Optimal Transport-driven Implicit Neural Network Design

Hongteng Xu

GSAI, RUC

Oct. 31, 2023

中國人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Outline

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, ...

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, ...

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

- The deeper, the larger, the better.
- Tricks: dropout, batchnorm, non-smooth activations, ...

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, ...

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

- The deeper, the larger, the better.
- Tricks: dropout, batchnorm, non-smooth activations, ...

Lead to a chaotic research field, with few artistic masterpieces (e.g., Transformer) + many imitations without originality.

Neural Network Design: Explicit v.s. Implicit

- **Explicit NN layer:**
  - Define and learn $f : \mathcal{X} \mapsto \mathcal{Z}$.
  - Deploy: $z = f(x)$

Neural Network Design: Explicit v.s. Implicit

- **Explicit NN layer:**
  - Define and learn $f : \mathcal{X} \mapsto \mathcal{Z}$.
  - Deploy: $z = f(x)$

- **Implicit NN layer:**
  - Define and learn $g : \mathcal{X} \times \mathcal{Z} \mapsto \mathbb{R}^n$
  - Deploy: Find z s.t. $g(x, z) = 0$

Neural Network Design: Explicit v.s. Implicit

- **Explicit NN layer:**
  - Define and learn $f : \mathcal{X} \mapsto \mathcal{Z}$.
  - Deploy: $z = f(x)$

- **Implicit NN layer:**
  - Define and learn $g : \mathcal{X} \times \mathcal{Z} \mapsto \mathbb{R}^n$
  - Deploy: Find z s.t. $g(x, z) = 0$

Implicit neural network design revisits neural networks from a more mathematical and interpretable perspective, e.g., **optimization**, differential equations, and dynamic systems.

- In general, g is more interpretable than f because its design (NOT learning!) is often determined by an optimization problem.

Optimization-based Implicit Neural Network Design

Layer	**Explicit Design**	**Implicit Design**
ReLU	$y = \max\{0, x\}$	$y = \arg\min_{z \geq 0} \lvert z - x \rvert^2$

Optimization-based Implicit Neural Network Design

Layer	**Explicit Design**	**Implicit Design**
ReLU	$y = \max\{0, x\}$	$y = \arg\min_{z \geq 0} \lvert z - x \rvert^2$
Linear+ReLU	$\boldsymbol{y} = \max\{\boldsymbol{0}, \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\}$	$\boldsymbol{y} = \arg\min_{\boldsymbol{z} \geq \boldsymbol{0}} \lVert \boldsymbol{z} - \boldsymbol{W}\boldsymbol{x} - \boldsymbol{b} \rVert^2$

Optimization-based Implicit Neural Network Design

Layer	**Explicit Design**	**Implicit Design**
ReLU	$y = \max\{0, x\}$	$y = \arg\min_{z \geq 0} \lvert z - x \rvert^2$
Linear+ReLU	$\boldsymbol{y} = \max\{\mathbf{0}, \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\}$	$\boldsymbol{y} = \arg\min_{\boldsymbol{z} \geq \mathbf{0}} \lVert \boldsymbol{z} - \boldsymbol{W}\boldsymbol{x} - \boldsymbol{b} \rVert^2$
Sigmoid	$y = \frac{1}{1+\exp(x)}$	$y = \arg\max_{z \in [0,1]} zx + z \log z + (1 - z) \log(1 - z)$

Optimization-based Implicit Neural Network Design

Layer	**Explicit Design**	**Implicit Design**
ReLU	$y = \max\{0, x\}$	$y = \arg\min_{z \geq 0} \lvert z - x \rvert^2$
Linear+ReLU	$\boldsymbol{y} = \max\{\boldsymbol{0}, \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\}$	$\boldsymbol{y} = \arg\min_{\boldsymbol{z} \geq \boldsymbol{0}} \lVert \boldsymbol{z} - \boldsymbol{W}\boldsymbol{x} - \boldsymbol{b} \rVert^2$
Sigmoid	$y = \frac{1}{1+\exp(x)}$	$y = \arg\max_{z \in [0,1]} zx + z \log z + (1-z)\log(1-z)$
Softmax	$y_i = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(x_j)}$	$\boldsymbol{y} = \arg\max_{\boldsymbol{z} \in \Delta^{N-1}} \langle \boldsymbol{z}, \boldsymbol{x} \rangle + \langle \boldsymbol{z}, \log \boldsymbol{z} \rangle$

Optimization-based Implicit Neural Network Design

Layer	Explicit Design	Implicit Design
ReLU	$y = \max\{0, x\}$	$y = \arg\min_{z \geq 0} \lvert z - x \rvert^2$
Linear+ReLU	$\boldsymbol{y} = \max\{\boldsymbol{0}, \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\}$	$\boldsymbol{y} = \arg\min_{\boldsymbol{z} \geq \boldsymbol{0}} \lVert \boldsymbol{z} - \boldsymbol{W}\boldsymbol{x} - \boldsymbol{b} \rVert^2$
Sigmoid	$y = \frac{1}{1+\exp(x)}$	$y = \arg\max_{z \in [0,1]} zx + z \log z + (1-z)\log(1-z)$
Softmax	$y_i = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(x_j)}$	$\boldsymbol{y} = \arg\max_{\boldsymbol{z} \in \Delta^{N-1}} \langle \boldsymbol{z}, \boldsymbol{x} \rangle + \langle \boldsymbol{z}, \log \boldsymbol{z} \rangle$

- Design a neural network layer $\Leftrightarrow$ Define an optimization problem + Select a solver
- Feed-forward computation $\Leftrightarrow$ Solve an optimization problem
- Backward propagation $\Leftrightarrow$ Adjust hyper-parameters in an optimization problem

Representative Implicit Neural Networks (Layers)

- Optimization-driven
  - Input Convex Neural Networks [1]
  - Deep Declarative Networks [2]
- Differential equation-driven
  - Neural ODEs [3]
  - Deep Equilibirum Models [4]

1 Amos, B., Xu, L., & Kolter, J. Z. Input convex neural networks. ICML, 2017.

2 Gould, S., Hartley, R., & Campbell, D. Deep declarative networks. IEEE TPAMI, 2021.

3 Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. Neural ordinary differential equations. NeurIPS, 2018.

4 Bai, S., Kolter, J. Z., & Koltun, V. Deep equilibrium models. NeurIPS, 2019.

5 https://github.com/bamos/thesis

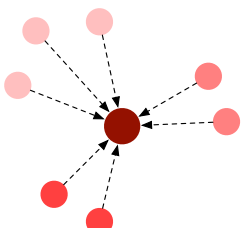
6 http://implicit-layers-tutorial.org

Motivation and Principle of Proposed Work

Essentially, many existing NN layers work for information fusion

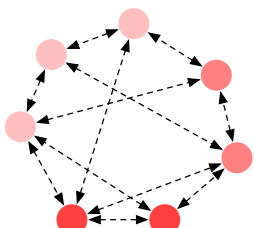
Global Pooling

$$\boldsymbol{y} = f_\theta(\boldsymbol{X})$$



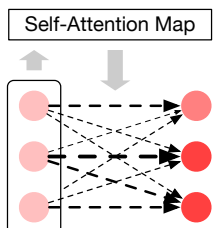
Message-Passing

$$\boldsymbol{Y} = \boldsymbol{A}_\theta \boldsymbol{X}$$



Self-Attention

$$\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{x}, \theta)\boldsymbol{X}$$

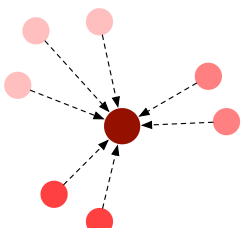


Motivation and Principle of Proposed Work

Essentially, many existing NN layers work for information fusion

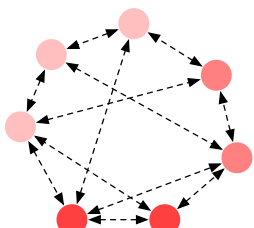
Global Pooling

$$\boldsymbol{y} = f_\theta(\boldsymbol{X})$$



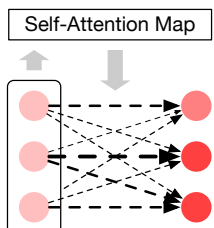
Message-Passing

$$\boldsymbol{Y} = \boldsymbol{A}_\theta \boldsymbol{X}$$



Self-Attention

$$\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{x}, \theta)\boldsymbol{X}$$



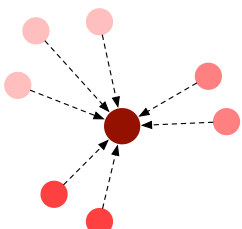
What is the design principle of the layers?

Motivation and Principle of Proposed Work

Essentially, many existing NN layers work for information fusion

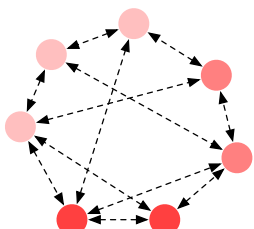
Global Pooling

$$\boldsymbol{y} = f_\theta(\boldsymbol{X})$$



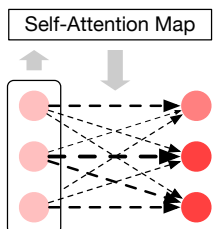
Message-Passing

$$\boldsymbol{Y} = \boldsymbol{A}_\theta \boldsymbol{X}$$



Self-Attention

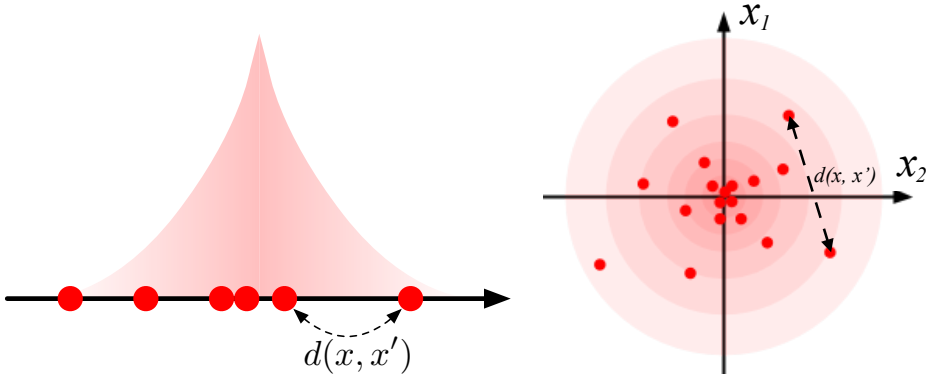
$$\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{x}, \theta)\boldsymbol{X}$$



What is the design principle of the layers?

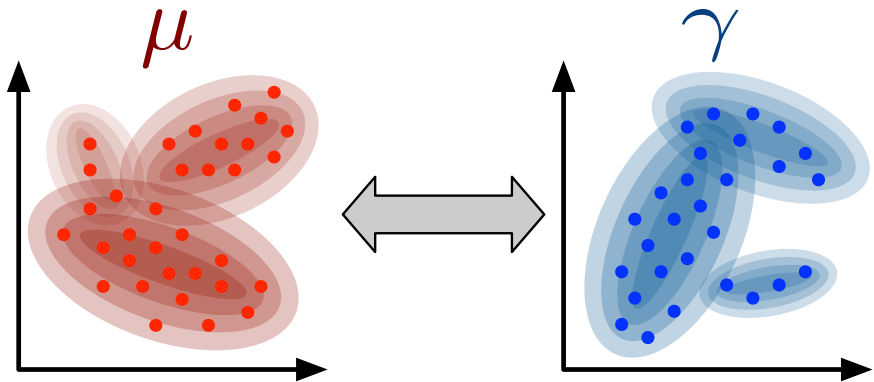
- Explore the **alignment** principle of information fusion through the lens of OT
- Develop OT-based implicit surrogates for above layers, improving interpretability and boosting performance

Preliminaries of OT: Metric-Measure Space



- $\mathcal{X}_{d,\mu} := (\mathcal{X}, d, \mu)$: A metric-measure space, where $x \in \mathcal{X}$ is a sample in the space.
- d: A distance metric of samples (e.g., Euclidean distance).
- $\mathbb{P}$: A space of (probability) measures defined on $\mathcal{X}$.
- $\mu \in \mathbb{P}$: a probability measure on $\mathcal{X}$.

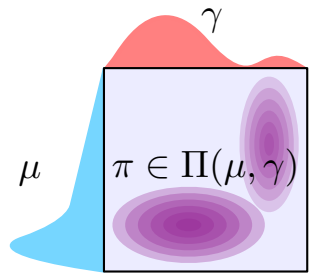
The Key ML Task: Distribution/Sample Matching



- Data Clustering, Domain Adaptation, Generative Modeling, Evaluation of Generative Model, ...

Distribution Matching Tool: The Kantorovich-form of OT

Leonid Kantorovich (1912-1986)

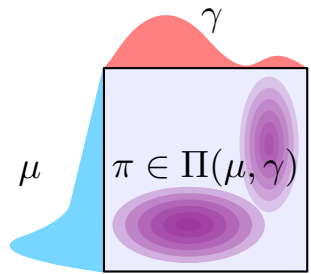


The Kantorovich-form of OT proposed in 1939

- $\Pi(\mu, \gamma) = \{\pi > 0 | \int_x \pi(x, y)dx = \gamma(y), \int_y \pi(x, y)dy = \mu(x)\}$ include all joint distributions taking μ and γ as marginals.
- $\pi \in \Pi(\mu, \gamma)$ is called **transport plan or coupling**.

