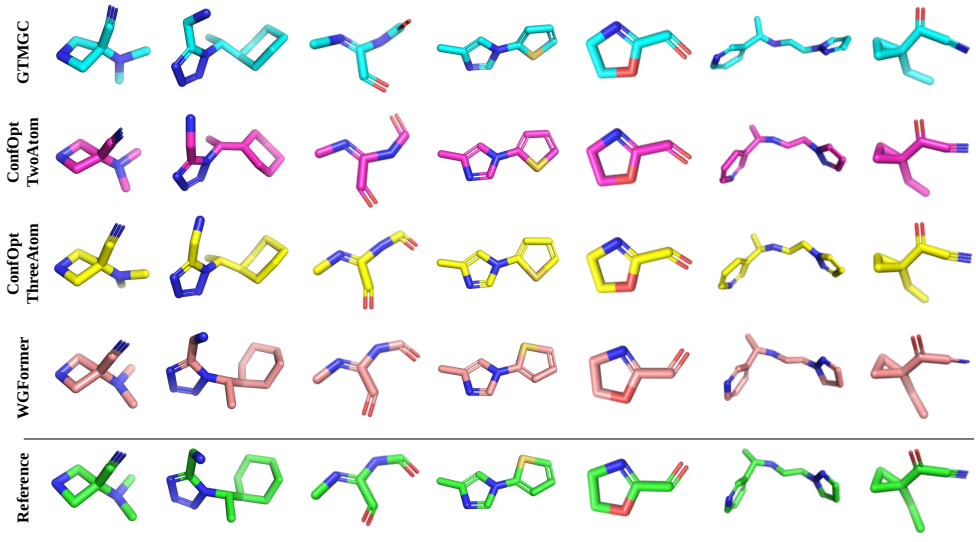
$$\hat{\boldsymbol{c}}_i = \tilde{\boldsymbol{c}}_i + \sum\nolimits_{j=1}^{N} \frac{\mathrm{MLP}(\boldsymbol{r}_{ij}^{(L)} - \boldsymbol{r}_{ij}^{(0)})(\tilde{\boldsymbol{c}}_i - \tilde{\boldsymbol{c}}_j)}{N}.$$

Supervised Learning of a model with 30-layer WGFormer.

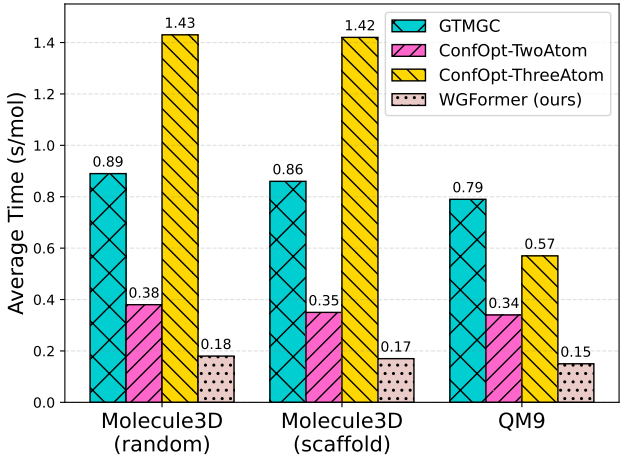
Applications of WGFormer: Ground-State Conformation Prediction

Dataset	Method	Model	Validation			Test		
			D-MAE↓	D-RMSE↓	C-RMSD↓	D-MAE↓	D-RMSE↓	C-RMSD↓
Molecule3D (random)	2D	GINE	0.590	1.014	1.116	0.592	1.018	1.116
		GATv2	0.563	0.983	1.082	0.564	0.986	1.083
		GPS	0.528	0.909	1.036	0.529	0.911	1.038
		GTMGC	0.432	0.719	0.712	0.433	0.721	0.713
	3D	SE(3)-Transformer	0.466	0.712	0.800	0.467	0.774	0.802
		EGNN	0.461	0.704	0.798	0.462	0.766	0.799
		ConfOpt-TwoAtom	0.438	0.668	0.748	0.438	0.670	0.749
		ConfOpt-ThreeAtom	0.429	0.659	0.734	0.430	0.661	0.736
		WGFormer (ours)	**0.391**	**0.649**	**0.662**	**0.392**	**0.652**	**0.664**
Molecule3D (scaffold)	2D	GINE	0.883	1.517	1.407	1.400	2.224	1.960
		GATv2	0.778	1.385	1.254	1.238	2.069	1.752
		GPS	0.538	0.885	1.031	0.657	1.091	1.136
		GTMGC	0.406	0.675	0.678	0.400	0.679	0.693
	3D	SE(3)-Transformer	0.460	0.676	0.775	0.456	0.678	0.747
		EGNN	0.448	0.666	0.758	0.442	0.670	0.741
		ConfOpt-TwoAtom	0.408	0.626	0.708	0.402	0.628	0.698
		ConfOpt-ThreeAtom	0.401	0.619	0.697	0.395	0.622	0.691
		WGFormer (ours)	**0.363**	**0.599**	**0.618**	**0.360**	**0.610**	**0.627**

Applications of WGFormer: Ground-State Conformation Prediction



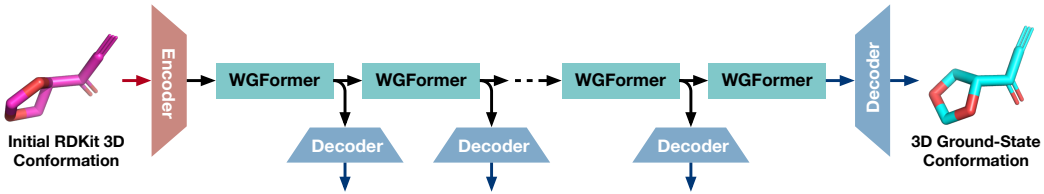
Applications of WGFormer: Ground-State Conformation Prediction

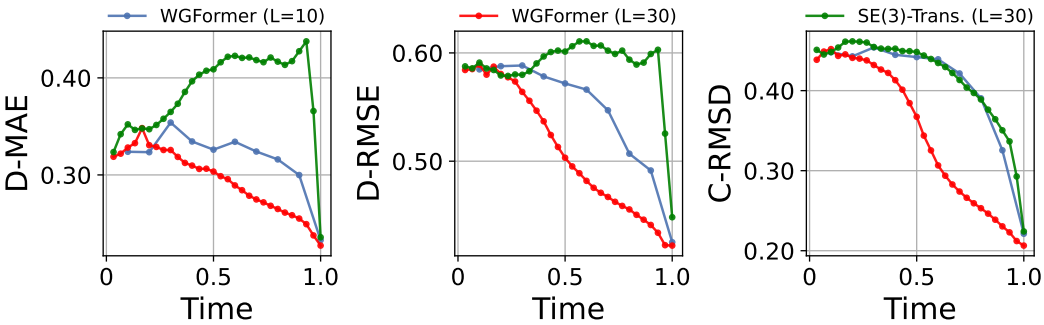


Both Sinkformer and WGFormer apply 3-5 Sinkhorn iterations per layer, achieving high efficiency.

Applications of WGFormer: Ground-State Conformation Prediction

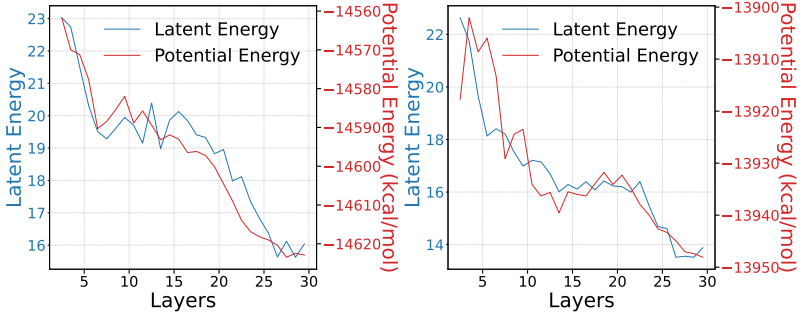
Feedforward computation = Euler step of latent energy optimization





Applications of WGFormer: Ground-State Conformation Prediction

The proposed latent energy is highly correlated with physical energy

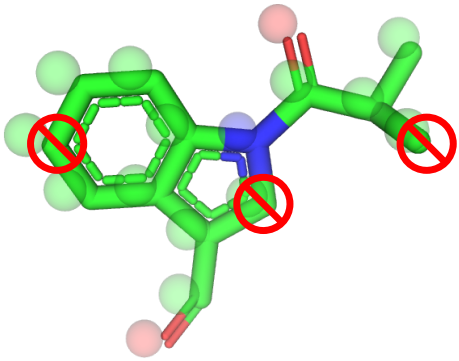


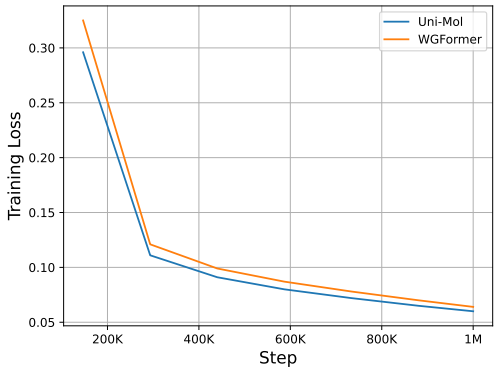
# Layers	5	10	15	20	25	30	Pearson Correlation
Physical Energy (kcal/mol)	-9.135	-18.199	-19.955	-34.814	-45.204	-52.378	**0.885 ± 0.033**
Proposed Latent Energy	-3.629	-7.729	-8.512	-8.932	-9.195	-10.385	

GFN2-xTB—An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. JCTC, 2019.

Applications of WGFormer: 3D Molecular Foundation Model

Replace the SE(3)-Transformer of Uni-Mol with WGFormer:





Applications of WGFormer: 3D Molecular Foundation Model

Classification Tasks

	ROC-AUC ↑								
Datasets # Molecules	BBBP 2039	BACE 1513	ClinTox 1478	Tox21 7831	ToxCast 8575	SIDER 1427	HIV 41127	PCBA 437929	MUV 93087
Uni-Mol (47.6M)	**0.702**	**0.837**	**0.794**	0.786	**0.687**	0.617	**0.804**	**0.885**	0.793
WGFormer (**39.7M**)	0.690	**0.837**	0.635	**0.790**	0.682	**0.623**	0.768	0.884	**0.816**

Regression Tasks

	RMSE ↓			MAE ↓		
Datasets # Molecules	ESOL 1128	FreeSolv 642	Lipo 4200	QM7 6830	QM8 21786	QM9 133885
Uni-Mol (47.6M)	0.884	1.756	0.598	**57.00**	**0.015**	**0.005**
WGFormer (**39.7M**)	**0.836**	**1.588**	**0.584**	58.70	0.016	**0.005**

The results are achieved under default hyperparameter settings of Uni-Mol.

Applications of WGFormer: 3D Molecular Foundation Model

Classification Tasks

	ROC-AUC ↑								
Datasets # Molecules	BBBP 2039	BACE 1513	ClinTox 1478	Tox21 7831	ToxCast 8575	SIDER 1427	HIV 41127	PCBA 437929	MUV 93087
Uni-Mol (47.6M)	**0.702**	**0.837**	0.794	0.786	**0.687**	0.617	**0.804**	**0.885**	0.793
WGFormer (**39.7M**)	0.690	**0.837**	**0.928**	**0.790**	0.682	**0.623**	0.768	0.884	**0.816**

Regression Tasks

	RMSE ↓			MAE ↓		
Datasets # Molecules	ESOL 1128	FreeSolv 642	Lipo 4200	QM7 6830	QM8 21786	QM9 133885
Uni-Mol (47.6M)	0.884	1.756	0.598	57.00	**0.015**	**0.005**
WGFormer (**39.7M**)	**0.836**	**1.588**	**0.584**	**46.65**	0.016	**0.005**

The results in red are achieved under non-default hyperparameter settings.

From Euclidean Space to Graph: Two Technical Routes

1. Reuse the above auto-encoding architecture with WGFormer
   - Take normalized adjacency/Laplacian matrix as initial $\boldsymbol{R}$.
   - Suitable for improving graph-oriented Transformers
   - Suitable for geometric deep learning

From Euclidean Space to Graph: Two Technical Routes

1. Reuse the above auto-encoding architecture with WGFormer
   - Take normalized adjacency/Laplacian matrix as initial $\boldsymbol{R}$.
   - Suitable for improving graph-oriented Transformers
   - Suitable for geometric deep learning

- However, without any acceleration, the dense computation of WGFormer is inapplicable for large-scale graphs, like social networks.

From Euclidean Space to Graph: Two Technical Routes

1. Reuse the above auto-encoding architecture with WGFormer
   - Take normalized adjacency/Laplacian matrix as initial $\boldsymbol{R}$.
   - Suitable for improving graph-oriented Transformers
   - Suitable for geometric deep learning
- However, without any acceleration, the dense computation of WGFormer is inapplicable for large-scale graphs, like social networks.

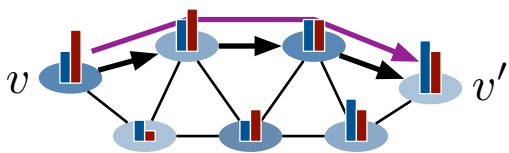
2. Consider the topological structure and the **optimal transport on graph** explicitly and efficiently
   - Connect OT to classic graph theory, rather than differential equations
   - Suitable for improving GNNs
   - Provide a new perspective to design GNNs and their learning paradigms.

