Machine Learning (2021) 110:2151–2186
https://doi.org/10.1007/s10994-021-06035-1

Sampled Gromov Wasserstein

Tanguy Kerdoncuff[1] · Rémi Emonet[1] · Marc Sebban[1]

Received: 29 January 2021 / Revised: 20 May 2021 / Accepted: 1 July 2021 /
Published online: 26 July 2021

Abstract
Optimal Transport (OT) has proven to be a powerful tool to compare probability distributions in machine learning, but dealing with probability measures lying in different spaces remains an open problem. To address this issue, the Gromov Wasserstein distance (GW) only considers intra-distribution pairwise (dis)similarities. However, for two (discrete) distributions with N points, the state of the art solvers have an iterative $O(N^4)$ complexity when using an arbitrary loss function, making most of the real world problems intractable. In this paper, we introduce a new iterative way to approximate GW, called *Sampled Gromov Wasserstein*, which uses the current estimate of the transport plan to guide the sampling of cost matrices. This simple idea, supported by theoretical convergence guarantees, comes with a $O(N^2)$ solver. A special case of Sampled Gromov Wasserstein, which can be seen as the natural extension of the well known Sliced Wasserstein to distributions lying in different spaces, reduces even further the complexity to $O(N \log N)$. Our contributions are supported by experiments on synthetic and real datasets.

Keywords Optimal transport · Gromov Wasserstein · Convergence guarantees

1 Introduction

Optimal Transport (OT) (Villani, 2008) and its associated Wasserstein distance allow the comparison of probability measures by aligning points between the distributions with respect to their masses and transportation costs. Recent advances from a computational perspective, notably with the entropic regularization introduced in (Cuturi, 2013) or the Sliced Wasserstein (Rabin & Peyré, 2011), led to some success stories of OT in the machine learning community, including the Wasserstein Generative Adversarial

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

Tanguy Kerdoncuff
tanguy.kerdoncuff@univ-st-etienne.fr

Rémi Emonet
remi.emonet@univ-st-etienne.fr

Marc Sebban
marc.sebban@univ-st-etienne.fr

[1] 18 Rue Professeur Benoît Lauras Bâtiment F, 42000 Saint-Étienne, France

Networks (Arjovsky et al., 2017), Domain Adaptation (Courty et al., 2014), Color Transfer (Rabin & Peyré, 2011), to cite a few. Even though the square Euclidean distance is used most of the time to compare points of the distributions, various other ground metrics can be naturally used or learned to better capture the idiosyncrasies of the application at hand: the Earth mover's distance in computer vision tasks, the Mahalanobis distance (Paty and Cuturi 2019), or concave functions in economy such as the square root of the Euclidean distance (Delon et al., 2012), etc.

Whatever the cost function, it is worth noting that the OT problem has been originally formulated so as to deal with distributions that are required to lie in the same space. To relax this constraint, a distance between metric spaces, named Gromov Wasserstein (GW),[1] has been introduced in (Memoli, 2007). It takes the form of the generalization of the well-known Quadratic Assignment problem (Beckman & Koopmans, 1957) with any distribution (Mémoli, 2011) and any loss function (Peyré et al., 2016). The intuition is still to align points between two distributions but the method only relies on pairwise distances, in each space separately. This allows notably to take into account the structure of each distribution while being invariant to rotation and translation. Therefore, GW is a relevant tool for matching and partitioning tasks involving graphs (Xu et al., 2019a, Xu et al., 2019b; Vayer et al., 2019a), by allowing *e.g.* to encode some structure like the shortest path between two vertices. GW has been further used in various other domains, such as Heterogeneous Domain Adaptation (Yan et al., 2018), Shape Matching (Mémoli, 2011; Bronstein et al., 2010; Vayer et al., 2019b), Object Modeling with Deep Learning (Ezuz et al., 2017), Generative Adversarial Networks (Bunne et al., 2019). The Wasserstein distance and the GW distance have also been jointly used in (Vayer et al., 2018) leading to the so-called Fused-Gromov Wasserstein distance.

From an algorithmic perspective, most of the previous methods resort to the entropic approximation (EGW) of the original GW formulation introduced in (Peyré et al., 2016) and based on a gradient descent followed by a projection step, both according to the Kullback Leibler (KL) divergence. While a naive implementation of the original GW problem leads to a $O\left(N^4\right)$ complexity, Peyré et al., (2016) further show that one can compute GW in $O\left(N^3\right)$ operations for a certain class of losses. Some other attempts have been recently proposed in the literature to speed-up the GW calculation. Sliced Gromov-Wasserstein (SGW) (Vayer et al., 2019b) takes inspiration from the Sliced Wasserstein distance (Rabin & Peyré, 2011) by projecting each distribution in an 1D line and then solving the 1D Gromov-Wasserstein problem efficiently in $O(N\log(N))$. The Anchor Energy (AE) distance from Sato et al., (2020),[2] is also related to the GW distance but simplifies the problem into N^2 linear sub-problems. The overall time complexity for solving AE is $O\left(N^2\log(N)\right)$. Scalable Gromov-Wasserstein Learning (S-GWL) (Xu et al., 2019a) decomposes recursively the two large probability measures into a set of small pairwise aligned distributions using a common Gromov-Wasserstein barycenter (Peyré et al., 2016). The final transport plan is the aggregation of the result of GW on each small aligned distributions.

In this paper, we aim at overcoming the main algorithmic bottleneck of EGW: the multiplication of a 4D tensor with a 2D matrix, which we interpret as an expectation over matrices. We leverage this interpretation, using sampling to approximate the expectation instead of

[1] By abuse of notation, we will use throughout this paper the term of Gromov Wasserstein *distance*, even if all the properties of an actual metric do not always hold.

[2] Since this work is not published yet and the code is not available, we will omit it in the related work section and in the experimental part.

computing it entirely, reducing the complexity to $O(N^2)$. Unlike SGW and AE which propose simplified distances, we optimize the *original GW distance*. Unlike EGW and S-GWL which have speedups for specific loss functions, we lower the complexity with *any loss function*. We obtain a generic algorithm, called *Sampled Gromov Wasserstein*, supported by theoretical convergence guarantees. We further show that when the number of sampled matrices is 1, the particular 1D case of the OT can be used to compute an update in $O(N\log(N))$. This version, called *Pointwise Gromov Wasserstein*, overcomes most of the limitations of SGW (Vayer et al., 2019b) detailed in Sect. 3, while still being very fast. Our contributions are supported by experiments on synthetic and real datasets. Interestingly, those experiments show evidence that our method outperforms the state of the art when it comes to finding the best compromise between the computation time and the quality of the distance. This behavior takes its origin from (i) the stochastic nature of our method which can reduce the risk to get stuck in local minima and (ii) the fact that the other approaches do not scale well. An experiment on a graph classification task shows that being able to change the loss function for free is of high interest for finding the one that best fits the problem at hand.

This article is organized as follows: Sect. 2 details the notations and the necessary background on GW. Section 3 covers the state of the art approaches for solving the underlying problem. Section 4 presents our *Sampled Gromov Wasserstein* algorithm, derives convergence guarantees for it, and introduces our very fast specialized variant called *Pointwise Gromov Wasserstein*. Experiments are detailed in Sect. 5.

2 Background on GW

In this section, we introduce the Optimal Transport (OT) problem with its associated Wasserstein distance, and the Gromov Wasserstein distance that allows the comparison of distributions lying in different spaces. Let $(\mathcal{X}, \mathcal{C}^{\mathcal{X}})$ be a compact metric space where $\mathcal{X}$ is a set and $\mathcal{C}^{\mathcal{X}}$ its associated metric. Let μ be a distribution with finite p-moment on $(\mathcal{X}, \mathcal{C}^{\mathcal{X}})$. Similarly, $(\mathcal{Y}, \mathcal{C}^{\mathcal{Y}})$ denotes another compact metric space and ν a distribution with finite p-moment on that space. We denote as $\Pi_{\mu\nu}$ the collection of coupling probability measures on $\mathcal{X}\times\mathcal{Y}$ constrained by the marginals μ and ν. $\Pi_{\mu\nu}$ defines the so-called set of admissible transport plans from μ to ν, used to define the OT problem.

Optimal Transport OT consists in finding the best mapping (or coupling or transport plan) between two distributions μ and ν on the same space, *i.e.*, $\mathcal{X} = \mathcal{Y}$ and $\mathcal{C}^{\mathcal{X}} = \mathcal{C}^{\mathcal{Y}}$. Denoting as $\mathcal{C}$ this common distance, one can define the p-Wasserstein distance (Kantorovich 1942) to the power of p, as follows:

$$W_p^p(\mathcal{C}) = \min_{\gamma\in\Pi_{\mu\nu}} \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{C}^p(x,y)d\gamma(x,y). \tag{1}$$

In the discrete version of Problem (1), μ and ν are empirical measures supported by two finite sets of points. In this context, $\mu = \sum_{i=1}^{I} a_i\delta_{x_i}$ defined by I points $(x_i)_{i\in[\![1,I]\!]}$ in $\mathcal{X}$ and the associated probability vector a. In the same way, we define $\nu = \sum_{k=1}^{K} b_k\delta_{y_k}$ in $\mathcal{Y}$ associated with the probability vector b. The set of admissible transport plans becomes $\Pi_{ab} = \{T \in \mathbb{R}_+^{I\times K} | T\mathbf{1}_K = a, T^{\mathrm{T}}\mathbf{1}_I = b\}$. In this discrete case, each distance function $\mathcal{C}$ can be considered as a matrix (or tensor) C. Therefore, the discrete p-Wasserstein distance to the power p is written as follows:

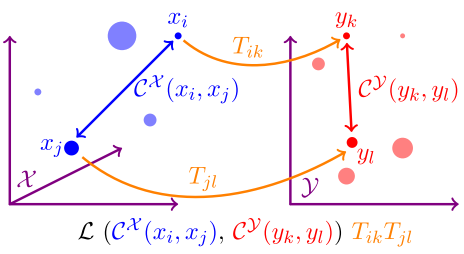
x_i
y_k
T_{ik}
$\mathcal{C}^{\mathcal{X}}(x_i, x_j)$
$\mathcal{C}^{\mathcal{Y}}(y_k, y_l)$
x_j
y_l
T_{jl}
$\mathcal{X}$
$\mathcal{Y}$
$\mathcal{L}\left(\mathcal{C}^{\mathcal{X}}(x_i, x_j), \mathcal{C}^{\mathcal{Y}}(y_k, y_l)\right) T_{ik}T_{jl}$

Fig. 1 Illustration of GW, with only one term L_{ijkl} of the quadruple sum of Eq. (5)

$$W_p^p(\mathcal{C}) = \min_{T\in\Pi_{ab}} \sum_{i,k=1}^{I,K} \mathcal{C}^p(x_i, y_k)T_{ik} = \min_{T\in\Pi_{ab}} \langle C^p, T\rangle \tag{2}$$

where $\langle ., .\rangle$ is the Frobenius dot product. To simplify the notations, it is often assumed that $I = K$ (same number of points in both sets) and N is used to denote this value. The optimal transport plan T^* can be found from (2) using a linear solver (Bonneel et al., 2011) with, at least, a complexity of $O\big(N^3 \log(N)\big)$ (Pele & Werman, 2009). To lower this complexity, an entropic regularization can be added (Cuturi, 2013) leading to a strongly convex problem that yields a smooth and unique solution in $O\big(PN^2\big)$ with P the number of Sinkhorn's iterations. Let $\epsilon \in \mathbb{R}_+$ be a regularization parameter and let $\mathcal{H}(T) = \sum_{ik} T_{ik} \log(T_{ik})$ be the negative entropy, the optimal plan T^* of Eq. (2) can be approximated by

$$T^* \approx \underset{T\in\Pi_{ab}}{\operatorname{argmin}} \langle C^p, T\rangle + \epsilon\mathcal{H}(T). \tag{3}$$

Gromov Wasserstein Distance (GW) While the OT problem requires the two distributions to lie in the same space, the GW distance allows to compare distributions in different metric spaces. Let $\mathcal{L}$ be a bounded loss function which allows the comparison of two distances. GW (Mémoli, 2011, 2009; Peyré et al., 2016) is defined as follows:

$$GW\big(\mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}}, \mu, \nu\big) = \min_{\gamma\in\Pi_{\mu\nu}} \int_{(\mathcal{X}\times\mathcal{Y})^2} \mathcal{L}\big(\mathcal{C}^{\mathcal{X}}(x, x'), \mathcal{C}^{\mathcal{Y}}(y, y')\big) d\gamma(x, y) d\gamma(x', y'). \tag{4}$$

The discrete case (see Fig. 1) can be formulated as:

$$GW\big(\mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}}, a, b\big) = \min_{T\in\Pi_{ab}} \sum_{i,j=1}^{I,I} \sum_{k,l=1}^{K,K} L_{ijkl} T_{ik} T_{jl}, \tag{5}$$

where $L_{ijkl} = \mathcal{L}\big(\mathcal{C}^{\mathcal{X}}(x_i, x_j), \mathcal{C}^{\mathcal{Y}}(y_k, y_l)\big)$. This formulation exhibits an important property of GW: only the pairwise distances are needed. This explains why the Gromov Wasserstein distance is often used to compare graphs, for which Chowdhury and Mémoli (2019) proves that GW is a pseudometric.

3 Approaches to solve GW

We describe here the most used method for solving GW, namely *Entropic Gromov Wasserstein*, as well as two other approaches that aim at lowering the time complexity of the former. As all these methods use an iterative optimization, for the sake of simplicity, we omit in this section the number S of iterations (of the outer loop).

Entropic Gromov Wasserstein (EGW) To solve an approximation of Problem (5), the authors of (Peyré et al., 2016) generalize the idea introduced in Solomon et al., (2016) by using a gradient descent step followed by a projection, both according to the Kullback Leibler (KL) divergence. This boil down to a two-step loop. First, from the current estimation of the transport plan T, a new matrix defined as $\Lambda_{jl} = \sum_{i,k=1}^{I,K} L_{ijkl}T_{ik}$ is computed, and which can be seen as an updated cost matrix. Second, a new estimate of the transport plan is obtained by solving the following entropic regularization-based OT problem:

$$\min_{T\in\Pi_{ab}} \langle \Lambda, T\rangle + \epsilon\mathcal{H}(T). \tag{6}$$

When the loss $\mathcal{L}(\mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}})$ can be decomposed as $f_1(\mathcal{C}^{\mathcal{X}}) + f_2(\mathcal{C}^{\mathcal{Y}}) - h_1(\mathcal{C}^{\mathcal{X}})h_2(\mathcal{C}^{\mathcal{Y}})$ for functions (f_1, f_2, h_1, h_2), it is shown that the Λ matrix can be computed in $O(N^3)$. This notably holds for the square loss and the KL divergence. However, in the general case, the complexity is $O(N^4)$, making this method intractable as N grows, as shown in our experiments.

