Uncovering Challenges of Solving the Continuous Gromov-Wasserstein Problem

Xavier Aramayo-Carrasco
Applied AI Institute
Moscow, Russia
xavier.aramayo2@gmail.com

Maksim Nekrashevich*
Nebius AI
Amsterdam, Netherlands

Petr Mokrov
Applied AI Institute
Moscow, Russia

Evgeny Burnaev
Applied AI Institute, AXXX
Moscow, Russia

Alexander Korotin
Applied AI Institute, AXXX
Moscow, Russia

Abstract

Recently, the Gromov-Wasserstein Optimal Transport (GWOT) problem has attracted the special attention of the ML community. In this problem, given two distributions supported on two (possibly different) spaces, one has to find the most isometric map between them. In the discrete variant of GWOT, the task is to learn an assignment between given discrete sets of points. In the more advanced continuous formulation, one aims at recovering a parametric mapping between unknown continuous distributions based on i.i.d. samples derived from them. The clear geometrical intuition behind the GWOT makes it a natural choice for several practical use cases, giving rise to a number of proposed solvers. Some of them claim to solve the continuous version of the problem. At the same time, GWOT is notoriously hard, both theoretically and numerically. Moreover, all existing continuous GWOT solvers still heavily rely on discrete techniques. Natural questions arise: to what extent do existing methods unravel the GWOT problem, what difficulties do they encounter, and under which conditions they are successful? Our benchmark paper is an attempt to answer these questions. We specifically focus on the continuous GWOT as the most interesting and debatable setup. We crash-test existing continuous GWOT approaches on different scenarios, carefully record and analyze the obtained results, and identify issues. Our findings experimentally testify that the scientific community is still missing a reliable continuous GWOT solver, which necessitates further research efforts. As a first step in this direction, we propose a new continuous GWOT solver specifically designed for the inner-product setting, which does not rely on discrete techniques and addresses some of the limitations of existing approaches. Code available at: https://github.com/Ark-130994/GWBench.git

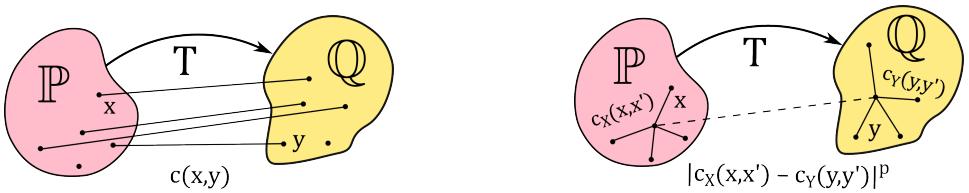
1 Introduction

Optimal Transport (OT) is a powerful framework that is widely used in machine learning (Montesuma et al., 2023). A popular application of OT is the domain adaptation of various modalities, including images (Courty et al., 2016; Luo et al., 2018; Redko et al., 2019), music transcription (Flamary et al., 2016), color transfer (Frigo et al., 2015), alignment of embedding spaces (Chen et al., 2020; Aboagye et al., 2022). Other applications include generative models (Salimans et al., 2018; Arjovsky et al., 2017), unpaired imgage-to-image transfer (Korotin et al., 2023b; Rout et al., 2022), etc.

In the conventional OT problem (Figure 1a), one needs to find a map between two data distributions that minimizes a certain "effort" expressed in the form of an *inter-domain* transport cost function. This cost function shows how hard it is to move a point in the source space to a given point in the

*Work was done while at Applied AI Institute

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Monge's OT between distributions $\mathbb{P}$ and $\mathbb{Q}$ with *inter-domain* cost function $c(x, y)$.

(b) Monge's GW between distributions $\mathbb{P}$ and $\mathbb{Q}$ with *intra-domain* costs $c_{\mathcal{X}}(x, x')$ and $c_{\mathcal{Y}}(y, y')$.

Figure 1: A schematic visualization of the OT problems and GW problems (Monge's form).

target space. Thus, in order for the resulting map to possess certain useful properties, one has to incorporate them into the cost function. Unfortunately, this is not always a straightforward task, especially when the data distributions are supported in different spaces.

A popular way to address the above-mentioned issue is to consider the Gromov-Wasserstein (GW) modification (Memoli, 2007; Mémoli, 2011; Peyré et al., 2016) of the OT problem (Figure 1b). Here one assumes that both the source and target spaces are equipped with a structure, e.g., with a metric, and one aims to find a transport map that maximally preserves this structure, i.e., the most isometric map. This clear geometrical intuition behind GW makes it natural in various applications: unsupervised data alignment (Alvarez-Melis & Jaakkola, 2018; Aboagye et al., 2022), single-cell data processing (Scetbon et al., 2022; Klein et al., 2023; Sebbouh et al., 2024; Demetci et al., 2024; Ryu et al., 2024), 2D and 3D shape analysis (Beier et al., 2022; Mémoli, 2009), graph data analysis (Xu et al., 2019, 2021; Vincent-Cuaz et al., 2022; Chowdhury & Needham, 2021; Vincent-Cuaz et al., 2021; Mazelet et al., 2025) and dimensionality reduction (Assel et al., 2024).

Discrete/continuous GW. The GWOT problem is about learning some specific translation that operates with source and target distributions. In practice, these distributions are typically given by empirical datasets. This leads to two possible ways of paving a GWOT map. In the **discrete** scenario, the learned translation is just a point(s)-to-point(s) assignment (transport matrix). In turn, the **continuous** GW is about learning a parametric mapping between the underlining (continuous) distributions. In this case, the datasets are treated as i.i.d. samples derived from them.

While existing computational approaches for the GW problem show considerable empirical success, the problem itself is highly non-trivial from different perspectives.