Optimal Transport on Graph

- Given two measures defined on a graph $G(\mathcal{V}, \mathcal{E})$ i.e., $\boldsymbol{\rho}_0 \in [0, \infty)^{|\mathcal{V}|}$ and $\boldsymbol{\rho}_1 \in [0, \infty)^{|\mathcal{V}|}$, the 1-order Wasserstein distance between them is

$$\mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) := \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \langle \boldsymbol{D}, \ \boldsymbol{P} \rangle = \min_{\boldsymbol{P} \in \Pi(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \sum_{v, v' \in \mathcal{V} \times \mathcal{V}} p_{vv'} d_{vv'}, \tag{74}$$

Mass transport along the shortest path



- $\boldsymbol{D} = [d_{vv'}]$, $d_{vv'}$ is the shortest path from v to v'.
- $\boldsymbol{P} = [p_{vv'}]$, $p_{vv'}$ the mass transported from v to v', along the shortest path.

Optimal Transport on Graph

$\mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)$ can be equivalently computed by **cost flow minimization**:

$$\begin{aligned} \mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = & \min_{\boldsymbol{f}} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \\ & s.t.\ \boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \{\boldsymbol{f} \mid \boldsymbol{S}_{\mathcal{V}}\boldsymbol{f} = \boldsymbol{\rho}_1 - \boldsymbol{\rho}_0\}. \end{aligned} \tag{75}$$

Optimal Transport on Graph

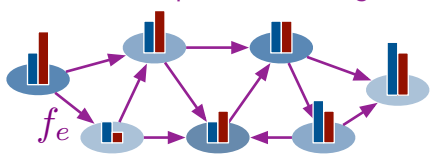
$\mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)$ can be equivalently computed by **cost flow minimization**:

$$\begin{aligned} \mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = &\min_{\boldsymbol{f}} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \\ &s.t.\ \boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \{\boldsymbol{f} \mid \boldsymbol{S}_{\mathcal{V}}\boldsymbol{f} = \boldsymbol{\rho}_1 - \boldsymbol{\rho}_0\}. \end{aligned} \tag{75}$$

- $\boldsymbol{S}_{\mathcal{V}} = [s_{ve}] \in \{0, \pm 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ is the **incidence matrix** of $G(\mathcal{V}, \mathcal{E})$:

$$s_{ve} = \begin{cases} 1 & \text{If } v \text{ is the head of } e \\ -1 & \text{If } v \text{ is the tail of } e \\ 0 & \text{Otherwise} \end{cases} \tag{76}$$

Mass transport flow on edges



- $\boldsymbol{a} \in \mathbb{R}^{|\mathcal{E}|}$ contains nonzero elements of adjacency matrix $\boldsymbol{A}$.

Optimal Transport on Graph

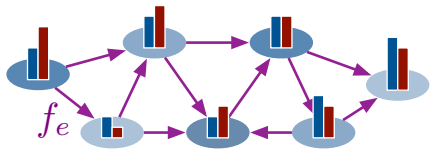
$\mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)$ can be equivalently computed by **cost flow minimization**:

$$\begin{aligned} \mathcal{W}_1(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = &\min_{\boldsymbol{f}} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \\ &s.t.\ \boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \{\boldsymbol{f} \mid \boldsymbol{S}_{\mathcal{V}}\boldsymbol{f} = \boldsymbol{\rho}_1 - \boldsymbol{\rho}_0\}. \end{aligned} \tag{75}$$

- $\boldsymbol{S}_{\mathcal{V}} = [s_{ve}] \in \{0, \pm 1\}^{|\mathcal{V}|\times|\mathcal{E}|}$ is the **incidence matrix** of $G(\mathcal{V}, \mathcal{E})$:

$$s_{ve} = \begin{cases} 1 & \text{If } v \text{ is the head of } e \\ -1 & \text{If } v \text{ is the tail of } e \\ 0 & \text{Otherwise} \end{cases} \tag{76}$$

Mass transport flow on edges



- $\boldsymbol{a} \in \mathbb{R}^{|\mathcal{E}|}$ contains nonzero elements of adjacency matrix $\boldsymbol{A}$.
- $\boldsymbol{f} \in \mathbb{R}^{|\mathcal{E}|}$ is the **cost flow on graph edges**, ensuring the transport from $\boldsymbol{\rho}_0$ to $\boldsymbol{\rho}_1$ (See the feasible domain).
- $\boldsymbol{S}_{\mathcal{V}}\boldsymbol{f}$ captures the mass difference on graph nodes.

Quasi-Wasserstein (QW) Distance for Measures on Graphs

- When only partial signals on $\mathcal{V}_L$ are given, we have **partial Wasserstein distance**: for $\boldsymbol{\rho}_0, \boldsymbol{\rho}_1 \in \mathbb{R}^{|\mathcal{V}_L|}$,

$$\mathcal{W}_1^{(P)}(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \tag{77}$$

Quasi-Wasserstein (QW) Distance for Measures on Graphs

- When only partial signals on $\mathcal{V}_L$ are given, we have **partial Wasserstein distance**: for $\boldsymbol{\rho}_0, \boldsymbol{\rho}_1 \in \mathbb{R}^{|\mathcal{V}_L|}$,

$$\mathcal{W}_1^{(P)}(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \tag{77}$$

- For partially-observed multi-dimensional signals, i.e., $\boldsymbol{Y}_0 = [\boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}], \boldsymbol{Y}_1 = [\boldsymbol{y}_{1,\mathcal{V}_L}^{(c)}] \in \mathbb{R}^{|\mathcal{V}_L| \times C}$, the QW distance between them is

$$\begin{aligned} QW(\boldsymbol{Y}_0, \boldsymbol{Y}_1) &:= \sum_{c=1}^{C} W_1^{(P)}(\boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{1,\mathcal{V}_L}^{(c)}) \\ &= \sum_{c=1}^{C} \min_{\boldsymbol{f}^{(c)} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{1,\mathcal{V}_L}^{(c)})} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}^{(c)}\|_1 \end{aligned}$$

Quasi-Wasserstein (QW) Distance for Measures on Graphs

- When only partial signals on $\mathcal{V}_L$ are given, we have **partial Wasserstein distance**: for $\boldsymbol{\rho}_0, \boldsymbol{\rho}_1 \in \mathbb{R}^{|\mathcal{V}_L|}$,

$$\mathcal{W}_1^{(P)}(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \tag{77}$$

- For partially-observed multi-dimensional signals, i.e., $\boldsymbol{Y}_0 = [\boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}], \boldsymbol{Y}_1 = [\boldsymbol{y}_{1,\mathcal{V}_L}^{(c)}] \in \mathbb{R}^{|\mathcal{V}_L| \times C}$, the QW distance between them is

$$\begin{aligned} QW(\boldsymbol{Y}_0, \boldsymbol{Y}_1) &:= \sum_{c=1}^{C} W_1^{(P)}(\boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{1,\mathcal{V}_L}^{(c)}) \\ &= \sum_{c=1}^{C} \min_{\boldsymbol{f}^{(c)} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{1,\mathcal{V}_L}^{(c)})} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}^{(c)}\|_1 \\ &= \min_{\boldsymbol{F} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \boldsymbol{Y}_0, \boldsymbol{Y}_1)} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{F}\|_1, \end{aligned} \tag{78}$$

Quasi-Wasserstein (QW) Distance for Measures on Graphs

- When only partial signals on $\mathcal{V}_L$ are given, we have **partial Wasserstein distance**: for $\boldsymbol{\rho}_0, \boldsymbol{\rho}_1 \in \mathbb{R}^{|\mathcal{V}_L|}$,

$$\mathcal{W}_1^{(P)}(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, \boldsymbol{\rho}_0, \boldsymbol{\rho}_1)} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}\|_1, \tag{77}$$

- For partially-observed multi-dimensional signals, i.e., $\boldsymbol{Y}_0 = [\boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}], \boldsymbol{Y}_1 = [\boldsymbol{y}_{1,\mathcal{V}_L}^{(c)}] \in \mathbb{R}^{|\mathcal{V}_L| \times C}$, the QW distance between them is

$$\begin{aligned} QW(\boldsymbol{Y}_0, \boldsymbol{Y}_1) &:= \sum_{c=1}^{C} W_1^{(P)}(\boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{1,\mathcal{V}_L}^{(c)}) \\ &= \sum_{c=1}^{C} \min_{\boldsymbol{f}^{(c)} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \boldsymbol{y}_{0,\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{1,\mathcal{V}_L}^{(c)})} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{f}^{(c)}\|_1 \\ &= \min_{\boldsymbol{F} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \boldsymbol{Y}_0, \boldsymbol{Y}_1)} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{F}\|_1, \end{aligned} \tag{78}$$

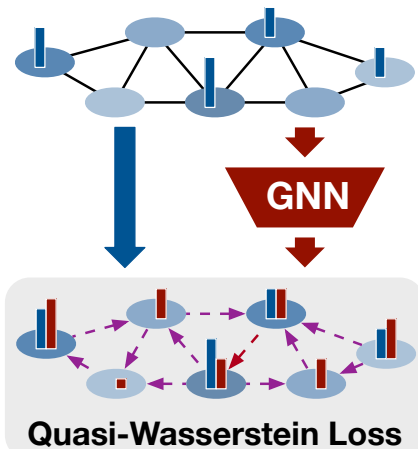
  - QW allows partial, unbalanced, and negative input signals.
  - Given a specific graph G and $\mathcal{V}_L$, QW is a valid distance metric.

Learning GNNs via Minimizing QW Distance

- Given partially observed node labels $\boldsymbol{Y}_{\mathcal{V}_L}$, we would like to predict them by a GNN, i.e., $\widehat{\boldsymbol{Y}}_{\mathcal{V}_L} := g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)$.
- We can learn the GNN via minimizing the QW distance:

$$\min_\theta QW(\underbrace{g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)}_{\widehat{\boldsymbol{Y}}_{\mathcal{V}_L}}, \boldsymbol{Y}_{\mathcal{V}_L}), \quad (79)$$

- **Lead to an implicit message-passing layer encoding label transport.**



Advantages over Traditional Losses (CE/MSE)

- **Traditional Node-Level Learning Paradigm of GNN:**

$$\max_\theta \prod_{v\in\mathcal{V}_L} p(\boldsymbol{y}_v|\boldsymbol{X},\boldsymbol{A};\theta) \Leftrightarrow \min_\theta \sum_{v\in\mathcal{V}_L} \psi(g_v(\boldsymbol{X},\boldsymbol{A};\theta),\ \boldsymbol{y}_v). \tag{80}$$

  - p is Gaussian $\Leftrightarrow$ ψ is MSE.
  - p is Sigmoid/Softmax $\Leftrightarrow$ ψ is Cross Entropy (CE) loss.

Advantages over Traditional Losses (CE/MSE)

- **Traditional Node-Level Learning Paradigm of GNN:**

$$\max_\theta \prod_{v\in\mathcal{V}_L} p(\boldsymbol{y}_v|\boldsymbol{X},\boldsymbol{A};\theta) \Leftrightarrow \min_\theta \sum_{v\in\mathcal{V}_L} \psi(g_v(\boldsymbol{X},\boldsymbol{A};\theta),\ \boldsymbol{y}_v). \tag{80}$$

  - p is Gaussian $\Leftrightarrow$ ψ is MSE.
  - p is Sigmoid/Softmax $\Leftrightarrow$ ψ is Cross Entropy (CE) loss.
- **What we really want to do is maximizing the joint probability of $\boldsymbol{Y}_{\mathcal{V}_L}$**, i.e.,

$$\max_\theta p(\boldsymbol{Y}_{\mathcal{V}_L}|\boldsymbol{X},\boldsymbol{A};\theta) \tag{81}$$

Advantages over Traditional Losses (CE/MSE)

- **Traditional Node-Level Learning Paradigm of GNN:**

$$\max_\theta \prod\nolimits_{v\in\mathcal{V}_L} p(\boldsymbol{y}_v|\boldsymbol{X},\boldsymbol{A};\theta) \Leftrightarrow \min_\theta \sum\nolimits_{v\in\mathcal{V}_L} \psi(g_v(\boldsymbol{X},\boldsymbol{A};\theta),\ \boldsymbol{y}_v). \tag{80}$$

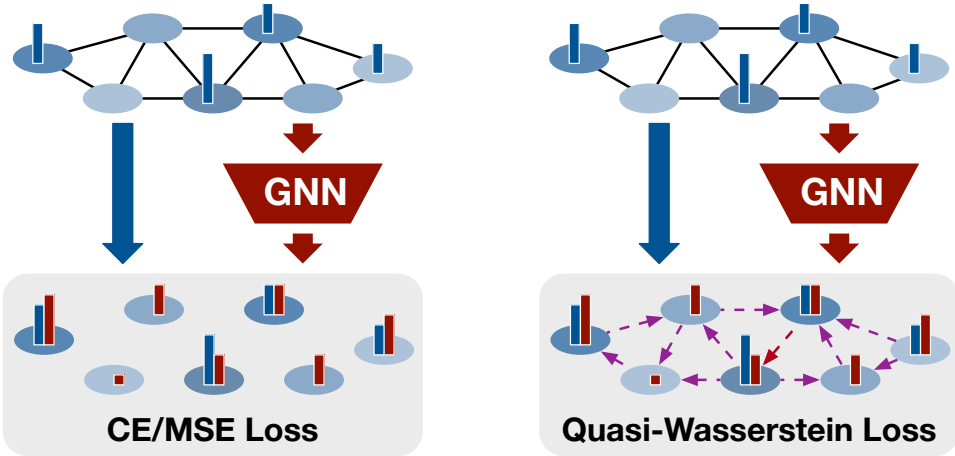
  - p is Gaussian $\Leftrightarrow$ ψ is MSE.
  - p is Sigmoid/Softmax $\Leftrightarrow$ ψ is Cross Entropy (CE) loss.
- **What we really want to do is maximizing the joint probability of $\boldsymbol{Y}_{\mathcal{V}_L}$**, i.e.,

$$\max_\theta p(\boldsymbol{Y}_{\mathcal{V}_L}|\boldsymbol{X},\boldsymbol{A};\theta) \tag{81}$$