Distribution Matching Tool: The Kantorovich-form of OT

Leonid Kantorovich (1912-1986)



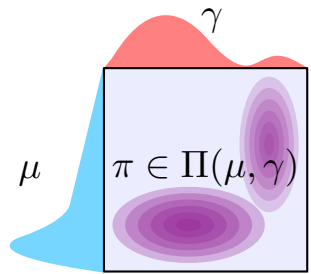
The Kantorovich-form of OT proposed in 1939

- $\Pi(\mu, \gamma) = \{\pi > 0 | \int_x \pi(x, y)dx = \gamma(y), \int_y \pi(x, y)dy = \mu(x)\}$ include all joint distributions taking μ and γ as marginals.
- $\pi \in \Pi(\mu, \gamma)$ is called **transport plan or coupling**.
- Find an **optimal** transport plan to minimize the expected cost.

$$W_p(\mu, \gamma) := \Big(\inf_{\pi \in \Pi(\mu,\gamma)} \int_{(x,y)\in\mathcal{X}^2} d^p(x, y) \mathrm{d}\pi(x, y) \Big)^{1/p}$$

Distribution Matching Tool: The Kantorovich-form of OT

Leonid Kantorovich (1912-1986)

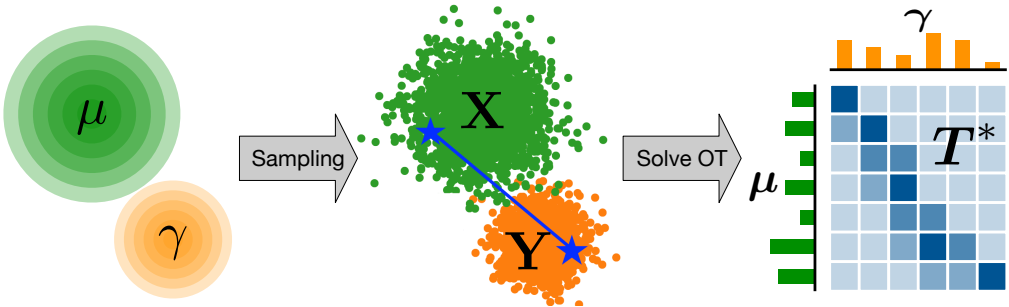


The Kantorovich-form of OT proposed in 1939

- $\Pi(\mu,\gamma) = \{\pi > 0 | \int_x \pi(x,y)dx = \gamma(y), \int_y \pi(x,y)dy = \mu(x)\}$ include all joint distributions taking μ and γ as marginals.
- $\pi \in \Pi(\mu,\gamma)$ is called **transport plan or coupling**.
- Find an **optimal** transport plan to minimize the expected cost.

$$W_p(\mu,\gamma) := \Big(\inf_{\pi\in\Pi(\mu,\gamma)} \int_{(x,y)\in\mathcal{X}^2} d^p(x,y)\mathrm{d}\pi(x,y)\Big)^{1/p} = \inf_{\pi\in\Pi(\mu,\gamma)} \mathbb{E}^{1/p}_{x,y\sim\pi}[d^p(x,y)]. \quad (1)$$

Empirical OT Problem Defined on Samples



Given $\boldsymbol{X} = \{x_m\}_{m=1}^M \sim \mu$, $\boldsymbol{Y} = \{y_n\}_{n=1}^N \sim \gamma$, $\boldsymbol{\mu} \in \Delta^{N-1}$ and $\boldsymbol{\gamma} \in \Delta^{M-1}$,

$$\widehat{W}_p(\boldsymbol{X}, \boldsymbol{Y}) := \Big(\min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \sum_{m=1}^{M} \sum_{n=1}^{N} d^p(x_m, y_n) t_{mn} \Big)^{1/p} = \Big(\min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \langle \boldsymbol{D}, \boldsymbol{T} \rangle \Big)^{1/p} \quad (2)$$

where $\boldsymbol{D} = [d^p(x_m, y_n)]$, $\boldsymbol{T} = [t_{mn}]$, $\Pi(\boldsymbol{\mu}, \boldsymbol{\gamma}) = \{\boldsymbol{T} > \mathbf{0} | \boldsymbol{T}\mathbf{1}_M = \boldsymbol{\mu}, \boldsymbol{T}^\top \mathbf{1}_N = \boldsymbol{\gamma}\}$.

Wasserstein Distance between 1D Samples

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

$$W_p(\mu, \gamma) = \left(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p \mathrm{d}z \right)^{1/p}, \tag{3}$$

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

Wasserstein Distance between 1D Samples

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

$$W_p(\mu, \gamma) = \Big(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p \mathrm{d}z \Big)^{1/p}, \tag{3}$$

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

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

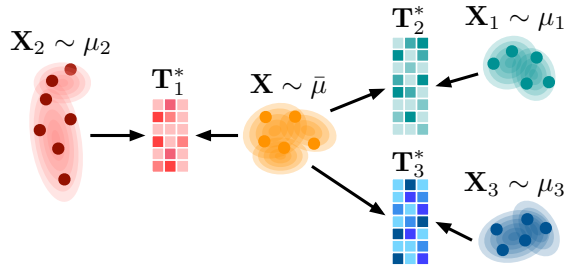
$$\widehat{W}_p(\boldsymbol{x}, \boldsymbol{y}) = \Big(\sum_{n=1}^N |x_n - y_{\sigma(n)}|^p \mathrm{d}z \Big)^{1/p}, \tag{4}$$

  where σ denotes the **sorting** operation.

- OT plan $\boldsymbol{T}^*$ is the permutation matrix corresponding to σ.

Theorem: For one dimensional $x_1 \leq ... \leq x_N$ and $y_1 \leq ... \leq y_N$, identity permutation ($\sigma(n) = n$ for $n = 1, ..., N$) leads to the optimal transport between them.

Wasserstein Barycenters



- Denote $\mathcal{P}_{\mathcal{X}_d}$ as the space of all probability measures in the metric space $\mathcal{X}_d$.
- $(\mathcal{P}_{\mathcal{X}_d}, W_p)$ becomes a metric space of probability measures.
- Given a set of probability measures $\{\mu_k\}_{k=1}^K \subset \mathcal{P}_{\mathcal{X}_d}$, we can define the p-**Wasserstein barycenter** [Agueh et al, 2021] as

$$\bar{\mu} := \arg\min_{\mu \in \mathcal{P}_{\mathcal{X}_d}} \sum_{k=1}^{K} W_p^p(\mu, \mu_k). \tag{5}$$

[Agueh et al, 2021] Agueh, M. and Carlier, G., Barycenters in the Wasserstein space. SIAM Journal on Mathematical Analysis, 2011.

Advantages of Optimal Transport

A valid metric for probability measures

- Apply to distribution comparison and fitting

Advantages of Optimal Transport

A valid metric for probability measures

- Apply to distribution comparison and fitting

The OT plan/matrix indicates the coherency of sample pairs

- Apply to point cloud matching and registration
- Achieve sample averaging and fusion

Advantages of Optimal Transport

A valid metric for probability measures

- Apply to distribution comparison and fitting

The OT plan/matrix indicates the coherency of sample pairs

- Apply to point cloud matching and registration
- Achieve sample averaging and fusion

Have potentials for designing implicit neural network and achieving interpretable information fusion.

Case 1: OT-based Implicit Global Pooling Design

Given $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, find a global pooling $f : \mathcal{X} \mapsto \mathbb{R}^D$.

Case 1: OT-based Implicit Global Pooling Design

Given $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, find a global pooling $f : \mathcal{X} \mapsto \mathbb{R}^D$.

- Mean-pooling

$$f(\boldsymbol{X}) = \frac{1}{N} \sum\nolimits_{n=1}^{N} \boldsymbol{x}_n \tag{6}$$

- Max-pooling

$$f(\boldsymbol{X}) = \|_{d=1}^{D} \max\nolimits_n \{x_{dn}\}_{n=1}^{N} \tag{7}$$

- Attention-pooling

$$f(\boldsymbol{X}) = \boldsymbol{X} \boldsymbol{a}_{\boldsymbol{X}} \tag{8}$$

Case 1: OT-based Implicit Global Pooling Design

Given $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, find a global pooling $f : \mathcal{X} \mapsto \mathbb{R}^D$.

- Mean-pooling

$$f(\boldsymbol{X}) = \frac{1}{N} \sum\nolimits_{n=1}^{N} \boldsymbol{x}_n \tag{6}$$

- Max-pooling

$$f(\boldsymbol{X}) = \|_{d=1}^{D} \max_n \{x_{dn}\}_{n=1}^{N} \tag{7}$$

- Attention-pooling

$$f(\boldsymbol{X}) = \boldsymbol{X} \boldsymbol{a}_{\boldsymbol{X}} \tag{8}$$

- The design and selection of f is **empirical and sub-optimal**.
- Can we build a generalized framework covering the above designs?

Case 1: OT-based Implicit Global Pooling Design

Determine the poolings by a "sample-feature dimension" distribution $\boldsymbol{P} = [p_{dn}]$.

Case 1: OT-based Implicit Global Pooling Design

Determine the poolings by a "sample-feature dimension" distribution $\boldsymbol{P} = [p_{dn}]$.

- A global pooling outputs the expectation of samples conditioned on different feature dimensions

$$f(\boldsymbol{X}) = \|_{d=1}^{D} \mathbb{E}_{n \sim p_{n|d}}[x_{dn}]$$

Case 1: OT-based Implicit Global Pooling Design

Determine the poolings by a "sample-feature dimension" distribution $\boldsymbol{P} = [p_{dn}]$.

- A global pooling outputs the expectation of samples conditioned on different feature dimensions

$$f(\boldsymbol{X}) = \|_{d=1}^{D} \mathbb{E}_{n \sim p_{n|d}}[x_{dn}] = (\boldsymbol{X} \odot \underbrace{\mathsf{diag}^{-1}(\overbrace{\boldsymbol{P}\mathbf{1}_N}^{\boldsymbol{p}=[p_d]})\boldsymbol{P}}_{\tilde{\boldsymbol{P}}=[p_{n|d}]})\mathbf{1}_N. \tag{9}$$

Case 1: OT-based Implicit Global Pooling Design

Determine the poolings by a "sample-feature dimension" distribution $\boldsymbol{P} = [p_{dn}]$.

- A global pooling outputs the expectation of samples conditioned on different feature dimensions

$$f(\boldsymbol{X}) = \|_{d=1}^{D} \mathbb{E}_{n\sim p_{n|d}}[x_{dn}] = (\boldsymbol{X} \odot \underbrace{\text{diag}^{-1}(\overbrace{\boldsymbol{P}\mathbf{1}_N}^{\boldsymbol{p}=[p_d]})\boldsymbol{P}}_{\tilde{\boldsymbol{P}}=[p_{n|d}]})\mathbf{1}_N. \tag{9}$$

Mean-pooling	$\frac{1}{N}\sum_{n=1}\boldsymbol{x}_n$	$\Leftrightarrow$	$\boldsymbol{P} = [\frac{1}{DN}]$
Max-pooling	$\|_{d=1}^{D}\max_n\{x_{dn}\}_{n=1}^{N}$	$\Leftrightarrow$	$\boldsymbol{P} \in \{0, \frac{1}{D}\}^{D\times N}$ and $\boldsymbol{P}\mathbf{1}_N = \frac{1}{D}\mathbf{1}_D$
Attention-pooling	$\boldsymbol{X}\boldsymbol{a}_{\boldsymbol{X}}$	$\Leftrightarrow$	$\boldsymbol{P} = \frac{1}{D}\mathbf{1}_D\boldsymbol{a}_{\boldsymbol{X}}^{\top}$

Case 1: OT-based Implicit Global Pooling Design

Determine the poolings by a "sample-feature dimension" distribution $\boldsymbol{P} = [p_{dn}]$.

- A global pooling outputs the expectation of samples conditioned on different feature dimensions

$$f(\boldsymbol{X}) = \|_{d=1}^{D} \mathbb{E}_{n\sim p_{n|d}}[x_{dn}] = (\boldsymbol{X} \odot \underbrace{\mathrm{diag}^{-1}(\overbrace{\boldsymbol{P}\mathbf{1}_N}^{\boldsymbol{p}=[p_d]})\boldsymbol{P}}_{\tilde{\boldsymbol{P}}=[p_{n|d}]})\mathbf{1}_N. \tag{9}$$

Mean-pooling	$\frac{1}{N}\sum_{n=1}\boldsymbol{x}_n$	$\Leftrightarrow$	$\boldsymbol{P} = [\frac{1}{DN}]$
Max-pooling	$\|_{d=1}^{D}\max_n\{x_{dn}\}_{n=1}^{N}$	$\Leftrightarrow$	$\boldsymbol{P} \in \{0, \frac{1}{D}\}^{D\times N}$ and $\boldsymbol{P}\mathbf{1}_N = \frac{1}{D}\mathbf{1}_D$
Attention-pooling	$\boldsymbol{X}\boldsymbol{a}_{\boldsymbol{X}}$	$\Leftrightarrow$	$\boldsymbol{P} = \frac{1}{D}\mathbf{1}_D\boldsymbol{a}_{\boldsymbol{X}}^{\top}$

Design global pooling is equivalent to design $\boldsymbol{P}$, leading to an OT-based implicit layer.