Sliced Gromov-Wassertein (SGW) In Rabin and Peyré (2011), the authors introduce an alternative metric, called Sliced Wasserstein distance, which uses random 1D-projections. The advantage of this method lies in the fact that the OT Problem (2) can be simply solved by sorting both empirical distributions (in $O(N\log(N))$) and matching the sorted lists. In a similar manner, Sliced Gromov-Wasserstein (SGW) (Vayer et al., 2019b) projects each distribution in a common 1D space, to solve the Gromov-Wasserstein problem (5) efficiently. While being very fast to compute, SGW comes with some limitations: (i) it cannot be used in general on graphs because a feature representation is needed to allow the 1D projection, (ii) it does not output an explicit transport plan which can be a pitfall in some applications like domain adaptation, (iii) it does not approximate the original GW distance and (iv) it is not *naturally* invariant to rotation (although the authors propose a solution by repeatedly calling SGW). Note that while SGW's theoretical result and the $O(N\log(N))$ time complexity are relying on the square loss, its algorithmic approach can be adapted to handle arbitrary losses. This adaptation results in a $O(N^2)$ time complexity.

Scalable GW Learning (S-GWL) Scalable Gromov-Wasserstein Learning (Xu et al., 2019a) aims at making GW tractable to large scale graph analysis. It recursively decomposes the two original graphs into a set of smaller sub-graph pairs, using Gromov-Wasserstein barycenters (Peyré et al., 2016). Then, these sub-graphs are matched. The transport plan is updated with a proximal gradient method regularized with a KL divergence. The time complexity is $O(N^2\log(N))$ when the cost matrices $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$ are not sparse and $\mathcal{L}$ is the square loss. However, with an arbitrary $\mathcal{L}$, the gain in complexity does not hold anymore because S-GWL cannot leverage the closed-form solution for the barycenter calculation.

4 Scalable GW optimization

We aim to address in this section the algorithmic bottleneck of EGW (Peyré et al., 2016) which prevents its use on large scale problems. By rewriting Eq. (5) as an alternating optimization problem, we propose to compute the GW distance by solving iteratively an OT problem from a cost matrix seen as the expectation of a random variable. This allows us to propose a sampling strategy to drastically reduce the algorithmic complexity of GW. We introduce our algorithm, called Sampled Gromov Wasserstein (**SaGroW**), and then derive its convergence guarantees.

We also present some special case and a variant of SaGroW: Pointwise Gromov Wasserstein (**PoGroW**) which leverages very efficient 1D OT solvers but does not exhibit the drawbacks of SGW, and *SaGroW*^{KL} a version using a Kullback-Leibler regularization. We finally show that an appropriate sampling strategy can be also be used to accurately and efficiently approximate the GW distance from a known transport plan.

4.1 Sampled Gromov Wasserstein (SaGroW)

It is known that the GW problem as described in Eq. (5) is not convex in general and thus difficult to solve. On the other hand, we can note that the transport plan T appears twice in the formulation. In the following, we suggest to treat these two instances differently and solve the problem with respect to two transport plan variables T and T', as follows:

$$\min_{T\in\Pi_{ab}} \min_{T'\in\Pi_{ab}} \sum_{i,k=1}^{I,K} \sum_{j,l=1}^{I,K} L_{ijkl} T'_{ik} T_{jl}. \tag{7}$$

Even though our sampling strategy leverages this decomposition into T and T', as if they were two different transport plans, note that we still solve the original GW problem. Indeed, as we will explain, our Algorithm fuses T and T' after each update, fulfilling the $T = T'$ constraint.

In an alternating optimization, with a fixed T, the optimal T' is thus the solution of the following OT problem:

$$\min_{T'\in\Pi_{ab}} \sum_{i,k,j,l=1}^{I,K,I,K} T_{jl} L_{ijkl} T'_{ik} = \min_{T'\in\Pi_{ab}} \left\langle \sum_{j,l=1}^{I,K} T_{jl} L_{\cdot j \cdot l}, T' \right\rangle \tag{8}$$

where $L_{\cdot j \cdot l}$ is an extracted matrix *i.e.*, $\left(L_{\cdot j \cdot l}\right)_{ik} = L_{ijkl}$.

As the transport plan T sums to 1, we can interpret it as (the parameters of) a categorical distribution on pairs of points (j, l), or equivalently on the associated matrices $L_{\cdot j \cdot l}$. We thus define a random variable $\mathbf{C}$ on matrices, defined[3] by the distribution $\mathbb{P}(\mathbf{C} = L_{\cdot j \cdot l}) = T_{jl}\ \forall (j,l) \in [\![1, N]\!]^2$. Leveraging this random variable, the cost matrix $\sum_{j,l} T_{jl} L_{\cdot j \cdot l}$ used in problem (8) can be seen as the expectation of $\mathbf{C}$. Therefore, the problem can be rewritten as follows:

$$\min_{T'\in\Pi_{ab}} \left\langle \mathbb{E}(\mathbf{C}), T' \right\rangle. \tag{9}$$

[3] The definition is not rigorous: two matrices $L_{\cdot j \cdot l}$ and $L_{\cdot j' \cdot l'}$ may be equal, and then the probabilities add up.

While solving this problem is still in $O\left(N^4\right)$ in general, it presents the advantage of opening the door to a sampling strategy allowing a reduction of the complexity. Indeed, rather than computing the entire expectation $\mathbb{E}(\mathbf{C})$, we suggest here to calculate an approximation by sampling M matrices $\{C^m\}_{m=1}^M$. To get a matrix C^m drawn according to the distribution of $\mathbf{C}$, it suffices to sample two indices (j_m, l_m) following the weights of the matrix T. Consequently, C^m takes the form of the matrix $L_{\cdot j_m \cdot l_m}$. Using these sampled matrices, Problem (9) can be approximated as follows:

$$\min_{T' \in \Pi_{ab}} \left\langle \frac{1}{M} \sum_{m=1}^{M} C^m, T' \right\rangle. \tag{10}$$

This approximation comes with two main advantages: (i) it allows a reduction of the computation time of the GW problem and (ii) similarly to a mini batch gradient descent, it might avoid being stuck in local minima and thus might lead to a better transport plan. Even though Problem (10) can be solved efficiently with any OT solver, our approach resorts to the Sinkhorn method (Cuturi, 2013) leading to a time complexity of $O\big((M+P)N^2\big)$ due to summing over M matrices and P iterations of the Sinkhorn algorithm.

Algorithm 1 gives the pseudo-code of Sampled Gromov Wasserstein (**SaGroW**). In the absence of prior, the transport plan T_0 is initialized to the joint distribution ab^T (line 1). At each iteration, M pairs of indices (j_m, l_m) are sampled from the current transport plan T_s (line 3). Then $\widehat{\Lambda}$, the approximation of $\mathbb{E}(\mathbf{C})$, is computed (line 4) and used in an entropic regularization-based OT problem (6) solved using the Sinkhorn algorithm, yielding the plan T_s' (line 5). As indicated before, Problem (7) inherently assumes that $T = T'$. To ensure that T' stays close to T and to mitigate the nature of the process, we perform a partial update $(1-\alpha)T_s + \alpha T_s'$. Given the symmetric roles of T and T' (as long as $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$ are symmetric) this partial update becomes our next plan T_{s+1} (line 6). This update, inspired by the Frank-Wolfe algorithm, allows us to derive theoretical guarantees (see next section). Notice that Algorithm 1 returns a single transport plan and thus aims at minimizing the original GW problem. In practice, other strategies can be used: as the previous plan T_s and the optimized T_s' can be interpreted as distributions, line 6 can be omitted and replaced by a KL regularization (on line 5) between them, as detailed in Sect. 4.4.

We end this section by noting that when the expectation is fully computed in SaGroW (*i.e.*, $M = \infty$ and "$M = N^2$" in terms of complexity as sampling becomes useless) and α is set to 1, our method is strictly equivalent to the two steps loop of EGW described in Sect. 3. This connection will be used advantageously in the next section by deriving new convergence guarantees for EGW when the GW problem is concave.

Algorithm 1 SaGroW

```
Require: $a$, $b$ (probability vectors of $\mu$ and $\nu$), $\mathcal{C}^{\mathcal{X}}$, $\mathcal{C}^{\mathcal{Y}}$ (cost matrices), $\mathcal{L}$ (loss function), $M$
    (number of samples), $\epsilon$ (entropy regularization), $\alpha$ (partial update weight)
1: $T_0 = ab^T$
2: for s= 0 to S-1 do
3:     $(j_m, l_m) \sim \text{Sample}(T_s)\ \forall m \in [\![1, M]\!]$
4:     $\widehat{\Lambda}_{ik} = \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}(C^{\mathcal{X}}_{i,j_m}, C^{\mathcal{Y}}_{k,l_m})\ \forall i, k \in [\![1, N]\!]$
5:     $T'_s$ = solve the regulazised OT problem $(a, b, \widehat{\Lambda}, \epsilon)$
6:     $T_{s+1} = (1-\alpha)T_s + \alpha T'_s$
7: end for
8: return $T_{S-1}$
```

4.2 Convergence analysis

In this section, we aim at studying the convergence of Algorithm 1. Note that convergence guarantees have been already derived for EGW in (Peyré et al., 2016). However, based on Rangarajan et al., (1999), this convergence has been proven only when L produces a convex problem. Unlike Peyré et al., (2016), the guarantees presented in this section have two main advantages: (i) they hold whatever the loss function, (ii) a convergence on average is proven to a *stationary point*. Note that other results related to the GW problem have been recently derived in the literature. The authors of Xu et al., (2019b) prove the convergence of their proximal point method to a stationary point as long as their regularized GW problem can be solved perfectly at each iteration. On the other hand, Redko et al., (2020) provides a guarantee on the convergence of Problem (7) under the condition that L yields a concave problem.

Assuming that the two cost functions $\mathcal{C}^{\mathcal{X}}$ and $\mathcal{C}^{\mathcal{Y}}$ are symmetric, we introduce the following notations: $\mathcal{E}(A, A') := \mathcal{E}(A', A) := \sum_{i,j=1}^{I}\sum_{k,l=1}^{K} L_{ijkl}A_{ik}A'_{jl}$ and $\mathcal{E}(A) := \mathcal{E}(A, A)$. Under these notations, our goal is to minimize (5), *i.e.*, to minimize $\mathcal{E}(T)$ under constraints on the marginals of T. Let us now define $G(T)$ as follows: $G(T) := \mathcal{E}(T, T) - \min_{T' \in \Pi_{ab}} \mathcal{E}(T, T')$. In a non convex setting, T is a stationary point of $\mathcal{E}(T)$ if and only if $G(T) = 0$ (Reddi et al., 2016). The goal of our Theorem 1 is to provide a guarantee on the convergence of $G(\overline{T})$ with $\overline{T}$ uniformly sampled from $(T_s)_{s \in [\![0, S-1]\!]}$. The convergence is proven on average over these sampling. A practical implementation will naturally take only the last transport plan, T_{S-1}, and avoid unnecessary computations.

Theorem 1 (Based on Reddi et al., (2016)) *For any* $L_{ijkl} \in [0, B]$, *for any distributions* μ *and* ν *with uniform weights* a *and* b *respectively, for any optimal solution* T^* *of Problem* (5), *on average for the transport plan* $\overline{T}$ *uniformly sampled from* $(T_s)_{s \in [\![0, S-1]\!]}$, *on average over all the samplings, the following bound holds:*

$$\mathbb{E}\left(G\left(\overline{T}\right)\right) \leq \sqrt{\frac{2B\left(\mathcal{E}(T_0) - \mathcal{E}(T^*)\right)N}{S}} + B\sqrt{\frac{2N}{M}} + \epsilon \log(N).$$

Proof The complete proof is available in the Appendix A.1. It requires a novel lemma that quantifies the difference between the Wasserstein distances obtained with and without the entropic regularization: $0 \leq \langle C, T^{\epsilon} \rangle - \left\langle C, T^{0} \right\rangle \leq \epsilon \log(N)$. We also prove that $\mathcal{E}(T)$ is $2N^2$-smooth and we bound the difference between two transport plans: $\|T - T'\|_F \leq \sqrt{\frac{2}{N}}$.

Those two results allow us to adapt the proof of Theorem 2 in (Reddi et al., 2016) where our new Lemma is useful as the entropy regularized solvers do not find the exact OT minimum. □

While our bound cannot be explicitly computed as T^* is unknown, it gives meaningful information about Algorithm 1. First of all, it prompts us to initialize T_0 so as to get $\mathcal{E}(T_0)$ as close to $\mathcal{E}(T^*)$ as possible. Without any prior information, ab^T (the uniform plan) appears to be a reasonable choice to avoid degenerated cases. Regarding the regularization parameter, if ϵ is not small enough, the convergence to a stationary point is not guaranteed. On the other hand, we can note that the number of sampled matrices M appears in only one term of the bound. Therefore, the costly complete computation of the expectation ($M = \infty$) would not guarantee the convergence while leading to a $O\left(N^4\right)$ complexity. Thus, our bound prompts us to find a compromise between reducing M and increasing the number of iterations S, allowing us to control the complexity while getting a reasonable bound.

As the GW problem has been shown in (Redko et al., 2020) to be often concave, especially with the square loss and the euclidean distance on both spaces, the following Theorem 2 gives a second bound dedicated to address the specific concave case. This result presents the major interest of providing an asymptotic convergence to a stationary point for EGW in this concave case, as the proofs proposed in (Peyré et al., 2016) only cover the convergence of EGW and only for high values of ϵ.