- **Theory.** Finding the most isometric map between probability spaces based just on the inner properties of these spaces may be poorly defined, e.g., the desired transform may be non-unique. This happens where the source (target) space permits some non-trivial isometries that preserve the corresponding source (target) distribution. A simple yet expressive example is the Gaussian case Delon et al. (2022). Intuitively, non-uniqueness may affect the stability of a GWOT solver.
- **Computations/algorithms.** It is known that the discrete GW yields a non-convex quadratic optimization problem (Vayer, 2020), which is computationally challenging. To partially alleviate the difficulty, one typical approach is to consider entropic regularization (Peyré et al., 2016; Alvarez-Melis & Jaakkola, 2018; Scetbon et al., 2022; Wang & Goldfeld, 2023). Fortunately, the regularized problem resorts to a sequence of tractable Sinkhorn OT assignments. However, the convergence of the procedure may not hold, see (Peyré et al., 2016, Remark 3). In addition, discrete GWOT scales poorly with the number of input (source or target) samples, making some problem setups unmanageable by such solvers. While there are techniques to reduce the computational burden w.r.t. data size (Scetbon et al., 2022), they come at the cost of additional restrictions and assumptions.
- **Methodology.** Majority of existing continuous GW methods are based on discrete GW techniques and inherit their computational challenges. Moreover, the transition from the discrete to the continuous setup may be questionable from a statistical point of view (Zhang et al., 2024).

Having said that, one naturally wonders: how do the current continuous GWOT methods manage to overcome these problems and show good practical results? What are the "bad cases" under which the aforementioned difficulties become critical and the solvers struggle? How to fight with these "bad cases"? In our paper, we shed light on these GWOT methods' ambiguities, specifically focusing on the continuous setup for the inner product and cosine distance cost functionals. **Our contributions:**

- We conduct a deep analysis of existing papers and reveal that the majority of works primarily consider datasets with some specific statistical relations between source and target samples. Formally speaking, such setups disobey the standard i.i.d. assumption on the data and may lead to ambiguous conclusions on the solvers' capabilities.

- By following the findings from the previous point, we evaluate the performance of existing continuous inner product and cosine distance based GWOT solvers in more statistically fair and practically realistic *statistically unrelated* data setups. Our experiments witness that the level of statistical relatedness indeed highly influences the quality of the learned GWOT maps.
- Motivated by these findings, we propose a novel continuous neural GW solver for the inner-product setting that does not rely on statistical relatedness between source and target training samples. Our method is not based on discrete GW and it may be learned on arbitrarily large datasets. Since it can be trained on a large number of samples, the method exhibits reasonably good performance even in the challenging *statistically unrelated* setup. On the other hand, the method is $\min$-$\max$-$\min$ adversarial, which impacts stability and requires plenty of data for training.

Overall, our findings reveal that existing GWOT solvers can be effective for certain applications when statistically related data is available—a condition typically impractical for many real-world GW tasks. However, these methods fail to learn meaningful transport plans when this statistical relation is absent. To address this limitation, we propose a solver capable of handling fully statistically unrelated setups, enabling the GWOT approach to be practically useful in more realistic and challenging scenarios.

Notations. Throughout the paper, $\mathbb{R}^{d_x}$ and $\mathbb{R}^{d_y}$ are the source and target data spaces, respectively. The set of Borel probability distributions on $\mathbb{R}^{d_x}$ is $\mathcal{P}(\mathbb{R}^{d_x})$. The dot product of vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{d_x}$ is $\langle \mathbf{x}, \mathbf{x}' \rangle_{d_x}$. For a measurable map $T\colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$, we denote the corresponding *push-forward* operator by $T_\sharp$. For $\mathbb{P} \in \mathcal{P}(\mathbb{R}^{d_x})$ and $\mathbb{Q} \in \mathcal{P}(\mathbb{R}^{d_y})$, we denote the set of all couplings between them by $\Pi(\mathbb{P}, \mathbb{Q})$, i.e., distributions π on $\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ with the corresponding marginals equal to $\mathbb{P}$ and $\mathbb{Q}$.

2 Background

In this section, we first explain the conventional OT setup (Villani, 2008; Santambrogio, 2015; Gozlan et al., 2017; Backhoff-Veraguas et al., 2019) and then introduce the GWOT formulation (Mémoli, 2011; Peyré et al., 2016). Finally, we clarify the practical learning setup for these problems.

2.1 Optimal Transport (OT) problem

Given two probability distributions $\mathbb{P} \in \mathcal{P}(\mathbb{R}^{d_x})$, $\mathbb{Q} \in \mathcal{P}(\mathbb{R}^{d_y})$ and a cost function $c\colon \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$, the OT problem is defined as follows:

$$\text{OT}_c(\mathbb{P}, \mathbb{Q}) \stackrel{\text{def}}{=} \inf_{T_\sharp \mathbb{P} = \mathbb{Q}} \int_{\mathbb{R}^{d_x}} c(\mathbf{x}, T(\mathbf{x})) d\mathbb{P}(\mathbf{x}). \tag{1}$$

This is known as Monge's formulation of the OT problem. Intuitively, it can be understood as finding an optimal transport map $T^* : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ that transforms $\mathbb{P}$ to $\mathbb{Q}$ and minimizes the total transportation expenses w.r.t. cost c, see Figure 1a. There have been developed a lot of methods for solving OT (1) in its discrete (Cuturi, 2013; Peyré et al., 2019) and continuous (Makkuva et al., 2020; Daniels et al., 2021; Korotin et al., 2023b; Choi et al., 2023; Fan et al., 2023; Uscidda & Cuturi, 2023; Gushchin et al., 2024; Mokrov et al., 2024; Asadulaev et al., 2024) variants.

The cost function c in (1) is commonly the squared Euclidean distance, making problem (1) exclusively defined for spaces of equal dimension. Handling incomparable spaces ($d_x \neq d_y$) requires a manually defined inter-domain cost, which is a nontrivial task (Wang & Zhang, 2025).

2.2 Gromov-Wasserstein OT (GWOT) problem

The GWOT problem is an extension of the optimal transport problem that aims to compare and transport probability distributions supported on different spaces. This problem is particularly useful when the underlying spaces do not align directly, but we still want to measure and align their intrinsic geometric structures. In what follows, we introduce the discrete and continuous variants of GWOT.

