GOT: An Optimal Transport framework for Graph comparison

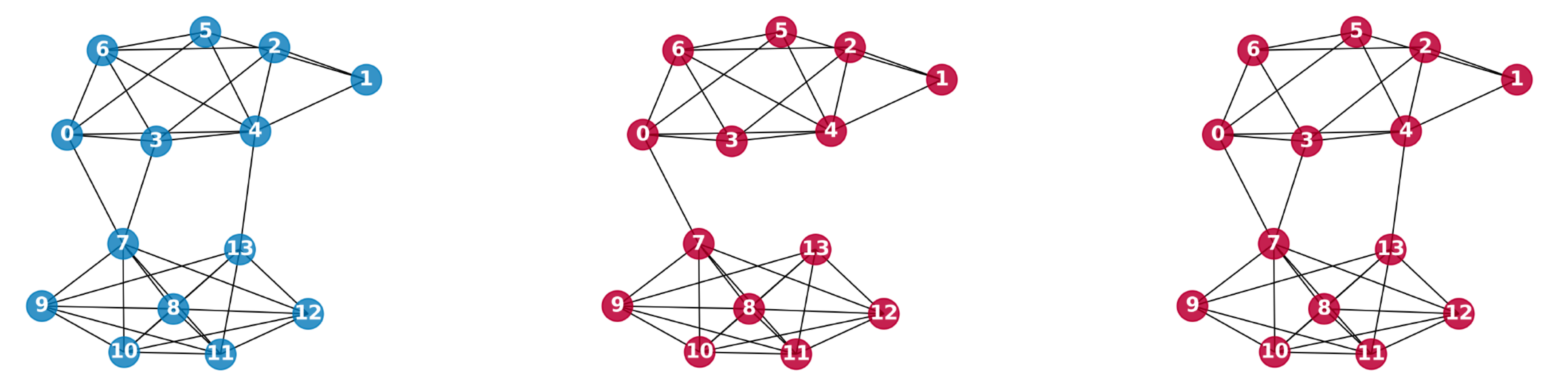
Hermina Petric Maretic*, Mireille EL Gheche*, Giovanni Chierchia** and Pascal Frossard*
*Ecole Polytechnique Fédérale de Lausanne, **Université Paris-Est

Summary

Problem		Solution
Graph representation	→	Modelling graphs through smooth signal distributions
Graph distance	→	Optimal transport: explicit expression of the Wasserstein distance between graphs
Graph signal prediction	→	Optimal transport: explicit signal transportation plan predicts signal behavior on another graph
Graph alignment	→	Novel graph alignment algorithm based on the new distance

Structurally meaningful graph distance

- GOT distance takes into account the entire graph structure



(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$: $\|L_1 - L_2\|_F = 2.828$, $\mathcal{W}_2^2(\nu^{\mathcal{G}_1}, \mu^{\mathcal{G}_2}) = 0.912$ (c) $\mathcal{G}_3$: $\|L_1 - L_3\|_F = 2.828$, $\mathcal{W}_2^2(\nu^{\mathcal{G}_1}, \mu^{\mathcal{G}_3}) = 0.013$

Graph signals and optimal transport

- Each graph defines a unique probability distribution of smooth signals [1]

$$\nu^{\mathcal{G}_1} = \mathcal{N}(0, L_1^\dagger) \quad \mu^{\mathcal{G}_2} = \mathcal{N}(0, L_2^\dagger)$$

- Explicit expression for the graph Wasserstein distance

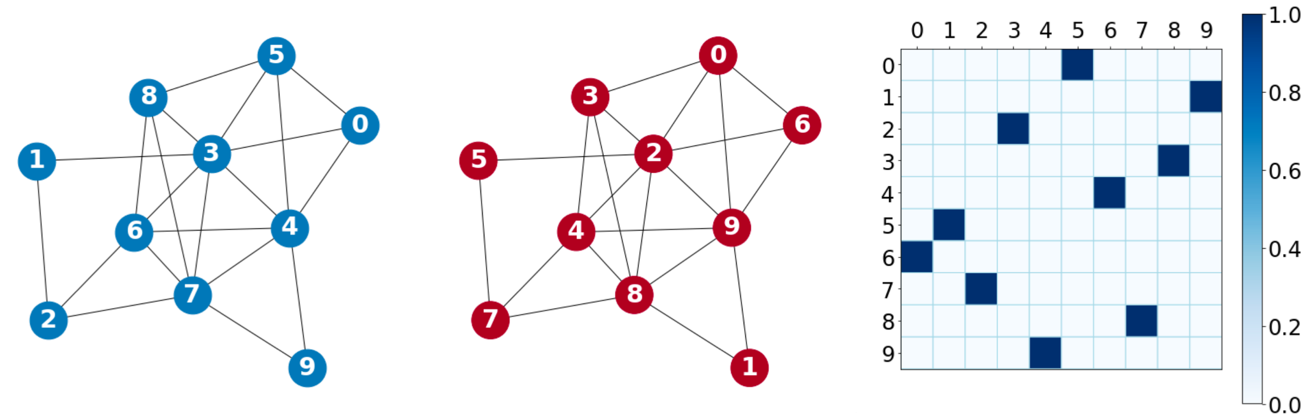
$$W_2^2(\nu^{\mathcal{G}_1}, \mu^{\mathcal{G}_2}) = \mathrm{Tr}\left(L_1^\dagger + L_2^\dagger\right) - 2\,\mathrm{Tr}\left(\sqrt{L_1^{\frac{\dagger}{2}} L_2^\dagger L_1^{\frac{\dagger}{2}}}\right)$$

