Semi-relaxed Gromov-Wasserstein divergence for graphs classification

Cédric VINCENT-CUAZ[1], Rémi FLAMARY[2], Marco CORNELI[1], Titouan VAYER[3], Nicolas COURTY[4]

[1] Université Côte d'Azur, Inria, Maasai, CNRS, LJAD

[2] Institut Polytechnique de Paris, CMAP, UMR 7641

[3]Université de Lyon, Inria, CNRS, ENS de Lyon, LIP UMR 5668

[4]Université Bretagne-Sud, CNRS, IRISA

cedric.vincent-cuaz@inria.fr, remi.flamary@polytechnique.edu,
marco.corneli@inria.fr,titouan.vayer@inria.fr,nicolas.courty@irisa.fr

Résumé – La comparaison d'objets structurés tels que les graphes est une opération fondamentale pour de nombreuses tâches d'apprentissage. À cette fin, la distance de Gromov-Wasserstein (GW) , basée sur le Transport Optimal (TO), s'est avérée efficace pour comparer de telles entités. GW opère sur les graphes, vus comme des mesures de probabilité sur des espaces décrits par les relations de connectivité de leurs noeuds. Au coeur du TO réside l'idée de conservation de masse, qui impose un couplage entre tous les noeuds des deux graphes considérés. Nous soutenons dans ce papier que cette propriété peut être préjudiciable pour des tâches telles que l'apprentissage de dictionnaire (AD), et nous la relaxons donc en proposant une nouvelle divergence issue de GW. Cette dernière amène des avantages computationnels immédiats et induit naturellement une nouvelle méthode d'AD, pertinente pour l'apprentissage non supervisé de représentations et la classification de graphes.

Abstract – Comparing structured objects such as graphs is a fundamental operation involved in many learning tasks. To this end, the Gromov-Wasserstein (GW) distance, based on Optimal Transport (OT), has been successful in providing meaningful comparison between such entities. GW operates on graphs, seen as probability measures over spaces depicted by their nodes connectivity relations. At the core of OT is the idea of mass conservation, which imposes a coupling between all the nodes from the two considered graphs. We argue in this paper that this property can be detrimental for tasks such as graph dictionary learning (DL), and we relax it by proposing a new semi-relaxed Gromov-Wasserstein divergence. The latter leads to immediate computational benefits and naturally induces a new graph DL method, shown to be relevant for unsupervised representation learning and classification of graphs.

1 Introduction

Learning from datasets containing non-vectorial objects such as graphs is a difficult task that involves many areas of data analysis such as signal processing [4] or more recently graph neural networks (GNN) [11]. Recently a novel way to model graphs has been proposed based on Optimal Transport (OT). These OT methods either consist in embedding the graphs in a space endowed with Wasserstein geometry [5], or rely on the Gromov-Wasserstein (GW) distance [3, 7]. The latter aims at comparing probability distributions whose supports lie on *different* metric spaces, by finding a matching of these distributions being as close as possible to an isometry. Limitations of GW based approaches include both the computational complexity [NP-hard non convex quadratic program 6] and the need to choose a probability mass function over the graph nodes, leading to suboptimal choices. In order to address these drawbacks, we introduce a new OT based divergence between graphs called the **semi-relaxed Gromov-Wasserstein (**srGW**)** divergence, faster to optimize (see Section 2.1) and providing a new dictionary learning method with discriminant denoising properties useful for graphs classification.

2 Semi-relaxed Gromov-Wasserstein

2.1 Gromov-Wasserstein distance

We model a graph $\mathcal{G}$ with n nodes as a couple $(\boldsymbol{C}, \boldsymbol{h})$ where $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is a matrix encoding the relation between nodes (*e.g.* adjacency) and $\boldsymbol{h} \in \Sigma_n$ in the probability simplex with N-bins, refers to a distribution modeling their relative importance within the graph (*e.g.* uniform or normalized degrees). Then given two observed graphs $\mathcal{G} = (\boldsymbol{C}, \boldsymbol{h})$ and $\overline{\mathcal{G}} = (\overline{\boldsymbol{C}}, \overline{\boldsymbol{h}})$, of respective orders n and m ($n \neq m$), The GW distance between $\mathcal{G}$ and $\overline{\mathcal{G}}$ is defined as :