Regularized Optimal Transport Pooling (ROTP) Layer

$$f_{\text{rot}}(\boldsymbol{X};\boldsymbol{\theta}) = (\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}})\mathbf{1}_N, \tag{10}$$

where

$$\boldsymbol{P}^*_{\text{rot}} = \arg\max_{\boldsymbol{P}\in\Omega} \overbrace{\underbrace{\langle \boldsymbol{X}, \boldsymbol{P}\rangle}_{\text{OT term}} + \underbrace{\alpha_0\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle}_{\text{Structural Reg.}}}^{\text{Fused Gromov-Wasserstein}} - \underbrace{\alpha_1 \text{R}(\boldsymbol{P})}_{\text{Smoothness}} - \underbrace{\alpha_2 \text{KL}(\boldsymbol{P}\mathbf{1}|\boldsymbol{p}_0) - \alpha_3 \text{KL}(\boldsymbol{P}^\top\mathbf{1}|\boldsymbol{q}_0)}_{\text{Marginal Prior}}$$

Regularized Optimal Transport Pooling (ROTP) Layer

$$f_{\text{rot}}(\boldsymbol{X};\boldsymbol{\theta}) = (\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}})\mathbf{1}_N, \tag{10}$$

where

$$\boldsymbol{P}^*_{\text{rot}} = \arg\max_{\boldsymbol{P}\in\Omega} \overbrace{\underbrace{\langle \boldsymbol{X}, \boldsymbol{P}\rangle}_{\text{OT term}} + \underbrace{\alpha_0\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle}_{\text{Structural Reg.}}}^{\text{Fused Gromov-Wasserstein}} - \underbrace{\alpha_1 \text{R}(\boldsymbol{P})}_{\text{Smoothness}} - \underbrace{\alpha_2 \text{KL}(\boldsymbol{P}\mathbf{1}|\boldsymbol{p}_0) - \alpha_3 \text{KL}(\boldsymbol{P}^\top\mathbf{1}|\boldsymbol{q}_0)}_{\text{Marginal Prior}}$$

Implementation	**Principle**	**Connections to OT**
$\max\langle \boldsymbol{X}, \boldsymbol{P}\rangle$	EM Principle (Max Output Energy)	OT loss

Regularized Optimal Transport Pooling (ROTP) Layer

$$f_{\text{rot}}(\boldsymbol{X};\boldsymbol{\theta}) = (\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}})\mathbf{1}_N, \tag{10}$$

where

$$\boldsymbol{P}^*_{\text{rot}} = \arg\max_{\boldsymbol{P}\in\Omega} \overbrace{\underbrace{\langle \boldsymbol{X}, \boldsymbol{P}\rangle}_{\text{OT term}} + \underbrace{\alpha_0\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle}_{\text{Structural Reg.}}}^{\text{Fused Gromov-Wasserstein}} - \underbrace{\alpha_1 \text{R}(\boldsymbol{P})}_{\text{Smoothness}} - \underbrace{\alpha_2 \text{KL}(\boldsymbol{P}\mathbf{1}|\boldsymbol{p}_0) - \alpha_3 \text{KL}(\boldsymbol{P}^\top\mathbf{1}|\boldsymbol{q}_0)}_{\text{Marginal Prior}}$$

Implementation	**Principle**	**Connections to OT**
$\max\langle \boldsymbol{X}, \boldsymbol{P}\rangle$	EM Principle (Max Output Energy)	OT loss
$+\max\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle$	Max Sample-Feature Correlation	FGW loss

Regularized Optimal Transport Pooling (ROTP) Layer

$$f_{\text{rot}}(\boldsymbol{X};\boldsymbol{\theta}) = (\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}})\mathbf{1}_N, \tag{10}$$

where

$$\boldsymbol{P}^*_{\text{rot}} = \arg\max_{\boldsymbol{P}\in\Omega} \overbrace{\underbrace{\langle \boldsymbol{X}, \boldsymbol{P}\rangle}_{\text{OT term}} + \underbrace{\alpha_0\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle}_{\text{Structural Reg.}}}^{\text{Fused Gromov-Wasserstein}} - \underbrace{\alpha_1 \text{R}(\boldsymbol{P})}_{\text{Smoothness}} - \underbrace{\alpha_2 \text{KL}(\boldsymbol{P}\mathbf{1}|\boldsymbol{p}_0) - \alpha_3 \text{KL}(\boldsymbol{P}^\top\mathbf{1}|\boldsymbol{q}_0)}_{\text{Marginal Prior}}$$

Implementation	**Principle**	**Connections to OT**
$\max\langle \boldsymbol{X}, \boldsymbol{P}\rangle$	EM Principle (Max Output Energy)	OT loss
$+\max\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle$	Max Sample-Feature Correlation	FGW loss
$+\min \text{R}(\boldsymbol{P})$	Avoid Over-sparse Distributions	Entropic/Quadratic FGW

Regularized Optimal Transport Pooling (ROTP) Layer

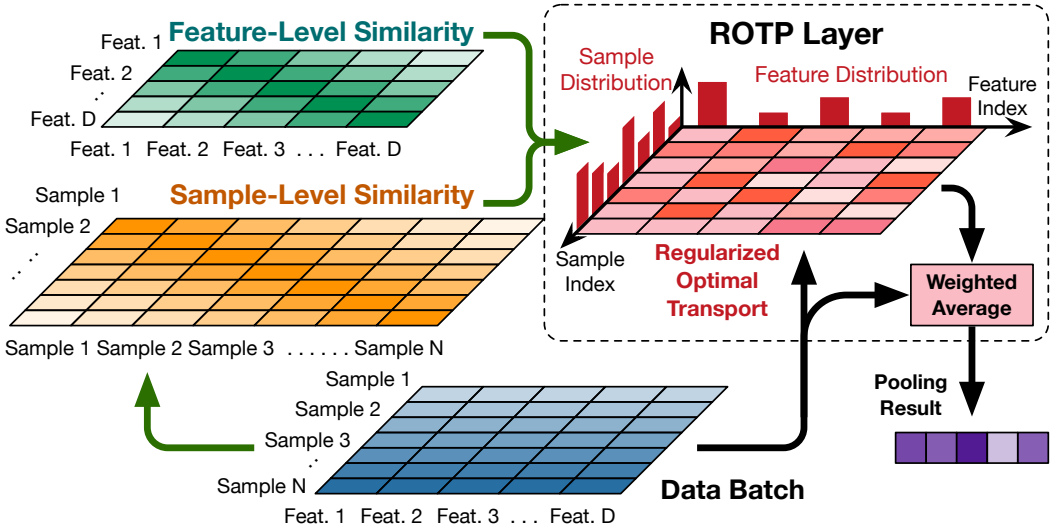
$$f_{\text{rot}}(\boldsymbol{X};\boldsymbol{\theta}) = (\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}})\mathbf{1}_N, \tag{10}$$

where

$$\boldsymbol{P}^*_{\text{rot}} = \arg\max_{\boldsymbol{P}\in\Omega} \overbrace{\underbrace{\langle \boldsymbol{X}, \boldsymbol{P}\rangle}_{\text{OT term}} + \underbrace{\alpha_0\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle}_{\text{Structural Reg.}}}^{\text{Fused Gromov-Wasserstein}} - \underbrace{\alpha_1 \text{R}(\boldsymbol{P})}_{\text{Smoothness}} - \underbrace{\alpha_2 \text{KL}(\boldsymbol{P}\mathbf{1}|\boldsymbol{p}_0) - \alpha_3 \text{KL}(\boldsymbol{P}^\top\mathbf{1}|\boldsymbol{q}_0)}_{\text{Marginal Prior}}$$

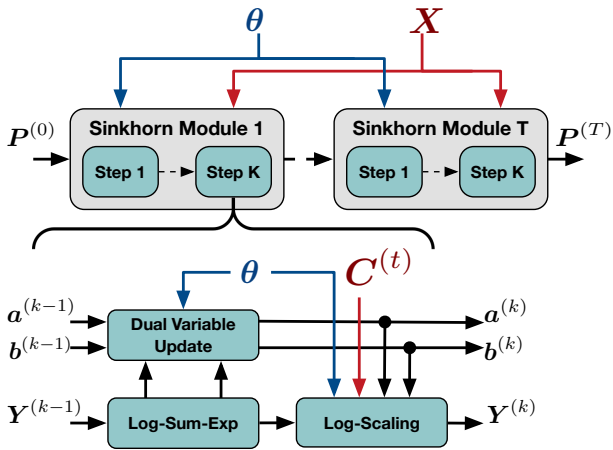
Implementation	**Principle**	**Connections to OT**
$\max\langle \boldsymbol{X}, \boldsymbol{P}\rangle$	EM Principle (Max Output Energy)	OT loss
$+\max\langle \boldsymbol{C}_D\boldsymbol{P}\boldsymbol{C}_N, \boldsymbol{P}\rangle$	Max Sample-Feature Correlation	FGW loss
$+\min \text{R}(\boldsymbol{P})$	Avoid Over-sparse Distributions	Entropic/Quadratic FGW
$+\min \text{KL}(\boldsymbol{P}\mathbf{1}\|\boldsymbol{p}_0)$ $+\min \text{KL}(\boldsymbol{P}^\top\mathbf{1}\|\boldsymbol{q}_0)$	Leverage Prior Knowledge	Unbalanced E/Q FGW

Regularized Optimal Transport Pooling (ROTP) Layer



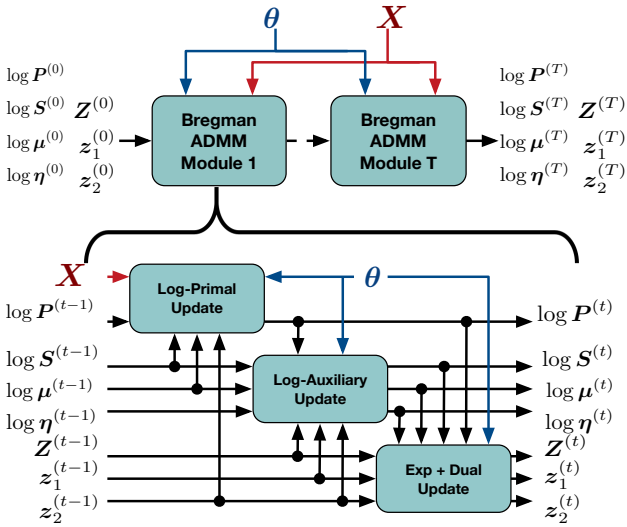
Regularized Optimal Transport Pooling (ROTP) Layer

Implement ROTP by **Unrolling Iterative Algorithms (e.g., Sinkhorn-scaling)**



Regularized Optimal Transport Pooling (ROTP) Layer

Implement ROTP by **Unrolling Iterative Algorithms (e.g., Bregman ADMM)**

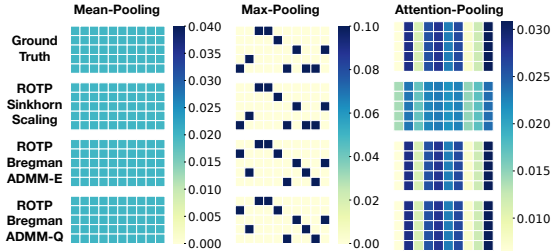


ROTP: Generalizability

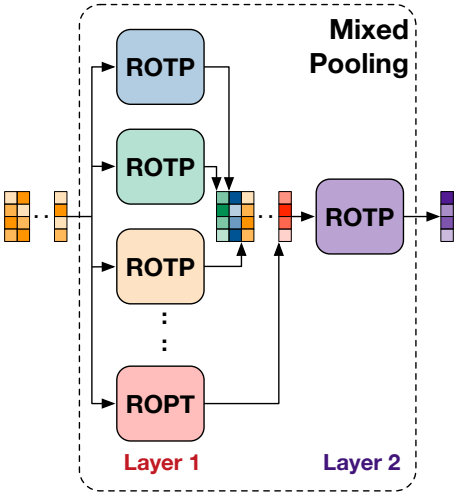
$$\boldsymbol{P}^*_{\text{rot}} = \arg\max_{\boldsymbol{P}\in\Omega} \overbrace{\langle \boldsymbol{X}, \boldsymbol{P}\rangle}^{\text{OT term}} + \overbrace{\alpha_0 \langle \boldsymbol{C}_D \boldsymbol{P} \boldsymbol{C}_N, \boldsymbol{P}\rangle}^{\text{Structural Reg.}} - \underbrace{\alpha_1 \mathsf{R}(\boldsymbol{P})}_{\text{Smoothness}} - \underbrace{\alpha_2 \mathsf{KL}(\boldsymbol{P}\mathbf{1}|\boldsymbol{p}_0) - \alpha_3 \mathsf{KL}(\boldsymbol{P}^\top \mathbf{1}|\boldsymbol{q}_0)}_{\text{Marginal Prior}}$$

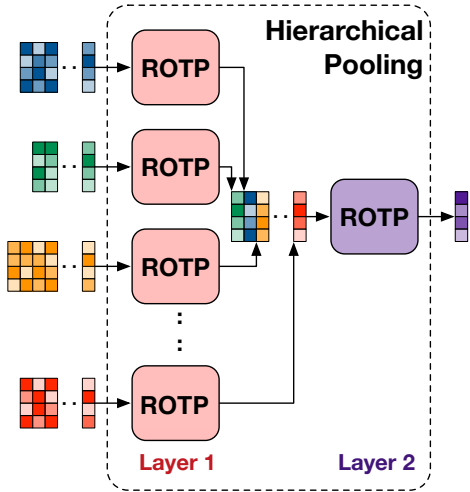
Fused Gromov-Wasserstein

$f_{\text{rot}}(\boldsymbol{X};\boldsymbol{\theta})$	α_0	α_1	α_2	α_3	$\boldsymbol{p}_0$	$\boldsymbol{q}_0$
Mean-pooling	0	$\to\infty$	$\to\infty$	$\to\infty$	$\frac{1}{D}\mathbf{1}_D$	$\frac{1}{N}\mathbf{1}_N$
Max-pooling	0	0	$\to\infty$	0	$\frac{1}{D}\mathbf{1}_D$	—
Attention-pooling	0	$\to\infty$	$\to\infty$	$\to\infty$	$\frac{1}{D}\mathbf{1}_D$	$\boldsymbol{a}_{\boldsymbol{X}}$