- Explicit transport plan expression for signal x

$$T(x) = L_1^{\frac{\dagger}{2}} \left(L_1^{\frac{\dagger}{2}} L_2^\dagger L_1^{\frac{\dagger}{2}}\right)^{\frac{\dagger}{2}} L_1^{\frac{\dagger}{2}} x$$

Graph alignment & GOT Algorithm

- Alignment problem – what if we enumerate the nodes differently?



$$\mu^{\mathcal{G}_1} = \mathcal{N}(0, L_1^\dagger) \neq \mathcal{N}(0, (P^\top L_1 P)^\dagger) = \mu^{\mathcal{G}_2}$$

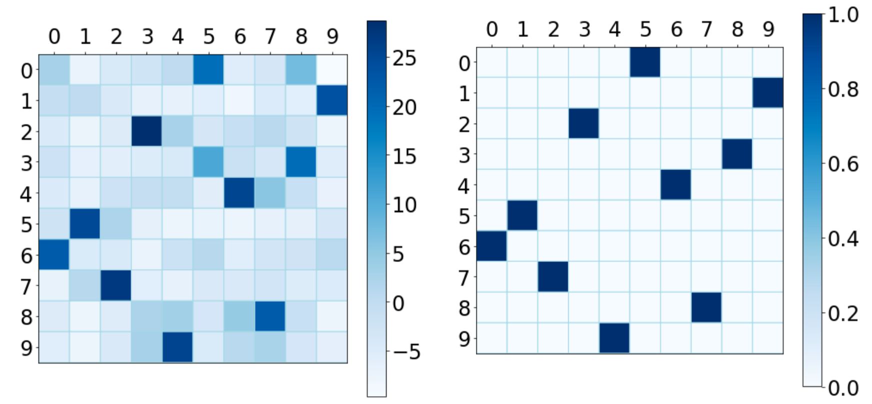
- The Wasserstein distance becomes

$$\mathcal{W}_2^2(\nu^{\mathcal{G}_1}, \mu_P^{\mathcal{G}_2}) = \mathrm{Tr}\left(L_1^\dagger + P^T L_2^\dagger P\right) - 2\,\mathrm{Tr}\left(\sqrt{L_1^{\frac{\dagger}{2}} P^T L_2^\dagger P L_1^{\frac{\dagger}{2}}}\right)$$

- Finding the right alignment

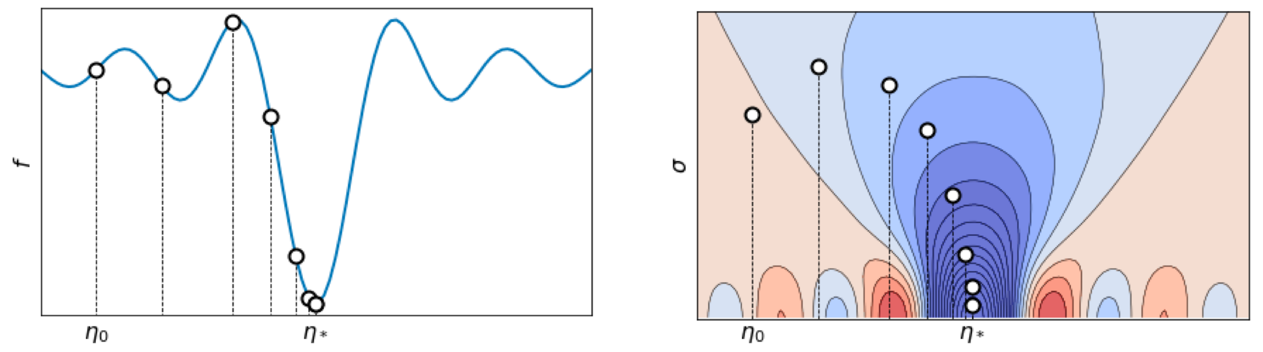
$$\underset{P \in R^{N\times N}}{\text{minimize}} \; \mathcal{W}_2^2(\nu^{\mathcal{G}_1}, \mu_P^{\mathcal{G}_2}) \quad \text{s.t.} \quad \begin{cases} P \in [0,1]^N \\ P\mathbb{1}_N = \mathbb{1}_N \\ \mathbb{1}_N^\top P = \mathbb{1}_N \\ P^\top P = \mathrm{I}_{N\times N} \end{cases}$$