$$\mathrm{GW}_2^2(\boldsymbol{C}, \boldsymbol{h}, \overline{\boldsymbol{C}}, \overline{\boldsymbol{h}}) = \min_{\substack{\boldsymbol{T}\mathbf{1}_m = \boldsymbol{h} \\ \boldsymbol{T}^\top \mathbf{1}_n = \overline{\boldsymbol{h}}}} \sum_{ijkl} \left| C_{ij} - \overline{C}_{kl} \right|^2 T_{ik} T_{jl} \quad (1)$$

with $\boldsymbol{T} \in \mathbb{R}_+^{n\times m}$, a coupling. The optimal coupling $\boldsymbol{T}^\star$ acts as a probabilistic matching of nodes which tends to associate pairs of nodes that have similar pairwise relations in $\boldsymbol{C}$ and $\overline{\boldsymbol{C}}$ respectively, while preserving masses $\boldsymbol{h}$ and $\overline{\boldsymbol{h}}$ through its marginals. GW defines a distance between graphs, invariant to measure preserving isometries [3], such as nodes permutation.

GW has also been extended to graphs with node attributes $(\boldsymbol{C}, \boldsymbol{F}, \boldsymbol{h})$, where $\boldsymbol{F} \in \mathbb{R}^{n\times d}$ is a matrix of node features, thanks

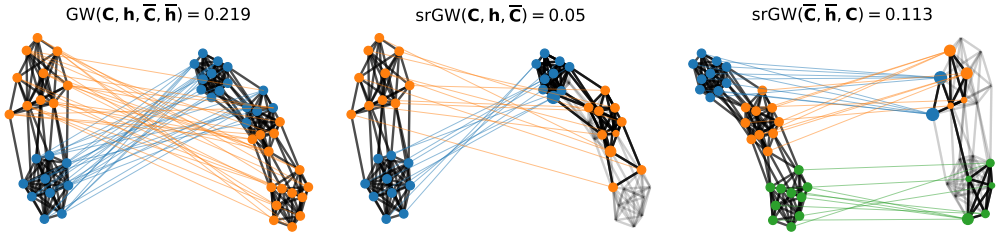

FIGURE 1 – Comparison of the GW matching (left) and asymmetric srGW matchings (middle and right) between graphs $\boldsymbol{C}$ and $\overline{\boldsymbol{C}}$ with uniform distributions. Nodes of the source graph are colored based on their clusters. The OT from the source to the target nodes is represented by arcs colored depending on the corresponding source node color. The nodes in the target graph are colored by averaging the (rgb) color of the source nodes, weighted by the entries of the OT plan.

to the Fused Gromov-Wasserstein distance (FGW) [8]. FGW between such two graphs looks for an OT minimizing a weighted mean of parameter $\alpha \in [0, 1]$, between a GW cost on structures and a linear OT cost on features. Most applications of GW can be extended with FGW to attributed graphs.

2.2 Semi-relaxed Gromov-Wasserstein divergence

We argue that enforcing unknown distributions over the source $\mathcal{G}$ and target $\overline{\mathcal{G}}$, as GW does, can be suboptimal in several cases. To this end we propose to find a correspondence between them while optimizing the weights $\overline{\boldsymbol{h}}$ on the second graph. Thus we introduce the *semi-relaxed Gromov-Wasserstein divergence* expressed as :

$$\mathrm{srGW}_2^2(\boldsymbol{C}, \boldsymbol{h}, \overline{\boldsymbol{C}}) = \min_{\overline{\boldsymbol{h}} \in \Sigma_m} \mathrm{GW}_2^2(\boldsymbol{C}, \boldsymbol{h}, \overline{\boldsymbol{C}}, \overline{\boldsymbol{h}}) \qquad (2)$$

This means that we search for a reweighing of the nodes of $\overline{\mathcal{G}}$ leading to a graph with structure $\overline{\boldsymbol{C}}$ with minimal GW distance from $\mathcal{G}$ [3, 7]. Explicitly, the problem (2) reads :

$$\mathrm{srGW}_2^2(\boldsymbol{C}, \boldsymbol{h}, \overline{\boldsymbol{C}}) = \min_{\boldsymbol{T}\mathbf{1}_m = \boldsymbol{h}, \boldsymbol{T} \geq 0} \sum_{ijkl} |C_{ij} - \overline{C}_{kl}|^2 T_{ik} T_{jl} \quad (3)$$

with $\boldsymbol{T} \in \mathbb{R}_+^{n \times m}$. From an optimal $\boldsymbol{T}^\star$ of problem (3), the optimal weights $\overline{\boldsymbol{h}}^\star$ expressed in problem (2) can be recovered by computing $\boldsymbol{T}^\star$'s second marginal, *i.e* $\overline{\boldsymbol{h}}^\star = \boldsymbol{T}^{\star\top} \mathbf{1}_n$.