Discrete Gromov-Wasserstein formulation. Let N_x and N_y be the number of training samples in the source and target domains, respectively. Let $\mathbf{C}^x \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{C}^y \in \mathbb{R}^{N_y \times N_y}$ be the corresponding source and target intra-domain cost matrices. These matrices measure the pairwise distance or similarity between the samples for a given function, i.e., cosine similarity, Euclidean distance, inner product, etc. The discrete GWOT problem is:

$$\mathbf{T}^* \stackrel{\text{def}}{=} \operatorname*{argmin}_{\mathbf{T} \in \mathcal{C}_{N_x, N_y}} \sum_{i,j,k,l} |\mathbf{C}^x_{i,k} - \mathbf{C}^y_{j,l}|^p \mathbf{T}_{i,j} \mathbf{T}_{k,l}, \tag{2}$$

here $\mathcal{C}_{N_x,N_y} \overset{\text{def}}{=} \{\mathbf{T} \in \mathbb{R}_+^{N_x \times N_y} \,|\, \mathbf{T}^T \mathbb{1} = \frac{1}{N_y}\mathbb{1}; \mathbf{T}\mathbb{1} = \frac{1}{N_x}\mathbb{1}\}$ is the set of coupling matrices between source and target spaces; $\mathbb{1} = [1, \dots, 1]^T \in \mathbb{R}^N$, $N \in \{N_x, N_y\}$. The loss function $|\cdot - \cdot|^p$ in (2) accounts for the misfit between the similarity matrices, a typical choice for the degree factor is $p = 2$. We refer to (Peyré et al., 2016; Mémoli, 2011; Chowdhury & Mémoli, 2019; Titouan et al., 2019b) for details.

Continuous Gromov-Wasserstein formulation. Let $\mathbb{P} \in \mathcal{P}(\mathbb{R}^{d_x})$, $\mathbb{Q} \in \mathcal{P}(\mathbb{R}^{d_y})$ be two distributions. Let $c_{\mathcal{X}} : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \to \mathbb{R}$ and $c_{\mathcal{Y}} : \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}$ be two intra-domain cost functions for the source ($\mathbb{R}^{d_x}$) and target ($\mathbb{R}^{d_y}$) domains, respectively. The Monge's GWOT problem is defined as:

$$\text{GWOT}_p^p(\mathbb{P}, \mathbb{Q}) \overset{\text{def}}{=} \inf_{T_\sharp \mathbb{P} = \mathbb{Q}} \int_{\mathbb{R}^{d_x} \times \mathbb{R}^{d_x}} |c_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') - c_{\mathcal{Y}}(T(\mathbf{x}), T(\mathbf{x}'))|^p d\mathbb{P}(\mathbf{x}) d\mathbb{P}(\mathbf{x}'). \tag{3}$$

Theoretical results on the existence and regularity of (3) under certain cases could be found in (Dumont et al., 2024; Mémoli & Needham, 2024, 2022). An illustration of problem (3) can be found in Figure 1b. In this continuous setup, the objective is to find an optimal transport map $T^* : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ that allows to transform (align) the source distribution to the target distribution. While in (1) we search for a map that sends $\mathbb{P}$ to $\mathbb{Q}$ minimizing the total transport cost, (3) aims to find the most isometric map w.r.t. the costs $c_{\mathcal{X}}$ and $c_{\mathcal{Y}}$, i.e., the map that maximally preserves the pairwise intra-domain costs. The commonly studied case (Vayer, 2020; Sebbouh et al., 2024) is $p = 2$ with the Euclidean distance $c(\cdot, \cdot) = \| \cdot - \cdot \|^2$ or inner product $c(\cdot, \cdot) = \langle \cdot, \cdot \rangle$ as intra-domain cost functions. In what follows, we will use **innerGW** to denote the latter case.

2.3 Practical Learning Setup

In practical scenarios, the source and target distributions $\mathbb{P}$ and $\mathbb{Q}$ are typically accessible by empirical samples (datasets) $X = \{\mathbf{x}_i\}_{i=1}^{N_x} \sim \mathbb{P}$ and $Y = \{\mathbf{y}_i\}_{i=1}^{N_y} \sim \mathbb{Q}$. Under the **discrete** GWOT formulation, these samples are directly used to compute intra-domain cost matrices $\mathbf{C}^x$, $\mathbf{C}^y$. These matrices are then fed to optimization problem (2). Having been solved, problem (2) yields a coupling matrix $\mathbf{T}^*$ which defines the GWOT correspondence between X and Y. Importantly, discrete GWOT operates with discrete empirical measures $\hat{\mathbb{P}} \overset{\text{def}}{=} \sum_{i=1}^{N_x} \frac{1}{N_x} \delta(\mathbf{x}_i)$, $\hat{\mathbb{Q}} \overset{\text{def}}{=} \sum_{i=1}^{N_y} \frac{1}{N_y} \delta(\mathbf{y}_i)$ rather than original ones.

In turn, under the **continuous** setup, the aim is to recover some parametric map $T^* : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ between the original source and target distributions $\mathbb{P}$ and $\mathbb{Q}$. In most practical scenarios, the latter is preferable, as it naturally allows *out-of-sample* estimation, i.e., provides GWOT mapping for new (unseen) samples $\mathbf{x} \sim \mathbb{P}$. In our paper, we deal with the continuous setup.

3 Continuous Gromov-Wasserstein solvers

Here we review recent advances in solving the continuous GWOT formulation. Most of these GWOT solvers are only discrete or adapted to *emulate a continuous behavior* by implementing some specific out-of-sample estimation method on top of the results of some discrete solver. The initial approach to solve the GWOT problem in discrete case (§2.2) was introduced in (Mémoli, 2011; Peyré et al., 2016). Below we only detail the methods which specifically aim to solve the continuous formulation and provide out-of-sample estimation. A sanity-check toy setup $\mathbb{R}^3 \to \mathbb{R}^2$ is available in Appendix B.