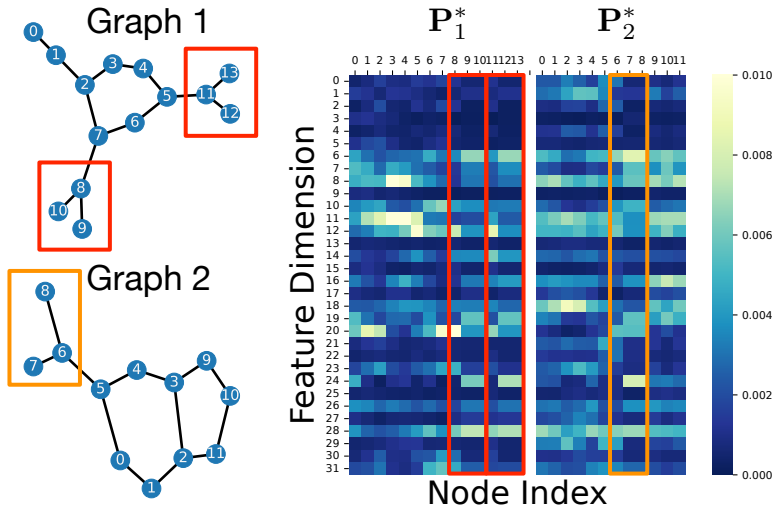


ROTP: Flexibility

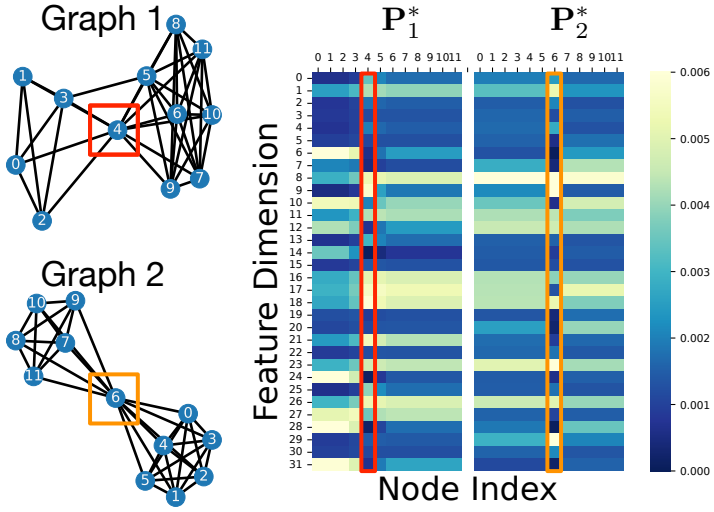




ROTP: Interpretablity



ROTP: Interpretablity



ROTP: Superiority on Data Fitting

Dataset	NCII	PROTEINS	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B	IMDB-M
#Graphs	4,110	1,113	188	5,000	2,000	4,999	1,000	1,500
Average #Nodes	29.87	39.06	17.93	74.49	429.63	508.52	19.77	13.00
Average #Edges	32.30	72.82	19.79	2,457.78	497.75	594.87	96.53	65.94
#Classes	2	2	2	3	2	5	2	3
Add	$67.96_{\pm 0.43}$	$72.97_{\pm 0.54}$	**$89.05_{\pm 0.86}$**	$71.06_{\pm 0.43}$	$80.00_{\pm 1.49}$	$50.16_{\pm 0.97}$	$70.18_{\pm 0.87}$	$47.56_{\pm 0.56}$
Mean	$64.82_{\pm 0.52}$	$66.09_{\pm 0.64}$	$86.53_{\pm 1.62}$	$72.35_{\pm 0.44}$	$83.62_{\pm 1.18}$	$52.44_{\pm 1.24}$	$70.34_{\pm 0.38}$	$48.65_{\pm 0.91}$
Max	$65.95_{\pm 0.76}$	$72.27_{\pm 0.33}$	$85.90_{\pm 1.68}$	$73.07_{\pm 0.57}$	$82.62_{\pm 1.25}$	$44.34_{\pm 1.93}$	$70.24_{\pm 0.54}$	$47.80_{\pm 0.54}$
DeepSet	$66.28_{\pm 0.72}$	**$73.76_{\pm 0.47}$**	$87.84_{\pm 0.71}$	$69.74_{\pm 0.66}$	$82.91_{\pm 1.37}$	$47.45_{\pm 0.54}$	$70.84_{\pm 0.71}$	$48.05_{\pm 0.71}$
Mixed	$66.46_{\pm 0.74}$	$72.25_{\pm 0.45}$	$87.30_{\pm 0.87}$	$73.22_{\pm 0.35}$	$84.36_{\pm 2.62}$	$46.67_{\pm 1.63}$	$71.28_{\pm 0.26}$	$48.07_{\pm 0.25}$
GatedMixed	$63.86_{\pm 0.76}$	$69.40_{\pm 1.93}$	$87.94_{\pm 1.28}$	$71.94_{\pm 0.40}$	$80.60_{\pm 3.89}$	$44.78_{\pm 4.53}$	$70.96_{\pm 0.60}$	$48.09_{\pm 0.44}$
Set2Set	$65.10_{\pm 1.12}$	$68.61_{\pm 1.44}$	$87.77_{\pm 0.86}$	$72.31_{\pm 0.73}$	$80.08_{\pm 5.72}$	$49.85_{\pm 2.77}$	$70.36_{\pm 0.85}$	$48.30_{\pm 0.54}$
Attention	$64.35_{\pm 0.61}$	$67.70_{\pm 0.95}$	$88.08_{\pm 1.22}$	$72.57_{\pm 0.41}$	$81.55_{\pm 4.39}$	$51.85_{\pm 0.66}$	$70.60_{\pm 0.38}$	$47.83_{\pm 0.78}$
GatedAtt	$64.66_{\pm 0.52}$	$68.16_{\pm 0.90}$	$86.91_{\pm 1.79}$	$72.31_{\pm 0.37}$	$82.55_{\pm 1.96}$	$51.47_{\pm 0.82}$	$70.52_{\pm 0.31}$	$48.67_{\pm 0.35}$
DynamicP	$62.11_{\pm 0.27}$	$65.86_{\pm 0.85}$	$85.40_{\pm 2.81}$	$70.78_{\pm 0.88}$	$67.51_{\pm 1.82}$	$32.11_{\pm 3.85}$	$69.84_{\pm 0.73}$	$47.59_{\pm 0.48}$
GNP	**$68.20_{\pm 0.48}$**	**$73.44_{\pm 0.61}$**	$88.37_{\pm 1.25}$	$72.80_{\pm 0.58}$	$81.93_{\pm 2.23}$	$51.80_{\pm 0.61}$	$70.34_{\pm 0.83}$	$48.85_{\pm 0.81}$
ASAP	$68.09_{\pm 0.42}$	$70.42_{\pm 1.45}$	$87.68_{\pm 1.42}$	$68.20_{\pm 2.37}$	$73.91_{\pm 1.50}$	$44.58_{\pm 0.44}$	$68.33_{\pm 2.50}$	$43.92_{\pm 1.13}$
SAGP	$67.48_{\pm 0.65}$	$72.63_{\pm 0.44}$	$87.88_{\pm 2.22}$	$70.19_{\pm 0.55}$	$74.12_{\pm 2.86}$	$46.00_{\pm 1.74}$	$70.34_{\pm 0.74}$	$47.04_{\pm 1.22}$
OTK	$67.96_{\pm 0.55}$	$69.52_{\pm 0.76}$	$86.90_{\pm 1.83}$	$71.35_{\pm 0.91}$	$74.28_{\pm 1.39}$	$50.57_{\pm 1.20}$	$70.94_{\pm 0.79}$	$48.41_{\pm 0.89}$
SWE	$68.06_{\pm 0.98}$	$70.09_{\pm 1.22}$	$85.68_{\pm 2.07}$	$72.17_{\pm 1.29}$	$79.30_{\pm 3.94}$	$51.11_{\pm 1.55}$	$70.34_{\pm 1.05}$	$48.93_{\pm 1.34}$
WEGL	**$68.16_{\pm 0.62}$**	$71.58_{\pm 0.94}$	**$88.68_{\pm 1.66}$**	$72.55_{\pm 0.69}$	$82.80_{\pm 1.73}$	$52.03_{\pm 0.60}$	**$71.94_{\pm 0.75}$**	$49.20_{\pm 0.87}$
ROTP$_{\text{S}}$	**$68.27_{\pm 1.06}$**	**$73.10_{\pm 0.22}$**	**$88.84_{\pm 1.21}$**	$71.20_{\pm 0.55}$	$81.54_{\pm 1.38}$	$51.00_{\pm 0.61}$	$70.74_{\pm 0.80}$	$47.87_{\pm 0.43}$
ROTP$_{\text{B-E}}$ ($\alpha_0 = 0$)	$66.23_{\pm 0.50}$	$67.71_{\pm 1.70}$	$86.82_{\pm 2.02}$	$73.86_{\pm 0.44}$	**$86.80_{\pm 1.19}$**	$52.25_{\pm 0.75}$	$71.72_{\pm 0.88}$	**$50.48_{\pm 0.14}$**
ROTP$_{\text{B-Q}}$ ($\alpha_0 = 0$)	$66.18_{\pm 0.76}$	$69.88_{\pm 0.87}$	$85.42_{\pm 1.10}$	**$74.14_{\pm 0.24}$**	**$87.72_{\pm 1.03}$**	**$52.79_{\pm 0.60}$**	**$72.34_{\pm 0.50}$**	$49.36_{\pm 0.52}$
ROTP$_{\text{B-E}}$ (learn α_0)	$65.90_{\pm 0.94}$	$70.19_{\pm 0.66}$	$88.01_{\pm 1.51}$	**$74.05_{\pm 0.34}$**	$86.78_{\pm 1.14}$	**$52.77_{\pm 0.69}$**	$71.76_{\pm 0.62}$	**$50.28_{\pm 0.86}$**
ROTP$_{\text{B-Q}}$ (learn α_0)	$65.96_{\pm 0.32}$	$70.12_{\pm 1.17}$	$86.79_{\pm 1.81}$	**$74.27_{\pm 0.47}$**	**$88.67_{\pm 0.99}$**	**$52.84_{\pm 0.60}$**	**$71.78_{\pm 1.00}$**	**$49.44_{\pm 0.46}$**

ROTP: Superiority on Data Fitting

Dataset	Messidor	Component	Function
D	687	200	200
#Positive bags	654	423	443
#Negative bags	546	2,707	4,799
#Instances	12,352	36,894	55,536
Min. bag size	8	1	1
Max. bag size	12	53	51
Add	$74.33_{\pm 2.56}$	$93.35_{\pm 0.98}$	$96.26_{\pm 0.48}$
Mean	$74.42_{\pm 2.47}$	$93.32_{\pm 0.99}$	$96.28_{\pm 0.66}$
Max	$73.92_{\pm 3.00}$	$93.23_{\pm 0.76}$	$95.94_{\pm 0.48}$
DeepSet	$74.42_{\pm 2.87}$	$93.29_{\pm 0.95}$	$96.45_{\pm 0.51}$
Mixed	$73.42_{\pm 2.29}$	**$93.45_{\pm 0.61}$**	$96.41_{\pm 0.53}$
GatedMixed	$73.25_{\pm 2.38}$	$93.03_{\pm 1.02}$	$96.22_{\pm 0.65}$
Set2Set	$73.58_{\pm 3.74}$	$93.19_{\pm 0.95}$	$96.43_{\pm 0.56}$
Attention	$74.25_{\pm 3.67}$	$93.22_{\pm 1.02}$	$96.31_{\pm 0.66}$
GatedAtt	$73.67_{\pm 2.23}$	**$93.42_{\pm 0.91}$**	**$96.51_{\pm 0.77}$**
DynamicP	$73.16_{\pm 2.12}$	$93.26_{\pm 1.30}$	**$96.47_{\pm 0.58}$**
GNP	$73.54_{\pm 3.68}$	$92.86_{\pm 1.96}$	$96.10_{\pm 1.03}$
OTK	$74.78_{\pm 2.89}$	$93.19_{\pm 0.93}$	$96.31_{\pm 1.02}$
SWE	$74.46_{\pm 3.72}$	$93.32_{\pm 1.26}$	$96.42_{\pm 0.88}$
ROTP$_{\text{S}}$	**$75.42_{\pm 2.96}$**	$93.29_{\pm 0.83}$	**$96.62_{\pm 0.48}$**
ROTP$_{\text{B-E}}$ ($\alpha_0 = 0$)	$74.83_{\pm 2.07}$	$93.16_{\pm 1.02}$	$96.17_{\pm 0.43}$
ROTP$_{\text{B-Q}}$ ($\alpha_0 = 0$)	$75.08_{\pm 2.06}$	$93.13_{\pm 0.94}$	$96.09_{\pm 0.46}$
ROTP$_{\text{B-E}}$ (learn α_0)	**$75.33_{\pm 1.96}$**	$93.16_{\pm 1.08}$	$96.22_{\pm 0.44}$
ROTP$_{\text{B-Q}}$ (learn α_0)	**$75.17_{\pm 2.45}$**	**$93.45_{\pm 0.96}$**	$96.22_{\pm 0.48}$