A first interesting property of srGW is that since $\overline{\boldsymbol{h}}$ is optimized in the simplex Σ_m, its optimal value $\overline{\boldsymbol{h}}^\star$ can be sparse. As a consequence, parts of the graph $\overline{\mathcal{G}}$ can be omitted in the comparison. This behavior is illustrated in the Figure 1, where two graphs with uniform distributions and structures $\boldsymbol{C}$ and $\overline{\boldsymbol{C}}$ forming respectively 2 and 3 clusters are matched. The GW matching (left) between both graphs forces nodes of different clusters from $\boldsymbol{C}$ to be transported on one of the three clusters of $\overline{\boldsymbol{C}}$, leading to a high GW cost where clusters are not preserved. Whereas srGW provides a reasonable approximation of the structure of the left graph by finding a subgraph within the target structure, forming a graph with as many clusters than the left graph. For a deeper inspection of the theoretical properties of this divergence, the reader is referred to [10].

2.3 Optimization and algorithms

The optimization problem in equation 3 is a non-convex quadratic program similar to the one of GW with the important difference that the linear constraints are independent. Consequently, we propose to solve (3) with a Conditional Gradient (CG) algorithm. This algorithm, provided in Alg. 1, consists in solving at each iteration (t) a linearization $\langle \boldsymbol{X}, \boldsymbol{G} \rangle$ of the problem (3) where $\boldsymbol{G}$ is the gradient of the objective in (3). The solution of the linearized problem provides a *descent direction* $\boldsymbol{X}^\star - \boldsymbol{T}$, and a linesearch whose optimal step can be found in closed form to update the current solution $\boldsymbol{T}$ [8]. The main source of efficiency of our algorithm comes from the computation of the descent directions. In the GW case, one needs to solve an exact linear OT problem, while in our case, one just needs to independently find the minimum on the rows of $\boldsymbol{G}$, within $O(mn)$ operations, significantly reducing the computing time [10].

Algorithm 1 CG solver for srGW

1: **repeat**
2: $\quad \boldsymbol{G}^{(t)} \leftarrow$ Compute gradient w.r.t $\boldsymbol{T}$ of (2) applied at $\boldsymbol{T}^{(t)}$.
3: $\quad \boldsymbol{X}^{(t)} \leftarrow \min_{\boldsymbol{X}\mathbf{1}_m = \boldsymbol{h}} \langle \boldsymbol{X}, \boldsymbol{G}^{(t)} \rangle$ with $\boldsymbol{X} \geq 0$.
4: $\quad \boldsymbol{T}^{(t+1)} \leftarrow (1 - \gamma^\star)\boldsymbol{T}^{(t)} + \gamma^\star \boldsymbol{X}^{(t)}$ with $\gamma^\star \in [0, 1]$ from exact-line search.
5: **until** convergence.

As illustrated in Figure 1, srGW naturally leads to sparse solutions in $\overline{\boldsymbol{h}}$. To compress even more the localization over a few nodes of $\overline{\boldsymbol{C}}$, we can promote the sparsity of $\overline{\boldsymbol{h}}$ which is equivalent to promoting the group-sparsity of the couplings at the column level. To this end, we propose to add a penalization $\Omega(\boldsymbol{T}) = \sum_j (\sum_i T_{ij})^{1/2} = \sum_j \overline{h}_j^{1/2}$ to the problem 3 leading to :

$$\min_{\boldsymbol{T}\mathbf{1}_m = \boldsymbol{h}, \boldsymbol{T} \geq 0} \sum_{ijkl} |C_{ij} - \overline{C}_{kl}|^2 T_{ik} T_{jl} + \lambda \Omega(\boldsymbol{T}) \qquad (4)$$

where $\lambda \in \mathbb{R}_+^*$. The resulting minimal value will be referred as $\mathrm{srGW}_{g,2}^2(\boldsymbol{C}, \boldsymbol{h}, \overline{\boldsymbol{C}}; \lambda)$. As Ω defines a concave function on the positive orthant $\mathbb{R}_+$, we propose to solve for this problem by using Alg 1 within the Majorisation-Minimisation framework described in [1], without changing the overall complexity.