StructuredGW (Sebbouh et al., 2024). It introduces an iterative algorithm for solving the discrete entropy-regularized inner product case of the GW problem with $p = 2$, using the reformulation from (Vayer, 2020, maxOT). At each step, the coupling matrix $\mathbf{T}$ is updated via Sinkhorn iterations, alongside an auxiliary rotation matrix is updated by various methods depending on the regularization. For out-of-sample estimation, the method uses entropic maps (Pooladian & Niles-Weed, 2024; Dumont et al., 2024), using a learned dual potential to perform the mapping.

FlowGW (Klein et al., 2023). The method consists in fitting a discrete GW solver, e.g., (Peyré et al., 2016), to obtain a coupling matrix $\mathbf{T}$. This coupling determines the best way to match source samples to target ones. Weighted pairs of source and target samples are constructed using the distribution described by the coupling matrix. Then these samples are used to train a Conditional Flow Matching (CFM) model (Lipman et al., 2023) with the noise outsourcing technique from (Kallenberg, 1997). The inference process is done by solving the ODE given by the CFM model.

AlignGW (Alvarez-Melis & Jaakkola, 2018). This work proposes a discrete solver for the alignment of word embeddings. The authors use Sinkhorn iterations to compute the updates on the coupling

matrix instead of a linear search implemented in the Python Optimal Transport library (Flamary et al., 2021) that is inspired by (Titouan et al., 2019a; Peyré et al., 2016). This change significantly improves the stability of the solver. In spite of being a totally discrete method, we consider it due to its performance in challenging tasks. In order to allow out-of-sample estimations, we train a Multi-Layer Perceptron (MLP) model on the argmax based matrix from the learned coupling matrix $\mathbf{T}$.

RegGW (Uscidda et al., 2024). This work extends the Monge Gap Regularizer (Uscidda & Cuturi, 2023) to the Gromov-Wasserstein setting by parameterizing the transport push-forward function T with a neural network. The model is trained using a regularized loss that combines a *fitting* term—chosen as the Sinkhorn divergence (Uscidda et al., 2024)—to align the mapped points with the target distribution, and the Gromov-Monge gap regularizer, defined as the difference between the distortion of T (i.e., its GW cost) and the true discrete GW solution. Unlike other methods such as (Sotiropoulou & Alvarez-Melis, 2024), this approach does not require access to an intermediate reference distribution, making it more practical for our experimental setting.

CycleGW (Zhang et al., 2021). The authors propose minimizing the Unbalanced Bidirectional Gromov–Monge Divergence (UBGMD) to recover two push-forward maps: f pushing $\mathbb{P}$ toward $\mathbb{Q}$, and g doing the reverse. This is conceptually related to the Unbalanced Gromov-Wasserstein divergence Séjourné et al. (2020). Their approach involves minimizing a Generalized Maximum Mean Discrepancy (GMMD), computed using MMD with Gaussian kernels. In our setup, f corresponds to our mapping function T. A related method from Hur et al. (2021) also incorporates an MMD term, but due to its similarity and lack of publicly available code, we focus on Zhang et al. (2021).

4 Limitations of existing methods

As mentioned before, the majority of existing Gromov-Wasserstein approaches rely on discrete GWOT formulation, see §2.2. Solving (2) becomes increasingly computationally expensive as the training sample sizes N_x and N_y grow. This renders some datasets to hardly be manageable by discrete solvers. This limitation gives rise to additional issues, which we describe in this section.

4.1 Pitfalls of practical data setup

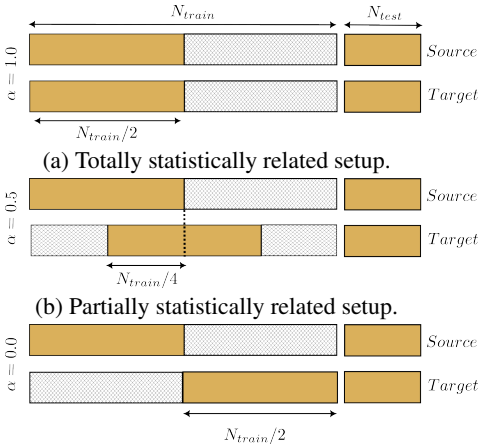
To begin with, we introduce the notion of **statistical (un)relatedness** of data which undergoes Gromov-Wasserstein alignment. To fit a GWOT solver, a practitioner typically has two training datasets, $X = \{\mathbf{x}_i\}_{i=1}^{N_x}$ and $Y = \{\mathbf{y}_i\}_{i=1}^{N_y}$. They are sampled from the source ($\mathbb{P}$) and target ($\mathbb{Q}$) distributions, see our training setup, §2.3. In what follows, without loss of generality, we will assume $N_x = N_y \overset{\text{def}}{=} N$. The natural statistical assumption on samples X and Y is that they are mutually independent. We define this data setup as *statistically unrelated*. Simultaneously, we introduce an alternative setup under which the source and target datasets X and Y turn out to be connected by some specific statistical relationships. Let us assume that samples X and Y are generated according to a joint distribution $\pi \in \Pi(\mathbb{P}, \mathbb{Q})$, $\pi \neq \mathbb{P} \otimes \mathbb{Q}$. Practically, we expect that samples from this distribution are meaningfully dependent; i.e., π is "significantly" different from the independent coupling $\mathbb{P} \otimes \mathbb{Q}$ due to the structural similarities present in the respective intra-domain geometries. In particular, coupling π may even set a one-to-one correspondence between the domains. Then, we call the training samples X and Y to be *statistically related* if they are obtained as follows:

1. First, we jointly sample X and $\widetilde{Y} = \{\widetilde{\mathbf{y}}_i\}_{i=1}^{N} \subset \mathbb{R}^{d_y}$ from coupling π, i.e.: $X \times \widetilde{Y} \sim \pi$.
2. Secondly, we apply *an unknown permutation* σ of indices to $\widetilde{Y}$ yielding Y, i.e.: $Y = \sigma \circ \widetilde{Y}$.