- **The above two equations are equivalent iff the labels are conditional independent, which is questionable in practice.**

$$p(\boldsymbol{y}_v|\boldsymbol{x}_v) \neq p(\boldsymbol{y}_v|\boldsymbol{x}_v,\boldsymbol{y}_{v'}),$$
$$p(\boldsymbol{y}_v,\boldsymbol{y}_{v'}|\boldsymbol{x}_v,\boldsymbol{x}_{v'}) = p(\boldsymbol{y}_v|\boldsymbol{x}_v,\boldsymbol{x}_{v'},\boldsymbol{y}_{v'})p(\boldsymbol{y}_{v'}|\boldsymbol{x}_v,\boldsymbol{x}_{v'}) \neq p(\boldsymbol{y}_v|\boldsymbol{x}_v,\boldsymbol{x}_{v'})p(\boldsymbol{y}_{v'}|\boldsymbol{x}_v,\boldsymbol{x}_{v'}),$$
$$p(\boldsymbol{Y}_{\mathcal{V}}|\boldsymbol{X},\boldsymbol{A}) \neq \prod\nolimits_{v\in\mathcal{V}} p(\boldsymbol{y}_v|\boldsymbol{X},\boldsymbol{A}).$$

Advantages over Traditional Losses (CE/MSE)



- **Relax the independency assumption, and lead to set-level prediction loss.**
- **In theory, the GNN minimizing QW loss fits labels better.**

A Quasi-Wasserstein loss for learning graph neural networks. WWW, 2024.

Solvers of QW Loss

- Learning GNN with the QW loss:

$$\min_{\theta} QW(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta), \boldsymbol{Y}_{\mathcal{V}_L}) = \min_{\theta} \min_{\boldsymbol{F} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{a})\boldsymbol{F}\|_1. \quad (82)$$

Solvers of QW Loss

- Learning GNN with the QW loss:

$$\min_{\theta} QW(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta), \boldsymbol{Y}_{\mathcal{V}_L}) = \min_{\theta} \min_{\boldsymbol{F} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{a})\boldsymbol{F}\|_1. \quad (82)$$

- **Keypoint:** Apply Bregman divergence $B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle$.
  - **An inexact solver based on Bregman divergence-based relaxation:**

$$\min_{\theta,\ \boldsymbol{F}} \|\mathsf{diag}(\boldsymbol{a})\boldsymbol{F}\|_1 + \lambda B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F},\ \boldsymbol{Y}_{\mathcal{V}_L}). \quad (83)$$

Solvers of QW Loss

- Learning GNN with the QW loss:

$$\min_{\theta} QW(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta), \boldsymbol{Y}_{\mathcal{V}_L}) = \min_{\theta} \min_{\boldsymbol{F} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{F}\|_1. \quad (82)$$

- **Keypoint:** Apply Bregman divergence $B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle$.
  - **An inexact solver based on Bregman divergence-based relaxation:**

$$\min_{\theta,\ \boldsymbol{F}} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{F}\|_1 + \lambda B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F},\ \boldsymbol{Y}_{\mathcal{V}_L}). \quad (83)$$

  - **An exact solver based on Bregman ADMM:** An augmented Lagrangian form with a dual variable $\boldsymbol{Z}$

$$\begin{aligned} \min_{\theta,\ \boldsymbol{F}} \max_{\boldsymbol{Z}} \|\mathrm{diag}(\boldsymbol{a})\boldsymbol{F}\|_1 + \langle \boldsymbol{Z},\ g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F} - \boldsymbol{Y}_{\mathcal{V}_L} \rangle \\ + \lambda B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F},\ \boldsymbol{Y}_{\mathcal{V}_L}). \end{aligned} \quad (84)$$

Compare with Traditional Learning Paradigm

Method	Setting	Node Classification	Node Regression
Apply the	ψ	Cross-entropy or KL	MSE
Traditional loss	Predicted $\boldsymbol{y}_v$	GNN: $g_v(\boldsymbol{X}, \boldsymbol{A}; \theta)$, $\forall v \in \mathcal{V} \setminus \mathcal{V}_L$	

Compare with Traditional Learning Paradigm

Method	Setting	Node Classification	Node Regression
Apply the Traditional loss	ψ	Cross-entropy or KL	MSE
	Predicted $\boldsymbol{y}_v$	GNN: $g_v(\boldsymbol{X}, \boldsymbol{A}; \theta)$, $\forall v \in \mathcal{V} \setminus \mathcal{V}_L$	
Apply the QW loss	ϕ	Entropy	$\frac{1}{2}\|\cdot\|_2^2$
	$B_\phi(=\psi)$	KL	MSE
	Predicted $\boldsymbol{y}_v$	$g_v(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_v \boldsymbol{F}^*$, $\forall v \in \mathcal{V} \setminus \mathcal{V}_L$	

Compare with Traditional Learning Paradigm

Method	Setting	Node Classification	Node Regression
Apply the Traditional loss	ψ	Cross-entropy or KL	MSE
	Predicted $\boldsymbol{y}_v$	GNN: $g_v(\boldsymbol{X}, \boldsymbol{A}; \theta)$, $\forall v \in \mathcal{V} \setminus \mathcal{V}_L$	
Apply the QW loss	ϕ	Entropy	$\frac{1}{2}\|\cdot\|_2^2$
	$B_\phi(=\psi)$	KL	MSE
	Predicted $\boldsymbol{y}_v$	$g_v(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_v\boldsymbol{F}^*$, $\forall v \in \mathcal{V} \setminus \mathcal{V}_L$	

- $\boldsymbol{F}^*$ captures the flow of labels.
- $\boldsymbol{S}_v\boldsymbol{F}^*$ works as **a nonparametric flow module captures the residue of GNN's prediction.**

Amortized Flow: From Loss to Model

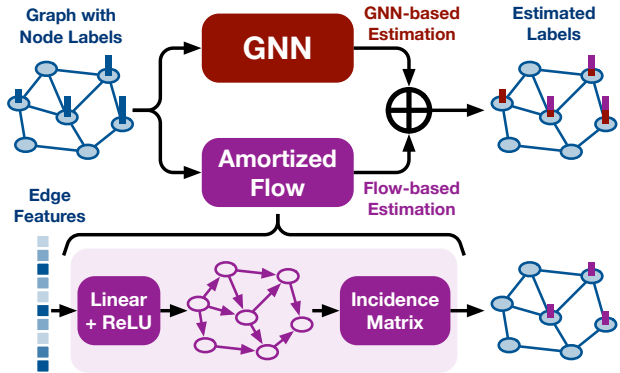
- We can parametrize the label residue by a simple NN:

$$\widehat{\boldsymbol{\Delta}}_{\mathcal{V}} = \boldsymbol{S}_{\mathcal{V}} \underbrace{\sigma(\boldsymbol{E}\boldsymbol{W})}_{\text{Amortized Flow}}. \qquad (85)$$

  $\boldsymbol{E} \in \mathbb{R}^{\mathcal{E} \times D}$ is edge feature.

- As a result, the GNN becomes

$$\widehat{\boldsymbol{Y}}_{\mathcal{V}} = g_{\mathcal{V}}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \widehat{\boldsymbol{\Delta}}_{\mathcal{V}}. \quad (86)$$



Amortized Flow: From Loss to Model

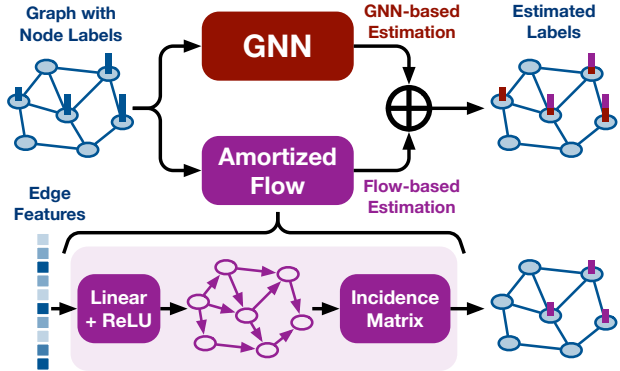
- We can parametrize the label residue by a simple NN:

$$\widehat{\boldsymbol{\Delta}}_{\mathcal{V}} = \boldsymbol{S}_{\mathcal{V}} \underbrace{\sigma(\boldsymbol{E}\boldsymbol{W})}_{\text{Amortized Flow}}. \qquad (85)$$

  $\boldsymbol{E} \in \mathbb{R}^{\mathcal{E}\times D}$ is edge feature.

- As a result, the GNN becomes

$$\widehat{\boldsymbol{Y}}_{\mathcal{V}} = g_{\mathcal{V}}(\boldsymbol{X}, \boldsymbol{A};\ \theta) + \widehat{\boldsymbol{\Delta}}_{\mathcal{V}}. \quad (86)$$



- When $\boldsymbol{E}$ is meaningful edge feature, the AF introduces additional information enhancing the GNN.
- When $\boldsymbol{E}$ is random noise, $\widehat{\boldsymbol{\Delta}}_{\mathcal{V}} \to \boldsymbol{0}$, and GNN+AF degrades to classic GNN.

Node Classification on Homophilic Graphs

Model	#Param.	Cora	Citeseer	Pubmed	Computers	Photo	Improve
GCN	49,802	87.44$_{\pm 0.96}$	79.98$_{\pm 0.84}$	86.93$_{\pm 0.29}$	88.42$_{\pm 0.45}$	93.24$_{\pm 0.43}$	—
GCN+F	2,508,412	87.88$_{\pm 0.79}$	**81.36$_{\pm 0.41}$**	**87.89$_{\pm 0.40}$**	89.20$_{\pm 0.41}$	93.81$_{\pm 0.36}$	+0.83
GCN+AF	49,822	**88.19$_{\pm 0.85}$**	80.61$_{\pm 0.56}$	87.84$_{\pm 0.25}$	**89.58$_{\pm 0.29}$**	**94.17 $_{\pm 0.48}$**	+0.88
GAT	399,390	89.20$_{\pm 0.79}$	80.75$_{\pm 0.78}$	87.42$_{\pm 0.33}$	90.08$_{\pm 0.36}$	94.38$_{\pm 0.25}$	—
GAT+F	2,858,000	89.11$_{\pm 0.66}$	80.19$_{\pm 0.64}$	**88.38$_{\pm 0.23}$**	90.41$_{\pm 0.28}$	94.65$_{\pm 0.24}$	+0.18
GAT+AF	399,410	**89.44 $_{\pm 0.69}$**	**81.39$_{\pm 0.94}$**	88.25$_{\pm 0.22}$	**90.62$_{\pm 0.35}$**	**94.67$_{\pm 0.30}$**	+0.51
GIN	58,122	86.22$_{\pm 0.95}$	76.18$_{\pm 0.78}$	87.87$_{\pm 0.23}$	80.87$_{\pm 1.43}$	89.83$_{\pm 0.72}$	—
GIN+F	2,516,732	86.24$_{\pm 0.90}$	76.13$_{\pm 1.09}$	87.53$_{\pm 0.34}$	**89.28$_{\pm 0.45}$**	**92.60$_{\pm 0.44}$**	+2.16
GIN+AF	58,142	**87.65$_{\pm 0.84}$**	**77.68$_{\pm 0.81}$**	**87.96$_{\pm 0.25}$**	87.88$_{\pm 0.59}$	92.25$_{\pm 0.33}$	+2.49
GraphSAGE	99,530	88.24$_{\pm 0.95}$	79.81$_{\pm 0.80}$	88.14$_{\pm 0.25}$	89.71$_{\pm 0.38}$	95.08$_{\pm 0.26}$	—
GraphSAGE+F	2,558,140	87.59$_{\pm 0.77}$	80.52$_{\pm 0.68}$	88.61$_{\pm 0.32}$	90.17$_{\pm 0.24}$	95.25$_{\pm 0.25}$	+0.23
GraphSAGE+AF	99,550	**88.34$_{\pm 0.74}$**	**80.71$_{\pm 0.70}$**	**88.78$_{\pm 0.16}$**	**90.60$_{\pm 0.45}$**	**95.49$_{\pm 0.34}$**	+0.59
APPNP	49,802	88.33$_{\pm 0.77}$	81.28$_{\pm 0.71}$	88.62$_{\pm 0.33}$	86.27$_{\pm 0.37}$	93.70$_{\pm 0.27}$	—
APPNP+F	2,508,412	88.74$_{\pm 0.84}$	80.94$_{\pm 0.61}$	89.48$_{\pm 0.28}$	86.95$_{\pm 0.82}$	94.43$_{\pm 0.24}$	+0.47
APPNP+AF	49,822	**89.38$_{\pm 0.77}$**	**82.09$_{\pm 0.74}$**	**89.70$_{\pm 0.32}$**	**87.55$_{\pm 0.59}$**	**94.50$_{\pm 0.43}$**	+1.00
BernNet	49,833	88.28$_{\pm 1.00}$	79.81$_{\pm 0.79}$	88.87$_{\pm 0.38}$	87.61$_{\pm 0.46}$	93.68$_{\pm 0.28}$	—
BernNet+F	2,508,443	**89.03$_{\pm 0.76}$**	**81.35$_{\pm 0.71}$**	89.03$_{\pm 0.38}$	**89.58$_{\pm 0.47}$**	**94.55$_{\pm 0.39}$**	+1.06
BernNet+AF	49,853	88.60$_{\pm 0.71}$	81.27$_{\pm 0.70}$	**89.37$_{\pm 0.47}$**	87.53$_{\pm 0.45}$	93.80$_{\pm 0.41}$	+0.46
ChebNetII	49,813	88.26$_{\pm 0.89}$	**80.00$_{\pm 0.74}$**	88.57$_{\pm 0.36}$	86.58$_{\pm 0.71}$	93.50$_{\pm 0.34}$	—
ChebNetII+F	2,508,423	88.54$_{\pm 0.76}$	79.47$_{\pm 0.70}$	**89.47$_{\pm 0.36}$**	90.43$_{\pm 0.22}$	**94.84$_{\pm 0.37}$**	+1.17
ChebNetII+AF	49,833	**88.64$_{\pm 0.81}$**	79.99$_{\pm 0.64}$	89.34$_{\pm 0.40}$	**90.46$_{\pm 0.39}$**	94.75$_{\pm 0.43}$	+1.25