Dataset	DECAGON DiBr-APND	DECAGON Anae-Fati	DECAGON PleuP-Diar	FEARS
#Graph sets	6,309	2,922	2,842	6,338
#Positive sets	3,189	1,526	1,422	3,169
Positive label	Difficulty breathing	Anaemia	Pleural pain	Non-myopathy
#Negative sets	3,120	1,396	1,420	3,169
Negative label	Pressure decreased	Fatigue	Diarrhea	Myopathy
Set size	2	2	2	$2\sim52$
Add	$50.86_{\pm 0.97}$	$63.15_{\pm 1.79}$	**$62.32_{\pm 1.08}$**	$75.89_{\pm 1.33}$
Mean	$51.10_{\pm 1.09}$	$61.95_{\pm 2.60}$	$61.30_{\pm 2.68}$	$72.42_{\pm 1.51}$
Max	$50.59_{\pm 0.77}$	$61.88_{\pm 2.03}$	$60.11_{\pm 2.03}$	**$82.02_{\pm 0.72}$**
DeepSet	$49.83_{\pm 1.07}$	$56.24_{\pm 5.20}$	$51.78_{\pm 3.10}$	**$82.40_{\pm 1.56}$**
Mixed	$51.13_{\pm 0.99}$	**$63.83_{\pm 1.19}$**	$60.91_{\pm 2.12}$	$81.54_{\pm 1.13}$
GatedMixed	$51.39_{\pm 0.63}$	$61.50_{\pm 1.61}$	$59.12_{\pm 2.12}$	$81.88_{\pm 1.14}$
Set2Set	$50.72_{\pm 1.71}$	$59.35_{\pm 2.04}$	$55.01_{\pm 3.59}$	$79.29_{\pm 0.84}$
Attention	$50.52_{\pm 1.10}$	$61.40_{\pm 2.03}$	$61.33_{\pm 2.40}$	$75.98_{\pm 0.74}$
GatedAtt	$50.74_{\pm 0.61}$	$62.15_{\pm 0.77}$	$58.80_{\pm 1.18}$	$75.84_{\pm 1.29}$
DynamicP	$51.01_{\pm 1.88}$	$55.93_{\pm 1.56}$	$52.58_{\pm 2.91}$	$74.00_{\pm 1.61}$
GNP	$50.00_{\pm 1.88}$	$53.98_{\pm 6.34}$	$52.58_{\pm 4.68}$	$62.71_{\pm 15.55}$
ASAP	$50.89_{\pm 0.82}$	$63.66_{\pm 1.81}$	$60.67_{\pm 2.69}$	$77.15_{\pm 1.13}$
SAGP	$49.87_{\pm 0.77}$	$63.62_{\pm 1.28}$	$59.86_{\pm 2.43}$	$77.29_{\pm 1.04}$
OTK	$50.96_{\pm 1.11}$	$63.68_{\pm 1.59}$	**$61.66_{\pm 2.39}$**	$79.40_{\pm 1.08}$
SWE	$51.05_{\pm 2.15}$	$63.21_{\pm 2.02}$	$61.37_{\pm 3.13}$	$80.64_{\pm 1.86}$
WEGL	**$51.67_{\pm 0.85}$**	**$63.79_{\pm 2.54}$**	$61.36_{\pm 2.30}$	$81.98_{\pm 0.77}$
ROTP$_{\text{S}}$	**$51.96_{\pm 0.71}$**	$62.91_{\pm 1.13}$	$59.40_{\pm 0.90}$	$79.75_{\pm 0.71}$
ROTP$_{\text{B-E}}$	$51.26_{\pm 0.84}$	**$63.86_{\pm 2.41}$**	**$62.57_{\pm 1.34}$**	**$82.55_{\pm 0.42}$**
ROTP$_{\text{B-Q}}$	**$52.72_{\pm 0.66}$**	$63.15_{\pm 1.27}$	$60.88_{\pm 1.65}$	$81.43_{\pm 1.12}$

Case 2: OT-based Implicit Message-Passing Design

Given a graph $G(\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{Y}_{\mathcal{V}_L}$ are labels of partial nodes.

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

$$\max_\theta \prod\nolimits_{v\in\mathcal{V}_L} p(\boldsymbol{y}_v|\boldsymbol{X}, \boldsymbol{A}; \theta) \Leftrightarrow \min_\theta \sum\nolimits_{v\in\mathcal{V}_L} \psi(g_v(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{y}_v). \tag{11}$$

  - p is Gaussian $\Leftrightarrow$ ψ is MSE.
  - p is Sigmoid/Softmax $\Leftrightarrow$ ψ is cross entropy loss.

Case 2: OT-based Implicit Message-Passing Design

Given a graph $G(\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{Y}_{\mathcal{V}_L}$ are labels of partial nodes.

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

$$\max_\theta \prod\nolimits_{v\in\mathcal{V}_L} p(\boldsymbol{y}_v|\boldsymbol{X}, \boldsymbol{A}; \theta) \Leftrightarrow \min_\theta \sum\nolimits_{v\in\mathcal{V}_L} \psi(g_v(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{y}_v). \tag{11}$$

  - p is Gaussian $\Leftrightarrow$ ψ is MSE.
  - p is Sigmoid/Softmax $\Leftrightarrow$ ψ is cross entropy loss.

- **What we really want to do is maximizing the probability of $\boldsymbol{Y}_{\mathcal{V}_L}$**, i.e.,

$$\max_\theta p(\boldsymbol{Y}_{\mathcal{V}_L}|\boldsymbol{X}, \boldsymbol{A}; \theta) \tag{12}$$

Case 2: OT-based Implicit Message-Passing Design

Given a graph $G(\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{Y}_{\mathcal{V}_L}$ are labels of partial nodes.

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

$$\max_\theta \prod\nolimits_{v\in\mathcal{V}_L} p(\boldsymbol{y}_v|\boldsymbol{X}, \boldsymbol{A}; \theta) \Leftrightarrow \min_\theta \sum\nolimits_{v\in\mathcal{V}_L} \psi(g_v(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{y}_v). \tag{11}$$

  - p is Gaussian $\Leftrightarrow$ ψ is MSE.
  - p is Sigmoid/Softmax $\Leftrightarrow$ ψ is cross entropy loss.

- **What we really want to do is maximizing the probability of $\boldsymbol{Y}_{\mathcal{V}_L}$**, i.e.,

$$\max_\theta p(\boldsymbol{Y}_{\mathcal{V}_L}|\boldsymbol{X}, \boldsymbol{A}; \theta) \tag{12}$$

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

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

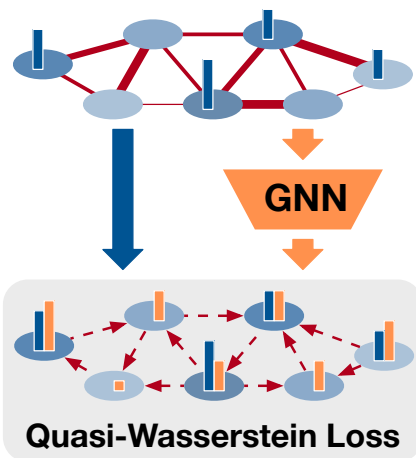
Proposed Learning Strategy

Ideally, we would like to define a loss for label sets, rather than individual labels, e.g.,

$$\min_\theta \text{Loss}(\underbrace{g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)}_{\widehat{\boldsymbol{Y}}_{\mathcal{V}_L}}, \boldsymbol{Y}_{\mathcal{V}_L}), \qquad (13)$$

which can be implemented as **an OT problem of labels defined on a graph.**

- **Moreover, lead to an implicit message-passing layer (with a QW loss) encoding label transport.**



Proposed Quasi-Wasserstein (QW) Loss of GNNs

- 1-Wasserstein distance between signals on a graph

$$W_1(\boldsymbol{\mu}, \boldsymbol{\gamma}) := \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \langle \boldsymbol{D},\ \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \sum\nolimits_{v, v' \in \mathcal{V} \times \mathcal{V}} t_{vv'} d_{vv'}, \quad (14)$$

  where $\boldsymbol{D}$ is the shortest-path distance matrix.

Proposed Quasi-Wasserstein (QW) Loss of GNNs

- 1-Wasserstein distance between signals on a graph

$$W_1(\boldsymbol{\mu}, \boldsymbol{\gamma}) := \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \langle \boldsymbol{D}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \sum\nolimits_{v, v' \in \mathcal{V} \times \mathcal{V}} t_{vv'} d_{vv'}, \tag{14}$$

  where $\boldsymbol{D}$ is the shortest-path distance matrix.

- W_1 **is equivalent to a minimum-cost flow problem:**

$$W_1(\boldsymbol{\mu}, \boldsymbol{\gamma}) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}}, \boldsymbol{\mu}, \boldsymbol{\gamma})} \|\mathrm{diag}(\boldsymbol{w}) \boldsymbol{f}\|_1, \tag{15}$$

  where $\boldsymbol{f} = [f_e] \in \mathbb{R}^{|\mathcal{E}|}$ is the flow on each edge.

$$\boldsymbol{S}_{\mathcal{V}} = [s_{ve}] \in \{0, \pm 1\}^{|\mathcal{V}| \times |\mathcal{E}|}, \quad s_{ve} = \begin{cases} 1 & \text{if } v \text{ is "head" of edge } e \\ -1 & \text{if } v \text{ is "tail" of edge } e \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Proposed Quasi-Wasserstein (QW) Loss of GNNs

- 1-Wasserstein distance between signals on a graph

$$W_1(\boldsymbol{\mu}, \boldsymbol{\gamma}) := \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \langle \boldsymbol{D}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\gamma})} \sum\nolimits_{v, v' \in \mathcal{V} \times \mathcal{V}} t_{vv'} d_{vv'}, \quad (14)$$

where $\boldsymbol{D}$ is the shortest-path distance matrix.

- W_1 **is equivalent to a minimum-cost flow problem:**

$$W_1(\boldsymbol{\mu}, \boldsymbol{\gamma}) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}}, \boldsymbol{\mu}, \boldsymbol{\gamma})} \|\mathrm{diag}(\boldsymbol{w}) \boldsymbol{f}\|_1, \quad (15)$$

where $\boldsymbol{f} = [f_e] \in \mathbb{R}^{|\mathcal{E}|}$ is the flow on each edge.

$$\boldsymbol{S}_{\mathcal{V}} = [s_{ve}] \in \{0, \pm 1\}^{|\mathcal{V}| \times |\mathcal{E}|}, \quad s_{ve} = \begin{cases} 1 & \text{if } v \text{ is "head" of edge } e \\ -1 & \text{if } v \text{ is "tail" of edge } e \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

$$\Omega(\boldsymbol{S}_{\mathcal{V}}, \boldsymbol{\mu}, \boldsymbol{\gamma}) = \mathcal{U}^{|\mathcal{E}|} \cap \{\boldsymbol{f} \mid \boldsymbol{S}_{\mathcal{V}} \boldsymbol{f} = \boldsymbol{\gamma} - \boldsymbol{\mu}\}, \quad \text{where } \mathcal{U} = \begin{cases} [0, \infty), & \text{Directed } G, \\ \mathbb{R}, & \text{Undirected } G. \end{cases}$$

Proposed Quasi-Wasserstein (QW) Loss of GNNs

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

$$W_1^{(P)}(\boldsymbol{\mu}, \boldsymbol{\gamma}) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, \boldsymbol{\mu}, \boldsymbol{\gamma})} \|\mathrm{diag}(\boldsymbol{w})\boldsymbol{f}\|_1, \tag{17}$$

- For partially-observed multi-dimensional labels, i.e., $\boldsymbol{Y}_{\mathcal{V}_L} = [\boldsymbol{y}_{\mathcal{V}_L}^{(c)}] \in \mathbb{R}^{|\mathcal{V}_L| \times C}$, **aggregating each dimension's $W_1^{(P)}$ leads to the QW loss:**

$$\begin{aligned} QW(\widehat{\boldsymbol{Y}}_{\mathcal{V}_L}, \boldsymbol{Y}_{\mathcal{V}_L}) &= \sum\nolimits_{c=1}^{C} W_1^{(P)}(\hat{\boldsymbol{y}}_{\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{\mathcal{V}_L}^{(c)}) \\ &= \sum\nolimits_{c=1}^{C} \min_{\boldsymbol{f}^{(c)} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \hat{\boldsymbol{y}}_{\mathcal{V}_L}^{(c)},\ \boldsymbol{y}_{\mathcal{V}_L}^{(c)})} \|\mathrm{diag}(\boldsymbol{w})\boldsymbol{f}^{(c)}\|_1 \end{aligned}$$

Proposed Quasi-Wasserstein (QW) Loss of GNNs

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

$$W_1^{(P)}(\boldsymbol{\mu}, \boldsymbol{\gamma}) = \min_{\boldsymbol{f} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L}, \boldsymbol{\mu}, \boldsymbol{\gamma})} \|\mathrm{diag}(\boldsymbol{w})\boldsymbol{f}\|_1, \tag{17}$$

- For partially-observed multi-dimensional labels, i.e., $\boldsymbol{Y}_{\mathcal{V}_L} = [\boldsymbol{y}_{\mathcal{V}_L}^{(c)}] \in \mathbb{R}^{|\mathcal{V}_L| \times C}$, **aggregating each dimension's $W_1^{(P)}$ leads to the QW loss:**

$$\begin{aligned} QW(\widehat{\boldsymbol{Y}}_{\mathcal{V}_L}, \boldsymbol{Y}_{\mathcal{V}_L}) &= \sum\nolimits_{c=1}^{C} W_1^{(P)}(\hat{\boldsymbol{y}}_{\mathcal{V}_L}^{(c)}, \boldsymbol{y}_{\mathcal{V}_L}^{(c)}) \\ &= \sum\nolimits_{c=1}^{C} \min_{\boldsymbol{f}^{(c)} \in \Omega(\boldsymbol{S}_{\mathcal{V}_L},\ \hat{\boldsymbol{y}}_{\mathcal{V}_L}^{(c)},\ \boldsymbol{y}_{\mathcal{V}_L}^{(c)})} \|\mathrm{diag}(\boldsymbol{w})\boldsymbol{f}^{(c)}\|_1 \\ &= \min_{\boldsymbol{F} \in \Omega_C(\boldsymbol{S}_{\mathcal{V}_L},\ g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathrm{diag}(\boldsymbol{w})\boldsymbol{F}\|_1, \end{aligned} \tag{18}$$

  - $\boldsymbol{F} = [\boldsymbol{f}^{(c)}]$ is the flow matrix with size $|\mathcal{E}| \times C$
  - Feasible domain $\Omega_C = \mathcal{U}^{|\mathcal{E}| \times C} \cap \{\boldsymbol{F} \mid \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F} = \boldsymbol{Y}_{\mathcal{V}_L} - g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)\}$.