We found that the majority of the experimental test cases, on which the existing GWOT solvers are validated, frequently follow exactly the *statistically related* data setup. For instance, in the problem of learning cross-lingual word embedding correspondence (Alvarez-Melis & Jaakkola, 2018; Grave et al., 2019), the underlining coupling π could be understood as the distribution of dictionary pairs. The other example is the bone marrow dataset (Luecken et al., 2021; Klein et al., 2023). In this case, the source and target samples are generated using two different methods to profile gene expressions on the same donors. Interestingly, the *statistically related* data setup suits well the discrete GWOT formulation, because the optimization problem in this case boils down to the search for the permutation σ that spawned the target dataset Y. This leads to the natural **hypothesis** that for *statistically unrelated* training datasets X and Y, the performance of existing GWOT solvers may be poor. To test this hypothesis, we propose the experimental framework described below.

Modeling statistical (un)relatedness in practice. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \sim \mathbb{P}$ and $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \sim \mathbb{Q}$ be the source and target datasets. We assume that X and Y are initially totally paired, i.e., every i-th vector in the source set X is the pair of the i-th vector in the target set, Y. Also, we suppose that the pairing is reasonable, i.e., dictated by the nature of the data on hand. For example, if X and Y are word embeddings, then $\mathbf{x}_i$ and $\mathbf{y}_i$ correspond to the same word. We propose a way to create training data with different levels of statistical relatedness given by $\alpha \in [0, 1]$. First, random batches of N paired source-target samples are split into train and test sets ($N = N_{\text{train}} + N_{\text{test}}$).

For training, source data takes the first $N_{\text{train}}/2$ samples. For target, we select $N_{\text{train}}/2$ consecutive samples from index $\lceil (1-\alpha)N_{\text{train}}/2 \rceil$ and then randomly shuffle them. $\alpha = 1$ indicates a *totally statistically related* setup, $0 < \alpha < 1$ *partially related*, and $\alpha = 0$ *statistically unrelated* (Figure 2).



(a) Totally statistically related setup.

(b) Partially statistically related setup.

(c) Statistically unrelated setup.

Figure 2: Data splitting and statistical (un)relatedness.

It is important to emphasize that our benchmark construction requires datasets with **(a)** known ground-truth pairs, **(b)** alignable structures (ground-truth pairs are optimal pairs, i.e., they yield minimal distortion) and **(c)** large amount of samples to train continuous models.

To conclude the subsection, we want to remark that both *statistically related* and *statistically unrelated* setups are practically important. The *statistically unrelated* setup naturally appears when aligning single-cell multi-omics data (Demetci et al., 2022), where the goal is to match different single-cell assays. In this setting, the assays are statistically unrelated, since applying multiple assays to the same single cell is typically infeasible.

4.2 Benchmarking GWOT solvers on statistically (un)related data

In order to check how continuous GW solvers perform under statistically unrelated, partially and totally statistically related setups, we utilize **word** and **voice** embeddings.

Word Embeddings. We consider two text corpora named Twitter[2] and MUSE[3] (Multilingual Unsupervised and Supervised Embeddings) (Conneau et al., 2017) bilingual vocabularies. We embed the Twitter vocabulary using the GloVe (Global Vectors for Word Representation) algorithm (Pennington et al., 2014) and the MUSE vocabulary using BPEmb (Byte-Pair) embeddings (Heinzerling & Strube, 2018), respectively. We refer to these as Twitter-GloVe and MUSE-BPEmb.

Voice Embeddings. We consider the VoxCeleb2 dataset[4]. To obtain the embeddings we use a pretrained WavLM model (Chen et al., 2021) and the embedding process from (Gengembre et al., 2024). We name it VoxCeleb2-WNTA, following the naming convention of the original work [5].

In this subsection we test **three** baseline solvers introduced in §3: StructuredGW, AlignGW, and FlowGW. To perform the training and evaluation of the baseline solvers, we introduce two types of target data pools: the *whole* data space and the *reference* space. The *whole* space consists on a big subset of samples (400K for Twitter-GloVe, 90K for MUSE-BPEmb from English dictionary and 600K for VoxCeleb2-WNTA). The *reference* space refers to the $N_{\text{train}} + N_{\text{test}}$ samples in Figure 2.

We fit each solver for different values of α (levels of statistical relatedness), ranging from 0.0 to 1.0 in increments of 0.1. For each α, we perform **ten** repetitions with different random seeds following the process described in § 4.1. We use $N_{\text{train}} = 6$K, i.e., each run uses source and target datasets containing $N_{\text{train}}/2 = 3$K samples; $N_{\text{test}} = 2048$. The only reason for this relatively small training size is the computational complexity of the three solvers. The discrete optimization procedures underlying them cannot scale to significantly larger N_{train}. The training of the RegGW and CycleGW solvers from §3 fails on small datasets, so we defer them to §5.2, where larger datasets are considered. The results are shown in Figure 3. We refer to Appendix D for additional experimental setups.

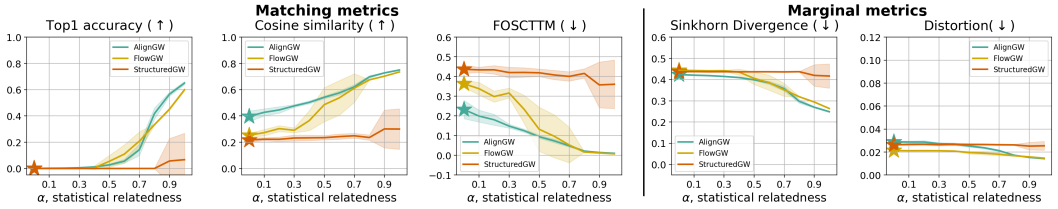
[2]https://radimrehurek.com/gensim/downloader.html

[3]https://github.com/facebookresearch/MUSE

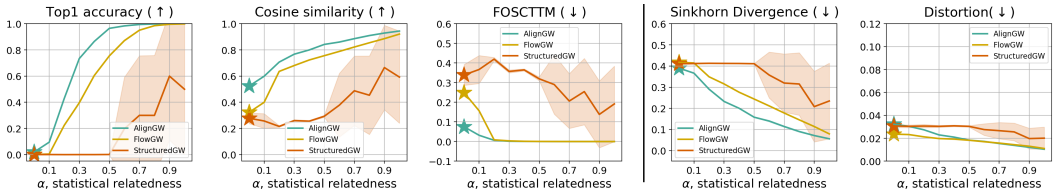
[4]https://www.robots.ox.ac.uk/ vgg/data/voxceleb/vox2.html