3 Learning the target structure

A dataset of K graphs $\mathcal{D} = \{(\boldsymbol{C}_k, \boldsymbol{h}_k)\}_{k \in [\![K]\!]}$ is now considered, with heterogeneous structures and a variable number of nodes, denoted by $\{n_k\}_{k \in [\![K]\!]}$. We propose to learn the graph

Algorithm 2 Stochastic update of the dictionary atom $\overline{\boldsymbol{C}}$

1: Sample a minibatch of graphs $\mathcal{B} := \{(\boldsymbol{C}^{(k)}, \boldsymbol{h}^{(k)})\}_k$.
2: Get OT $\{\boldsymbol{T}_k^\star\}_{k\in\mathcal{B}}$ from $\mathrm{srGW}(\boldsymbol{C}_k, \boldsymbol{h}_k, \overline{\boldsymbol{C}})$ with Alg.1.
3: Get gradient $\widetilde{\nabla}_{\overline{\boldsymbol{C}}}$ of srGW with fixed $\{\boldsymbol{T}_k^\star\}_{k\in\mathcal{B}}$ and (optionally) perform a projected gradient step on chosen set $\mathcal{S}$:

$$\overline{\boldsymbol{C}} \leftarrow \mathrm{Proj}_{\mathcal{S}}(\overline{\boldsymbol{C}} - \eta\widetilde{\nabla}_{\overline{\boldsymbol{C}}}) \tag{5}$$

dictionary $\overline{\boldsymbol{C}} \in \mathbb{R}^{m\times m}$ from the observed data, by optimizing :

$$\min_{\overline{\boldsymbol{C}}\in\mathbb{R}^{m\times m}} \frac{1}{K}\sum_{k=1}^{K} \mathrm{srGW}_2^2(\boldsymbol{C}_k, \boldsymbol{h}_k, \overline{\boldsymbol{C}}). \tag{6}$$

This problem is denoted as **srGW Dictionary Learning**. It can be seen as a srGW barycenter problem [6] where we look for a graph structure $\overline{\boldsymbol{C}}$ for which there exists node weights $(\overline{\boldsymbol{h}}_k^\star)_{k\in[\![K]\!]}$ leading to a minimal GW error. Then the embedded graph $(\overline{\boldsymbol{C}}, \overline{\boldsymbol{h}}_k^*)$ comes down to a projection of the input graph $(\boldsymbol{C}_k, \boldsymbol{h}_k)$ in the GW sense (by minimizing the weights $\overline{\boldsymbol{h}}$ in srGW) and the optimal weights $\overline{\boldsymbol{h}}_k^*$ relates to the embedding of this input. Interestingly this DL model requires only to solve the srGW problem to compute the embedding $\overline{\boldsymbol{h}}_k^\star$ of $(\boldsymbol{C}_k, \boldsymbol{h}_k)$, since it can be recovered from the solution $\boldsymbol{T}_k^\star$ of the problem 3, with $\overline{\boldsymbol{h}}_k^\star = \boldsymbol{T}_k^{\star\top}\mathbf{1}_{n_k}$.

We solve the non-convex optimization problem 6 with an online algorithm similar to the one first proposed in [2] for vectorial data and adapted by [9] for graph data. The core of the stochastic algorithm is depicted in Algorithm 2. Since the embedding $\overline{\boldsymbol{h}}_k^\star$ is a by-product of computing the different srGW, we do not need an iterative solver to estimate it. Consequently, it leads to a speed up on CPU of 2 to 3 orders of magnitude compared to our main competitors (see Section 4) whose DL methods, instead, require such iterative scheme.

4 Numerical experiments

We first illustrate the behavior of our srGW DL on the dataset of social networks IMDB-B in Figure 2, for a learned dictionary $\overline{\boldsymbol{C}}$ of 10 nodes. The projection of the left graph onto the dictionary results in a subgraph with 4 nodes, which highlight key components within the input graph, *i.e* 3 clusters of variable proportions and a central node.