Theorem 2 *With the same notations as in Theorem 1 with the entropy regularization parameter ϵ_s that may now change along the iterations s, when L yields a concave GW problem, the following bound holds:*

$$\mathbb{E}\left(G\left(\overline{T}\right)\right) \leq \frac{\mathcal{E}(T_0) - \mathcal{E}(T^*)}{2S} + B\sqrt{\frac{2N}{M}} + \frac{1}{S}\sum_{s=0}^{S-1} \epsilon_s log(N)$$

We can make the following comments from this bound. First, the convergence is better in the concave case as, unlike in Theorem 1, the first term is now linear in S. Second, as it can be seen in the proof (see Appendix A.1), it can be shown that in this concave scenario, the best value for α is 1. Thus, if we completely compute the matrix Λ ($M = \infty$), this bound applies to EGW. For any sequence $(\epsilon_s)_{s\in\mathbb{N}}$ such that $\sum_{s=0}^{S-1} \epsilon_s$ is $o(S)$, the convergence of EGW to a stationary point is guaranteed.

Relationship between SaGroW and the Frank-Wolfe algorithm At first sight, SaGroW seems akin to a Frank-Wolfe algorithm (Frank & Wolfe, 1956). In fact, when the regularization parameter $\epsilon = 0$, SaGroW is strictly equivalent to a Stochastic Frank-Wolfe (Reddi et al., 2016). The convergence analysis of this general non-convex setting is thus very similar, except for the term that depends on ϵ which quantifies the error due to the entropy regularization. Moreover, note that if $\epsilon = 0$, EGW becomes equivalent to the Frank-Wolfe algorithm (Frank & Wolfe, 1956) when its step size α is set to 1. Since the α parameter in our algorithm plays the same role as that of the step size of the Frank-Wolfe algorithm, we might wonder why SaGroW does not compute the optimal value using a line search. To the best of your knowledge, in this general non convex setting, there is no convergence guarantees towards a stationary point for a stochastic Frank-Wolfe algorithm that would make use of the optimal step. Moreover, it is worth noting that this optimal step is expensive ($O\left(N^4\right)$ complexity) to calculate without approximation. Considering an approximation would make the derivation of theoretical guarantees even more challenging.

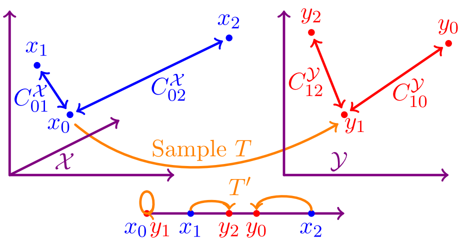

Fig. 2 Intuition behind PoGroW when $j, l = 0, 1$ are sampled from T: only the distances to x_0 in $\mathcal{X}$ (on the left) and to y_1 in $\mathcal{Y}$ (on the right) characterize a pair, and then T' can be computed in $O(N \log N)$ like in 1D OT

4.3 Particular case: pointwise GW

We focus in this section on the special case of SaGroW where only one matrix C is sampled (*i.e.*, $M = 1$) at each iteration. This variant, called Pointwise Gromov Wasserstein (**PoGroW**), makes it possible to leverage a dedicated solver to reduce the algorithmic complexity of GW.

When $M = 1$, if we sample a position j, l from T, then we seek to minimize the following problem:

$$\min_{T' \in \Pi_{ab}} \sum_{i,k=1}^{I,K} \mathcal{L}(C^{\mathcal{X}}(x_i, x_j), C^{\mathcal{Y}}(y_k, y_l)) T'_{ik}. \tag{11}$$

As illustrated in Fig. 2, each point in $\mathcal{X}$ (resp. $\mathcal{Y}$) is simply defined by its distance to x_j (resp. y_l), as done in papers that define a distribution using a distance to a point (Gelfand et al., 2005; Sato et al., 2020). With a single feature per point, Problem (11) can be solved very efficiently in $O(N \log(N))$ like a 1D OT problem: the two lists of distances can be sorted and matched. With non-convex losses, this sorting approach is only an approximation. PoGroW can be seen as a natural GW extension of Sliced Wasserstein where each point is described by its distance to a chosen "anchor" (instead of a position on a line). Recall that the output of Problem (11) is a transport plan. If needed for the application at hand, the GW value can be computed in $O(N^2)$ (see Sect. 4.5).

In summary, PoGroW has the same low complexity as Sliced Gromov Wasserstein (Vayer et al., 2019b) but it overcomes its main limitations: PoGroW is naturally invariant to rotation; it returns a transport plan; it approximates the actual GW distance; it works with graphs.

4.4 A KL regularization-based variant

As the transport plan T is a distribution and most GW algorithms progressively update T, an interesting idea is to encourage the next plan T' to be close (in terms of KL divergence) to the current estimate T. This idea, already used in Xu et al., (2019b) based on Xie et al., (2020), can be applied to our SaGroW algorithm: we name this approach SaGroW^{KL} and describe it below.

In Algorithm 1, we used partial updates to explore the transport plan space while encouraging the new value of T to be close to the preceding one, as reflected in line 6. We suggest here a slight modification, consisting in using a Kullback Leibler (KL) regularization between T and T' in line 5 and removing line 6. This allows to account, in a natural way, for the requirement for T and T' to stay close to each other during the optimization. This leads to the following sampled optimization problem,

$$\min_{T' \in \Pi_{ab}} \left\langle \frac{1}{M} \sum_{m=1}^{M} C^m, T' \right\rangle + \epsilon KL(T' || T), \tag{12}$$

which can be rearranged into,

$$\min_{T' \in \Pi_{ab}} \left\langle \frac{1}{M} \sum_{m=1}^{M} C^m - \epsilon log(T), T' \right\rangle + \epsilon \mathcal{H}(T'). \tag{13}$$

This regularization allows to take advantage of the Sinkhorn-Knopps solver (Cuturi, 2013) as it is similar to Eq. (3) with a cost function modified to take into account the current prior T. Even if ϵ is high, the optimization might lead to a solution close to the edge of the polytope with enough iterations which is not the case with a classical entropy regularization without prior. The time complexity does not increase as it is still $O\big((P+M)N^2\big)$. As this regularization is not specific to our method, we will also use it for EGW during the experiments to allow a fair comparison. On the other hand, note that this regularization cannot be used with PoGroW as it currently does not seem possible to solve 1D entropy-regularized OT in $O(Nlog(N))$ (Cuturi et al., 2019). Note also that the convergence Theorem 1 does not hold anymore with this regularization.

4.5 Efficient computation the GW distance from a transport plan

This section introduces and evaluates a low-complexity high-accuracy method for the estimation of $\mathcal{E}(T)$. Indeed, while SaGroW and PoGroW provide important complexity improvements, one might argue that they only find a good transport plan T and do not provide a value for $\mathcal{E}(T)$. An exact computation of $\mathcal{E}(T)$ has a $O\big(N^4\big)$ time complexity, and it would dominate the complexity of our algorithms in applications where $\mathcal{E}(T)$ is required, for example when GW is used as a dissimilarity measure between graphs. Additionally, having an efficient way of estimating $\mathcal{E}(T)$ opens the door to selecting the best transport plan among a set of plans, *e.g.*, obtained by varying the hyper-parameters or the random seed of an algorithm.

We address this issue in this section. Similar to Eq. (9), we propose to interpret the sums in the definition of $\mathcal{E}(T)$ as the expectation of a random variable $\mathbf{R}$ (this time real-valued instead of matrix-valued, so with a quadruple sum), with $\mathbb{P}(\mathbf{R} = L_{ijkl}) = T_{ij}T_{kl}$:

$$\mathbb{E}(\mathbf{R}) = \sum_{i,j,k,l=1}^{I,J,K,K} L_{ijkl} T_{ij} T_{kl}. \tag{14}$$

Instead of simply sampling this expectation, we propose to stratify by each index i, k to improve the quality of the estimate. Let U_i be the event "*i is chosen for the first dimension of L*" and U'_k be the event "*k is chosen for the third dimension of L*". Based on the marginal a and using the law of total expectation, $\mathbb{E}(\mathbf{R})$ can be rewritten as:

$$\mathbb{E}(\mathbf{R}) = \sum_{i,k=1}^{I,K} \mathbb{P}(U_i \cap U'_k)\mathbb{E}(\mathbf{R}|U_i \cap U'_k) = \sum_{i,k=1}^{I,K} a_i a_k \mathbb{E}(\mathbf{R}|U_i \cap U'_k). \tag{15}$$

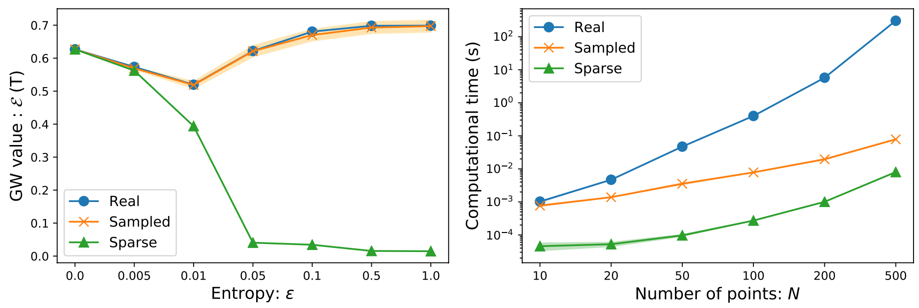

Fig. 3 Estimated value of $\mathcal{E}(T)$ as sparsity decreases due to an increasing ϵ regularization in EGW (left) and evolution of the time required for its estimation as N grows (right). The absolute loss is used in these experiments and the distributions take the form of two graphs generated using a gaussian random partition graph (Brandes et al., 2003). For a given ϵ and N, the same T (obtained using EGW) is passed to the three considered methods: *Real*) an exact one which computes completely $\mathcal{E}(T)$, *Sampled*) our sampling method described in Sect. 4.5, and, *Sparse*) a sparse approximation which keeps only the $2N$ largest values of T and sets the other entries to 0. The mean and 2 standard deviations over 10 runs are displayed on both figures. When the standard deviation is not visible, it corresponds either to a deterministic method or a value very close to 0.

Table 1 Complexity of each method with an arbitrary loss function, with S iterations, P Sinkhorn iterations, N points in the dataset and M matrix samples

Methods	GW approximation	Transport Plan	$\mathcal{E}(T)$	Total
EGW	Yes	$S(P+N^2)N^2$	N^4	$S(P+N^2)N^2$
SaGroW	Yes	$S(P+M)N^2$	N^2	$S(P+M)N^2$
S-GWL	Yes	Unknown	SN^2	Unknown
SGW	No	Unavailable	SN^2	SN^2
PoGroW	Yes	$SN\log(N)$	N^2	N^2

Note that the complexity can be different for specific loss functions

For each (i, k), the conditional expectation is approximated using M samples of a random variable X_{ik}, defined by $\mathbb{P}(X_{ik} = L_{ijkl}) = \mathbb{P}(\mathbf{R} = L_{ijkl} | U_i \cap U'_k) = T_{ij}T_{kl}$. Finally, $\hat{\mathbf{R}} = \sum_{ik} a_i a_k \frac{1}{M} \sum_{m=1}^{M} X_{ik}^m$ defines an unbiased estimate of the GW distance which can be computed in $O(MN^2)$ (details about the variance estimate are provided in the Appendix A.3).

As shown in Fig. 3 (left), the prediction is perfect for a sparse transport plan ($\epsilon = 0$), while still being almost perfect and much better than a naive sparse approximation of the OT plan as ϵ increases. Fig. 3 (right) confirms that this approximation is clearly faster than the exact computation which becomes quickly intractable as N grows.

Having at our disposal an efficient method for estimating $\mathcal{E}(T)$, we can now fully compare, in Table 1, the complexity of the state of the art methods with that of SaGroW and PoGroW, for the general case of an arbitrary loss function. From this table, we have evidence that SaGroW allows a drastic reduction of the algorithmic complexity of EGW. On the other hand, PoGroW fully benefits from the 1D projections. But unlike SGW, it provides a transport plan and does approximate the original GW problem.

5 Experiments

In this section,[4] we first compare different GW methods on both their *speed* and their *accuracy*. We use here the term *accuracy* to express the capability of the methods to minimize $\mathcal{E}$(T). Indeed, as the exact (optimal) GW distance is unknown for a given dataset (solving this problem is known to be NP-hard), the best method will be the one with the smallest value of $\mathcal{E}(T)$. Then, we analyze the impact of the hyperparameters, illustrating that our approach covers a range of very good trade-offs between speed and accuracy. Using a real graph-classification task, we finally illustrate why being able to solve GW for various loss functions is important.

5.1 General setup and methods

We compare SaGroW^{KL} and PoGroW with: (I) EGW (Peyré et al., 2016; II) EGW^{KL}, a KL regularized version of EGW described in Xu et al., (2019b); (III) EMD-GW, which is similar to EGW$_0$, but uses the OT solver of (Bonneel et al.,, 2011) as the Sinkhorn algorithm (Cuturi, 2013) cannot handle a null value for ϵ; (IV) S-GWL (Xu et al., 2019a), adapted for arbitrary loss functions using the optimizer of Wright (1996) to update the barycenter; (V) SGW when the points are available, with an adaptation to arbitrary losses; (VI) the uniform transport plan, used as a baseline.

While Sect. 5.3 will detail the impact of the hyperparameters, the next section reports, for each method, the results obtained by the set of parameters with the lowest GW estimation. To take into account the stochasticity of some methods the GW estimation for each hyperparameter set is taken on average over 10 runs. ϵ is chosen among {0.001, 0.005, 0.01, 0.005, 0.1} for EGW and EGW^{KL}, and in {0.001, 0.01, 0.1, 1, 10, 100} for S-GWL. To have comparable sets of hyperparameters, we fix some of our parameters: in PoGroW, a step of $\alpha = 0.8$, and in SaGroW, the number of samples $M = 10$ and a KL regularization $\epsilon = 1$. Experiments in the Appendices B.5 and B.6 show that: SaGroW is much less sensitive to ϵ than EGW and $\alpha = 0.8$ is a reasonable choice. The number of iterations S is chosen among {10, 100, 500, 1000} to obtain a reasonable accuracy-speed trade-off.

This experiment compares the quality of the transport plan and the computational time of the methods for an increasing number of points N. Each method minimizes Problem (5) and returns a transport plan T (besides SGW, see below). In order to assess the quality of this transport plan, $\mathcal{E}(T)$ is then computed exactly. Notably, our GW distance approximation (see Sect. 4.5) is *not* used in this first experiment. The mean and standard deviation of $\mathcal{E}(T)$ over ten runs are reported.