[5]https://huggingface.co/Orange/Speaker-wavLM-pro

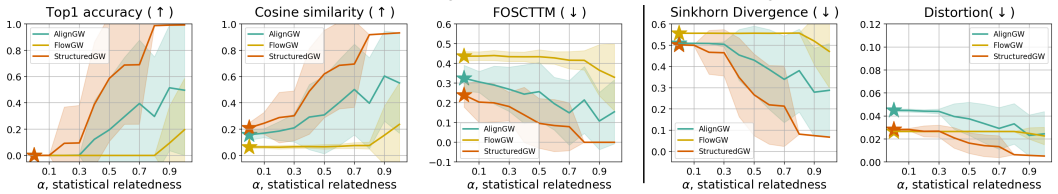
In the main text, we show results for different source and target embedding dimensions. For Twitter-GloVe, we use the setup $100 \to 50$ (Figure 3a), and for MUSE-BPEmb, the same $100 \to 50$ setup applies (Figure 3b), where 100 is the source dimension and 50 the target. For VoxCeleb2-WNTA, we consider the $128 \to 64$ setup (see Figure 3c). Additional configurations are presented in Appendix D.



(a) Twitter-GloVe dataset (Source:100 $\to$ Target:50)



(b) MUSE-BPEmb (English) dataset (Source:100 $\to$ Target:50)



(c) VoxCeleb2-WNTA dataset (Source:128 $\to$ Target:64)

Figure 3: Performance of the baseline GWOT solvers for **Twitter-GloVe** (3a), **MUSE-BPEmb (English)** (3b) and **VoxCeleb2-WNTA** (3c) at different levels of statistical relatedness α. Solvers were trained with $N_{\text{train}}/2 = 3000$ samples from spaces of 400K, 90K and 600K, respectively. The plot shows results on a 2048-sample test subset. Accuracy was computed on the *reference* space of $N_{\text{train}} + N_{\text{test}} = 8048$ samples.

Evaluation. In the main text we report matching metrics (Top-1 accuracy ↑, cosine similarity ↑, and FOSCTTM ↓) and marginal metrics (Distortion ↓ and Sinkhorn divergence ↓). Additional marginal metrics and a justification for the chosen metrics are given in Appendices D.1 and C.1, respectively. All metrics are computed on unseen test data. Top-1 accuracy is computed using the *reference* space; results for the *whole* data space are in Appendix D.2.

Conclusions. The results indicate that almost all the baseline solvers perform well in totally statistically related scenarios, even when evaluated on unseen data. StructuredGW presents a less robust behavior. This demonstrates their ability to learn and capture the inner structures when the data is highly statistically related. However, their performance drops significantly as the value of α decreases. This finding highlights a key limitation: current GWOT methods are restricted to setups where source and target datasets are statistically related. This is particularly significant, as it shows that these methods fail to learn meaningful structure-preserving mappings when optimal source–target pairs are absent from the training data. Consequently, the applicability of GW solvers is limited, since their goal is to recover relational structure rather than individual optimal pairings.

An attempt to solve the pitfalls of existing GWOT solvers. We conjecture that the observed behavior is mainly due to the small sizes of training sets dictated by the discrete nature of the solvers. Indeed, relatively small samples hardly could fully express the intrinsic geometry of the data, which complicates the faithful GW alignment of the domains, see Appendix F.1 for the broader discussion on the issue. Therefore, one possible way to improve how GW solvers perform on statistically unrelated data is to create GW solvers capable of handling large datasets. A natural direction is to construct solvers parameterized by neural networks, which scale well to large datasets via gradient-based optimization. In the following section (§5) we propose a new continuous solver (§5.1) which does not rely on discrete techniques. Therefore, it can better capture the inner geometry and structure of the data without the strict need of training on statistically related data.

5 GWOT solvers at large scale

In this section, we start by introducing NeuralGW, a novel scalable method to solve the continuous GWOT problem (§5.1). Then we proceed to the practical performance of NeuralGW and the baselines in large-scale GloVe benchmark (§5.2).

5.1 Neural Gromov-Wasserstein Solver

In this subsection, we conduct the theoretical and algorithmic derivation of our proposed approach. In what follows, we restrict to the case $d_x \geq d_y$; source ($\mathbb{P}$) and target ($\mathbb{Q}$) distributions are absolutely continuous and supported on some compact subsets $\mathcal{X} \subset \mathbb{R}^{d_x}$, $\mathcal{Y} \subset \mathbb{R}^{d_y}$ respectively. Our method is developed for innerGW, i.e., problem (3) with $c_{\mathcal{X}} = \langle \cdot, \cdot \rangle_{d_x}$, $c_{\mathcal{Y}} = \langle \cdot, \cdot \rangle_{d_y}$, $p = 2$:

$$\text{innerGW}_2^2(\mathbb{P}, \mathbb{Q}) \stackrel{\text{def}}{=} \min_{T_\sharp \mathbb{P} = \mathbb{Q}} \int\limits_{\mathbb{R}^{d_x} \times \mathbb{R}^{d_x}} \left| \langle \mathbf{x}, \mathbf{x}' \rangle_{d_x} - \langle T(\mathbf{x}), T(\mathbf{x}') \rangle_{d_y} \right|^2 d\mathbb{P}(\mathbf{x}) d\mathbb{P}(\mathbf{x}'). \tag{4}$$