Node Classification on Heterophilic Graphs

Model	#Param.	Squirrel	Chameleon	Actor	Texas	Cornell	Arxiv-year	Improve
GCN	134,085	46.55±1.15	63.57±1.16	34.00±1.28	77.21±3.28	61.91±5.11	44.40±0.16	—
GCN+F	1,125,850	52.62±0.49	68.10±1.01	38.09±0.50	84.10±2.95	84.26±2.98	44.70±0.30	+7.37
GCN+AF	134,095	**53.40±1.35**	**68.25±1.01**	**38.77±0.70**	**89.51±1.64**	**87.02±2.98**	**44.72±0.36**	+9.00
GAT	1,073,679	48.20±1.67	64.31±2.01	35.68±0.60	80.00±3.11	68.09±2.13	44.21±0.30	—
GAT+F	2,065,444	**55.03±1.35**	67.35±1.42	33.86±2.13	80.33±1.80	70.21±2.13	**44.41±0.28**	+1.78
GAT+AF	1,073,689	51.47±1.06	**67.40±1.27**	**35.94±0.78**	**80.82±2.63**	**70.64±2.55**	44.24±0.25	+1.67
GIN	142,405	39.11±2.23	64.29±1.51	32.37±1.56	72.79±4.92	62.55±4.80	44.39±0.28	—
GIN+F	1,134,170	**65.29±0.68**	**73.26±1.12**	32.32±1.93	77.54±2.60	64.04±3.62	**44.56±0.26**	+6.92
GIN+AF	142,415	50.76±1.17	71.25±1.45	**34.43±1.03**	**78.69±3.11**	**67.87±5.96**	44.51±0.30	+5.34
GraphSAGE	268,101	43.79±0.59	63.26±1.09	38.99±0.85	90.00±2.30	84.26±2.98	42.58±0.18	—
GraphSAGE+F	1,259,866	**54.37±0.89**	**68.32±0.68**	37.82±0.45	**90.33±1.97**	**86.38±2.13**	42.63±0.21	+2.82
GraphSAGE+AF	268,111	53.09±0.78	67.05±0.94	**40.02±0.56**	90.16±2.30	86.17±2.98	**42.74±0.23**	+2.72
APPNP	134,085	36.15±0.75	52.93±1.71	40.46±0.64	91.31±1.97	87.66±2.13	**41.05±0.32**	—
APPNP+F	1,125,850	**38.73±1.06**	53.76±1.25	**40.78±0.74**	**91.48±2.30**	**87.87±2.34**	40.98±0.28	+0.67
APPNP+AF	134,095	37.81±1.52	**53.85±1.44**	40.61±0.74	**91.48±1.97**	86.81±2.77	40.99±2.77	+0.33
BernNet	134,106	51.15±1.09	67.96±1.05	40.72±0.80	93.28±1.48	90.21±2.35	**41.36±0.44**	—
BernNet+F	1,125,871	**55.22±0.64**	**71.66±1.18**	40.91±0.71	**93.44±1.80**	**90.85±2.34**	41.34±0.37	+1.46
BernNet+AF	134,116	50.51±1.15	70.59±1.01	**41.70±1.14**	90.98±1.97	90.64±2.55	41.30±0.39	+0.17
ChebNetII	134,096	57.78±0.84	71.71±1.40	40.70±0.77	92.79±1.48	88.94±2.78	48.60±0.17	—
ChebNetII+F	1,125,861	**60.55±0.64**	**74.05±0.68**	**41.37±0.67**	93.93±0.98	87.23±3.62	48.82±0.19	+0.91
ChebNetII+AF	134,106	56.81±0.95	73.41±0.74	41.07±1.06	**94.10±1.48**	**89.15±2.77**	**49.06±0.31**	+0.51

Summary

- Most advanced neural networks can be revisited and improved through the lens of optimal transport
- Lead to interpretable and strong models for various applications
- Scalability and efficiency are main bottlenecks.

WGFormer [ICML'25]: `https://arxiv.org/abs/2410.09795`

- Code: `https://github.com/SDS-Lab/WGFormer`

QW Loss [WWW'24, 26]: `https://arxiv.org/abs/2310.11762`

- Code: `https://github.com/SDS-Lab/QW_Loss`

OT Pooling [TPAMI'23]:
`https://ieeexplore.ieee.org/abstract/document/10247589/`

- Code: `https://github.com/SDS-Lab/ROT-Pooling`

Thanks!
5-min Break and QA

Part 1 **Computational Optimal Transport (Hongteng Xu)**

- Preliminaries and basic concepts
- Typical computation methods

Part 2 **Representation Learning Driven by OT (Dixin Luo)**

- OT-based multi-modal learning
- Monge gap and its Gromovization for information bottleneck

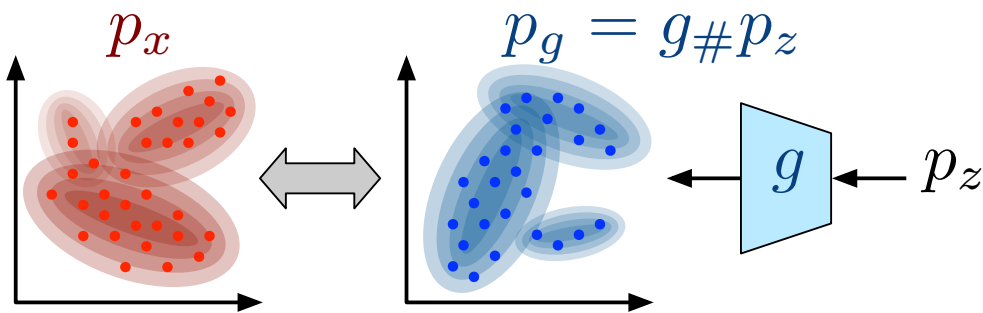
Part 3 **Neural Network Design Driven by OT (Minjie Cheng)**

- OT-based Transformer
- OT-based graph neural network

Part 4 **Recent Progress in Generative Modeling (Hongteng Xu)**

- OT-based flow matching
- Applications of optimal acceleration transport

Generative Modeling = Distribution Fitting and Matching



- $g : \mathcal{Z} \mapsto \mathcal{X}$ is the generator/decoder.
- p_z is the (predefined) latent distribution, and $p_g = g_\# p_z$ is the model distribution.
- Learn g to fit data distribution p_x by p_g.

Outline

1. **A Quick Review of Generative Modeling Based on Static OT**
   - Wasserstein GAN (WGAN)
   - Wasserstein Autoencoder (WAE)
   - Recent Variants

2. **Recent Generative Modeling Methods Based on Dynamic OT**
   - OT-based conditional flow matching
   - Improved flow matching based on Optimal Acceleration Transport (OAT)

Classic OT-based Generative Modeling Paradigms

Solution 1: Minimize $\mathcal{W}_1$ approximately in its dual-form or its SW surrogates:

- **WGAN:** Wasserstein generative adversarial networks. ICML, 2017.
- **WGAN-GP:** Improved training of Wasserstein GANs. NeurIPS, 2017.
- **Max-SWG:** Max-sliced Wasserstein distance and its use for GANs. CVPR, 2019.
- **Amortized Max-SWG** Amortized projection optimization for sliced Wasserstein generative models. NeurIPS, 2022.

Classic OT-based Generative Modeling Paradigms

Solution 1: Minimize $\mathcal{W}_1$ approximately in its dual-form or its SW surrogates:

- **WGAN:** Wasserstein generative adversarial networks. ICML, 2017.
- **WGAN-GP:** Improved training of Wasserstein GANs. NeurIPS, 2017.
- **Max-SWG:** Max-sliced Wasserstein distance and its use for GANs. CVPR, 2019.
- **Amortized Max-SWG** Amortized projection optimization for sliced Wasserstein generative models. NeurIPS, 2022.

Solution 2: Minimize $\mathcal{W}_2$ approximately in its primal-form:

- **WAE:** Wasserstein Auto-Encoders. ICLR, 2018.
- **SinkDiff:** Learning generative models with Sinkhorn divergences. AISTATS, 2018.
- **SWAE:** Sliced Wasserstein auto-encoders. ICLR, 2018.
- **RAE:** Learning autoencoders with relational regularization. ICML, 2020.
- **Conditional Transport:** Exploiting Chain Rule and Bayes' Theorem to Compare Probability Distributions. NeurIPS, 2021.
- **HCP-AE:** Hilbert curve projection distance for distribution comparison. TPAMI, 2024.

Wasserstein Generative Adversarial Network (WGAN)

Wasserstein Generative Adversarial Network (WGAN) [Arjovsky et al., 2017]: Fit the model distribution p_g by minimizing its 1-Wasserstein distance to the data distribution p_x **in the dual-form**:

$$\mathcal{W}_1(p_x, p_g) = \inf_{\pi \in \Pi(p_x, p_g)} \mathbb{E}_{(x, g(z)) \sim \pi}[\|x - g(z)\|_1] = \sup_{f \in L_1} \mathbb{E}_x[f(x)] - \mathbb{E}_z[f(g(z))] \quad (87)$$

Wasserstein Generative Adversarial Network (WGAN)

Wasserstein Generative Adversarial Network (WGAN) [Arjovsky et al., 2017]: Fit the model distribution p_g by minimizing its 1-Wasserstein distance to the data distribution p_x **in the dual-form**:

$$\mathcal{W}_1(p_x, p_g) = \inf_{\pi \in \Pi(p_x, p_g)} \mathbb{E}_{(x, g(z)) \sim \pi}[\|x - g(z)\|_1] = \sup_{f \in L_1} \mathbb{E}_x[f(x)] - \mathbb{E}_z[f(g(z))] \tag{87}$$

Therefore, we have

$$\inf_g \mathcal{W}_1(p_x, p_g) \Longleftrightarrow \inf_g \sup_{f \in L_1} \mathbb{E}_x[f(x)] - \mathbb{E}_z[f(g(z))] \tag{88}$$

Wasserstein Generative Adversarial Network (WGAN)

Wasserstein Generative Adversarial Network (WGAN) [Arjovsky et al., 2017]: Fit the model distribution p_g by minimizing its 1-Wasserstein distance to the data distribution p_x **in the dual-form**:

$$\mathcal{W}_1(p_x, p_g) = \inf_{\pi \in \Pi(p_x, p_g)} \mathbb{E}_{(x, g(z)) \sim \pi}[\|x - g(z)\|_1] = \sup_{f \in L_1} \mathbb{E}_x[f(x)] - \mathbb{E}_z[f(g(z))] \tag{87}$$

Therefore, we have

$$\inf_g \mathcal{W}_1(p_x, p_g) \Longleftrightarrow \inf_g \sup_{f \in L_1} \mathbb{E}_x[f(x)] - \mathbb{E}_z[f(g(z))] \tag{88}$$

Given a set of samples $X = \{x_n\}_{n=1}^N$ and a set of latent code $Z = \{z_n\}_{n=1}^N$, we have

$$\min_g \max_{f \in L_1} \sum_n [f(x_n)] - \sum_n [f(g(z_n))] \tag{89}$$

Wasserstein Autoencoder (WAE)

Besides approximate the primal form of W_p by EOT, another way is applying the autoencoding architecture.

Wasserstein Autoencoder (WAE)

Besides approximate the primal form of W_p by EOT, another way is applying the autoencoding architecture.

- **Wasserstein autoencoder (WAE)** fits the model distribution p_g by minimizing its W_2 distance to the data distribution p_x approximately.

Wasserstein Autoencoder (WAE)

Besides approximate the primal form of W_p by EOT, another way is applying the autoencoding architecture.