An Implicit Message Passing Associated with The QW Loss

- Learning GNN with the QW loss:

$$\begin{aligned} &\min_\theta QW(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta), \boldsymbol{Y}_{\mathcal{V}_L}) \\ =&\min_\theta \min_{\boldsymbol{F} \in \Omega_C(\boldsymbol{S}_{\mathcal{V}_L},\ g_{\mathcal{V}_L}(\boldsymbol{X},\boldsymbol{A};\theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1. \end{aligned} \tag{19}$$

An Implicit Message Passing Associated with The QW Loss

- Learning GNN with the QW loss:

$$\begin{aligned} &\min_\theta QW(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta), \boldsymbol{Y}_{\mathcal{V}_L}) \\ =&\min_\theta \min_{\boldsymbol{F} \in \Omega_C(\boldsymbol{S}_{\mathcal{V}_L},\ g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1. \end{aligned} \tag{19}$$

- Optionally, the flow can be used to parameterize the adjacency matrix

$$\min_{\theta,\xi} \min_{\boldsymbol{F} \in \Omega_C(\boldsymbol{S}_{\mathcal{V}_L}, g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}(\boldsymbol{F};\xi);\theta), \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1, \tag{20}$$

An Implicit Message Passing Associated with The QW Loss

- Learning GNN with the QW loss:

$$\begin{aligned} &\min_\theta QW(g_{\mathcal{V}_L}(\boldsymbol{X},\boldsymbol{A};\theta),\boldsymbol{Y}_{\mathcal{V}_L}) \\ =&\min_\theta \min_{\boldsymbol{F}\in\Omega_C(\boldsymbol{S}_{\mathcal{V}_L},\ g_{\mathcal{V}_L}(\boldsymbol{X},\boldsymbol{A};\theta),\ \boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1. \end{aligned} \tag{19}$$

- Optionally, the flow can be used to parameterize the adjacency matrix

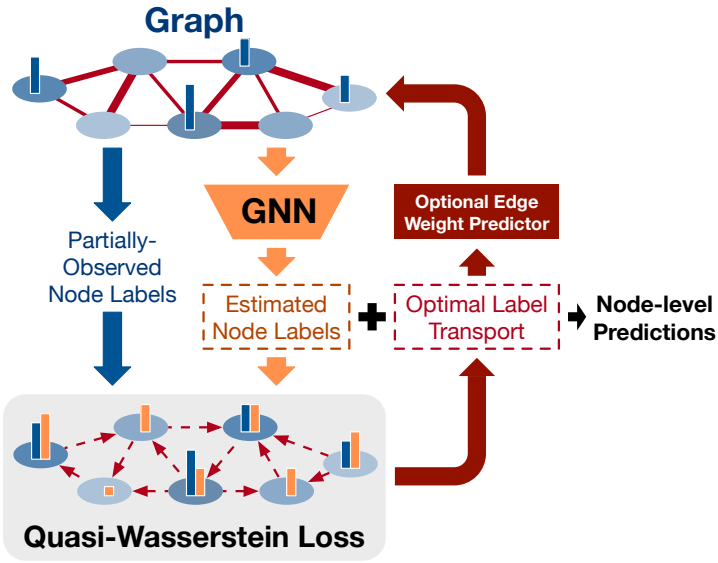
$$\min_{\theta,\xi} \min_{\boldsymbol{F}\in\Omega_C(\boldsymbol{S}_{\mathcal{V}_L},g_{\mathcal{V}_L}(\boldsymbol{X},\boldsymbol{A}(\boldsymbol{F};\xi);\theta),\boldsymbol{Y}_{\mathcal{V}_L})} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1, \tag{20}$$

- Adding the optimal label transport in the testing phase: For $v \in \mathcal{V}\setminus\mathcal{V}_L$, we predict its label as

$$\tilde{\boldsymbol{y}}_v := g_v(\boldsymbol{X},\boldsymbol{A};\theta^*) + \boldsymbol{S}_v\boldsymbol{F}^*,\ \text{s.t. } \boldsymbol{F}^* \Leftarrow QW(g_{\mathcal{V}_L}(\boldsymbol{X},\boldsymbol{A};\theta),\boldsymbol{Y}_{\mathcal{V}_L}). \tag{21}$$

An implicit message passing layer encoding training labels' optimal transport.

An Implicit Message Passing Associated with The QW Loss



Solvers of QW Loss

$$\begin{aligned} &\min_{\theta} \min_{\boldsymbol{F}} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1. \\ &s.t.\ \boldsymbol{F} \in \mathcal{U}^{|\mathcal{E}|\times C} \cap \{\boldsymbol{F} \mid \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F} = \boldsymbol{Y}_{\mathcal{V}_L} - g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)\} \end{aligned} \tag{22}$$

where $B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle$.

Solvers of QW Loss

$$\begin{aligned} &\min_{\theta} \min_{\boldsymbol{F}} \|\text{diag}(\boldsymbol{w})\boldsymbol{F}\|_1. \\ &s.t.\ \boldsymbol{F} \in \mathcal{U}^{|\mathcal{E}|\times C} \cap \{\boldsymbol{F} \mid \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F} = \boldsymbol{Y}_{\mathcal{V}_L} - g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)\} \end{aligned} \tag{22}$$

where $B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle$.

- An inexact solver based on Bregman Divergence-based Relaxation:

$$\min_{\theta,\ \boldsymbol{F}\in\mathcal{U}^{|\mathcal{E}|\times C}} \|\text{diag}(\boldsymbol{w})\boldsymbol{F}\|_1 + \lambda B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F},\ \boldsymbol{Y}_{\mathcal{V}_L}). \tag{23}$$

Solvers of QW Loss

$$\begin{aligned} &\min_{\theta} \min_{\boldsymbol{F}} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1. \\ &s.t.\ \boldsymbol{F} \in \mathcal{U}^{|\mathcal{E}|\times C} \cap \{\boldsymbol{F} \mid \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F} = \boldsymbol{Y}_{\mathcal{V}_L} - g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta)\} \end{aligned} \tag{22}$$

where $B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle$.

- An inexact solver based on Bregman Divergence-based Relaxation:

$$\min_{\theta,\ \boldsymbol{F}\in\mathcal{U}^{|\mathcal{E}|\times C}} \|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1 + \lambda B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F},\ \boldsymbol{Y}_{\mathcal{V}_L}). \tag{23}$$

- An exact solver based on Bregman ADMM:

$$\begin{aligned} \min_{\theta,\ \boldsymbol{F}\in\mathcal{U}^{|\mathcal{E}|\times C}} \max_{\boldsymbol{Z}\in\mathbb{R}^{|\mathcal{V}_L|\times C}} &\|\mathsf{diag}(\boldsymbol{w})\boldsymbol{F}\|_1 \\ &+ \langle \boldsymbol{Z},\ g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F} - \boldsymbol{Y}_{\mathcal{V}_L}\rangle \\ &+ \lambda B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F},\ \boldsymbol{Y}_{\mathcal{V}_L}). \end{aligned} \tag{24}$$

Compare with Traditional Learning Paradigm

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

Theorem

Let $\{\theta^\star, \boldsymbol{F}^\star, \boldsymbol{Z}^\star\}$ *be the optimal solution of* (24), $\{\theta^\dagger, \boldsymbol{F}^\dagger\}$ *be the optimal solution of* (23)*, and* $\theta^\ddagger$ *be the optimal solution of* $\min_\theta B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta),\ \boldsymbol{Y}_{\mathcal{V}_L})$*, we have*

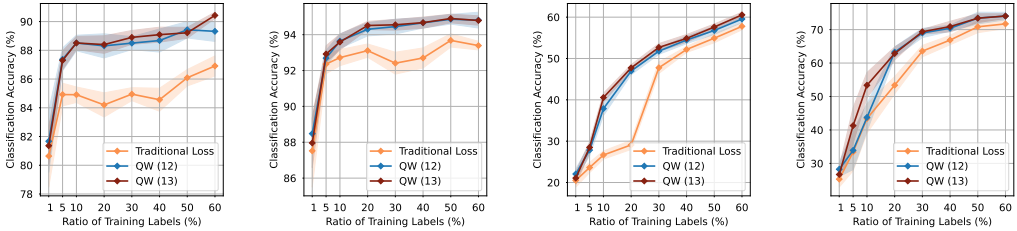
$$\begin{aligned} B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta^\star) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F}^\star,\ \boldsymbol{Y}_{\mathcal{V}_L}) &\le B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta^\dagger) + \boldsymbol{S}_{\mathcal{V}_L}\boldsymbol{F}^\dagger,\ \boldsymbol{Y}_{\mathcal{V}_L}) \\ &\le B_\phi(g_{\mathcal{V}_L}(\boldsymbol{X}, \boldsymbol{A}; \theta^\ddagger),\ \boldsymbol{Y}_{\mathcal{V}_L}) \end{aligned}$$

QW Loss and Implicit MP: Superiority

Node classification tasks:

Model	Method	Homophilic graphs					Heterophilic graphs					Overall Improve
		Cora	Citeseer	Pubmed	Computers	Photo	Squirrel	Chameleon	Actor	Texas	Cornell	
#Nodes ($\|\mathcal{V}\|$)		2,708	3,327	19,717	13,752	5,201	7,650	2,277	7,600	183	183	
#Features (D)		1,433	3,703	500	767	754	2,089	2,325	932	1,703	1,703	
#Edges ($\|\mathcal{E}\|$)		5,278	4,552	44,324	245,861	119,081	198,358	31,371	26,659	279	277	
Intra-edge rate		81.0%	73.6%	80.2%	77.7%	82.7%	22.2%	23.0%	21.8%	6.1%	12.3%	
#Classes (C)		7	6	5	10	8	5	5	5	5	5	
GCN	(2)	87.44$_{\pm 0.96}$	79.98$_{\pm 0.84}$	86.93$_{\pm 0.29}$	88.42$_{\pm 0.45}$	93.24$_{\pm 0.43}$	46.55$_{\pm 1.15}$	63.57$_{\pm 1.16}$	34.00$_{\pm 1.28}$	77.21$_{\pm 3.28}$	61.91$_{\pm 5.11}$	—
	(2)+LPA	86.34$_{\pm 1.45}$	78.51$_{\pm 1.22}$	84.72$_{\pm 0.70}$	82.48$_{\pm 0.69}$	88.10$_{\pm 1.31}$	44.81$_{\pm 1.81}$	60.90$_{\pm 1.63}$	32.43$_{\pm 1.59}$	78.69$_{\pm 6.47}$	68.72$_{\pm 5.95}$	-1.36
	QW	**87.88$_{\pm 0.79}$**	**81.36$_{\pm 0.41}$**	**87.89$_{\pm 0.40}$**	**89.20$_{\pm 0.41}$**	**93.81$_{\pm 0.36}$**	**52.62$_{\pm 0.49}$**	**68.10$_{\pm 1.01}$**	**38.09$_{\pm 0.50}$**	**84.10$_{\pm 2.95}$**	**84.26$_{\pm 2.98}$**	+4.81
GAT	(2)	**89.20$_{\pm 0.79}$**	**80.75$_{\pm 0.78}$**	87.42$_{\pm 0.33}$	90.08$_{\pm 0.36}$	94.38$_{\pm 0.25}$	48.20$_{\pm 1.67}$	64.31$_{\pm 2.01}$	**35.68$_{\pm 0.60}$**	80.00$_{\pm 3.11}$	68.09$_{\pm 2.13}$	—
	QW	89.11$_{\pm 0.66}$	80.19$_{\pm 0.64}$	**88.38$_{\pm 0.23}$**	**90.41$_{\pm 0.28}$**	**94.65$_{\pm 0.24}$**	**55.03$_{\pm 1.35}$**	**67.35$_{\pm 1.42}$**	33.86$_{\pm 2.13}$	**80.33$_{\pm 1.80}$**	**70.21$_{\pm 2.13}$**	+1.14
GIN	(2)	86.22$_{\pm 0.95}$	**76.18$_{\pm 0.78}$**	**87.87$_{\pm 0.23}$**	80.87$_{\pm 1.43}$	89.83$_{\pm 0.72}$	39.11$_{\pm 2.23}$	64.29$_{\pm 1.51}$	**32.37$_{\pm 1.56}$**	72.79$_{\pm 4.92}$	62.55$_{\pm 4.80}$	—
	QW	**86.24$_{\pm 0.90}$**	76.13$_{\pm 1.09}$	87.53$_{\pm 0.34}$	**89.28$_{\pm 0.45}$**	**92.60$_{\pm 0.44}$**	**65.29$_{\pm 0.68}$**	**73.26$_{\pm 1.12}$**	32.32$_{\pm 1.93}$	**77.54$_{\pm 2.60}$**	**64.04$_{\pm 3.62}$**	+5.22
GraphSAGE	(2)	**88.24$_{\pm 0.95}$**	79.81$_{\pm 0.80}$	88.14$_{\pm 0.25}$	89.71$_{\pm 0.38}$	95.08$_{\pm 0.26}$	43.79$_{\pm 0.59}$	63.26$_{\pm 1.09}$	**38.99$_{\pm 0.85}$**	90.00$_{\pm 2.30}$	84.26$_{\pm 2.98}$	—
	QW	87.59$_{\pm 0.77}$	**80.52$_{\pm 0.68}$**	**88.61$_{\pm 0.32}$**	**90.17$_{\pm 0.24}$**	**95.25$_{\pm 0.25}$**	**54.37$_{\pm 0.89}$**	**68.32$_{\pm 0.68}$**	37.82$_{\pm 0.45}$	**90.33$_{\pm 1.97}$**	**86.38$_{\pm 2.13}$**	+1.18
APPNP	(2)	88.14$_{\pm 0.73}$	80.47$_{\pm 0.74}$	88.12$_{\pm 0.31}$	85.32$_{\pm 0.37}$	88.51$_{\pm 0.31}$	36.15$_{\pm 0.75}$	52.93$_{\pm 1.71}$	40.46$_{\pm 0.64}$	91.31$_{\pm 1.97}$	87.66$_{\pm 2.13}$	—
	QW	**88.74$_{\pm 0.84}$**	**80.94$_{\pm 0.61}$**	**89.48$_{\pm 0.28}$**	**86.95 $_{\pm 0.82}$**	**94.43$_{\pm 0.24}$**	**38.73$_{\pm 1.06}$**	**53.76$_{\pm 1.25}$**	**40.78$_{\pm 0.74}$**	**91.48$_{\pm 2.30}$**	**87.87$_{\pm 2.34}$**	+1.41
BernNet	(2)	88.28$_{\pm 1.00}$	79.81$_{\pm 0.79}$	88.87$_{\pm 0.38}$	87.61$_{\pm 0.46}$	93.68$_{\pm 0.28}$	51.15$_{\pm 1.09}$	67.96$_{\pm 1.05}$	40.72$_{\pm 0.80}$	93.28$_{\pm 1.48}$	90.21$_{\pm 2.35}$	—
	QW	**89.03$_{\pm 0.76}$**	**81.35$_{\pm 0.71}$**	**89.03$_{\pm 0.38}$**	**89.58$_{\pm 0.47}$**	**94.55$_{\pm 0.39}$**	**55.22$_{\pm 0.64}$**	**71.66$_{\pm 1.18}$**	**40.91$_{\pm 0.71}$**	**93.44$_{\pm 1.80}$**	**90.85$_{\pm 2.34}$**	+1.41
ChebNetII	(2)	88.26$_{\pm 0.89}$	**80.00$_{\pm 0.74}$**	88.57$_{\pm 0.36}$	86.58$_{\pm 0.71}$	93.50$_{\pm 0.34}$	57.78$_{\pm 0.84}$	71.71$_{\pm 1.40}$	40.70$_{\pm 0.77}$	92.79$_{\pm 1.48}$	**88.94$_{\pm 2.78}$**	—
	QW	**88.54$_{\pm 0.76}$**	79.47$_{\pm 0.70}$	**89.47$_{\pm 0.36}$**	**90.43$_{\pm 0.22}$**	**94.84$_{\pm 0.37}$**	**60.55$_{\pm 0.64}$**	**74.05$_{\pm 0.68}$**	**41.37$_{\pm 0.67}$**	**93.93$_{\pm 0.98}$**	87.23$_{\pm 3.62}$	+1.11

QW Loss and Implicit MP: Superiority



(a) Computers (b) Photo (c) Squirrel (d) Chameleon

Figure: Illustrations of the methods' performance given different amounts of labeled nodes.

Node regression tasks:

Model	Method	Homophilic graphs		Heterophilic graphs	
		Computers	Photo	Actor	Cornell
GIN	(2)	$0.0605_{\pm 0.0018}$	$0.0459_{\pm 0.0044}$	$0.1570_{\pm 0.0014}$	$0.1609_{\pm 0.0359}$
	QW	$\mathbf{0.0244_{\pm 0.0028}}$	$\mathbf{0.0203_{\pm 0.0012}}$	$\mathbf{0.1564_{\pm 0.0012}}$	$\mathbf{0.1524_{\pm 0.0043}}$
BernNet	(2)	$0.0871_{\pm 0.0002}$	$0.0488_{\pm 0.0009}$	$\mathbf{0.1661_{\pm 0.0020}}$	$0.0989_{\pm 0.0076}$
	QW	$\mathbf{0.0364_{\pm 0.0038}}$	$\mathbf{0.0297_{\pm 0.0014}}$	$0.1671_{\pm 0.0008}$	$\mathbf{0.0753_{\pm 0.0024}}$

Case 3: OT-based Implicit Structured Attention Layer Design

Empirically, the power of Transformer is attributed to its Multi-head Attention (MHA) Architecture:

$$\mathsf{MHA}_\theta(\boldsymbol{X}) := \mathsf{Concat}_{\mathsf{col}}(\{\mathsf{Att}(\boldsymbol{V}_m; \boldsymbol{Q}_m, \boldsymbol{K}_m)\}_{m=1}^{M}), \tag{25}$$

For $m = 1, ..., M$, we have $\boldsymbol{V}_m = \boldsymbol{X}\boldsymbol{W}_{V,m}$, $\boldsymbol{Q}_m = \boldsymbol{X}\boldsymbol{W}_{Q,m}$, and $\boldsymbol{K}_m = \boldsymbol{X}\boldsymbol{W}_{K,m}$.

$$\boldsymbol{P}(\boldsymbol{Q}, \boldsymbol{K}) = \mathsf{Softmax}\Big(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\Big), \tag{26}$$

Case 3: OT-based Implicit Structured Attention Layer Design

Empirically, the power of Transformer is attributed to its Multi-head Attention (MHA) Architecture:

$$\mathrm{MHA}_\theta(\boldsymbol{X}) := \mathrm{Concat}_{\mathrm{col}}(\{\mathrm{Att}(\boldsymbol{V}_m; \boldsymbol{Q}_m, \boldsymbol{K}_m)\}_{m=1}^M), \tag{25}$$

For $m = 1, ..., M$, we have $\boldsymbol{V}_m = \boldsymbol{X}\boldsymbol{W}_{V,m}$, $\boldsymbol{Q}_m = \boldsymbol{X}\boldsymbol{W}_{Q,m}$, and $\boldsymbol{K}_m = \boldsymbol{X}\boldsymbol{W}_{K,m}$.

$$\boldsymbol{P}(\boldsymbol{Q}, \boldsymbol{K}) = \mathrm{Softmax}\Big(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\Big), \tag{26}$$

Challenges:

- **Over-smoothness** when dealing with long sequences.
- **Quadratic complexity** w.r.t. the sequence length.

Development Tendency of Attention Layer

Model	Attention($\boldsymbol{V}; \boldsymbol{Q}, \boldsymbol{K}$)	Complexity	Attention Structure	Faster?	Better?
Transformer	Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^2)$	Dense + Row-wisely normalized	—	—

Development Tendency of Attention Layer

Model	Attention($\boldsymbol{V};\boldsymbol{Q},\boldsymbol{K}$)	Complexity	Attention Structure	Faster?	Better?
Transformer	Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^2)$	Dense + Row-wisely normalized	—	—
SparseTrans	Local2D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^{1.5})$	Sparse + Row-wisely normalized	Yes	Comparable
Longformer	Local1D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DNL)$	Sparse + Row-wisely normalized	Yes	Comparable
Reformer	LSH-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN\log N)$	Sparse + Row-wisely normalized	Yes	Comparable
CosFormer	$(\boldsymbol{Q}_{\cos}\boldsymbol{K}_{\cos}^\top+\boldsymbol{Q}_{\sin}\boldsymbol{K}_{\sin}^\top)\boldsymbol{V}$	$\mathcal{O}(DE_{QK})$	Sparse	Yes	Comparable

Development Tendency of Attention Layer

Model	Attention($\boldsymbol{V};\boldsymbol{Q},\boldsymbol{K}$)	Complexity	Attention Structure	Faster?	Better?
Transformer	Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^2)$	Dense + Row-wisely normalized	—	—
SparseTrans	Local2D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^{1.5})$	Sparse + Row-wisely normalized	Yes	Comparable
Longformer	Local1D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DNL)$	Sparse + Row-wisely normalized	Yes	Comparable
Reformer	LSH-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN\log N)$	Sparse + Row-wisely normalized	Yes	Comparable
CosFormer	$(\boldsymbol{Q}_{\cos}\boldsymbol{K}_{\cos}^\top+\boldsymbol{Q}_{\sin}\boldsymbol{K}_{\sin}^\top)\boldsymbol{V}$	$\mathcal{O}(DE_{QK})$	Sparse	Yes	Comparable
Performer	$\phi_r(\boldsymbol{Q})\phi_r(\boldsymbol{K})^\top\boldsymbol{V}$	$\mathcal{O}(DNr)$	Low-rank	Yes	No
Linformer	Softmax $\left(\frac{\boldsymbol{Q}\psi_r(\boldsymbol{K})^\top}{\sqrt{D}}\right)\psi_r(\boldsymbol{V})$	$\mathcal{O}(DNr)$	Low-rank + Row-wisely normalized	Yes	No

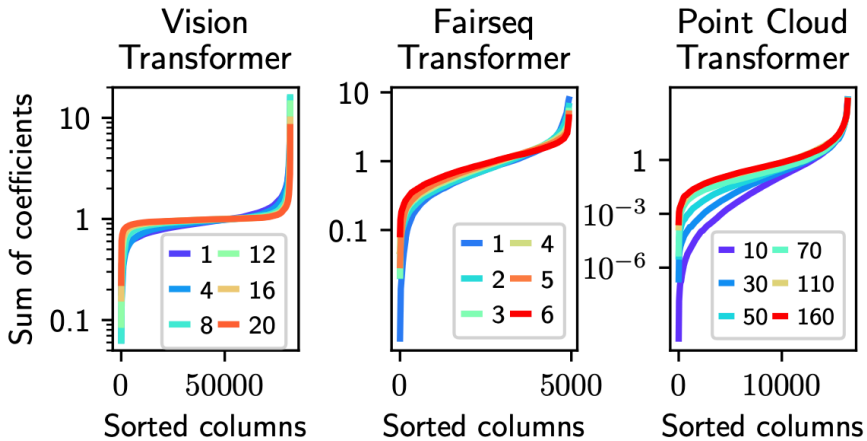
Development Tendency of Attention Layer

Model	Attention($\boldsymbol{V}; \boldsymbol{Q}, \boldsymbol{K}$)	Complexity	Attention Structure	Faster?	Better?
Transformer	Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^2)$	Dense + Row-wisely normalized	—	—
SparseTrans	Local2D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^{1.5})$	Sparse + Row-wisely normalized	Yes	Comparable
Longformer	Local1D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DNL)$	Sparse + Row-wisely normalized	Yes	Comparable
Reformer	LSH-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN\log N)$	Sparse + Row-wisely normalized	Yes	Comparable
CosFormer	$(\boldsymbol{Q}_{\cos}\boldsymbol{K}_{\cos}^\top + \boldsymbol{Q}_{\sin}\boldsymbol{K}_{\sin}^\top)\boldsymbol{V}$	$\mathcal{O}(DE_{QK})$	Sparse	Yes	Comparable
Performer	$\phi_r(\boldsymbol{Q})\phi_r(\boldsymbol{K})^\top\boldsymbol{V}$	$\mathcal{O}(DNr)$	Low-rank	Yes	No
Linformer	Softmax $\left(\frac{\boldsymbol{Q}\psi_r(\boldsymbol{K})^\top}{\sqrt{D}}\right)\psi_r(\boldsymbol{V})$	$\mathcal{O}(DNr)$	Low-rank + Row-wisely normalized	Yes	No
Sinkformer	Sinkhorn$_K$ $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(KDN^2)$	Dense + Doubly stochastic	No	Yes

Development Tendency of Attention Layer

Model	Attention($\boldsymbol{V};\boldsymbol{Q},\boldsymbol{K}$)	Complexity	Attention Structure	Faster?	Better?
Transformer	Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^2)$	Dense + Row-wisely normalized	—	—
SparseTrans	Local2D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN^{1.5})$	Sparse + Row-wisely normalized	Yes	Comparable
Longformer	Local1D-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DNL)$	Sparse + Row-wisely normalized	Yes	Comparable
Reformer	LSH-Softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(DN\log N)$	Sparse + Row-wisely normalized	Yes	Comparable
CosFormer	$(\boldsymbol{Q}_{\cos}\boldsymbol{K}_{\cos}^\top+\boldsymbol{Q}_{\sin}\boldsymbol{K}_{\sin}^\top)\boldsymbol{V}$	$\mathcal{O}(DE_{QK})$	Sparse	Yes	Comparable
Performer	$\phi_r(\boldsymbol{Q})\phi_r(\boldsymbol{K})^\top\boldsymbol{V}$	$\mathcal{O}(DNr)$	Low-rank	Yes	No
Linformer	Softmax $\left(\frac{\boldsymbol{Q}\psi_r(\boldsymbol{K})^\top}{\sqrt{D}}\right)\psi_r(\boldsymbol{V})$	$\mathcal{O}(DNr)$	Low-rank + Row-wisely normalized	Yes	No
Sinkformer	Sinkhorn$_K$ $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V}$	$\mathcal{O}(KDN^2)$	Dense + Doubly stochastic	No	Yes
Sliceformer	Sort$_{\text{col}}(\boldsymbol{V})$	$\mathcal{O}(DN\log N)$	Full-rank+Sparse+Doubly stochastic	Yes	Yes

Doubly Stochastic Tendency of Attention Map



Sander, Michael E., et al. Sinkformers: Transformers with doubly stochastic attention. AISTATS, 2022.