Note that the existence of minimizer for (4) is due to (Dumont et al., 2024, Theorem 3.2). We base our method on the theoretical insights about GW from (Vayer, 2020). According to (Vayer, 2020, Theorem 4.2.1), when $\int \|\mathbf{x}\|_2^4 d\mathbb{P}(\mathbf{x}) < +\infty$, $\int \|\mathbf{y}\|_2^4 d\mathbb{Q}(\mathbf{y}) < +\infty$, problem (4) is equivalent to

$$\text{innerGW}_2^2(\mathbb{P}, \mathbb{Q}) = \text{Const}(\mathbb{P}, \mathbb{Q}) - \max_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \max_{P \in F_{d_x, d_y}} \int \langle \mathbf{P}\mathbf{x}, \mathbf{y} \rangle_{d_y} d\pi(\mathbf{x}, \mathbf{y}), \tag{5}$$

where $F_{d_x, d_y} \stackrel{\text{def}}{=} \{\mathbf{P} \in \mathbb{R}^{d_y \times d_x} \mid \|\mathbf{P}\|_{\mathcal{F}} = \min(\sqrt{d_x}, \sqrt{d_y})\}$ are the matrices of fixed Frobenius norm. Note that (5) admits a solution $\pi^* \in \Pi(\mathbb{P}, \mathbb{Q})$, $P^* \in F_{d_x, d_y}$ (Vayer, 2020, Lemmas 6.2.7; 4.2.2).

Our following lemma reformulates the innerGW problem as a minimax optimization problem. This reformulation is inspired by well-celebrated dual OT solvers such as (Korotin et al., 2021b; Fan et al., 2023; Korotin et al., 2023b).

Lemma 5.1 (InnerGW as a minimax optimization) *It holds that* (5) *is equivalent to*

$$\text{innerGW}_2^2(\mathbb{P}, \mathbb{Q}) = \text{Const}\,(\mathbb{P}, \mathbb{Q}) + \min_{P \in F_{d_x, d_y}} \max_{f: \mathbb{R}^{d_y} \to \mathbb{R}} \min_{T: \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}} \mathcal{L}(\boldsymbol{P}, f, T); \tag{6}$$

$$\mathcal{L}(\boldsymbol{P}, f, T) \stackrel{\text{def}}{=} \int_{\mathbb{R}^{d_y}} f(\boldsymbol{y}) d\mathbb{Q}(\boldsymbol{y}) - \int_{\mathbb{R}^{d_x}} [\langle \boldsymbol{P}\boldsymbol{x}, T(\boldsymbol{x}) \rangle_{d_y} + f(T(\boldsymbol{x}))] d\mathbb{P}(\boldsymbol{x}).$$

The follwoign theorem shows that, for any existing optimal GW map T^*, one can select a compatible P^* so that T^* minimizes (6) for every optimal dual potential f^*.

Theorem 5.2 (Optimal maps solve the minimax problem) *Assume that there exists at least one optimal GW map T^*. For any such map, let $\boldsymbol{P}^*$ be a matrix satisfying*

$$\boldsymbol{P}^* \in \underset{\boldsymbol{P} \in F_{d_x, d_y}}{\text{argmax}} \int_{\mathbb{R}^{d_x}} \langle \boldsymbol{P}\boldsymbol{x}, T^*(\boldsymbol{x}) \rangle \, d\mu(\boldsymbol{x}).$$

Then $\boldsymbol{P}^*$ solves the outer minimization in (6)*. For any potential f^* that solves*

$$f^* \in \underset{f}{\text{argmax}} \min_{T:\, \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}} \mathcal{L}(\boldsymbol{P}^*, f, T),$$

we have:

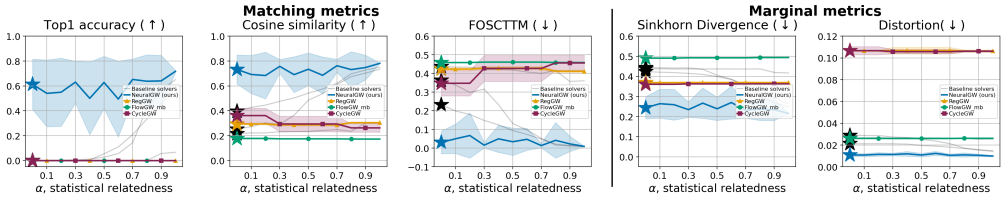
$$T^* \in \underset{T:\, \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}}{\text{argmin}} \mathcal{L}(\boldsymbol{P}^*, f^*, T). \tag{7}$$

To optimize (6) we follow the best practices from the field of continuous OT (Korotin et al., 2021c; Fan et al., 2023; Korotin et al., 2023a,b; Asadulaev et al., 2024; Choi et al., 2023; Gushchin et al., 2024) and simply parameterize f and T with neural networks. In turn, $\mathbf{P}$ is a learnable matrix of fixed Frobenius norm. We use the alternating stochastic gradient descent/ascent/descent method to train their parameters. The learning algorithm is detailed in Appendix E. We call the method NeuralGW.

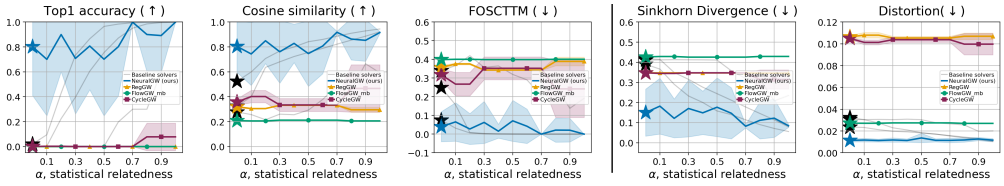
The hard min-max-min optimization inherent to the method (Equation (6)) can lead to stability challenges. We alleviate these by adopting a specific initialization strategy for the parameterized networks T, f, and $\mathbf{P}$ (see Appendix G). As a sanity check, we run our proposed approach on toy 3D→2D setup, see Figure 5b in Appendix B. Note that in comparison to other continuous GWOT approaches, our method *does not rely on discrete OT* in any form. In particular, the training process assumes access to just random samples from $\mathbb{P}, \mathbb{Q}$; it does not use/need any pairing between them.