- **Wasserstein autoencoder (WAE)** fits the model distribution p_g by minimizing its W_2 distance to the data distribution p_x approximately.

$$\inf_g \mathcal{W}_2(p_x, p_g) \approx \inf_{g,f} \underbrace{\mathbb{E}_{p_x}\mathbb{E}_{q_{z|x;f}}[d_x(x, g(z))]}_{\text{reconstruction loss}} + \underbrace{\gamma d_p(\overbrace{\mathbb{E}_{p_x}[q_{z|x;f}]}^{q_{z;f}}, p_z)}_{\text{distance(posterior, prior)}}, \tag{90}$$

- $q_{z|x;f}$ is the posterior of z given x, parameterized by an **encoder** $f : \mathcal{X} \mapsto \mathcal{Z}$.
- $q_{z;f} = \mathbb{E}_{p_x}[q_{z|x;f}]$ is the expectation of the posterior distributions.
- p_z is the prior of z.

Wasserstein Auto-Encoders. ICLR 2018.

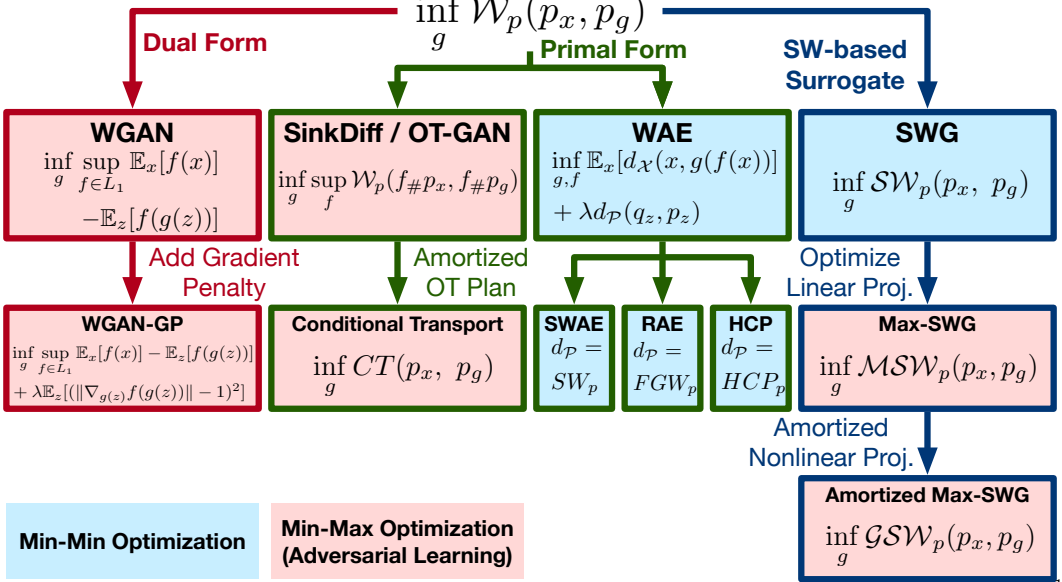
Comparisons with Other Autoencoders

Method	$f : \mathcal{X} \mapsto \mathcal{Z}$	Prior p_z	Learn p_z	$d_p(q_{z;Q}, p_z)$
VAE	Probabilistic	$\mathcal{N}(z; 0, I)$	No	KL
GMVAE	Probabilistic	$\frac{1}{K}\sum_k \mathcal{N}(z; u_k, \Sigma_k)$	No	KL
VampPrior	Probabilistic	$\frac{1}{K}\sum_k \mathcal{N}(z; Q(x_k))$	Yes	KL

Comparisons with Other Autoencoders

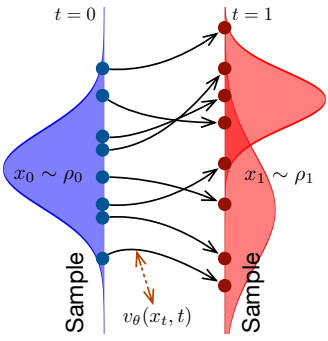
Method	$f : \mathcal{X} \mapsto \mathcal{Z}$	Prior p_z	Learn p_z	$d_p(q_{z;Q}, p_z)$
VAE	Probabilistic	$\mathcal{N}(z; 0, I)$	No	KL
GMVAE	Probabilistic	$\frac{1}{K}\sum_k \mathcal{N}(z; u_k, \Sigma_k)$	No	KL
VampPrior	Probabilistic	$\frac{1}{K}\sum_k \mathcal{N}(z; Q(x_k))$	Yes	KL
WAE	Deterministic	$\mathcal{N}(z; 0, I)$	No	MMD/GAN
SWAE	Deterministic	$\mathcal{N}(z; 0, I)$	No	SW_2
RAE	Probabilistic/Deterministic	$\frac{1}{K}\sum_k \mathcal{N}(z; u_k, \Sigma_k)$	Yes	FGW_2
HCP-AE	Probabilistic/Deterministic	$\mathcal{N}(z; 0, I)$	No	HCP_2

A (Partial) Family Tree of OT-based Generative Models



The above methods are based on the static definition of OT (i.e., Kantorovich-form OT). The dynamic-form OT triggers more recent generative modeling methods — flow matching.

Flow Matching (FM) and Classical Two-Phase FM

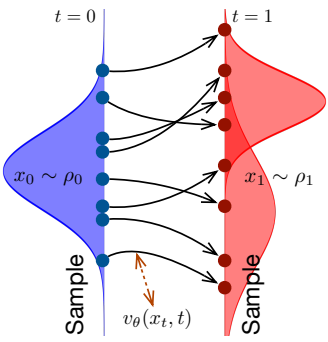


1) Flow Matching (FM) (Sample Space): Learn a velocity field $v_\theta(x, t)$ capturing the transport of probability mass from a prior ρ_0 to a data ρ_1.

Flow Matching for Generative Modeling. ICLR, 2023.

Improving and generalizing flow-based generative models with minibatch optimal transport. TMLR, 2024.

Flow Matching (FM) and Classical Two-Phase FM



1) Flow Matching (FM) (Sample Space): Learn a velocity field $v_\theta(x, t)$ capturing the transport of probability mass from a prior ρ_0 to a data ρ_1.

- **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0, 1)$, with an auxiliary variable $z \sim \pi$:

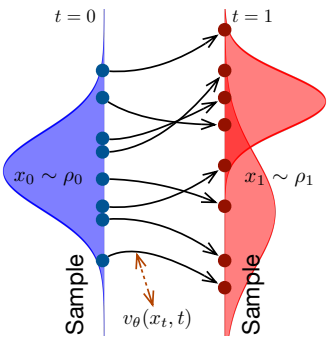
$$\min_\theta \ \mathbb{E}_{z,\ t,\ x}[\|v_\theta(x, t) - v_t(x|z)\|^2], \tag{91}$$

Generate new data by $\hat{x}_1 = x_0 + \int_0^1 v_\theta(x_t, t)\mathrm{d}t$. In practice, $x_{t+\Delta t} = x_t + \Delta t \cdot v_\theta(x_t, t)$.

Flow Matching for Generative Modeling. ICLR, 2023.

Improving and generalizing flow-based generative models with minibatch optimal transport. TMLR, 2024.

Flow Matching (FM) and Classical Two-Phase FM



1) Flow Matching (FM) (Sample Space): Learn a velocity field $v_\theta(x, t)$ capturing the transport of probability mass from a prior ρ_0 to a data ρ_1.

- **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0, 1)$, with an auxiliary variable $z \sim \pi$:

$$\min_\theta \ \mathbb{E}_{z,\, t,\, x}[\|v_\theta(x, t) - v_t(x|z)\|^2], \tag{91}$$

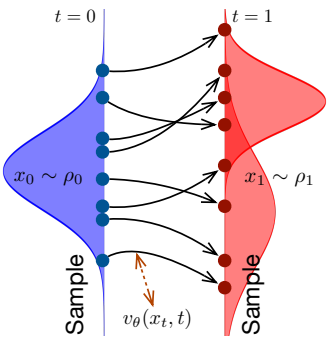
  Generate new data by $\hat{x}_1 = x_0 + \int_0^1 v_\theta(x_t, t)\mathrm{d}t$. In practice, $x_{t+\Delta t} = x_t + \Delta t \cdot v_\theta(x_t, t)$.

- **FM (Lipman et al.)**:
  $p_t(x|z) = \mathcal{N}\big(tz, (t\sigma - t + 1)^2\big),\ \pi = \rho_1$

Flow Matching for Generative Modeling. ICLR, 2023.

Improving and generalizing flow-based generative models with minibatch optimal transport. TMLR, 2024.

Flow Matching (FM) and Classical Two-Phase FM



Flow Matching for Generative Modeling. ICLR, 2023.

Improving and generalizing flow-based generative models with minibatch optimal transport. TMLR, 2024.

1) Flow Matching (FM) (Sample Space): Learn a velocity field $v_\theta(x, t)$ capturing the transport of probability mass from a prior ρ_0 to a data ρ_1.

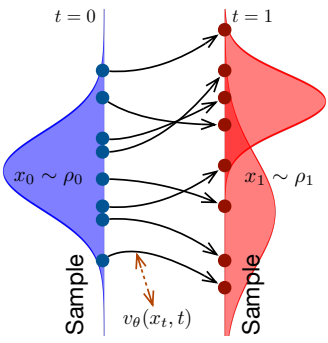
- **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0, 1)$, with an auxiliary variable $z \sim \pi$:

$$\min_\theta \ \mathbb{E}_{z,\ t,\ x}[\|v_\theta(x, t) - v_t(x|z)\|^2], \tag{91}$$

  Generate new data by $\hat{x}_1 = x_0 + \int_0^1 v_\theta(x_t, t)\mathrm{d}t$. In practice, $x_{t+\Delta t} = x_t + \Delta t \cdot v_\theta(x_t, t)$.

- **FM (Lipman et al.)**: $p_t(x|z) = \mathcal{N}\big(tz, (t\sigma - t + 1)^2\big),\ \pi = \rho_1$
- **I-CFM**: $x_t = (1 - t) \cdot x_0 + t \cdot x_1,\ \pi = \rho_0 \times \rho_1$

Flow Matching (FM) and Classical Two-Phase FM



Flow Matching for Generative Modeling. ICLR, 2023.

Improving and generalizing flow-based generative models with minibatch optimal transport. TMLR, 2024.

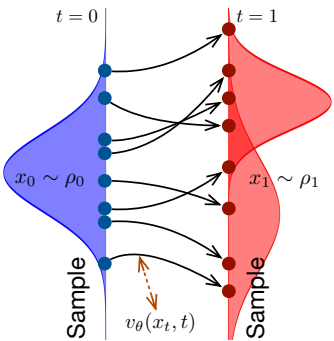
1) Flow Matching (FM) (Sample Space): Learn a velocity field $v_\theta(x, t)$ capturing the transport of probability mass from a prior ρ_0 to a data ρ_1.

- **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0, 1)$, with an auxiliary variable $z \sim \pi$:

$$\min_\theta \ \mathbb{E}_{z,\ t,\ x}[\|v_\theta(x, t) - v_t(x|z)\|^2], \tag{91}$$

  Generate new data by $\hat{x}_1 = x_0 + \int_0^1 v_\theta(x_t, t)\mathrm{d}t$. In practice, $x_{t+\Delta t} = x_t + \Delta t \cdot v_\theta(x_t, t)$.
- **FM (Lipman et al.)**: $p_t(x|z) = \mathcal{N}\big(tz, (t\sigma - t + 1)^2\big),\ \pi = \rho_1$
- **I-CFM**: $x_t = (1 - t) \cdot x_0 + t \cdot x_1,\ \pi = \rho_0 \times \rho_1$
- **OT-CFM**: **Optimal Transport (OT)** perspective...

Flow Matching (FM) and Classical Two-Phase FM



1) Flow Matching (FM) (Sample Space): Learn a velocity field $v_\theta(x, t)$ capturing the transport of probability mass from a prior ρ_0 to a data ρ_1.

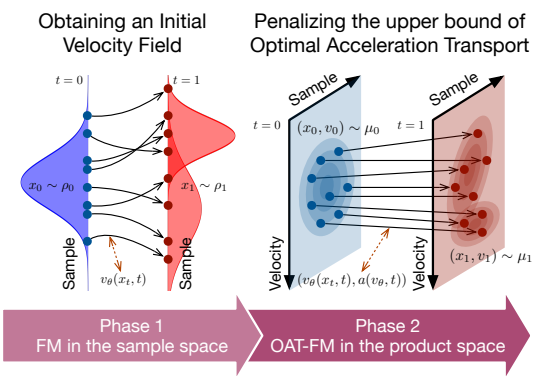
2) Classical Two-Phase FM (Sample Space): ReFlow, Consistency Distillation

- **Pros:** Few sampling steps, competitive results, ...
- **Cons:** Require a large number of noise data pair, the risk of distribution drift, ...

Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. ICLR, 2023.

Consistency Models. ICML, 2023.