To further emphasize the relevance of our factorizations, we benchmarked our embeddings on the task of graphs classification considering three types of datasets : i) social networks from IMDB-B and IMDB-M ; ii) graphs with discrete features representing chemical compounds from MUTAG and cuneiform signs from PTC-MR ; iii) graphs with continuous features, namely BZR, COX2, PROTEINS and ENZYMES. We learn our dictionaries on each dataset validating their shapes $M \in \{10, 20, ..., 50\}$, the vanilla srGW is distinguished from its regularized version srGW_g whose additional parameter is validated within $\{1., 0.1, 0.01, 0.001\}$. We benchmark our embeddings with those produced by the SOTA GW based DL

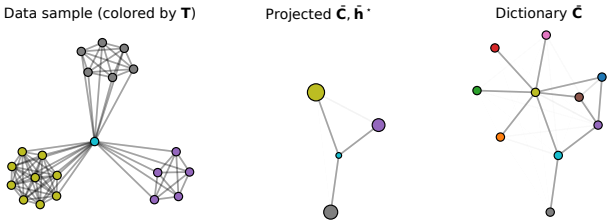

FIGURE 2 – Illustration of the embedding on a sample $(\boldsymbol{C}_k, \boldsymbol{h}_k)$ from the IMDB-B dataset on the estimated dictionary $\overline{\boldsymbol{C}}$. We show the input graph with nodes colored using correspondences from the srGW OT plan (left), the embedded graph $(\overline{\boldsymbol{C}}, \overline{\boldsymbol{h}}_k^*)$ (center) and $\overline{\boldsymbol{C}}$ with uniform mass (right).

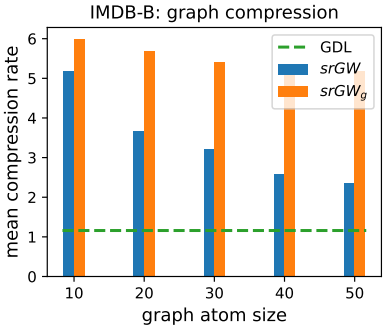

FIGURE 3 – Averaged ratios of input and corresponding embedded graph sizes using benchmarked DL methods. For srGW_g we fixed here $\lambda = 0.01$.

method GDL [9] and its regularized version GDL_λ. All dictionaries are learned over 100 epochs with the same learning rate 0.01 and batch size 32, using Adam optimizer.

We analyse if both *unsupervised* DL methods, produce representations which help to discriminate between graphs. As their resulting graph subspace is endowed with the GW geometry, we perform classification as a downstream task, using SVM with GW kernels computed on embedded graph, *e.g.* $\{(\overline{\boldsymbol{C}}, \overline{\boldsymbol{h}}_k^*)\}_k$ for srGW DL. For all experiments we mimic the benchmark proposed in [9], where FGWK refers to GW kernels between raw input graphs, and other methods are SOTA graph kernels unrelated with OT. We perform 10-fold nested cross-validations repeated over 10 train/test splits, using same folds across methods, and same validated values for SVM's hyperparameters.

Classification performances measured by means of accuracy are reported in Table 1. All variants of srGW DL lead to more discriminant graph representations than those of GDL, while consistently improving performances provided by FGWK opereting on raw graphs represented by their adjacency matrix. Notably srGW_g consistently outperforms all benchmarked methods.

Moreover, our srGW DL naturally leads to embedded graphs of variable resolutions, contrary to GDL, with considerably small number of nodes relatively to their input representations. This behavior illustrated on IMDB-B in the figure 3 helps to drastically reduce the runtimes required to compute the GW pairwise matrices used in SVM, especially while promoting

TABLE 1 – Classification performances on real datasets : We highlight the 1^{st} (resp. 2^{nd}) best method in bold (resp. italic). Unfilled values (-) when methods are specific to certain type of graph features.