The loss $\mathcal{L}$ chosen here is the absolute loss in order to show the capacity of our methods to deal with any arbitrary loss function. We remind that EGW, S-GWL and SGW are much faster (with speeds that are comparable to our approach) for some specific losses, such as the square loss (see Appendix B.2 and Sect. 5.4).

To include SGW (which needs points to project) in this comparative study, a first dataset uses μ and ν that are composed of N points sampled from two different mixtures of gaussians. Details about the generation of the datasets are available in the Appendix B.1.

[4] The code to reproduce all the experiments, figures and tables is available in the GitHub https://github.com/Hv0nnus/Sampled-Gromov-Wasserstein

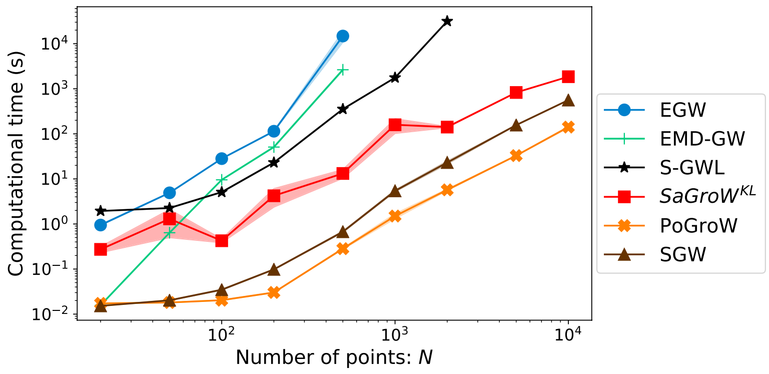

Fig. 4 Computational time of various methods to compute the distance between samples from two mixtures of gaussians. The mean and the standard deviation over 10 runs are reported

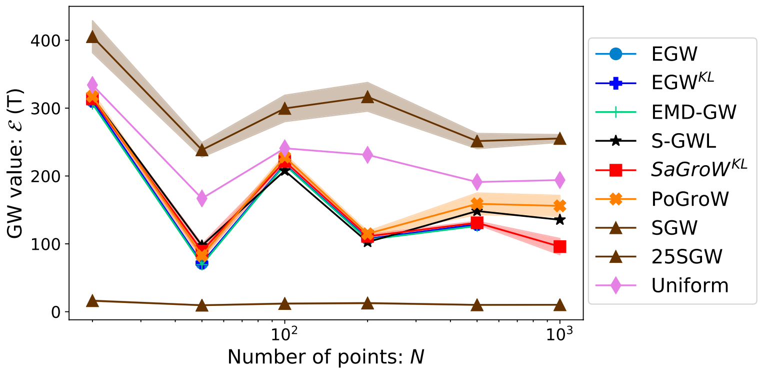

Fig. 5 GW distance estimation between samples from two mixtures of gaussians. The mean and standard deviation over 10 runs are reported for the stochastic methods

5.2 Speed and accuracy of the GW estimate

Figure 4 shows, in a log-log representation, that EGW and EMD-GW become quickly intractable when the number of points increases and that S-GWL is slightly faster. We exclude EGW^{KL} for the clarity of the figure as it has a computational time similar to EGW. SaGroW, PoGroW and SGW behave better, with a quadratic complexity (linear slope of 2 in log-log) but with different multiplicative factors (offsets in the log-log plot).

Figure 5 reports the quality of the obtained GW value. Comparing SGW to other methods is complicated as it does not return a transport plan, nor aims at computing an approximation of the GW distance. We thus report the distance it computes and also the same rescaled by a factor 25. With rescaling, we see that SGW seems to behave more like the uniform transport plan than like the GW methods (which produce better-than-uniform plans). While all other methods predict very similar GW distances, EGW-based methods have often the best accuracy. However, when N reaches 1000 points, we can observe interesting behaviors: EGW is not able to provide any result, PoGroW is the fastest with a lesser

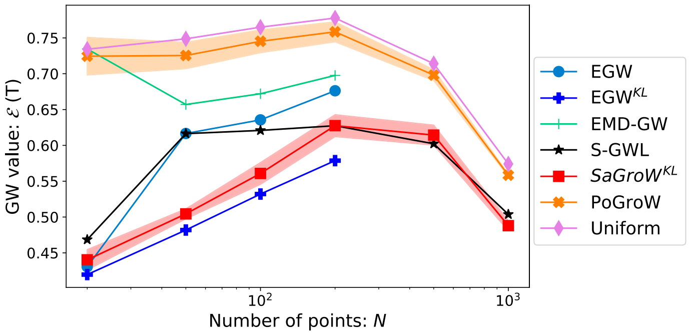

Fig. 6 GW distance estimation between synthetic graphs (Brandes et al., 2003). The mean and standard deviation over 10 runs are reported for the stochastic methods

accuracy than S-GWL, and SaGroW provides the best value while being much faster than S-GWL.

In a second series of experiments, we make use of graphs that are generated using a gaussian random partition graph (Brandes et al., 2003). On this more difficult dataset, we see in Fig. 6 that SaGroW is very competitive with the best method EGW^{KL} while being able to scale to more than 200 nodes, which is the limit for all EGW-based methods. With more nodes, SaGroW is as accurate as S-GWL but remains much faster and scalable (computation times are similar to the ones from the first dataset). In this experiment, a key factor of success seems to be the KL regularization, used in EGW^{KL}, S-GWL and SaGroW. This can explain why PoGroW stays close to the uniform baseline.

5.3 Hyperparameters analysis

We now focus on the impact of the numbers of iterations S and samples M, showing that these allow our approach to cover a variety of trade-offs between speed and accuracy. More experiments (in the Appendix B) consider other parameters such as different loss functions or dataset size. We also study, in this experiment, the impact of the ϵ parameter of other methods.

Figure 7 shows that increasing the number of iterations S yields a strong improvement for SaGroW, independently of the number of samples M. Interestingly, the accuracy of SaGroW is similar regardless the value of M. This remark supports the key assumption of this paper that the entire computation of the expectation is not needed. The standard deviation displayed in Fig. 7 shows that most runs provide similar GW distances, with enough iterations. However, there is a high variance with less iterations which tends to highlight that the different runs of SaGroW take different paths during the optimization. As shown in Fig. 8, the speed of EGW and S-GWL does not vary much with ϵ but this parameter needs to be chosen carefully for those methods to reach a good accuracy.

On Fig. 9 we can see that PoGroW is even faster than SaGroW: it can provide a reasonable approximation in a second, compared to the three hours required by EGW. Because PoGroW does not resort to a KL regularization, it is more impacted by stochasticity: two runs can yield very different results. This can be used advantageously by keeping the plan that gives the lowest GW among ten runs (crosses on Fig. 9). The combination of SaGroW and PoGroW allows to obtain a good trade-off between speed and accuracy.

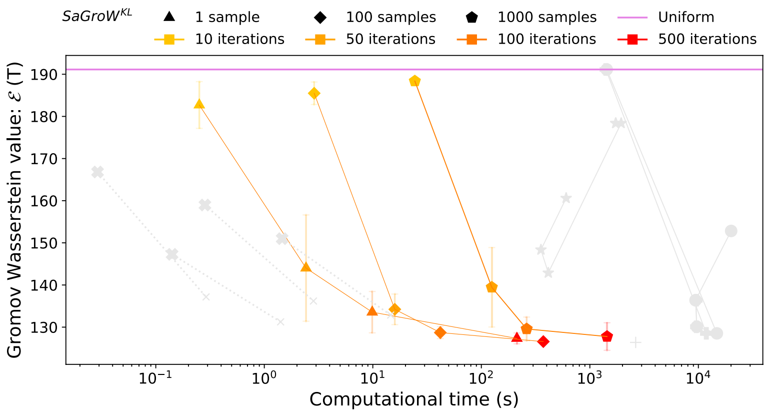

Fig. 7 Impact of the number of sample M and the number of iterations S for SaGroW on the GW distance estimation and computational time, for two sets of 500 points sampled from two mixtures of gaussians. The mean and standard deviation over 10 runs are display

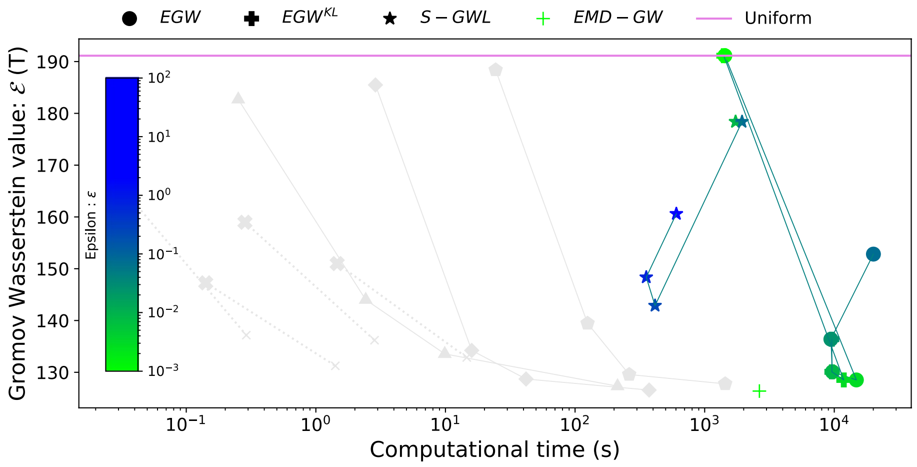

Fig. 8 Impact of the Kullback-Leiber regularization ϵ for EGW and S-GWL on the GW distance estimation and computational time, for two sets of 500 points sampled from two mixtures of gaussians

Beyond the algorithmic advantages shown above, one last key question remains: is it useful, in an application, to compute the GW distance for other losses than the widely used square loss?

5.4 Graph classification

We illustrate here the usefulness of using different loss functions in a context of graph classification. We take the FIRSTMM-DB graph dataset (Neumann et al., 2013) which is the one with the biggest average nodes number (1377) over the database of (Kersting et al., 2016). Each of the 41 graphs of the dataset describes an object from one of the 11 classes

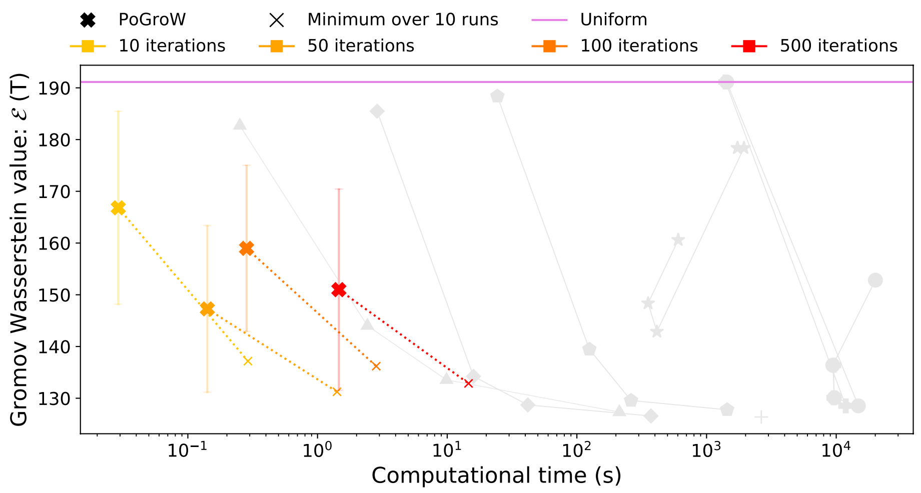

Fig. 9 Impact of the number of iterations S for PoGroW on the GW distance estimation and computational time, for two sets of 500 points sampled from two mixtures of gaussians. The mean and standard deviation over 10 runs are display. To take advantage of the large stochasticity, the minimum over 10 runs is also display

Table 2 Classification accuracy and computation time of various methods on a 11-class graph classification task

Methods	Accuracy	Time (s)
S-GWL$_5$	0.44	23.4
EGW$^{KL}_{0.005}$	0.24	41.1
EMD-GW	0.37	16.6
EGW$_{0.001}$	0.22	36.2
Uniform	0.07	0.1
SaGroW$_{p=1}$	**0.49**	11.6
SaGroW$_{p=2}$	0.39	12.7
PoGroW$_{p=2}$	0.39	**0.5**

In this summary table, only the hyperparameters yielding the best classification accuracy are reported, for each considered method

(cup, knife, etc.). The distance matrix of each graph $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$ is computed using the shortest path length, similarly to Mémoli (2011). For each method, we compute the pairwise GW distance matrix. Finally, a 1-Nearest-Neighbor classifier is used to predict the class of each graph (using a leave-one-graph-out scheme).

Section 5.2 showed that EGW, EGW^{KL} and S-GWL are very slow with arbitrary loss functions on graphs (with around 1000 nodes). Therefore, we use for them the square loss to allow them to be competitive from a time complexity perspective. We consider ten values for the entropic regularization, $\epsilon \in [10^{-4}, 10^{2}]$. SGW is excluded as it is unable to handle graphs. For our methods, we set $\epsilon = 0.1$ for SaGroW and $\alpha = 0.8$ for PoGroW and keep $M = 1$, $S = 100$ for both methods. However, ten different loss functions $\mathcal{L}$ are tested, notably $|C^{\mathcal{X}}_{ij} - C^{\mathcal{Y}}_{kl}|^p$ for different values of $p \in [0.5, 3]$.

The results are reported in Table 2. Looking at SaGroW, we see that the classical square loss ($p = 2$) is outperformed, *e.g.*, by the absolute loss ($p = 1$) which yields a better classification accuracy. Beyond that, the ability of SaGroW to handle arbitrary losses allows it to get the best overall accuracy, across all the methods. The explanation can be that the $L1$ loss is more robust to outlier nodes, which might be important on this real dataset. Note that while EGW and S-GWL are fast as they are computed with the square loss for $\mathcal{L}$, SaGroW is still slightly faster. PoGroW has a competitive accuracy and even outperforms EGW while being very fast. The complete table with every hyperparameter run is available in the Appendix B.7.

While the goal of this experiment is to correctly classify graphs, we can still compare the GW distances obtained from the transport plans returned by all methods. This comparison only makes sense with the same (square) loss for all methods. Averaged over 41^2 distances, SaGroW gets the lowest value of 336, followed by EMD-GW with 341. This highlights the fact that, on a real dataset, the stochasticity used by our method can lead to a better GW distance estimation.