Implement Attention Maps Implicitly via Slicing-Sorting

The attention layer in Sliceformer is implemented by slicing-sorting:

$$\text{SliceSort}(\boldsymbol{X}) := \text{Sort}_{\text{col}}(\underbrace{\boldsymbol{X}\boldsymbol{W}_V}_{\boldsymbol{V}=[\boldsymbol{v}_i]}) = \text{Concat}_{\text{col}}(\{\boldsymbol{P}_i\boldsymbol{v}_i\}_{i=1}^{MD}), \tag{27}$$

$\boldsymbol{P}_i$ is a permutation matrix corresponding to the sorting of $\boldsymbol{v}_i$.

- Full-rank, sparse, and doubly stochastic.

Implement Attention Maps Implicitly via Slicing-Sorting

The attention layer in Sliceformer is implemented by slicing-sorting:

$$\mathsf{SliceSort}(\boldsymbol{X}) := \mathsf{Sort}_{\mathsf{col}}(\underbrace{\boldsymbol{X}\boldsymbol{W}_V}_{\boldsymbol{V}=[\boldsymbol{v}_i]}) = \mathsf{Concat}_{\mathsf{col}}(\{\boldsymbol{P}_i\boldsymbol{v}_i\}_{i=1}^{MD}), \tag{27}$$

$\boldsymbol{P}_i$ is a permutation matrix corresponding to the sorting of $\boldsymbol{v}_i$.

- Full-rank, sparse, and doubly stochastic.

Slicing-sorting operation = computing the Wasserstein barycenter of $\{\boldsymbol{v}_i\}_{i=1}^{MD}$:

$$\min_{\boldsymbol{v}} \sum_{i=1}^{MD} W_1(\boldsymbol{v}, \boldsymbol{v}_i)$$

Implement Attention Maps Implicitly via Slicing-Sorting

The attention layer in Sliceformer is implemented by slicing-sorting:

$$\mathsf{SliceSort}(\boldsymbol{X}) := \mathsf{Sort}_{\mathsf{col}}(\underbrace{\boldsymbol{X}\boldsymbol{W}_V}_{\boldsymbol{V}=[\boldsymbol{v}_i]}) = \mathsf{Concat}_{\mathsf{col}}(\{\boldsymbol{P}_i\boldsymbol{v}_i\}_{i=1}^{MD}), \tag{27}$$

$\boldsymbol{P}_i$ is a permutation matrix corresponding to the sorting of $\boldsymbol{v}_i$.

- Full-rank, sparse, and doubly stochastic.

Slicing-sorting operation = computing the Wasserstein barycenter of $\{\boldsymbol{v}_i\}_{i=1}^{MD}$:

$$\min_{\boldsymbol{v}} \sum_{i=1}^{MD} W_1(\boldsymbol{v}, \boldsymbol{v}_i) \xRightarrow{\text{1D case}} \sum_{i=1}^{MD} W_1(\boldsymbol{v}_1, \boldsymbol{v}_i)$$

Implement Attention Maps Implicitly via Slicing-Sorting

The attention layer in Sliceformer is implemented by slicing-sorting:

$$\mathsf{SliceSort}(\boldsymbol{X}) := \mathsf{Sort}_{\mathsf{col}}(\underbrace{\boldsymbol{X}\boldsymbol{W}_V}_{\boldsymbol{V}=[\boldsymbol{v}_i]}) = \mathsf{Concat}_{\mathsf{col}}(\{\boldsymbol{P}_i\boldsymbol{v}_i\}_{i=1}^{MD}), \tag{27}$$

$\boldsymbol{P}_i$ is a permutation matrix corresponding to the sorting of $\boldsymbol{v}_i$.

- Full-rank, sparse, and doubly stochastic.

Slicing-sorting operation = computing the Wasserstein barycenter of $\{\boldsymbol{v}_i\}_{i=1}^{MD}$:

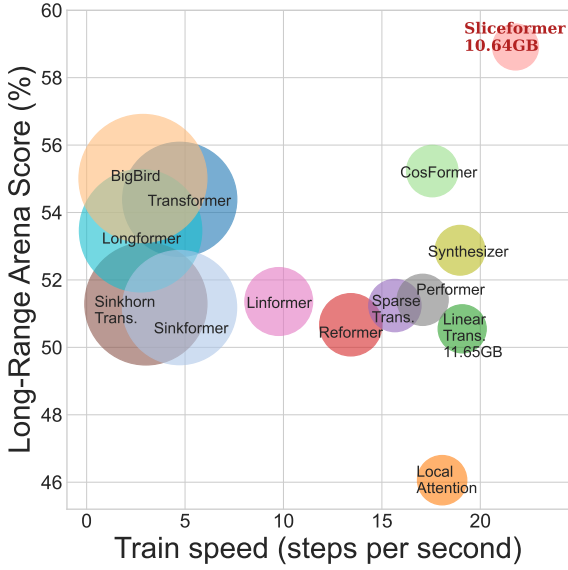
$$\min_{\boldsymbol{v}} \sum_{i=1}^{MD} W_1(\boldsymbol{v}, \boldsymbol{v}_i) \xRightarrow{\text{1D case}} \sum_{i=1}^{MD} W_1(\boldsymbol{v}_1, \boldsymbol{v}_i) \xRightarrow{\text{Alignment}} \sum_{i=1}^{MD}\sum_{n=1}^{N} |v_{1n} - v_{i\sigma_i(n)}|^2 \tag{28}$$

$\sigma_i \Leftrightarrow \boldsymbol{P}_i,\ \forall i = 1, ..., MD.$

Sliceformer: Superiority on LRA

Model	ListOps	Text	Retrieval	Image	Path	Path-X	Avg.
Transformer	36.37	64.27	57.46	42.44	71.40	FAIL	54.39
Local Att.	15.82	52.98	53.39	41.46	66.63	FAIL	46.06
Linear Trans.	16.13	**65.90**	53.09	42.34	75.30	FAIL	50.55
Reformer	37.27	56.10	53.40	38.07	68.50	FAIL	50.67
Sinkformer	30.70	64.03	55.45	41.08	64.65	FAIL	51.18
SparseTrans	17.07	63.58	59.59	44.24	71.71	FAIL	51.24
SinkhornTrans	33.67	61.20	53.83	41.23	67.45	FAIL	51.29
Linformer	35.70	53.94	52.27	38.56	76.34	FAIL	51.36
Performer	18.01	65.40	53.82	42.77	77.05	FAIL	51.41
Synthesizer	36.99	61.68	54.67	41.61	69.45	FAIL	52.88
Longformer	35.63	62.85	56.89	42.22	69.71	FAIL	53.46
BigBird	36.05	64.02	59.29	40.83	74.87	FAIL	55.01
Cosformer	**37.90**	63.41	61.36	43.17	70.33	FAIL	55.23
Sliceformer	37.65	64.60	**62.23**	**48.02**	**82.04**	FAIL	**58.91**

Sliceformer: Superiority on LRA



Sliceformer: Generalizability on Other Tasks

Image Classification: Model size ($\times 10^6$) and Top-1 accuracy (%)

Data	Dogs vs. Cats		MNIST		CIFAR-10		CIFAR-100		Tiny-ImageNet	
Metric	Size	Top-1	Size	Top-1	Size	Top-1	Size	Top-1	Size	Top-1
ViT	1.90	79.03	9.60	98.78	9.60	80.98	9.65	53.99	22.05	**52.74**
Sliceformer	**1.11**	**79.71**	**6.50**	**98.81**	**6.46**	**82.16**	**6.50**	**54.24**	**18.50**	51.77

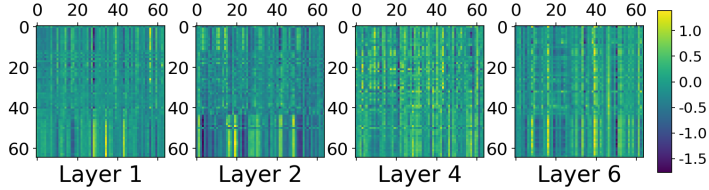
Text Classification:

Data	IMDB	
Metric	Size	Top-1
Transformer	8.84	83.05
Sliceformer	**8.05**	**84.91**

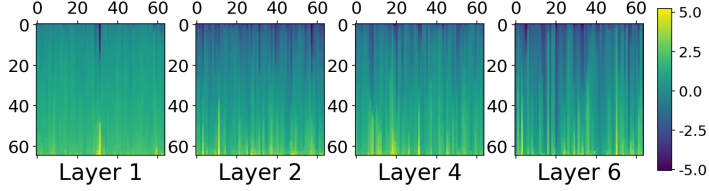
Molecular Property Prediction:

Data	PCQM4M-LSC	
Metric	Size	MAE
Graphormer	47.09	**0.1287**
Sliceformer	**32.91**	0.1308

Empirical Rationality of Sliceformer: Suppressing Mode Collapse



(a) ViT



(b) Sliceformer

Figure: $\mathrm{MHA}_\theta(\boldsymbol{X})$ of ViT v.s. $\mathrm{SliceSort}(\boldsymbol{X})$ of Sliceformer.

Empirical Rationality of Sliceformer: Suppressing Mode Collapse

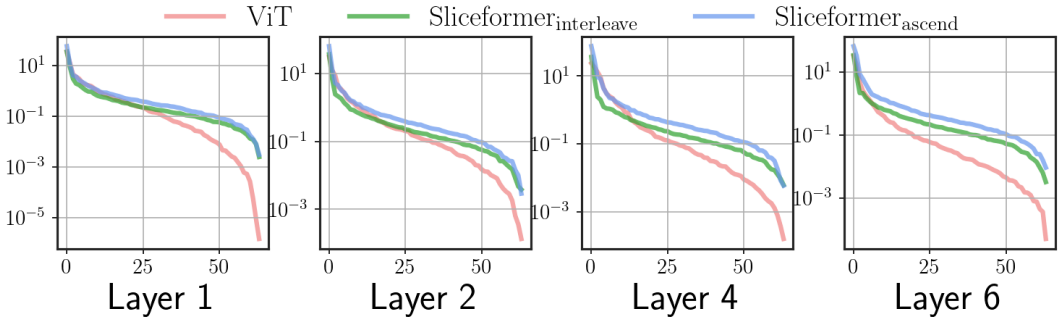


Figure: The comparisons on the singular spectrum.

Summary

Study the implicit design of information fusion layers and propose OT-driven solutions:

Layer	Design	OT Problem
Global Pooling	$(\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}(\theta)\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}}(\theta))\mathbf{1}_N$	ROTP
Message-Passing	$g_{\mathcal{V}}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}}\boldsymbol{F}^*$	W_1 on graph
MHA Layer	$\text{Concat}_{\text{col}}(\{\boldsymbol{P}_i\boldsymbol{X}\boldsymbol{w}_i\}_{i=1}^{MD})$	1D WB

Summary

Study the implicit design of information fusion layers and propose OT-driven solutions:

Layer	Design	OT Problem
Global Pooling	$(\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}(\theta)\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}}(\theta))\mathbf{1}_N$	ROTP
Message-Passing	$g_{\mathcal{V}}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}}\boldsymbol{F}^*$	W_1 on graph
MHA Layer	$\text{Concat}_{\text{col}}(\{\boldsymbol{P}_i\boldsymbol{X}\boldsymbol{w}_i\}_{i=1}^{MD})$	1D WB

Their common advantages:

- Simplify the design, even computation
- Feed-forward computation with better interprebility
- Advantages on data fitting

Summary

Study the implicit design of information fusion layers and propose OT-driven solutions:

Layer	Design	OT Problem
Global Pooling	$(\boldsymbol{X} \odot \text{diag}^{-1}(\boldsymbol{P}^*_{\text{rot}}(\theta)\mathbf{1}_N)\boldsymbol{P}^*_{\text{rot}}(\theta))\mathbf{1}_N$	ROTP
Message-Passing	$g_{\mathcal{V}}(\boldsymbol{X}, \boldsymbol{A}; \theta) + \boldsymbol{S}_{\mathcal{V}}\boldsymbol{F}^*$	W_1 on graph
MHA Layer	$\text{Concat}_{\text{col}}(\{\boldsymbol{P}_i\boldsymbol{X}\boldsymbol{w}_i\}_{i=1}^{MD})$	1D WB

Their common advantages:

- Simplify the design, even computation
- Feed-forward computation with better interprebility
- Advantages on data fitting

Current limitations:

- Over-fitting when distribution shifts
- The potentials in generative tasks are not investigated
- Some designs have limitations on model capacity (e.g., The Top-1 Acc. of Sliceformer on full-sized ImageNet is $< 70\%$.)

References and Resources

ROTP

- Paper: https://ieeexplore.ieee.org/abstract/document/10247589/
- Code: https://github.com/SDS-Lab/ROT-Pooling

QW Loss

- Paper: https://arxiv.org/abs/2310.11762

Sliceformer

- Paper: https://arxiv.org/abs/2310.17683
- Code: https://github.com/SDS-Lab/sliceformer

More OT-driven Work

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

Thank you!

https://hongtengxu.github.io

https://github.com/HongtengXu

hongtengxu@ruc.edu.cn