- Implicit projections with the Sinkhorn operator



$$\mathcal{S}_\tau(P) = A \exp(P/\tau) B$$

- Stochastic exploration



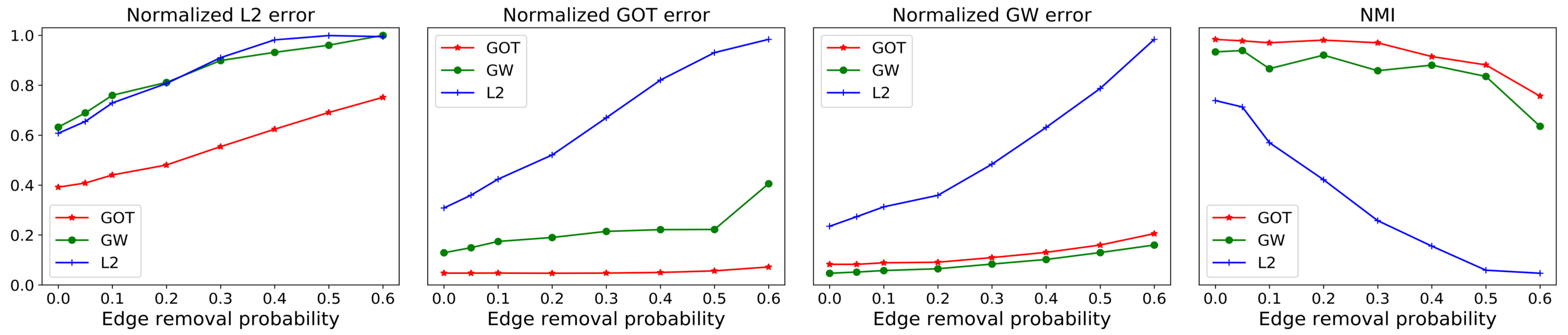
(a) Plot of $f(t) = -\mathrm{sinc}(t)$ (b) Contours of $J(\eta,\sigma) = \mathbb{E}_{t\sim\mathcal{N}(\eta,\sigma^2)}\{f(t)\}$

$$\underset{\theta}{\text{minimize}} \; \mathbb{E}_{P\sim q_\theta}\left\{\mathcal{W}_2^2\left(\nu^{\mathcal{G}_1}, \mu_{\mathcal{S}_\tau(P)}^{\mathcal{G}_2}\right)\right\}$$

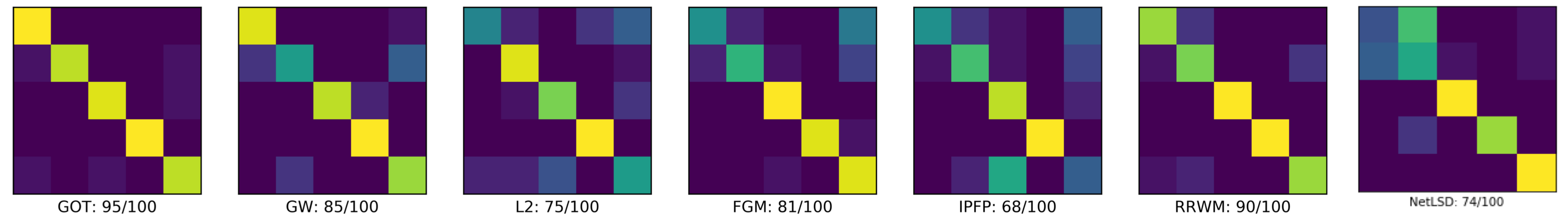
$$\theta = (\eta, \sigma) \in R^{N\times N} \times R^{N\times N} \qquad q_\theta = \prod_{i,j} \mathcal{N}(\eta_{ij}, \sigma_{ij}^2)$$