		No attribute		Discrete attributes		Real attributes			
Categories	Models	IMDB-B	IMDB-M	MUTAG	PTC-MR	BZR	COX2	ENZYMES	PROTEIN
OT DL	srGW (ours)	*72.1(4.1)*	49.2(3.6)	*89.1(5.9)*	*64.5(7.8)*	*88.0(4.2)*	77.7(2.7)	*72.3(5.7)*	*72.9(5.1)*
	srGW_g (ours)	**73.2(4.3)**	**51.3(3.4)**	**90.3(5.4)**	**64.5(6.9)**	**88.5(3.9)**	**79.8(1.8)**	**73.6(4.3)**	**74.1(4.8)**
	GDL	70.1(3.3)	49.1(4.6)	87.4(5.0)	56.4(6.5)	85.9(4.3)	77.4(3.1)	70.7(3.9)	71.6(3.9)
	GDL_λ	71.5(4.1)	*50.1(4.8)*	88.1(7.8)	59.5(8.4)	86.5(5.4)	*78.1(4.4)*	71.5(4.2)	72.9(5.8)
OT kernel	FGWK	70.8(3.5)	48.9(3.9)	82.6(7.2)	56.2(8.9)	85.6(5.2)	77.0(4.2)	72.2(4.0)	72.4(4.7)
Kernels	SPK	56.2(2.9)	39.1(4.9)	83.3(8.0)	60.6(6.4)	-	-	-	-
	WLK	-	-	86.4(8.0)	63.1(6.6)	-	-	-	-
	HOPPERK	-	-	-	-	84.5(5.2)	79.7(3.5)	46.2(3.8)	72.1(3.1)
	PROPAK	-	-	-	-	80.0(5.1)	77.8(3.8)	71.8(5.8)	61.7(4.5)

TABLE 2 – GW Kernel computation times (in ms) on different graph embeddings and input graphs, averaged over all corresponding pairs of graphs (499500 symmetric pairs in IMDB-B).

Models	Runtimes (ms)	
	min	max
srGW (ours)	4.7	11.1
srGW_g (ours)	1.7	3.7
GDL	19.1	19.9
FGWK	24.7	

sparsity of our embeddings, as reported in Table 2. Therefore, the coupled discriminant denoising abilities and computational efficiency of our methods show that it could even be considered as a pre-processing step for GW based analysis.

5 Conclusion

We introduce a new OT based divergence between structured data by relaxing the mass constraint on the second distribution of the GW problem. After designing efficient solvers to estimate this divergence, called the semi-relaxed Gromov-Wasserstein (srGW), we suggest to learn a unique structure to describe a dataset of graphs in the srGW sense. This novel modeling can be seen as a Dictionary Learning approach where graphs are embedded as a subgraph of a single atom. Numerical experiments highlight the interest of our methods for graph unsupervised representation learning whose evaluation is conducted through classification of graphs.

We believe that this new divergence will unlock the potential of GW for graphs with unbalanced proportions of nodes. The associated fast numerical solvers allow to handle large size graph datasets, which was not possible with current GW solvers. Also, as relaxing the second marginal constraint in the original optimization problem gives more degrees of freedom to the underlying problem, one can expect dedicated regularization schemes, over *e.g.* the level of sparsity of $\overline{\boldsymbol{h}}$, to address a variety of application needs. Finally, our method can be seen as a special relaxation of the subgraph isomorphism problem. It remains to be understood theoretically in which sense this relaxation holds.

Références

[1] N. Courty, R. Flamary, and D. Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer, 2014.

[2] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.

[3] F. Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4) :417–487, 2011.

[4] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing : Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5) :808–828, 2018.

[5] H. Petric Maretic, M. El Gheche, G. Chierchia, and P. Frossard. Got : An optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*, 32 :13876–13887, 2019.

[6] G. Peyré, M. Cuturi, and J. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pages 2664–2672, 2016.

[7] K.-T. Sturm. The space of spaces : curvature bounds and gradient flows on the space of metric measure spaces. *arXiv preprint arXiv :1208.0434*, 2012.

[8] T. Vayer, N. Courty, R. Tavenard, and R. Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.

[9] C. Vincent-Cuaz, T. Vayer, R. Flamary, M. Corneli, and N. Courty. Online graph dictionary learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10564–10574. PMLR, 18–24 Jul 2021.

[10] C. Vincent-Cuaz, R. Flamary, M. Corneli, T. Vayer, and N. Courty. Semi-relaxed gromov-wasserstein divergence and applications on graphs. In *International Conference on Learning Representations*, 2022.

[11] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.