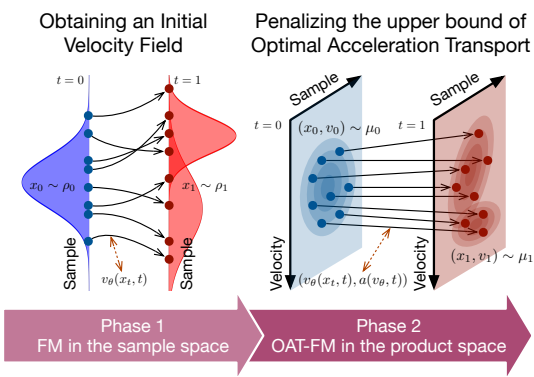
OAT-FM: A Novel Two-Phase FM



3) OAT-FM (Sample $\times$ Velocity Space): A novel two-phase FM based on **Optimal Acceleration Transport (OAT)**

- Given a pre-trained flow-based generator $\boldsymbol{v}_\theta$

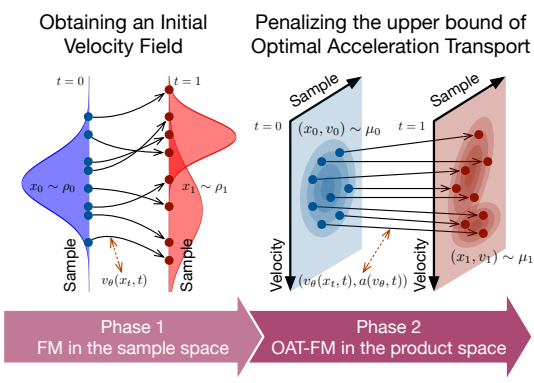
OAT-FM: A Novel Two-Phase FM



3) OAT-FM (Sample $\times$ Velocity Space): A novel two-phase FM based on **Optimal Acceleration Transport (OAT)**

- Given a pre-trained flow-based generator $\boldsymbol{v}_\theta$
- Minimizes the acceleration transport between μ_0 and μ_1

OAT-FM: A Novel Two-Phase FM



Theoretical Guarantee, **Practical Computation**, **Efficient Training**, **Consistent Improvement**

OAT-FM: Optimal Acceleration Transport for Improved Flow Matching. arXiv, 2025.

The Motivations: Recall Dynamic Optimal Transport

Intuition: The "Least Effort" Principle

- **The Task:** Moving a pile of sand (source ρ_0) to a target shape (target ρ_1).
- **The Goal:** Find the most efficient flow that minimizes the total energy spent.

The Motivations: Recall Dynamic Optimal Transport

Intuition: The "Least Effort" Principle

- **The Task:** Moving a pile of sand (source ρ_0) to a target shape (target ρ_1).
- **The Goal:** Find the most efficient flow that minimizes the total energy spent.

Dynamic Formulation (Benamou-Brenier): The Wasserstein-2 distance finds the path of **Least Kinetic Energy**:

$$\mathcal{W}_2^2(\rho_0, \rho_1) = \min_{\rho, v} \int_0^1 \int_{\mathcal{X}} \underbrace{\frac{1}{2}\rho(x,t)\|v(x,t)\|_2^2}_{\text{Kinetic Energy Density}} \mathrm{d}x\mathrm{d}t, \tag{92}$$

subject to:

- $\underbrace{\partial_t \rho + \nabla_x \cdot (v\rho) = 0}_{\text{Conservation of Mass}}$: No mass is created or destroyed.
- $\rho(\cdot, 0) = \rho_0, \rho(\cdot, 1) = \rho_1$: Start at noise, end at data.

A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. Numerische Mathematik, 2000.

The Motivations: Optimal Transport Perspective of FM

1. **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0,1)$, with an auxiliary variable $z \sim \pi$:

$$\min_\theta \quad \mathbb{E}_{z\sim\pi,\ t\sim\mathsf{Unif}[0,1],\ x\sim p_t(\cdot|z)}[\|v_\theta(x,t) - v_t(x|z)\|^2], \tag{93}$$

The Motivations: Optimal Transport Perspective of FM

1. **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0,1)$, with an auxiliary variable $z \sim \pi$:

$$\min_\theta \quad \mathbb{E}_{z\sim\pi,\ t\sim\mathsf{Unif}[0,1],\ x\sim p_t(\cdot|z)}[\|v_\theta(x,t) - v_t(x|z)\|^2], \tag{93}$$

2. **OT-CFM**: implements CFM by setting the distribution π in (93) as the OT plan corresponding to $\mathcal{W}_2^2(\rho_0,\rho_1)$ and $x_t = (1-t)\cdot x_0 + t\cdot x_1$.

The Motivations: Optimal Transport Perspective of FM

1. **Conditional FM (CFM)**: Set $\rho_0 = \mathcal{N}(0,1)$, with an auxiliary variable $z \sim \pi$:

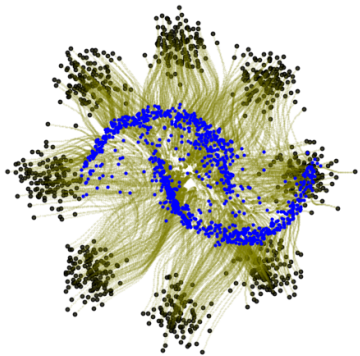
$$\min_\theta \quad \mathbb{E}_{z\sim\pi,\ t\sim\mathsf{Unif}[0,1],\ x\sim p_t(\cdot|z)}[\|v_\theta(x,t) - v_t(x|z)\|^2], \tag{93}$$

2. **OT-CFM**: implements CFM by setting the distribution π in (93) as the OT plan corresponding to $\mathcal{W}_2^2(\rho_0,\rho_1)$ and $x_t = (1-t)\cdot x_0 + t\cdot x_1$.

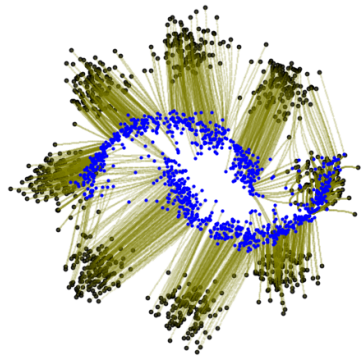
$$\begin{aligned} &\min_\theta \overbrace{\mathbb{E}_{(x_0,x_1)\sim\pi^*,\ t\sim\mathsf{Unif}[0,1]}[\|v_\theta(x_t,t) - (x_1 - x_0)\|^2]}^{\textbf{Upper-level: } \mathcal{L}_{\mathsf{CFM}}}, \\ &\text{s.t. } \pi^* = \overbrace{\underset{\pi\in\Pi(\rho_0,\rho_1)}{\arg\min}\ \mathbb{E}_\pi[\|x_1 - x_0\|_2^2]}^{\textbf{Lower-level: } \mathcal{W}_2^2(\rho_0,\rho_1)}, \end{aligned} \tag{94}$$

This is a **Bi-level Optimization Problem**.

The Motivations: Optimal Transport Perspective of FM

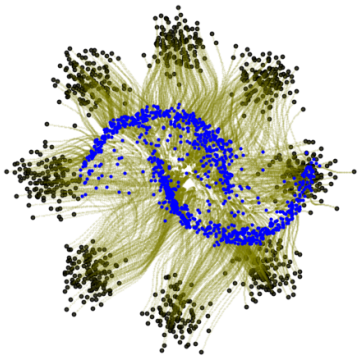


(a) I-CFM, $\pi = \rho_0 \times \rho_1$

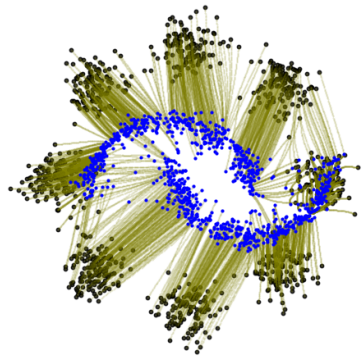


(b) OT-CFM, $\pi = \pi^*$

The Motivations: Optimal Transport Perspective of FM



(a) I-CFM, $\pi = \rho_0 \times \rho_1$



(b) OT-CFM, $\pi = \pi^*$

The objective of OT-CFM regresses $v_\theta(x_t, t)$ to the **constant velocity** $(x_1 - x_0)$. **However, constant velocity is sufficient but not necessary for straightening flows**.

The Motivations: Optimal Transport Perspective of FM

Proposition 3 (Straightness Criterion)

The trajectory is straight if and only if the velocity direction is time invariant and the ***acceleration*** *is everywhere parallel to the velocity. The classical (first-order) dynamical optimal transport is recovered as the special case with zero acceleration.*

Proposition 3 (Straightness Criterion)

The trajectory is straight if and only if the velocity direction is time invariant and the ***acceleration*** *is everywhere parallel to the velocity. The classical (first-order) dynamical optimal transport is recovered as the special case with zero acceleration.*

Can we leverage the advantages of OT by minimizing acceleration instead of velocity?

Optimal Acceleration Transport (OAT)

Optimal Acceleration Transport: Bridges μ_0 and μ_1 in the product space $(\mathcal{X} \times \mathcal{V})$ by finding a path that minimizes total squared acceleration under second-order dynamics.

Optimal Acceleration Transport (OAT)

Optimal Acceleration Transport: Bridges μ_0 and μ_1 in the product space $(\mathcal{X} \times \mathcal{V})$ by finding a path that minimizes total squared acceleration under second-order dynamics.

Definition 4 (Dynamic Formulation of Optimal Acceleration Transport [1])

Let $\mathcal{X} \subset \mathbb{R}^d$ be the sample space and $\mathcal{V} \subset \mathbb{R}^d$ the velocity space (by default $\mathcal{V} = \mathbb{R}^d$). For $\mu_0, \mu_1 \in \mathbb{P}(\mathcal{X} \times \mathcal{V})$, the optimal acceleration transport between them is defined as

$$\mathcal{A}_2^2(\mu_0, \mu_1) := \min_{\mu,\, a} \int_0^1 \int_{\mathcal{X}\times\mathcal{V}} \frac{1}{2}\, \mu(x, v, t)\, \|a(x, v, t)\|_2^2 \, \mathrm{d}x\, \mathrm{d}v\, \mathrm{d}t, \tag{95}$$

subject to the Vlasov equation $\partial_t \mu + v \cdot \nabla_x \mu + \nabla_v \cdot \big(a\, \mu\big) = 0$, with boundary conditions $\mu(\cdot, \cdot, 0) = \mu_0$ and $\mu(\cdot, \cdot, 1) = \mu_1$. Here, $a : \mathcal{X} \times \mathcal{V} \times [0, 1] \mapsto \mathbb{R}^d$ is the acceleration field, and the Vlasov equation expresses conservation of mass in the product space.

Optimal Acceleration Transport (OAT)

Definition 5 (Kantorovich formulation of OAT [2,3,4])

Given $z_0 = (x_0, v_0) \sim \mu_0$ and $z_1 = (x_1, v_1) \sim \mu_1$, the OAT problem is equivalent to solving an optimal coupling w.r.t. squared acceleration cost, i.e.,

$$\begin{aligned} &\mathcal{A}_2^2(\mu_0,\, \mu_1) \\ &= \min_{\pi \in \Pi(\mu_0,\mu_1)} \mathbb{E}_{(z_0,z_1)\sim\pi}[c_{\mathsf{A}}^2(z_0, z_1)] \\ &= \min_{\pi \in \Pi(\mu_0,\mu_1)} \mathbb{E}_{(z_0,z_1)\sim\pi}\Big[12 \underbrace{\Big\| \frac{x_1 - x_0}{T} - \frac{v_1 + v_0}{2} \Big\|^2}_{\text{velocity alignment}} + \underbrace{\|v_1 - v_0\|^2}_{\text{acceleration penalty}} \Big], \end{aligned} \tag{96}$$

where $T > 0$ denotes the time horizon between μ_0 and μ_1, which is 1 in our case.

Optimal Acceleration Transport (OAT)

Joint Matching: Couples samples in $\mathcal{X} \times \mathcal{V}$, aligning both position and velocity.

Theorem 6 (Straightening Flow via OAT)

Given two boundary distributions $\mu_0, \mu_1 \in \mathbb{P}(\mathcal{X} \times \mathcal{V})$, OAT admits an optimal coupling $\pi^* \in \Pi(\mu_0, \mu_1)$ for the static problem in (96). For every (x_0, v_0), $(x_1, v_1) \sim \pi^*$, the corresponding trajectory is straight iff v_0 and v_1 are collinear with $x_1 - x_0$. Otherwise, it bends exactly to match the endpoints' orthogonal components.

OAT-FM: From Constant Velocity to Acceleration Control

Standard FM: Enforces $v_\theta(x, t) \approx$ constant velocity.

OAT-FM: From Constant Velocity to Acceleration Control

Standard FM: Enforces $v_\theta(x, t) \approx$ constant velocity.

OAT-FM Motivation:

- Shift to ***acceleration minimization***.
- Desideratum: For pre-trained v_θ, refine using OAT for better performance.

OAT-FM: From Constant Velocity to Acceleration Control

Standard FM: Enforces $v_\theta(x,t) \approx$ constant velocity.

OAT-FM Motivation:

- Shift to ***acceleration minimization***.
- Desideratum: For pre-trained v_θ, refine using OAT for better performance.

Problem Setup:

- Trajectory endpoints: $z_0 = (x_0, v_0)$ and $z_1 = (x_1, v_1)$.
- Path x_t: Linear interpolation $x_t = (1-t)x_0 + tx_1$.
- Model state: $z_t(\theta) = (x_t, v_\theta(x_t,t))$.