6 Conclusion

In this paper, we present both algorithmic and theoretical contributions to address the still open problem related to the calculation of the Gromov Wasserstein distance. We propose a method to reduce drastically the time complexity of GW for arbitrary loss functions. To do so, we tackle the bottleneck of the mostly used GW solver, namely EGW, by using a sampling strategy to efficiently approximate the costly sum of N^2 matrices. Our SaGroW algorithm is supported with theoretical convergence guarantees to a stationary point in the general non-convex setting. We also introduce PoGroW, an algorithm which samples only one matrix and allows us to benefit from a very low complexity by using 1D OT. We show that PoGroW overcomes the main issues related to SGW. Experiments on synthetic datasets show that our method are tractable for a large number of points and offer a good trade-off between speed and accuracy. Finally, a real world experiment on graph classification illustrates the interest of choosing different loss functions. In order to deal with potential outliers, we show that the absolute loss associated with SaGroW gives the highest classification accuracy. We claim that this capacity to choose *ad-hoc loss functions* will push the state of the art in various graph applications by unlocking their use with large graphs.

Appendix A: Scalable GW optimization

A.1: Detailed derivations for the convergence

This section gives the (very) detailed derivations used to obtain the convergence properties of Sect. 4.2 of Algorithm 1 (from the paper).

Goal and context

First, let's give a few reminders of the context and the final result. The proposed algorithm runs for S iterations, and averages M sampled cost matrices (obtained by sampling pairs of indices), at each iteration. We provide here a proof of convergence to a stationary point for any arbitrary loss $\mathcal{L}$. Previous algorithms relied on having some particular loss $\mathcal{L}$ to be efficient. When $M = \infty$ and $\alpha = 1$, the proposed algorithm is equivalent to EGW.

We are interested in $G(T) \overset{\text{def}}{=} \mathcal{E}(T,T) - \min_{T'} \mathcal{E}(T,T')$. In a non convex setting, T is a stationary point of $\mathcal{E}(T)$ if and only if $G(T) = 0$ (Reddi et al., 2016). We recall the assumptions and notations:

- We suppose $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$ symmetric. This assumption is notably satisfied if $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$ are metrics.
- We define $\mathcal{E}(A,A') \overset{\text{def}}{=} \mathcal{E}(A',A) = \sum_{i,j=1}^{I} \sum_{k,l=1}^{K} L_{ijkl} A_{ik} A'_{jl}$
- We overload the notation if the two parameters are the same: $\mathcal{E}(A) \overset{\text{def}}{=} \mathcal{E}(A,A)$
- We assume that $0 \leq L_{ijkl} \leq B$. This value B can be found in $O\left(N^2\right)$ with any losses $\mathcal{L}$ that increase when $|C^{\mathcal{X}}_{ij}$ - $C^{\mathcal{Y}}_{kl}|$ increases, by looking at the extreme values of the two matrix $C^{\mathcal{Y}}$ and $C^{\mathcal{X}}$.

More precisely, the bound that we will prove here is the following (Theorem 1 of the paper):

$$\mathbb{E}\left(G\left(\overline{T}\right)\right) \leq \sqrt{\frac{2B\left(\mathcal{E}(T_0) - \mathcal{E}(T^*)\right)N}{S}} + B\sqrt{\frac{2N}{M}} + \epsilon \log(N).$$

Where T^* is the optimal (unknown) solution of GW, i.e., $T^* = \underset{T \in \Pi_{ab}}{\operatorname{argmin}} \mathcal{E}(T)$ and the expectation is taken on all the sampling done during the algorithm and on $\overline{T}$.

The notation of the Algorithm 1 are slightly different in this appendix, as we make the distinction between T'^{ϵ}_s and T'_s. T'^{ϵ}_s is the transport plan given by the OT-Sinkhorn solver, while T'_s is the exact minimum transport plan.

Our proof is inspired by the Theorem 2 from Reddi et al., (2016) but we additionally consider the entropy regularization with notably the Lemma 1 which is specific to the OT problem. To give all details while trying to improve readability, we first prove some intermediate results.

Necessary intermediate results

We first prove the following new lemma which quantifies the difference between the Wasserstein distance with and without the entropy regularization, for a generic OT problem with a cost matrix C. Note that a related bound was proposed by Genevay et al., (2019) or Blondel et al., (2018) but include the entropy regularization while here we are only concerned about the difference between the scalar product.

Lemma 1 *Let T^{ϵ} (resp. T^0) be the optimal solution of the a discrete OT problem with (resp. without) entropic regularization. We suppose the simplified case with N points in each empirical distribution and with uniform marginal distributions. We will note C the $N \times N$ cost matrix of this problem.*

$$0 \leq \langle C, T^{\epsilon} \rangle - \left\langle C, T^0 \right\rangle \leq \epsilon \log(N) \tag{16}$$

Proof The positivity is obtained by definition of T^0 (it minimizes $\langle C, T \rangle$). The right-hand side inequality can be derived as follows (where $-\mathcal{H}(T)$ denotes the entropy of T):

$$\langle C, T^\epsilon \rangle - \left\langle C, T^0 \right\rangle = \langle C, T^\epsilon \rangle - \left\langle C, T^0 \right\rangle + \epsilon\mathcal{H}(T^\epsilon) - \epsilon\mathcal{H}(T^\epsilon) \tag{17}$$

$$\leq \left\langle C, T^0 \right\rangle - \left\langle C, T^0 \right\rangle + \epsilon\mathcal{H}(T^0) - \epsilon\mathcal{H}(T^\epsilon) \tag{18}$$

$$\leq \epsilon\mathcal{H}(T^0) - \epsilon\mathcal{H}(T^\epsilon) \tag{19}$$

$$\leq \epsilon \log(N^{-1}) - \epsilon \log(N^{-2}) \tag{20}$$

$$= \epsilon \log(N) \tag{21}$$

Line 18 : by definition, T^ϵ minimizes $\langle C, T \rangle + \epsilon\mathcal{H}(T)$. Line 20 T^0 is a permutation and T^ϵ is at worse (in terms of $\mathcal{H}()$) uniform.

□

Interestingly this bound does not depend directly depend on C (still, C impacts the value of T^0, T^ϵ). A scale increase of C will virtually reduce ϵ in comparison, thus T^ϵ will be closer to T^0. Note that the bound can be adapted to the general case (arbitrary distributions), then the bound is $-\epsilon(\mathcal{H}(\mu) + \mathcal{H}(\nu))$ as we bound $\mathcal{H}(T^0)$ by 0 and $-\mathcal{H}(T^\epsilon)$ by $-\mathcal{H}(\mu \times \nu)$.

Let $(T, T') \in \Pi^2_{ab}$. We now derive several intermediate results with these arbitrary transport plans T and T', in a simplified case when $I = J = N$ (same number of points in each empirical distribution).

We start with a bound on the maximal distance between these transport plans (in term of Frobenius norm):

$$\|T - T'\|_F = \sqrt{\|T - T'\|_F^2} \tag{22}$$

$$\leq \sqrt{\|T\|_F^2 + \|T'\|_F^2} \tag{23}$$

$$\leq \sqrt{\sum_{i,k=1}^{I,K} T_{ik}^2 + \sum_{i,k=1}^{I,K} T'^2_{ik}} \tag{24}$$

$$\leq \sqrt{N\left(\frac{1}{N}\right)^2 + N\left(\frac{1}{N}\right)^2} \tag{25}$$

$$= \sqrt{\frac{2}{N}}. \tag{26}$$

Line 23: the triangular inequality is used. Line 25: for doubly stochastic matrices, the highest Frobenius norm is obtained with a permutation (fewer and thus bigger values give a bigger norm), the permutation has N non-zero values equal to $\frac{1}{N}$.

For completeness, we prove that the gradient of $\mathcal{E}(T)$ is expressed in terms of T. We prove it with L symmetric, in the sense that $L_{ijkl} = L_{jilk}$, which is implied if the cost matrices are symmetric. For all indices a, b in T, we have:

$$\frac{\partial \mathcal{E}}{\partial T_{ab}}(T) = \frac{\partial}{\partial T_{ab}} \sum_{i,j=1}^{I,I} \sum_{k,l=1}^{K,K} L_{ijkl} T_{ik}\, T_{jl} \tag{27}$$

$$= \frac{\partial}{\partial T_{ab}} \left(L_{aabb}\, T_{ab}^2 + \left(2 \sum_{cd \neq ab} L_{acbd} T_{cd} \right) T_{ab} \right) \tag{28}$$

$$= 2\, L_{aabb} T_{ab} + 2 \sum_{cd \neq ab} L_{acbd} T_{cd} \tag{29}$$

$$= 2 \sum_{cd} L_{acbd} T_{cd} \tag{30}$$

$$\nabla \mathcal{E}(T) = 2 \sum_{jl} L_{.j.l} T_{jl} \tag{31}$$

$$= \sum_{jl} (L_{.j.l} + L_{j.l.}) T_{jl} \text{ in the case where L is not symmetric.} \tag{32}$$

We can also prove that $\mathcal{E}$ is $2BN^2$-smooth, as follows:

$$\left\| \nabla \mathcal{E}(T) - \nabla \mathcal{E}(T') \right\|_F = \left\| 2 \sum_{jl} L_{.j.l} T_{jl} - 2 \sum_{jl} L_{.j.l} T'_{jl} \right\|_F \tag{33}$$

$$= \left\| 2 \sum_{jl} L_{.j.l} \left(T_{jl} - T'_{jl} \right) \right\|_F \tag{34}$$

$$= \sqrt{ \sum_{ik} \left(2 \sum_{jl} L_{ijkl} \left(T_{jl} - T'_{jl} \right) \right)^2 } \tag{35}$$

$$= \sqrt{ \sum_{ik} \left(2 \langle L_{i.k.}, T - T' \rangle \right)^2 } \tag{36}$$

$$\leq \sqrt{ \sum_{ik} \left(2 \| L_{i.k.} \|_F \| T - T' \|_F \right)^2 } \tag{37}$$

$$\leq \sqrt{4 \sum_{jl} B^2 N^2 \|T - T'\|_F^2} \tag{38}$$

$$\leq 2B\sqrt{N^4 \|T - T'\|_F^2} \tag{39}$$

$$\leq 2BN^2 \|T - T'\|_F. \tag{40}$$

Line 37 uses the Cauchy–Schwarz inequality. Line 38 uses $0 \leq L_{ijkl} \leq B$.

The following Lemma 2 is the same as the one provided in Reddi et al., (2016) and will allow to start the proof.

Lemma 2 *If $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ is L-smooth, then for all $x, y \in \mathbb{R}^d$.*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2$$

Proof of the theorem

Theorem 3 (Based on (Reddi et al., 2016)) *For any $L_{ijkl} \in [0, 1]$, for any distributions μ and ν with uniform weights a and b respectively, for any optimal solution T^* of Problem (5), on average for the transport plan $\overline{T}$ uniformly sampled from $(T_s)_{s \in [\![0, S-1]\!]}$, on average over all the samplings, the following bound holds:*

$$\mathbb{E}\Big(G\big(\overline{T}\big)\Big) \leq \sqrt{\frac{2B\big(\mathcal{E}(T_0) - \mathcal{E}(T^*)\big)N}{S}} + B\sqrt{\frac{2N}{M}} + \epsilon \log(N).$$

Proof T_s and $T_s'^{\epsilon}$ are the transport plan obtain in the Algorithm 1. $T_s' = T_s'^{0}$ is the solution without entropy regularization.

Let $\hat{T}_s' = \underset{T_s' \in \Pi_{ab}}{\text{argmin}} \big\langle T_s', \nabla \mathcal{E}(T_s) \big\rangle = \underset{T_s' \in \Pi_{ab}}{\text{argmax}} \big\langle T_s', -\nabla \mathcal{E}(T_s) \big\rangle$ and $\widehat{\Lambda}_s$ the sum of matrices sampled M times at iteration s.

$$\mathcal{E}(T_{s+1}) \leq \mathcal{E}(T_s) + \big\langle \nabla \mathcal{E}(T_s), T_{s+1} - T_s \big\rangle + \frac{2BN^2}{2} \|T_{s+1} - T_s\|^2 \tag{41}$$

$$\leq \mathcal{E}(T_s) + \big\langle \nabla \mathcal{E}(T_s), \alpha(T_s'^{\epsilon} - T_s) \big\rangle + BN^2 \|\alpha(T_s'^{\epsilon} - T_s)\|^2 \tag{42}$$

$$\leq \mathcal{E}(T_s) + \big\langle \nabla \mathcal{E}(T_s), \alpha(T_s'^{\epsilon} - T_s) \big\rangle + BN^2 \alpha^2 \sqrt{\frac{2}{N}}^2 \tag{43}$$

$$= \mathcal{E}(T_s) + \alpha\Big\langle 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s \Big\rangle + \alpha\Big\langle \nabla \mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s \Big\rangle + BN^2 \alpha^2 \sqrt{\frac{2}{N}}^2 \tag{44}$$

$$= \mathcal{E}(T_s) + \alpha\left\langle 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s + T_s' - T_s' \right\rangle + \alpha\left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s \right\rangle + BN^2\alpha^2\sqrt{\frac{2}{N}}^{-2} \tag{45}$$

$$= \mathcal{E}(T_s) + \alpha\left\langle 2\widehat{\Lambda}_s, T_s' - T_s \right\rangle + \alpha 2\left\langle \widehat{\Lambda}_s, T_s'^{\epsilon} - T_s' \right\rangle + \alpha\left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s \right\rangle + BN^2\alpha^2\sqrt{\frac{2}{N}}^{-2} \tag{46}$$

$$\le \mathcal{E}(T_s) + \alpha\left\langle 2\widehat{\Lambda}_s, \hat{T}_s' - T_s \right\rangle + \alpha 2\epsilon\log(N) + \alpha\left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s \right\rangle + BN^2\alpha^2\sqrt{\frac{2}{N}}^{-2} \tag{47}$$

$$= \mathcal{E}(T_s) + \alpha\left\langle \nabla\mathcal{E}(T_s), \hat{T}_s' - T_s \right\rangle + \alpha\left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - \hat{T}_s' \right\rangle + \alpha 2\epsilon\log(N) + BN^2\alpha^2\sqrt{\frac{2}{N}}^{-2} \tag{48}$$

$$= \mathcal{E}(T_s) - 2\alpha G(T_s) + \alpha\left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - \hat{T}_s' \right\rangle + \alpha 2\epsilon\log(N) + BN^2\alpha^2\sqrt{\frac{2}{N}}^{-2} \tag{49}$$

$$\le \mathcal{E}(T_s) - 2\alpha G(T_s) + \sqrt{\frac{2}{N}}\alpha\|\nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s\| + \alpha 2\epsilon\log(N) + BN^2\alpha^2\sqrt{\frac{2}{N}}^{-2} \tag{50}$$