Results

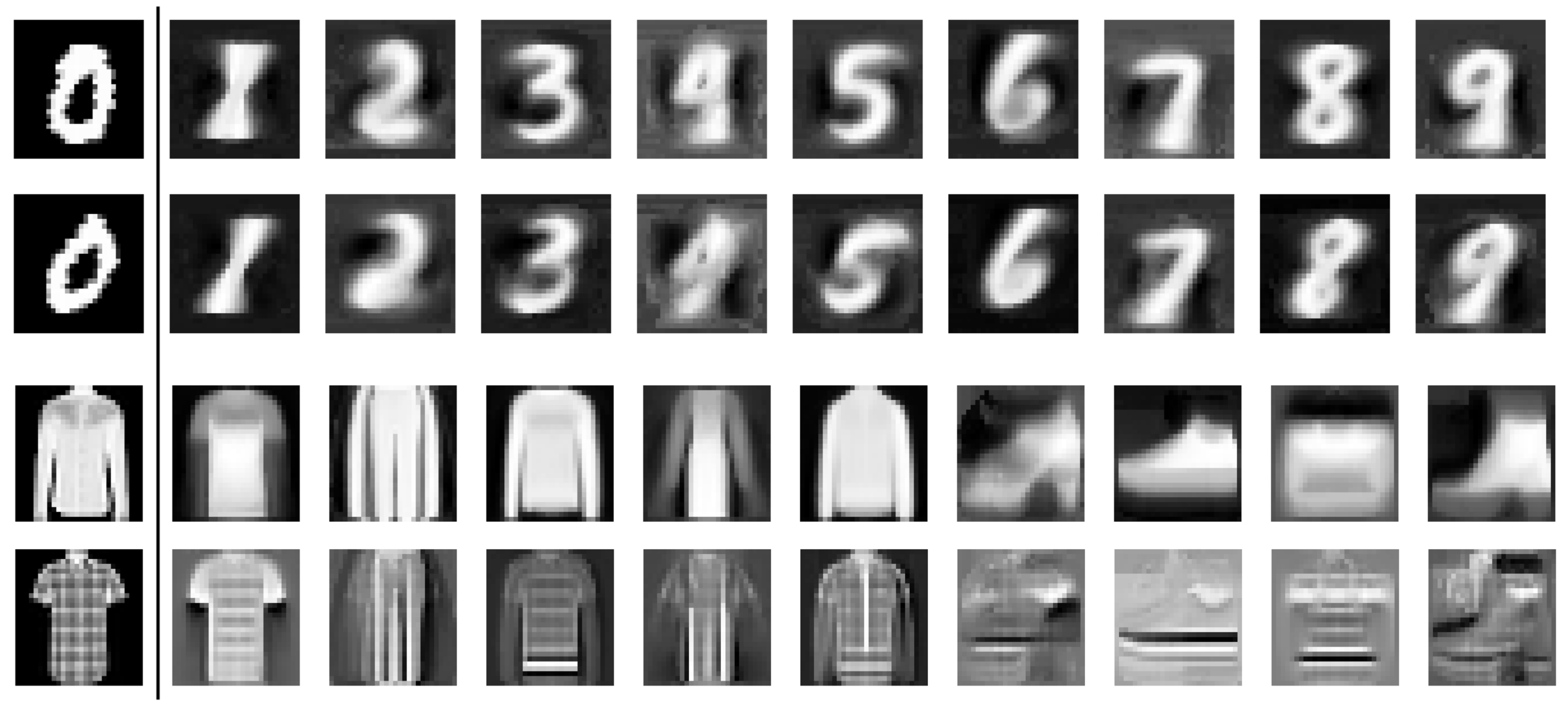
- Alignment and community detection in structured graphs
  - Influence of distance definition on alignment recovery: L2, GOT and GW [2, 3]
  - Alignment of graphs to their perturbed copies with removed edges: within communities (p = 0.5), between communities (p = [0, 0.6])
  - Comparison in terms of all three distances and community recovery NMI



- Graph classification
  - Instances of 5 different random graph models (SBM2, SBM3, RG, BA, WS)
  - Confusion matrices for different methods (GOT, GW [2,3], L2, FGM [4], IPFP [5], RRWM [6] and NetLSD [7])



- Graph signal transportation
  - For each class of images, we construct a K-NN graph between pixels (784 nodes)
  - Each image is a signal on the corresponding graph
  - Images generated as a transportation of a signal from the source graph (0) to the target graph (1…9)



[1] X. Dong et al. Learning laplacian matrix in smooth graph signal representations, IEEE Transactions on Signal Processing, 2016.
[2] G. Peyré et al. Gromov-wasserstein averaging of kernel and distance matrices, International Conference on Machine Learning, 2016.
[3] T. Vayer et al. Optimal Transport for structured data with application on graphs. International Conference on Machine Learning. 2019.
[4] F. Zhou et al. Deformable graph matching, In IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[5] M. Leordeanu et al. An integer projected fixed point method for graph matching and map inference, Advances in Neural Information Processing Systems, 2009.
[6] M. Cho et al. Reweighted random walks for graph matching, European conference on Computer vision, 2010.
[7] A. Tsitsulin et al. Netlsd: hearing the shape of a graph, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.