OAT-FM: The Objective Function

Cost Function $\ell_{\mathcal{A}}$:

$$\ell_{\mathcal{A}}(z_0, z_1, t; \theta) = \alpha \underbrace{\left\| \frac{x_t - x_0}{t} - \frac{v_0 + v_\theta}{2} \right\|_2^2}_{\text{Velocity Alignment } (0 \to t)} + (1-\alpha) \underbrace{\|v_\theta - v_0\|_2^2}_{\text{Accel. Penalty } (0 \to t)}$$
$$+ \alpha \underbrace{\left\| \frac{x_1 - x_t}{1-t} - \frac{v_\theta + v_1}{2} \right\|_2^2}_{\text{Velocity Alignment } (t \to 1)} + (1-\alpha) \underbrace{\|v_1 - v_\theta\|_2^2}_{\text{Accel. Penalty } (t \to 1)} \tag{97}$$

OAT-FM: The Objective Function

Cost Function $\ell_{\mathcal{A}}$:

$$\ell_{\mathcal{A}}(z_0, z_1, t; \theta) = \alpha \underbrace{\left\| \frac{x_t - x_0}{t} - \frac{v_0 + v_\theta}{2} \right\|_2^2}_{\text{Velocity Alignment } (0 \to t)} + (1-\alpha) \underbrace{\|v_\theta - v_0\|_2^2}_{\text{Accel. Penalty } (0 \to t)}$$
$$+ \alpha \underbrace{\left\| \frac{x_1 - x_t}{1-t} - \frac{v_\theta + v_1}{2} \right\|_2^2}_{\text{Velocity Alignment } (t \to 1)} + (1-\alpha) \underbrace{\|v_1 - v_\theta\|_2^2}_{\text{Accel. Penalty } (t \to 1)} \tag{97}$$

Key Properties:

- Hyperparameter α balances *alignment* vs. *acceleration*.
- With $\alpha = \frac{12}{13}$, recovers OAT cost structure: $\ell_{\mathcal{A}} = \frac{1}{13}(c_{\mathcal{A}}^2(z_0, z_t) + c_{\mathcal{A}}^2(z_t, z_1))$.

OAT-FM: The Bi-level Optimization Problem

OAT-FM Objective: We fine-tune the flow model by solving the following **Bi-level Optimization Problem**:

$$\min_{\theta} \overbrace{\mathbb{E}_{(z_0,z_1)\sim\pi^*,\ t\sim\mathsf{Unif}[0,1]}[\ell_{\mathcal{A}}(z_0,z_1,t;\,\theta)]}^{\textbf{Upper-level: } \mathcal{L}_{\mathsf{OAT}}(\mu_0,\mu_1;\alpha)},$$
$$\text{s.t. } \pi^* = \overbrace{\underset{\pi\in\Pi(\mu_0,\mu_1)}{\arg\min}\ \mathbb{E}_{(z_0,z_1)\sim\pi}[c^2_{\mathcal{A}}(z_0,\,z_1)]}^{\textbf{Lower-level: } \mathcal{A}_2^2(\mu_0,\mu_1)}. \tag{98}$$

OAT-FM: The Bi-level Optimization Problem

OAT-FM Objective: We fine-tune the flow model by solving the following **Bi-level Optimization Problem**:

$$\min_{\theta} \overbrace{\mathbb{E}_{(z_0,z_1)\sim\pi^*,\ t\sim\mathsf{Unif}[0,1]}[\ell_{\mathcal{A}}(z_0,z_1,t;\,\theta)]}^{\textbf{Upper-level: } \mathcal{L}_{\mathsf{OAT}}(\mu_0,\mu_1;\alpha)},$$
$$\text{s.t. } \pi^* = \overbrace{\underset{\pi\in\Pi(\mu_0,\mu_1)}{\arg\min}\ \mathbb{E}_{(z_0,z_1)\sim\pi}[c^2_{\mathcal{A}}(z_0,\,z_1)]}^{\textbf{Lower-level: } \mathcal{A}_2^2(\mu_0,\mu_1)}. \tag{98}$$

- **Lower-level**: Finds the optimal coupling π^* that minimizes total acceleration in the product space.
- **Upper-level**: Aligns the learned flow with the OAT geodesics via $\ell_{\mathcal{A}}$.
- **Parameter** α: Balances *directional alignment* and *acceleration minimization*.

OAT-FM vs. OT-CFM

Component	OT-CFM	OAT-FM (Proposed)
Space	Sample Space $\mathcal{X}$	Product Space $\mathcal{X} \times \mathcal{V}$

OAT-FM vs. OT-CFM

Component	**OT-CFM**	**OAT-FM (Proposed)**
Space	Sample Space $\mathcal{X}$	Product Space $\mathcal{X} \times \mathcal{V}$
Dynamics	Continuity Equation	Vlasov Equation
	$\partial_t \rho + \nabla_x \cdot (v\rho) = 0$	$\partial_t \mu + \nabla_x \cdot (v\mu) + \nabla_v \cdot (a\mu) = 0$

OAT-FM vs. OT-CFM

Component	**OT-CFM**	**OAT-FM (Proposed)**
Space	Sample Space $\mathcal{X}$	Product Space $\mathcal{X} \times \mathcal{V}$
Dynamics	Continuity Equation	Vlasov Equation
	$\partial_t \rho + \nabla_x \cdot (v\rho) = 0$	$\partial_t \mu + \nabla_x \cdot (v\mu) + \nabla_v \cdot (a\mu) = 0$
Lower-level	Optimal Transport (OT)	Optimal Acceleration Transport (OAT)
(Coupling)	$\pi^* = \arg\min \mathbb{E}[\|x_1 - x_0\|^2]$	$\pi^* = \arg\min \mathbb{E}[c^2_{\mathcal{A}}(z_0, z_1)]$

OAT-FM vs. OT-CFM

Component	**OT-CFM**	**OAT-FM (Proposed)**
Space	Sample Space $\mathcal{X}$	Product Space $\mathcal{X} \times \mathcal{V}$
Dynamics	Continuity Equation	Vlasov Equation
	$\partial_t \rho + \nabla_x \cdot (v\rho) = 0$	$\partial_t \mu + \nabla_x \cdot (v\mu) + \nabla_v \cdot (a\mu) = 0$
Lower-level	Optimal Transport (OT)	Optimal Acceleration Transport (OAT)
(Coupling)	$\pi^* = \arg\min \mathbb{E}[\|x_1 - x_0\|^2]$	$\pi^* = \arg\min \mathbb{E}[c^2_{\mathcal{A}}(z_0, z_1)]$
Upper-level	Velocity Matching	Acceleration Matching Proxy
(Objective)	$\min \|v_\theta - (x_1 - x_0)\|^2$	$\min \ell_{\mathcal{A}}(z_0, z_1, t; \theta)$

OAT-FM vs. OT-CFM

Component	OT-CFM	OAT-FM (Proposed)
Space	Sample Space $\mathcal{X}$	Product Space $\mathcal{X} \times \mathcal{V}$
Dynamics	Continuity Equation $\partial_t \rho + \nabla_x \cdot (v\rho) = 0$	Vlasov Equation $\partial_t \mu + \nabla_x \cdot (v\mu) + \nabla_v \cdot (a\mu) = 0$
Lower-level (Coupling)	Optimal Transport (OT) $\pi^* = \arg\min \mathbb{E}[\|x_1 - x_0\|^2]$	Optimal Acceleration Transport (OAT) $\pi^* = \arg\min \mathbb{E}[c^2_{\mathcal{A}}(z_0, z_1)]$
Upper-level (Objective)	Velocity Matching $\min \|v_\theta - (x_1 - x_0)\|^2$	Acceleration Matching Proxy $\min \ell_{\mathcal{A}}(z_0, z_1, t; \theta)$
Mechanism (Straightening)	**Constant Velocity** $\min \int_0^1 \|v_t\|^2 \mathrm{d}t \implies \ddot{x} = 0$	**Minimized Acceleration** $\min \int_0^1 \|a_t\|^2 \mathrm{d}t \implies \ddot{v} = 0$

OAT Bound of OAT-FM

Theorem 7 (OAT Bound of OAT-FM)

The OAT-FM objective $\mathcal{L}_{OAT}(\mu_0, \mu_1; \alpha)$ is lower-bounded by a scaled version of the true OAT second-order discrepancy, i.e.,

$$\mathcal{L}_{OAT}(\mu_0, \mu_1; \alpha) \geq \frac{2}{27}\mathcal{A}_2^2(\mu_0, \mu_1), \tag{99}$$

with $\alpha = 2/3$, and the equality held if and only if $v_1 = v_0$ for π^*-almost every pair.

Efficient Implementation via Decomposable Structure

The Challenge: Solving OAT requires coupling in a 4D product space: $\pi(z_0, z_1) \in \Pi(\mu_0, \mu_1)$.

Efficient Implementation via Decomposable Structure

The Challenge: Solving OAT requires coupling in a 4D product space: $\pi(z_0, z_1) \in \Pi(\mu_0, \mu_1)$.

The Simplification (Decomposition): In FM, velocities are deterministic given samples: $v = v_\theta(x, t)$. This implies a *decomposable structure* for the coupling:

$$\pi(z_0, z_1) = \underbrace{\pi_x(x_0, x_1)}_{\text{Sample Coupling}} \cdot \underbrace{\delta_{v_\theta(x_0,0)}(v_0) \cdot \delta_{v_\theta(x_1,1)}(v_1)}_{\text{Deterministic Velocity Assignment}} . \tag{100}$$

Efficient Implementation via Decomposable Structure

The Resulting Lower-Level Problem: We reduce the OAT problem to a classic OT problem on samples:

$$\underset{\pi_x \in \Pi(\rho_0, \rho_1)}{\arg\min} \ \mathbb{E}_{(x_0,x_1)\sim\pi_x}\Big[12\|x_1 - x_0 - \bar{v}_{x_0,x_1}\|^2 + \|\tilde{v}_{x_0,x_1}\|_2^2\Big], \tag{101}$$

where ρ_0, ρ_1 are marginals on $\mathcal{X}$, and velocities are fixed by the current model:

- $\bar{v}_{x_0,x_1} = \frac{1}{2}(v_\theta(x_0, 0) + v_\theta(x_1, 1))$ (Mean Velocity)
- $\tilde{v}_{x_0,x_1} = v_\theta(x_1, 1) - v_\theta(x_0, 0)$ (Velocity Difference)

Efficient Implementation via Decomposable Structure

The Resulting Lower-Level Problem: We reduce the OAT problem to a classic OT problem on samples:

$$\underset{\pi_x \in \Pi(\rho_0,\rho_1)}{\arg\min} \ \mathbb{E}_{(x_0,x_1)\sim\pi_x}\Big[12\|x_1 - x_0 - \bar{v}_{x_0,x_1}\|^2 + \|\tilde{v}_{x_0,x_1}\|_2^2\Big], \tag{101}$$

where ρ_0, ρ_1 are marginals on $\mathcal{X}$, and velocities are fixed by the current model:

- $\bar{v}_{x_0,x_1} = \frac{1}{2}(v_\theta(x_0,0) + v_\theta(x_1,1))$ (Mean Velocity)
- $\tilde{v}_{x_0,x_1} = v_\theta(x_1,1) - v_\theta(x_0,0)$ (Velocity Difference)

Computational Complexity Analysis:

- **Exact OT (Linear Program):** $\mathcal{O}(B^3 \log \|\mathbf{C}\|_\infty)$.
- **Sinkhorn Algorithm (Approximation):** $\mathcal{O}(B^2 \log B)$.
  - Solved efficiently via iterative matrix scaling (highly parallelizable).
  - Recovers exact OT solution when $\epsilon \to 0$.

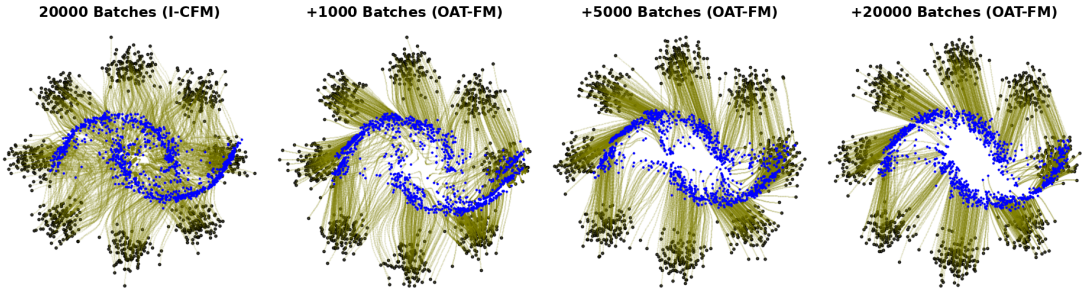
Algorithm Scheme: OAT-FM Training Loop

Algorithm 1 OAT-FM Training Procedure

Require: Pre-trained model v_{θ_0}, Dataset $\mathcal{D}$, Batch size B, EMA rate λ.
Ensure: Refined velocity field v_θ.