The line 41 uses the smoothness of $\mathcal{E}$. The line 42 uses the definition of the update. The line 43 uses the bound between transports plans. The line 44 adds artificially the $2\widehat{\Lambda}_s$ term. The line 45 adds artificially the T_s' term. The line 46 separate two terms. The line 47 uses the Lemma 1 with $\widehat{\Lambda}_s$ as cost matrix and use the definition of T_s'. The line 48 uses the following equalities,

$$\left\langle 2\widehat{\Lambda}_s, \hat{T}_s' - T_s \right\rangle + \left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - T_s \right\rangle \tag{51}$$

$$= \left\langle 2\widehat{\Lambda}_s, \hat{T}_s' - T_s - T_s'^{\epsilon} + T_s \right\rangle + \left\langle \nabla\mathcal{E}(T_s), T_s'^{\epsilon} - T_s + \hat{T}_s' - \hat{T}_s' \right\rangle \tag{52}$$

$$= \left\langle 2\widehat{\Lambda}_s, \hat{T}_s' - T_s'^{\epsilon} \right\rangle + \left\langle \nabla\mathcal{E}(T_s), T_s'^{\epsilon} - \hat{T}_s' \right\rangle + \left\langle \nabla\mathcal{E}(T_s), \hat{T}_s' - T_s \right\rangle \tag{53}$$

$$= \left\langle \nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s, T_s'^{\epsilon} - \hat{T}_s' \right\rangle + \left\langle \nabla\mathcal{E}(T_s), \hat{T}_s' - T_s \right\rangle. \tag{54}$$

The line 49 uses the definition of $G(T_s)$. The line 50 applies Cauchy Schwartz inequality and bound the difference between OT plan.

To bound the difference between the real expectation $\nabla\mathcal{E}(T_s)$ and the sampling $2\widehat{\Lambda}_s$, the following result is needed. Let define M random variable, $z_m = L_{.j_m.l_m} - \sum_{jl} L_{.j.l}T_{jl}$. They have 0 mean and each $\underline{z_m}$ are independent from each other. Moreover, $\|z_m\| = \|L_{.j_m.l_m} - \sum_{jl} L_{.j.l}T_{jl}\| \le \sqrt{\sum_{ik} B^2} = BN$.

$$\mathbb{E}(\|\nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s\|) = \mathbb{E}(\|2\sum_{jl} L_{.j.l}T_{jl} - \frac{2}{M}\sum_{m=1}^{M} L_{.j_m.l_m}\|) \tag{55}$$

$$= \frac{1}{M}2\mathbb{E}(\|\sum_{m=1}^{M} z_m\|) \tag{56}$$

$$= \frac{2}{M}\sqrt{(\mathbb{E}(\|\sum_{m=1}^{M} z_m\|))^2} \tag{57}$$

$$\le \frac{2}{M}\sqrt{\mathbb{E}(\|\sum_{m=1}^{M} z_m\|^2)} \qquad \text{Jensen Inequality} \tag{58}$$

$$= \frac{2}{M}\sqrt{\sum_{m=1}^{M}\sum_{r=1}^{M} \mathbb{E}(\langle z_m, z_r\rangle)} \tag{59}$$

$$= \frac{2}{M}\sqrt{\sum_{m=1}^{M} \mathbb{E}(\|z_m\|^2)} \tag{60}$$

$$\le \frac{2}{M}\sqrt{\sum_{m=1}^{M} \mathbb{E}(B^2N^2)} \tag{61}$$

$$= \frac{2}{M}\sqrt{\sum_{m=1}^{M} B^2N^2} \tag{62}$$

$$= \frac{2}{M}\sqrt{MB^2N^2} \tag{63}$$

$$= \frac{2BN}{\sqrt{M}} \tag{64}$$

This result can be used directly on the bound, after averaging over all the sampling.

$$\mathbb{E}(\mathcal{E}(T_{s+1})) \leq \mathbb{E}(\mathcal{E}(T_s)) - 2\alpha\mathbb{E}(G(T_s)) + \sqrt{\frac{2}{N}}\alpha\mathbb{E}(\|\nabla\mathcal{E}(T_s) - 2\widehat{\Lambda}_s\|) + \alpha 2\epsilon\log(N) + BN^2\alpha^2\sqrt{\frac{2}{N}}^2 \tag{65}$$

$$\leq \mathbb{E}(\mathcal{E}(T_s)) - 2\alpha\mathbb{E}(G(T_s)) + \sqrt{\frac{2}{N}}\alpha\frac{2BN}{\sqrt{M}} + \alpha 2\epsilon\log(N) + BN^2\alpha^2\sqrt{\frac{2}{N}}^2 \tag{66}$$

Thus,

$$2\alpha\mathbb{E}(G(T_s)) \leq \mathbb{E}(\mathcal{E}(T_s)) - \mathbb{E}(\mathcal{E}(T_{s+1})) + 2\sqrt{\frac{2}{N}}\alpha\frac{BN}{\sqrt{M}} + \alpha 2\epsilon\log(N) + BN^2\alpha^2\sqrt{\frac{2}{N}}^2. \tag{67}$$

We set sum over all s on both side.

$$2\alpha\sum_{s=0}^{S-1}\mathbb{E}(G(T_s)) \leq \mathcal{E}(T_0) - \mathbb{E}(\mathcal{E}(T_{S-1})) + S2\sqrt{\frac{2}{N}}\alpha\frac{BN}{\sqrt{M}} + S\alpha 2\epsilon\log(N) + SBN^2\alpha^2\sqrt{\frac{2}{N}}^2 \tag{68}$$

$$\leq \mathcal{E}(T_0) - \mathbb{E}(\mathcal{E}(T_{S-1})) + S2\sqrt{\frac{2}{N}}\alpha\frac{BN}{\sqrt{M}} + S\alpha 2\epsilon\log(N) + SBN^2\alpha^2\sqrt{\frac{2}{N}}^2 \tag{69}$$

We use the definition of $\overline{T}$ for $G(\overline{T})$. Notice that the following line is correct only on average for the random variable $\overline{T}$. This part is not clearly specified in the original proof of Reddi et al., (2016). We use also the definition of T^* for the second inequality.

$$\mathbb{E}(G(\overline{T})) \leq \frac{\mathcal{E}(T_0) - \mathbb{E}(\mathcal{E}(T_{S-1}))}{2S\alpha} + \sqrt{\frac{2}{N}}\frac{BN}{\sqrt{M}} + \epsilon\log(N) + BN^2\alpha\sqrt{\frac{1}{N}}^2 \tag{70}$$

$$\leq \frac{\mathcal{E}(T_0) - \mathcal{E}(T^*)}{2S\alpha} + B\sqrt{\frac{2N}{M}} + \epsilon\log(N) + BN\alpha \tag{71}$$

We derive the function $f(\alpha) = \frac{\mathcal{E}(T_0)-\mathcal{E}(T^*)}{2S\alpha} + B\sqrt{\frac{2N}{M}} + \epsilon\log(N) + BN\alpha$ with respect to α.

$$\frac{df}{d\alpha}(\alpha) = 0 \iff -\frac{\mathcal{E}(T_0) - \mathcal{E}(T^*)}{2S\alpha^2} + BN = 0 \tag{72}$$

$$\iff \frac{1}{\alpha^2} = \frac{2S}{\mathcal{E}(T_0) - \mathcal{E}(T^*)}BN \tag{73}$$

$$\iff \alpha = \sqrt{\frac{\mathcal{E}(T_0) - \mathcal{E}(T^*)}{2SBN}} \tag{74}$$

As $\mathcal{E}(T_0) - \mathcal{E}(T^*) \geq 0$, the second derivative is positive, thus f is convex, therefore we have the minimum. We can replace α and find the final bound,

$$\mathbb{E}(G(\overline{T})) \leq \frac{\mathcal{E}(T_0)-\mathcal{E}(T^*)}{2S\sqrt{\frac{\mathcal{E}(T_0)-\mathcal{E}(T^*)}{2BSN}}} + B\sqrt{\frac{2N}{M}} + \epsilon\log(N) + BN\sqrt{\frac{\mathcal{E}(T_0)-\mathcal{E}(T^*)}{2BSN}} \tag{75}$$

$$\leq \frac{\mathcal{E}(T_0)-\mathcal{E}(T^*)}{2S}\sqrt{\frac{2BSN}{\mathcal{E}(T_0)-\mathcal{E}(T^*)}} + B\sqrt{\frac{2N}{M}} + \epsilon\log(N) + \sqrt{\frac{(\mathcal{E}(T_0)-\mathcal{E}(T^*))BN}{2S}} \tag{76}$$

$$\leq \sqrt{\frac{(\mathcal{E}(T_0)-\mathcal{E}(T^*))BN}{2S}} + B\sqrt{\frac{2N}{M}} + \epsilon\log(N) + \sqrt{\frac{(\mathcal{E}(T_0)-\mathcal{E}(T^*))BN}{2S}} \tag{77}$$

$$\leq \sqrt{\frac{2(\mathcal{E}(T_0)-\mathcal{E}(T^*))BN}{S}} + B\sqrt{\frac{2N}{M}} + \epsilon\log(N). \tag{78}$$

□

We will now prove the second theorem using the same proof.

Theorem 4 *With the same notations as in Theorem 3 with the entropy ϵ_s that may now change along the iterations, when L yields a concave GW problem the following bound holds:*

$$\mathbb{E}\left(G\left(\overline{T}\right)\right) \leq \frac{\mathcal{E}(T_0)-\mathcal{E}(T^*)}{2S} + B\sqrt{\frac{2N}{M}} + \frac{1}{S}\sum_{s=0}^{S-1}\epsilon_s\log(N)$$

Proof The first difference is line 68, the sum from 0 to $S-1$ cannot be changed to S as ϵ_s may change along the iterations and is now a sum over $\sum_{s=0}^{S-1}$. The second difference is in Lemma 2, were the last term disappear as GW is concave. Thus, in line 71, as the last term is not present the optimal value of α is 1, which gives the proposed bound. □

A.2: A KL regularization-based variant

In this section we will discuss the convergence of the KL variant. A related convergence proof is proposed in Xu et al., (2019b) were the authors aim at solving GW using a proximal points method,

$$\min_{T\in\Pi_{ab}} \mathcal{E}(T,T) + \epsilon KL(T||T^n).$$

However it does not cover our case were we optimize $\min_{T\in\Pi_{ab}} \mathcal{E}(T,T^n) + \epsilon KL(T||T^n)$ at each iteration with the expectation approximated by a sampling. Without sampling, this optimization can be seen as a Majorization-Minization method (Sun et al., 2016),

$$\mathcal{E}(T) \leq \mathcal{E}(T^n) + \langle\nabla\mathcal{E}(T^n), T-T^n\rangle + \frac{2BN^2}{2}\|T-T^n\|^2 \tag{79}$$

$$\leq \mathcal{E}(T^n) + \langle\nabla\mathcal{E}(T^n), T-T^n\rangle + BN^2|T-T^n|_1^2 \tag{80}$$

$$\leq \mathcal{E}(T^n) + \langle \nabla \mathcal{E}(T^n), T - T^n \rangle + BN^2 KL(T||T^n). \tag{81}$$

Were the first line is the line 41 in the proof of Theorem 3. The second line use the fact that the L2 norm is bigger than the L1 norm. The last line uses the Pinsker's inequality.

While the last inequality seems to be a good starting point, we could not directly derive (or find in the literature) a bound that applies with sampling and the KL term (that makes the use of Sinkhorn-Knopps possible). Thus, while this interpretation seems interesting, the question of the convergence is left open and would need to be studied in a future work.

A.3: Approximating the Gromov Wasserstein distance

This section gives mathematical details for the estimation of the Gromov Wasserstein distance from a given transport plan. Our approach to compute the GW distance will take inspiration from the idea of sampling $T \in \Pi_{ab}$ (i.e., with marginals a and b).

Let define a new random variable $P(\mathbf{R} = L_{ijkl}) = T_{ij}T_{kl}$. This definition is not totally rigorous: two values L_{ijkl} and $L_{i'j'k'l'}$ may be equal, the actual probability is then the sum of the probabilities. The GW distance can now be seen as an expectation,

$$\sum_{ijkl} L_{ijkl} T_{ij} T_{kl} = \mathbb{E}(\mathbf{R}). \tag{82}$$

Instead of simply sampling this expectation, we propose to stratify by each index i, k to improve the quality of the estimator. Let U_i be the event "*i is chosen for the first dimension of L*" and U'_k be the event "*k is chosen for the third dimension of L*". Using the rule of total expectation, the expectation can be transformed to,

$$\mathbb{E}(\mathbf{R}) = \sum_{ik} \mathbb{P}(U_i \cap U'_k) \mathbb{E}(\mathbf{R} | U_i \cap U'_k) \tag{83}$$

$$= \sum_{ik} a_i a_k \mathbb{E}(\mathbf{R} | U_i \cap U'_k). \tag{84}$$

For any $(i, k) \in [\![1, N]\!]^2$, we denote as X_{ik} the random variable defined by: $\mathbb{P}(X_{ik} = L_{ijkl}) = \mathbb{P}(\mathbf{R} = L_{ijkl} | U_i \cap U'_k)$. Thus, we use $\hat{\mathbf{R}} = \sum_{ik} a_i a_k \frac{1}{M} \sum_{m=1}^{M} X_{ik}^m$ to estimate the Gromov Wasserstein distance. This estimator is unbiased and comes with a tight estimator of the standard deviation as shown on the Fig. 3 of the paper,

$$\hat{\sigma}_{\hat{\mathbf{R}}} = \sqrt{\sum_{ik} a_i^2 a_k^2 \frac{1}{M-1} \sum_{m=1}^{M} \left(X_{ik}^m - \left(\frac{1}{M} \sum_{m'=1}^{M} X_{ik}^{m'} \right) \right)^2}. \tag{85}$$

We recommend to take at least $M = 2$, to have access to the standard deviation. Note that in theory we could only look at a sub-sample of the index i, k ($\sqrt{N} \log(N)$ instead of all the N points), to have an approximation of the distance in $N \log(N)$. This might be useful when coupled with Pointwise Gromov Wasserstein, however the predicted distance might be far from the real one without any standard deviation to quantify the error.