Initialize $v_\theta \leftarrow v_{\theta_0}$.
while training **do**
    // Step 1: Data Preparation
    Sample batch $\{x_{1,i}\}_{i=1}^{B} \sim \mathcal{D}$, $\{x_{0,i}\}_{i=1}^{B} \sim \mathcal{N}(0, I)$, $t \sim \mathcal{U}[0, 1]$.
    Estimate boundary velocities using current model:
    $\{v_{0,i} \leftarrow v_\theta(x_{0,i}, 0)\}_{i=1}^{B}, \quad \{v_{1,i} \leftarrow v_\theta(x_{1,i}, 1)\}_{i=1}^{B}$.
    // Step 2: Lower-Level (Coupling)
    Compute optimal coupling $\mathbf{T}^*$ by solving the reduced classic OT.
    Sample pairs $(x_1, x_0) \sim \mathbf{T}^*$ to get aligned batches.
    // Step 3: Upper-Level (Optimization)
    Interpolate $x_t \leftarrow (1 - t)x_0 + tx_1$, predict $v_t \leftarrow v_\theta(x_t, t)$.
    Compute $\mathcal{L}_{\text{OAT}}$ and update: $\theta' \leftarrow \theta - \nabla_\theta \mathcal{L}_{\text{OAT}}$.
    Update EMA: $\theta \leftarrow \text{stopgrad}(\lambda\theta + (1 - \lambda)\theta')$.
end while

Application 1: Low-dimensional OT Benchmark



Experimental Setup:

- **Tasks:** 5 standard 2D distribution mapping tasks (e.g., 8gaussians $\rightarrow$ moons).
- **Evaluation Metric:** 2-Wasserstein distance and Normalized Path Energy (NPE):

$$\text{NPE}(v_\theta) = \frac{|\text{PE}(v_\theta) - \mathcal{W}_2^2(\rho_0, \rho_1)|}{\mathcal{W}_2^2(\rho_0, \rho_1)}, \quad \text{with } \text{PE}(v_\theta) = \mathbb{E}_{x_0} \int_0^1 \|v_\theta(x_t, t)\|^2 \mathrm{d}t. \tag{102}$$

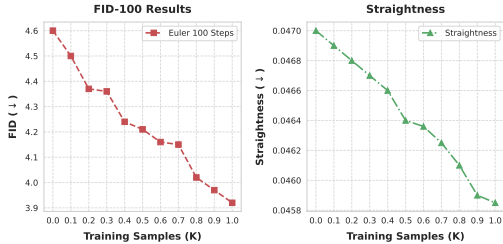
Application 1: Low-dimensional OT Benchmark

Task	$\mathcal{N}$ →8gs		8gs→moons		$\mathcal{N}$ →moons		$\mathcal{N}$ →scurve		moons→8gs	
Method	$\mathcal{W}_2^2 \downarrow$	NPE↓	$\mathcal{W}_2^2 \downarrow$	NPE↓	$\mathcal{W}_2^2 \downarrow$	NPE↓	$\mathcal{W}_2^2 \downarrow$	NPE↓	$\mathcal{W}_2^2 \downarrow$	NPE↓
FM	$0.58_{\pm 0.16}$	$0.24_{\pm 0.01}$	$5.80_{\pm 0.06}$	$0.05_{\pm 0.02}$	$0.15_{\pm 0.07}$	$0.27_{\pm 0.05}$	$\mathbf{0.81}_{\pm 0.39}$	$0.08_{\pm 0.04}$	$7.39_{\pm 0.45}$	$0.96_{\pm 0.05}$
+OAT-FM	$\mathbf{0.31}_{\pm 0.09}$	$\mathbf{0.02}_{\pm 0.01}$	$\mathbf{0.08}_{\pm 0.03}$	$\mathbf{0.01}_{\pm 0.01}$	$\mathbf{0.08}_{\pm 0.03}$	$\mathbf{0.03}_{\pm 0.01}$	$0.90_{\pm 0.18}$	$\mathbf{0.03}_{\pm 0.02}$	$\mathbf{0.28}_{\pm 0.10}$	$\mathbf{0.04}_{\pm 0.02}$
I-CFM	$0.45_{\pm 0.18}$	$0.30_{\pm 0.01}$	$0.18_{\pm 0.05}$	$1.40_{\pm 0.05}$	$0.11_{\pm 0.03}$	$0.52_{\pm 0.06}$	$1.16_{\pm 0.47}$	$0.03_{\pm 0.03}$	$0.74_{\pm 0.12}$	$1.19_{\pm 0.06}$
+OAT-FM	$\mathbf{0.32}_{\pm 0.10}$	$\mathbf{0.04}_{\pm 0.01}$	$\mathbf{0.15}_{\pm 0.03}$	$\mathbf{0.13}_{\pm 0.01}$	$\mathbf{0.07}_{\pm 0.02}$	$\mathbf{0.04}_{\pm 0.04}$	$\mathbf{1.12}_{\pm 0.45}$	$\mathbf{0.03}_{\pm 0.02}$	$\mathbf{0.50}_{\pm 0.11}$	$\mathbf{0.44}_{\pm 0.03}$
VP-CFM	$0.43_{\pm 0.14}$	$0.24_{\pm 0.01}$	$0.15_{\pm 0.02}$	$1.24_{\pm 0.05}$	$0.10_{\pm 0.03}$	$0.31_{\pm 0.07}$	$\mathbf{1.05}_{\pm 0.41}$	$0.22_{\pm 0.04}$	$1.39_{\pm 0.35}$	$1.22_{\pm 0.05}$
+OAT-FM	$\mathbf{0.31}_{\pm 0.12}$	$\mathbf{0.03}_{\pm 0.01}$	$\mathbf{0.09}_{\pm 0.01}$	$\mathbf{0.02}_{\pm 0.01}$	$\mathbf{0.07}_{\pm 0.02}$	$\mathbf{0.04}_{\pm 0.01}$	$1.10_{\pm 0.34}$	$\mathbf{0.03}_{\pm 0.02}$	$\mathbf{0.32}_{\pm 0.10}$	$\mathbf{0.10}_{\pm 0.02}$
SB-CFM	$0.51_{\pm 0.10}$	$\mathbf{0.01}_{\pm 0.01}$	$0.13_{\pm 0.04}$	$0.03_{\pm 0.01}$	$\mathbf{0.08}_{\pm 0.03}$	$\mathbf{0.04}_{\pm 0.03}$	$\mathbf{0.79}_{\pm 0.29}$	$0.04_{\pm 0.02}$	$0.36_{\pm 0.14}$	$\mathbf{0.03}_{\pm 0.02}$
+OAT-FM	$\mathbf{0.34}_{\pm 0.08}$	$0.03_{\pm 0.01}$	$\mathbf{0.07}_{\pm 0.01}$	$\mathbf{0.01}_{\pm 0.01}$	$0.09_{\pm 0.04}$	$0.10_{\pm 0.04}$	$0.80_{\pm 0.18}$	$\mathbf{0.02}_{\pm 0.02}$	$\mathbf{0.25}_{\pm 0.08}$	$\mathbf{0.03}_{\pm 0.02}$
OT-CFM	$0.35_{\pm 0.09}$	$\mathbf{0.01}_{\pm 0.01}$	$\mathbf{0.07}_{\pm 0.02}$	$\mathbf{0.01}_{\pm 0.01}$	$0.07_{\pm 0.02}$	$\mathbf{0.04}_{\pm 0.02}$	$0.87_{\pm 0.33}$	$\mathbf{0.03}_{\pm 0.03}$	$0.31_{\pm 0.10}$	$\mathbf{0.02}_{\pm 0.02}$
+OAT-FM	$\mathbf{0.32}_{\pm 0.10}$	$0.04_{\pm 0.01}$	$\mathbf{0.07}_{\pm 0.01}$	$\mathbf{0.01}_{\pm 0.01}$	$\mathbf{0.06}_{\pm 0.01}$	$\mathbf{0.04}_{\pm 0.01}$	$\mathbf{0.83}_{\pm 0.34}$	$0.04_{\pm 0.02}$	$\mathbf{0.29}_{\pm 0.09}$	$0.10_{\pm 0.02}$

Application 2: Unconditional Image Generation (CIFAR-10)

Method	#Batch	NFE↓	FID↓
FM	400K	147	3.71
FM + OAT-FM	+1K	135	**3.54**
I-CFM	400K	149	3.67
I-CFM + OAT-FM	+1K	138	**3.48**
OT-CFM	400K	132	3.64
OT-CFM + OAT-FM	+1K	126	**3.46**
DDPM*		1K	3.17
Score SDE*		2K	2.38
LSGM*		147	2.10
2-ReFlow++*		35	2.30
EDM		35	1.96
EDM + OAT-FM	+12K	35	**1.93**

Lower-level Problem	Upper-level Problem	Phase-1 Method: FM	Phase-1 Method: EDM
Without Phase-2 Training		3.71	1.96
$\mathcal{W}_2^2$ in (94)	$\mathcal{L}_{\text{CFM}}$ in (94)	3.75	8.77
$\mathcal{W}_2^2$ in (94)	$\mathcal{L}_{\text{OAT}}$ in (101)	3.55	8.68
$\mathcal{A}_2^2$ in (101)	$\mathcal{L}_{\text{CFM}}$ in (94)	3.81	1.95
$\mathcal{A}_2^2$ in (101)	$\mathcal{L}_{\text{OAT}}$ in (101)	**3.54**	**1.93**



Application 3: Large-scale Conditional Image Generation

(a) SiT-XL (Left) v.s. + OAT-FM (Right)

(b) SiT-XL (Left) v.s. + OAT-FM (Right)

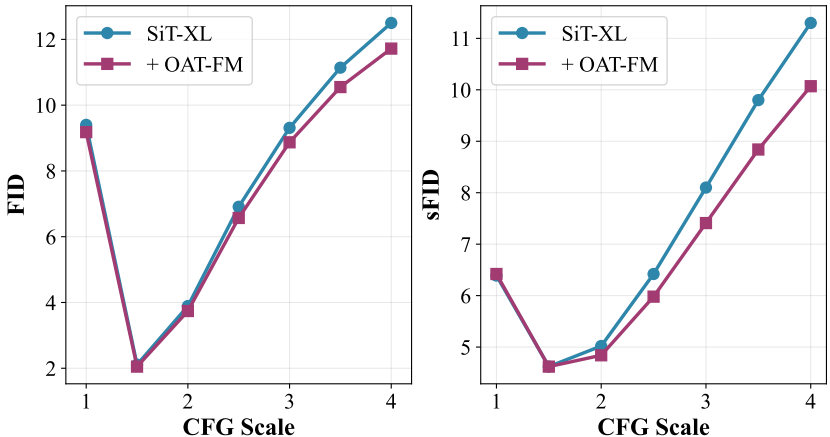
(c) SiT-XL (Left) v.s. + OAT-FM (Right)

(d) SiT-XL (Left) v.s. + OAT-FM (Right)

Application 3: Large-scale Conditional Image Generation

Method	#Epochs	FID↓	sFID↓	IS↑	P↑	R↑
BigGAN-deep		6.95	7.36	171.4	0.87	0.28
StyleGAN-XL		2.30	4.02	265.1	0.78	0.53
Mask-GIT		6.18	-	182.1	-	-
ADM-G/U		3.94	6.14	215.8	0.83	0.53
CDM		4.88	-	158.7	-	-
RIN		3.42	-	182.0	-	-
Simple Diffusion$_{\text{U-ViT, L}}$		2.77	-	211.8	-	-
VDM++		2.12	-	267.7	-	-
DiT-XL$_{\text{CFG=1.5}}$		2.27	4.60	278.2	0.83	0.57
SiT-XL$_{\text{CFG=1.5, Sampler=ODE}}$	1,400	2.11	**4.62**	256.0	**0.81**	**0.61**
SiT-XL$_{\text{CFG=1.5, Sampler=ODE}}$ + OAT-FM	+5	**2.05**	**4.62**	**259.4**	0.80	**0.61**
SiT-XL$_{\text{CFG=2.5, Sampler=ODE}}$	1,400	6.91	6.42	391.5	**0.89**	0.47
SiT-XL$_{\text{CFG=2.5, Sampler=ODE}}$ + OAT-FM	+5	**6.57**	**5.98**	**394.8**	**0.89**	**0.49**
SiT-XL$_{\text{CFG=1.5, Sampler=SDE}}$	1,400	2.05	4.50	269.6	**0.82**	**0.59**
SiT-XL$_{\text{CFG=1.5, Sampler=SDE}}$ + OAT-FM	+5	**2.00**	**4.43**	**275.1**	**0.82**	**0.59**
SiT-XL$_{\text{CFG=2.5, Sampler=SDE}}$	1,400	7.75	6.64	405.0	**0.90**	0.45
SiT-XL$_{\text{CFG=2.5, Sampler=SDE}}$ + OAT-FM	+5	**7.44**	**5.77**	**409.9**	**0.90**	**0.46**

Application 3: Large-scale Conditional Image Generation



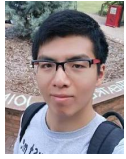
Summary

- OT-CFM shows the potential dynamic OT in generative modeling.
- Proposes OAT-FM to straighten flow trajectories by minimizing acceleration in the joint sample-velocity space
- Introduces an efficient two-phase fine-tuning paradigm that improves pre-trained models without distribution drift
- Achieves superior generation quality on high-dimensional tasks (e.g., CIFAR-10, ImageNet) with minimal training overhead
- *Paper:* `https://arxiv.org/pdf/2509.24936`
- *Code:* `https://github.com/AngxiaoYue/OAT-FM`

Acknowledgment

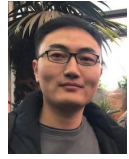
Computational Optimal Transport


Xiangfeng Wang
ECNU


Cheng Meng
RUC


Jun Yu
BIT


Anqi Dong
KTH


Tao Li
RUC

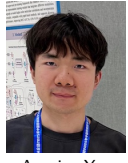

Mengyu Li
RUC


Moyi Yang
ECNU

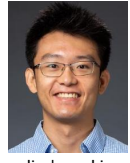
OT-based Machine Learning


Lawrence Carin
Duke

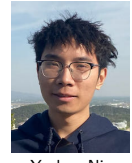

Hongyuan Zha
CUHK-SZ


Angxiao Yue
RUC


Fanmeng Wang
RUC


Jiachang Liu
Cornell


Fengjiao Gong
RUC


Yuzhou Nie
UCSB

Thank you!

https://hongtengxu.github.io

https://github.com/HongtengXu

hongtengxu@ruc.edu.cn

AAAI'22 Tutorial on Gromov-Wasserstein Learning
IJCAI'23 Tutorial on OT-based Machine Learning
AAAI'26 Tutorial on OT-based Machine Learning
https://hongtengxu.github.io/talks.html