Appendix B: Experiments

B.1: General setup and methods

We remind that the code to reproduce all the experiments, figures and tables is available in the GitHub repository: https://github.com/Hv0nnus/Sampled-Gromov-Wasserstein.

Gaussians mixtures

This section explains how the gaussians mixtures are created with a Gaussian Random Partition Graph (Brandes et al., 2003) based on Stochastic Block Model (Holland et al., 1983). The Algorithm 2 describe how to sample N points. This algorithm will create some gaussians separated from each other and some values will be sample from those gaussians.

For the experiment, the dimension space d is set to 10 and 20 for the distributions μ and ν. The Euclidean distance is used on both spaces to compute $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$.

Algorithm 2 Gaussians mixtures dataset

```
Require: $N$, $d$, $v_{in} = 2$, $v_{out} = 10$, $v_{points} = 5$, $c = \min(200, \frac{N}{2})$
1: $n = 0$
2: $Positions = [\,]$
3: while $n < N$ do
4:     $mean = \mathcal{N}(\mathbf{0}, v_{out} \times \mathbf{I}_d)$
5:     $n' = \mathcal{N}(c, v_{points})$
6:     Add $n'$ sample of $\mathcal{N}(mean, v_{in} \times \mathbf{I}_d)$ to $Positions$
7:     $n = n + n'$
8: end while
9: return $N$ first points of $Positions$
```

Gaussian Random Partition Graph

For the second experiment, we generate graphs using a Gaussian Random Partition Graph (Brandes et al., 2003) with intra-cluster probability of 0.5, extra-cluster probability of 0.1, the number of nodes in each cluster is sampled from a Gaussian with mean $min(\frac{N}{2}, 200)$ and a variance of 5. The adjacency matrix of each graph is used for $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$. We set a and b to the uniform distribution.

B.2: Speed and accuracy of the GW estimate

We reproduce the figures available in the paper in Figs. 10 and 11. The left part of Fig. 11 is omitted from the paper: it shows similar time complexity compared to the left part of Fig. 10.

Figure 12 shows that the computation time iclearly different with the square loss. To facilitate the comparison, we keep, for every method, the same hyperparameter used for the absolute loss. Those parameters may not be optimal, especially S-GWL which seems to perform poorly. On the left Fig. 12, SGW is faster that PoGroW because the distance can

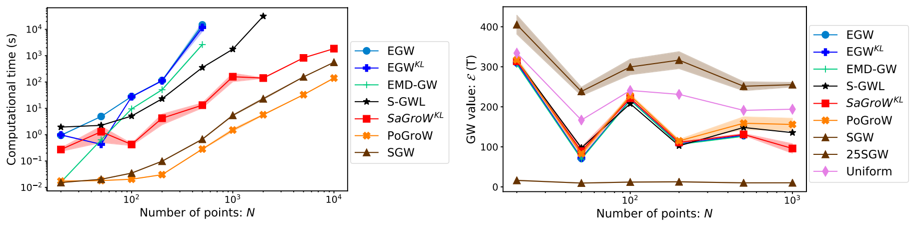

Fig. 10 Computational time (left) and GW distance estimation (right) between points sampled from mixtures of gaussians

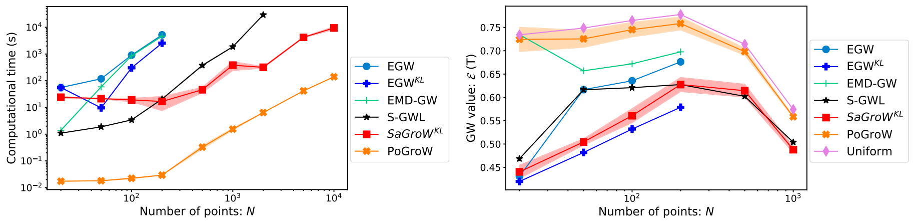

Fig. 11 Computational time (left) and GW distance estimation (right) on synthetic graphs (Brandes et al., 2003)

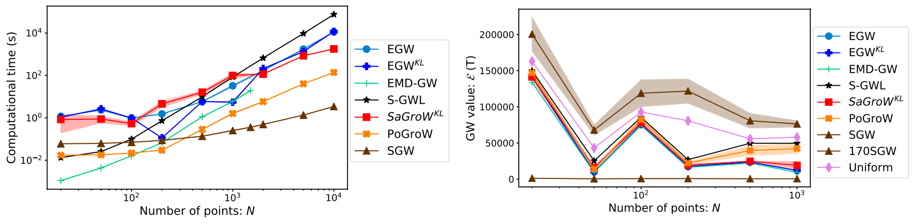

Fig. 12 Computational time (left) and GW distance estimation (right) between sampled points from mixtures of gaussians with the square loss

be computed in $O(N\log(N))$, and the entire algorithm is efficiently parallelized. Because of its $O\left(N^2\right)$ complexity, SaGroW is still faster than S-GWL and EGW for a high number of points.

B.3: Hyperparameter analysis

In this section, we plot figures similar to Figs. 7, 8 and 9 from the paper.

Figure 13 shows the difference between the square loss and the absolute one, to compare computational times. While our method remains the same, the other method improve their computational time.

Figure 14 shows similar expected behavior for a graphs dataset.

Figure 15 shows a very easy situations, where every method probably finds the right GW distance. In this case PoGroW is very competitive even for the square loss.

Figure 16 highlights the interest of our method even for a very small N (20 nodes in each graphs). In this case, SaGroW obtains the best transport plan for the square loss.

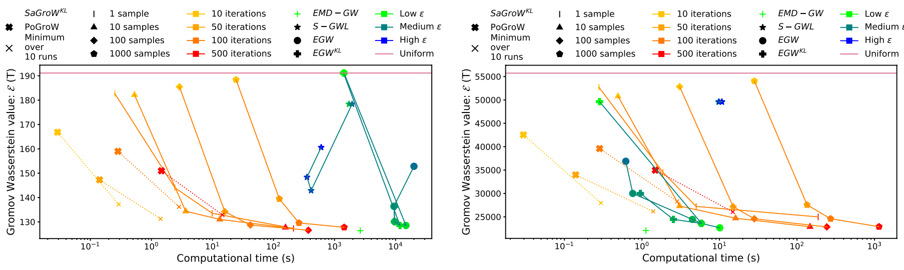

Fig. 13 Similar to the Figs. 7, 8 and 9 in the paper. (Left) Absolute loss. (Right) Square loss

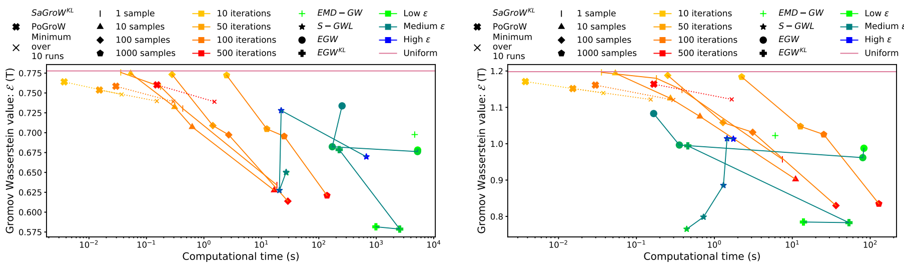

Fig. 14 Hyperparameters analysis on a Stochastic Block Model dataset with 200 nodes for each graphs. (Left) Absolute loss. (Right) Square loss

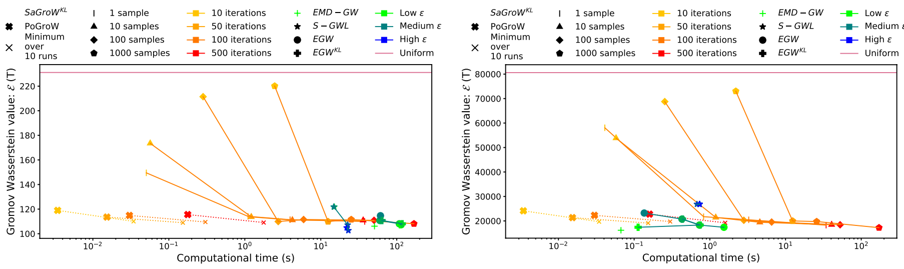

Fig. 15 Hyperparameters analysis on a mixture of Gaussians with 200 points sampled for each distributions. (Left) Absolute loss. (Right) Square loss

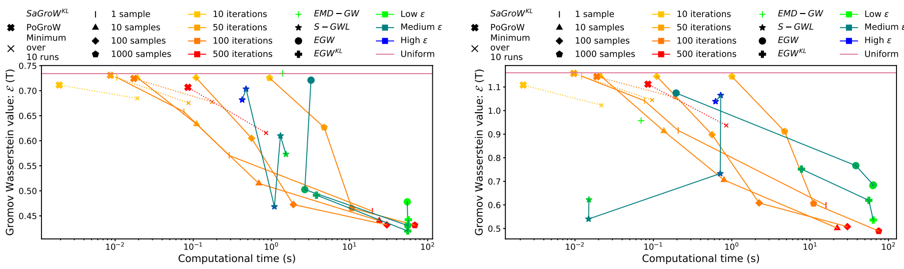

Fig. 16 Hyperparameters analysis on a Stochastic Block Model dataset with only 20 nodes for each graphs. (Left) Absolute loss. (Right) Square loss

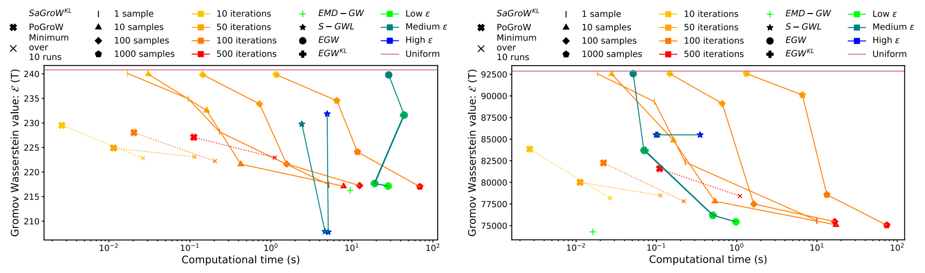

Fig. 17 Hyperparameters analysis on a mixture of Gaussians with 100 points sampled for each distributions. (Left) Absolute loss. (Right) Square loss

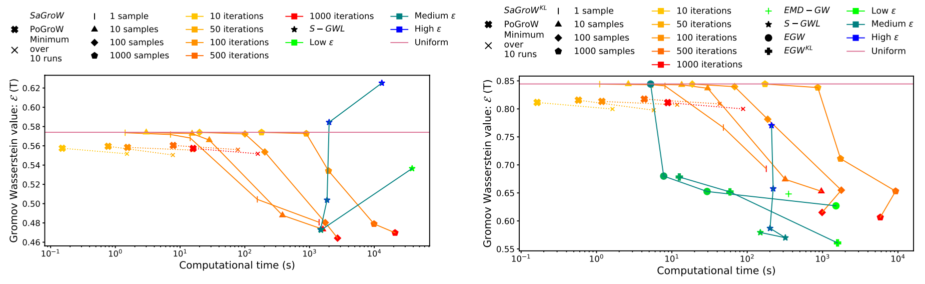

Fig. 18 Hyperparameters analysis on a Stochastic Block Model dataset with 1000 nodes for each graphs. (Left) Absolute loss. (Right) Square loss

Figure 17 (left) shows an interesting example when every methods seem stuck in the same local minima and S-GWL finds a better transport plan which is probably the global minima.

Lastly, Fig. 18 shows that even with 1000 iterations, SaGroW doesn't seem to converge. The value of ϵ is too high in this case and needed to be lowered to avoid too much iterations. However, SaGroW still obtains a better plan than S-GWL for the absolute loss.

Small experiment on SaGroW without the KL regularization

In this experiment, we replace SaGroW^{KL} by SaGroW and reproduce Figs. 7, 8 and 9 in the paper. We use $\epsilon = 0.1$ for this experiment and $\alpha = 0.8$. The Fig. 19 shows that the value of $\epsilon = 0.1$ is too high on this dataset. Section B.5 highlights the difficulty to chose a good value of entropy regularization while the KL regularization is much more robust to this choice.

As the number of sample increases, the performance of SaGroW tend to EGW$_{0.1}$, which is the left-most point. This behaviour is expected as SaGroW become similar to EGW when the expectation is completely computed. However, the performance improves slowly with the number of iterations. This might be due to the lack of memory from one iteration to the other, as the transport plan T_s may vary a lot between two iterations. This illustrates the advantage of the KL regularization which completely take into account the previous transport plan.

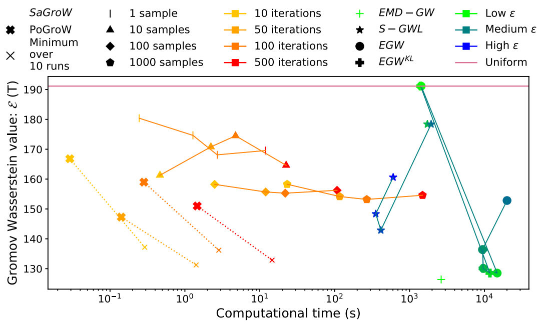

Fig. 19 Similar to the Figs. 7, 8 and 9 in the paper, with SaGroW^{KL} replaced by SaGroW

Table 3 Gromov Wasserstein estimation for different values of ϵ

ϵ	SaGroW^{KL}	EGW
0.001	0.73	0.75
0.005	0.59	0.63
0.01	0.55	0.62
0.05	0.51	0.67
0.1	0.51	0.71
0.5	0.52	0.75
1	0.52	0.75
5	0.62	0.75
10	0.68	0.75

The dataset is composed of 2 graphs created with a Gaussian Random Partition Graph (Brandes et al., 2003) with 50 points each. The mean cluster size is set to 25 and the variance to 5. The probability of intra-cluster connection is 0.5 while the inter-cluster is set to 0.1. The GW distances reported are averaged over 10 iterations. The absolute distance is used for $\mathcal{L}$. The number of iteration of SaGroW^{KL} is 1000 with one sample per iteration

Small experiment on the entropy parameter

Table 3 shows that EGW is really sensitive to the entropy regularization, with only 4 values of ϵ that give a Gromov Wasserstein distance different from 0.75. This value of 0.75 corresponds to the uniform matrix. In contrast, due to the KL regularization instead of the classical entropy, SaGroW^{KL} never returns the uniform matrix. Moreover, SaGroW^{KL} gives a reasonable value for a large range of parameters (from 0.05 to 1).

The entropy regularization ensure to stay close to the uniform. Thus, for high value of ϵ it will always stay close to the uniform. The KL regularization ensure than the next value will be close to the previous one. In such a case, with enough iteration we can still converge to a local minima. This is intuition was given in Xu et al., (2019b) based on Xie et al., (2020).

Small experiment on the α parameter

Tables 4 and 5 analyze the impact of α and the number of iterations. The most important information is that a high value of α seems a good choice. A high value of α ensure to

Table 4 Gromov Wasserstein distance for PoGroW with different values of α and different number of iterations

S\α	0.001	0.01	0.1	0.25	0.5	0.75	0.9	0.99	0.999	1
1	74.87	74.87	74.82	74.63	74.03	73.07	72.33	71.82	71.77	71.76
10	74.87	74.82	74.73	74.52	74.88	74.69	72.28	71.84	71.80	71.79
100	74.82	74.35	73.36	73.94	74.67	68.34	68.76	71.91	71.93	71.94
1000	74.37	73.76	72.90	73.32	73.09	72.12	70.62	72.97	73.01	70.69

The dataset is composed of 2 graphs created with a Gaussian Random Partition Graph (Brandes et al., 2003) with 50 points. The absolute distance is used for $\mathcal{L}$

Table 5 Gromov Wasserstein distance for PoGroW with different values of α and different number of iterations

S\α	0.001	0.01	0.1	0.25	0.5	0.75	0.9	0.99	0.999	1
1	166.88	166.87	166.28	163.53	154.03	138.40	126.08	117.62	116.73	116.63
10	166.83	166.37	163.81	160.98	117.23	85.41	85.27	85.24	79.84	79.84
100	166.48	163.31	105.36	90.42	137.56	76.93	79.02	73.80	73.64	73.62
1000	163.38	134.57	86.59	80.47	79.43	78.28	77.45	77.14	77.11	77.11

The dataset is composed of 2 samples of 50 points of mixtures of gaussians. The absolute distance is used for $\mathcal{L}$

always be close to the edge of the polytope, were the optimal value is assumed to be. For a concave problem (Table 5), the best value to choose is 1. In Table 4 the best value around 0.75 - 0.9. Thus, it might not be very interesting to cross validate this parameter, and a value around 0.8 seems a reasonable choice.

On average, it is better to apply many iterations. This is especially true for small value of α were the GW distance changes very slowly. We see on this experiment, the limit of the convergence proof. In practice, we will never use a small value of α, even if the convergence is ensured.

Graph classification

Table 6 gives the complete table of the graphs classification experiment. The best parameter taken for each of the method is not on the edge on the parameter range. Thus, a good parameter is found for each method. Notice that the performance of PoGroW are very similar for different value of power p. This can be explain by the fact that the transport plan found does not depend on the loss used. The 1D optimal transport plan is the same for all convex loss functions. For the case $p = 0.5$, PoGroW does not find the perfect transport plan at each iteration as we suppose the loss convex. For $p = 1$ the problem might be degenerated, many different transport plans can be optimal. We can suppose p slightly higher than one to avoid the problem. Caracciolo et al., (2020) proposes a bound for the 1D OT concave case.

Other losses than the absolute loss at power p have been tested. Only the exponential square $(1 - e^{-(C^{\mathcal{X}}-C^{\mathcal{Y}})^2})$ has a reasonable accuracy.

Table 6 Complete table of the classification experiment in the paper

Dataset	Accuracy	GW Distance	Time (s)
S-GWL$_{0.005}$	0.1	400	14.7
S-GWL$_{0.01}$	0.1	400	13.9
S-GWL$_{0.05}$	0.1	400	13.9
S-GWL$_{0.1}$	0.1	400	12.6
S-GWL$_{0.5}$	0.17	390	12.0
S-GWL$_{1}$	0.29	374	11.0
S-GWL$_{5}$	0.44	362	23.4
S-GWL$_{10}$	0.41	377	27.8
S-GWL$_{50}$	0.41	374	34.1
S-GWL$_{100}$	0.39	372	33.2
EGW$_{0.0001}$	0.07	430	0.1
EGW$_{0.0005}$	0.07	429	4.5
EGW$_{0.001}$	0.22	412	36.2
EGW$_{0.005}$	0.22	375	42.6
EGW$_{0.01}$	0.12	383	25.1
EGW$_{0.05}$	0.15	408	6.2
EGW$_{0.1}$	0.12	420	2.0
EGW$_{0.5}$	0.07	429	0.3
EGW$_{1}$	0.07	429	0.3
EGW$^{KL}_{0.0001}$	0.07	430	0.1
EGW$^{KL}_{0.0005}$	0.07	429	0.1
EGW$^{KL}_{0.001}$	0.15	419	0.2
EGW$^{KL}_{0.005}$	0.24	375	41.1
EGW$^{KL}_{0.01}$	0.12	383	25.8
EGW$^{KL}_{0.05}$	0.15	408	7.1
EGW$^{KL}_{0.1}$	0.12	420	2.4
EGW$^{KL}_{0.5}$	0.07	429	0.4
EGW$^{KL}_{1}$	0.07	429	0.4
EMD-GW	0.37	341	16.6
SaGroW$_{p=0.5}$	0.41		12.3
SaGroW$_{p=1}$	**0.49**		11.6
SaGroW$_{p=1.5}$	**0.49**		13.6
SaGroW$_{p=2}$	0.39	**336**	12.7
SaGroW$_{p=2.5}$	0.37		12.3
SaGroW$_{p=3}$	0.27		11.8
SaGroW $(1 - e^{-\lvert C^{\mathcal{X}} - C^{\mathcal{Y}}\rvert})$	0.05		0.7
SaGroW $(1 - e^{-\frac{\lvert C^{\mathcal{X}} - C^{\mathcal{Y}}\rvert}{10}})$	0.05		0.8
SaGroW $(1 - e^{-(C^{\mathcal{X}} - C^{\mathcal{Y}})^2})$	0.27		13.9
SaGroW $(1 - e^{\frac{(C^{\mathcal{X}} - C^{\mathcal{Y}})^2}{100}})$	0.05		0.8
PoGroW$_{p=0.5}$	0.37		0.5
PoGroW$_{p=1}$	0.37		0.5
PoGroW$_{p=1.5}$	0.37		0.5
PoGroW$_{p=2}$	0.39	373	0.5
PoGroW$_{p=2.5}$	0.32		0.6

Table 6 (continued)

Dataset	Accuracy	GW Distance	Time (s)
PoGroW$_{p=3}$	0.27		0.5
PoGroW $(1 - e^{-(C^{\mathcal{X}}-C^{\mathcal{Y}})^2})$	0.2		0.5
PoGroW $(1 - e^{\lvert C^{\mathcal{X}}-C^{\mathcal{Y}}\rvert})$	0.05		0.5
PoGroW $(1 - e^{\frac{\lvert C^{\mathcal{X}}-C^{\mathcal{Y}}\rvert}{10}})$	0.05		0.5
PoGroW $(\frac{1}{e^{-(C^{\mathcal{X}}-C^{\mathcal{Y}})^2}} - 1)$	0.1		0.2
Uniform	0.07	430	0.1

References

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 214–223).

Beckman, M., & Koopmans, T. (1957). Assignment problems and the location of economic activities. *Econometrica, 25,* 53–76.

Blondel, M., Seguy, V., & Rolet, A. (2018). Smooth and sparse optimal transport. In *International conference on artificial intelligence and statistics* (pp. 880–889), PMLR.

Bonneel, N., Van De Panne, M., Paris, S., & Heidrich, W. (2011). Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference* (pp. 1–12).

Brandes, U., Gaertler, M., & Wagner, D. (2003). Experiments on graph clustering algorithms. In *European symposium on algorithms* (pp. 568–579), Springer.

Bronstein, A. M., Bronstein, M. M., Kimmel, R., Mahmoudi, M., & Sapiro, G. (2010). A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision, 89*(2–3), 266–286.

Bunne, C., Alvarez-Melis, D., Krause, A., & Jegelka, S. (2019). Learning generative models across incomparable spaces. In *International conference on machine learning* (pp. 851–861).

Caracciolo, S., D'Achille, M. P., Erba, V., & Sportiello, A. (2020). The dyck bound in the concave 1-dimensional random assignment model. *Journal of Physics A Mathematical and Theoretical 53*(6), 064001.

Chowdhury, S., & Mémoli, F. (2019). The gromov-wasserstein distance between networks and stable network invariants. *Information and Inference: A Journal of the IMA 8*(4), 757–787.

Courty, N., Flamary, R., & Tuia, D. (2014). Domain adaptation with regularized optimal transport. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 274–289), Springer.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems* (pp. 2292–2300).

Cuturi, M., Teboul, O., & Vert, J. P. (2019). Differentiable ranking and sorting using optimal transport. In *Advances in neural information processing systems* (pp. 6861–6871).

Delon, J., Salomon, J., & Sobolevski, A. (2012). Local matching indicators for transport problems with concave costs. *SIAM Journal on Discrete Mathematics 26*(2), 801–827.

Ezuz, D., Solomon, J., Kim, V. G., & Ben-Chen, M. (2017). Gwcnn: A metric alignment layer for deep shape analysis. *Computer Graphics Forum 36,* 49–57.

Frank, M., Wolfe, P., et al. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly, 3*(1–2), 95–110.

Gelfand, N., Mitra, N. J., Guibas, L.J., & Pottmann, H. (2005). Robust global registration. In *Symposium on geometry processing* (Vol. 2, pp. 5), Vienna, Austria.

Genevay, A., Chizat, L., Bach, F., Cuturi, M., & Peyré, G. (2019). Sample complexity of sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics* (pp. 1574–1583).

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks, 5*(2), 109–137.

Kantorovich, L. (1942). On the transfer of masses. In *Dokl Acad Nauk USSR* (Vol. 37, pp. 7–8).

Kersting, K., Kriege, N. M., Morris, C., Mutzel, P., & Neumann, M. (2016). Benchmark data sets for graph kernels. http://graphkernels.cs.tu-dortmund.de.

Memoli, F. (2007). On the use of Gromov-Hausdorff distances for shape comparison. In: Botsch M, Pajarola R, Chen B, Zwicker M (eds) *Eurographics symposium on point-based graphics*. The Eurographics Association. https://doi.org/10.2312/SPBG/SPBG07/081-090.

Mémoli, F. (2009). Spectral gromov-wasserstein distances for shape matching. In *IEEE 12th international conference on computer vision workshops* (pp. 256–263). IEEE: ICCV Workshops.

Mémoli, F. (2011). Gromov-wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics 11*(4), 417–487.

Neumann, M., Moreno, P., Antanas, L., Garnett, R., & Kersting, K. (2013). Graph kernels for object category prediction in task-dependent robot grasping. In *Online proceedings of the eleventh workshop on mining and learning with graphs* (pp. 0–6).

Paty, F. P., & Cuturi, M. (2019). Subspace robust wasserstein distances. In *International conference on machine learning* (pp. 5072–5081), PMLR.

Pele, O., & Werman, M. (2009). Fast and robust earth mover's distances. In *2009 IEEE 12th international conference on computer vision* (pp. 460–467), IEEE.

Peyré, G., Cuturi, M., & Solomon, J. (2016). Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning* (pp. 2664–2672).

Rabin, J., & Peyré, G. (2011). Wasserstein regularization of imaging problem. In *2011 18th IEEE international conference on image processing* (pp. 1541–1544), IEEE.

Rangarajan, A., Yuille, A., & Mjolsness, E. (1999). Convergence properties of the softassign quadratic assignment algorithm. *Neural Computation, 11*(6), 1455–1474.

Reddi, S. J., Sra, S., Póczos, B., & Smola, A. (2016). Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th annual Allerton conference on communication, control, and computing (Allerton)* (pp. 1244–1251), IEEE.

Redko, I., Vayer, T., Flamary, R., & Courty, N. (2020). Co-optimal transport. In *NeurIPS 2020-thirty-four conference on neural information processing systems*.

Sato, R., Cuturi, M., Yamada, M., & Kashima, H. (2020). Fast and robust comparison of probability measures in heterogeneous spaces. arXiv preprint arXiv:200201615.

Solomon, J., Peyré, G., Kim, V. G., & Sra, S. (2016). Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG), 35*(4), 1–13.

Sun, Y., Babu, P., & Palomar, D. P. (2016). Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing 65*(3), 794–816.

Vayer, T., Chapel, L., Flamary, R., Tavenard, R., & Courty, N. (2018). Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties. arXiv preprint arXiv:181102834.

Vayer, T., Chapel, L., Flamary, R., Tavenard, R., & Courty, N. (2019a). Optimal transport for structured data with application on graphs. In *ICML 2019-36th international conference on machine learning* (pp. 1–16).

Vayer, T., Flamary, R., Tavenard, R., Chapel, L., & Courty, N. (2019b). Sliced gromov-wasserstein. In *NeurIPS 2019-thirty-third conference on neural information processing systems* (vol. 32).

Villani, C. (2008). *Optimal transport: old and new*. Springer.

Wright, M.H. (1996). Direct search methods: Once scorned, now respectable. Pitman Research Notes in Mathematics Series (pp. 191–208).

Xie, Y., Wang, X., Wang, R., & Zha, H. (2020). A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence* (pp. 433–453), PMLR.

Xu, H., Luo, D., & Carin, L. (2019a). Scalable gromov-wasserstein learning for graph partitioning and matching. In *Advances in neural information processing systems* (pp. 3046–3056).

Xu, H., Luo, D., Zha, H., & Duke, L.C. (2019b). Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning* (pp. 6932–6941).

Yan, Y., Li, W., Wu, H., Min, H., Tan, M., & Wu, Q. (2018). Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI* (pp. 2969–2975).