5.2 Practical performance of NeuralGW and baselines at large scale

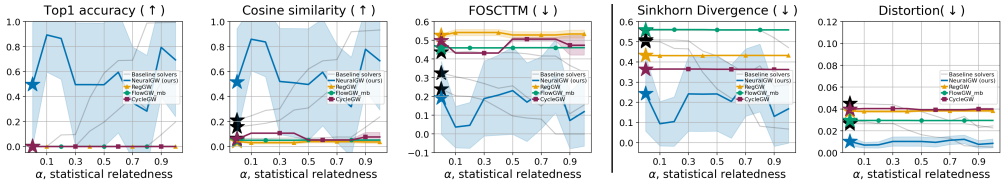
We start by introducing our **large-scale** Twitter-GloVe, MUSE-BPEmb and VoxCeleb2-WNTA setups. We consider the same data split process as in §4.2, but we respectively take $N_{train} \in \{398\text{K}, 88\text{K}, 598\text{K}\}$ samples instead of the 6K used in §4.2 for the baseline solvers; $N_{test} = 2048$ is left the same. Each repetition consists in training the models with the same data but using different initialization parameters for, e.g., the neural networks. We computed accuracy on the *whole* target space. In the large-scale setup, the *whole* space coincides with the *reference* space.



(a) Twitter-GloVe dataset (Source:100 → Target:50)



(b) MUSE-BPEmb (English) dataset (Source:100 → Target:50)



(c) VoxCeleb2-WNTA dataset (Source:128 → Target:64)

Figure 4: Performance of the GWOT solvers for the large-scale **Twitter-GloVe** (4a) **MUSE-BPEmb (English)** (4b) and **VoxCeleb2-WNTA** (4c) embeddings at different levels of statistical relatedness α. The solvers were trained with $N_{\text{train}}/2 = 198\text{K}$, $N_{\text{train}}/2 = 44\text{K}$ and $N_{\text{train}}/2 = 298\text{K}$ samples, respectively. The plot shows results on a 2048-sample test subset. Accuracy was computed on the *whole space* of $N_{\text{train}} + N_{\text{test}}$ of 400K, 90K and 600K samples, respectively.

The methods under comparison for the **large-scale** setups are: NeuralGW (§5.1); RegGW; CycleGW and FlowGW (§3). The latter is trained in a minibatch manner, i.e., it fits flow matching on top of discrete GW solutions for minibatches. For completeness, we additionally report the performance of the baselines from §4.2 (in gray, labeled as "Baseline solvers", Figure 4). In this plot, the baseline methods are trained on a small subset ($N_{train} = 6\text{K}$), but the metrics are computed with respect to the *whole* data space, similar to NeuralGW, RegGW, CycleGW and FlowGW.

The colored charts for baselines' metrics computed on the *whole* data spaces of Twitter-GloVe, MUSE-BPEmb, and VoxCeleb2-WNTA are shown in Figure 8 (Appendix D.4). While comparing baseline methods trained on 3K samples (shown in gray) with our approach trained on larger datasets (200K for Twitter, 90K for MUSE, and 300K for VoxCeleb2) may seem unfair, we emphasize that the dataset sizes were selected according to the computational capabilities of each solver.

Based on the results in Figure 4, we note the following:

1. NeuralGW is the only method that handles statistically unrelated setups ($\alpha = 0$). In spite of the hard optimization problem, NeuralGW succeeds in finding a GWOT map in 7 out of 10 repeats.
2. The advanced baselines (RegGW, CycleGW and FlowGW) failed to achieve reasonable performance for any value of α. This happens because, although the source and target datasets are globally statistically related, the random permutation applied to the target data (see §4.1) leads to batches with little or no statistical relation. Consequently, these models are trained on batches with $\alpha_{\text{batch}} \approx 0$. As these methods somehow rely on discrete methods when finding the GW coupling for every batch, they inherit this limitation.

For the sake of completeness, we provide some additional experiments and metrics in Appendix D:

- NeuralGW trained on $3K$ samples (Appendix D.4). The results are bad, which is expected considering NeuralGW requires larger datasets due to adversarial training.
- Statistically related batches (Appendix D.3). We train RegGW, CycleGW and FlowGW (minibatch) on statistically related batches, then $\alpha_{\text{batch}} = \alpha$. This improves the performance; however, this setup is unfair due to this artificial construction, and the results should be interpreted with caution.
- Low-to-high experiments (Appendix D.4). While the main text focuses on settings where the source dimension exceeds the target dimension ($d_x > d_y$), we also consider the reverse case ($d_x < d_y$) for completeness. Results for the Twitter-GloVe $50 \to 100$ setup using baseline and large-scale solvers are presented in Figures 11a and 10c, respectively. NeuralGW fails in this setting because a regular deterministic neural map T generally cannot generate the additional degrees of freedom required to satisfy $T_\sharp \mu = \nu$ when the target distribution is full-dimensional.

Conclusions. As shown in Figures 4 and 3, neither the baseline methods nor the proposed solver (NeuralGW) can fully handle statistically unrelated setups. A key limitation of the baseline solvers (§4.2) is their reliance on discrete GW solvers: while effective on small datasets, these approaches fail in statistically unrelated settings. Learning the mapping in such scenarios requires training on large scale datasets to properly capture the intrinsic geometries. We therefore examine more advanced solvers, which nevertheless also fail due to their dependence on discrete solvers at some stage of the learning process. Finally, we propose NeuralGW, a fully continuous method that avoids discrete solvers and is the only solver capable of successfully handling the statistically unrelated setup.

6 Discussion

The main goal of this paper is to highlight a previously overlooked issue in the evaluation pipelines of GWOT solvers. In particular, statistically unrelated setups have not been explored in prior work, despite representing a potentially more impactful application of GWOT methods, as they introduce a clear separation between data alignment and matching problems. Besides, it was recently noticed that the problem of correct evaluation pipeline does affect the related fields of Generative Modeling, namely, population dynamics with backward Euler scheme (JKO) (Persiianov et al., 2026, Appendix A.3) which underlines the importance of the presented study.

Acknowledgments

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4F0002 and the agreement №139-10-2025-033.

References

Aboagye, P. O., Zheng, Y., Yeh, M., Wang, J., Zhuang, Z., Chen, H., Wang, L., Zhang, W., and Phillips, J. Quantized wasserstein procrustes alignment of word embedding spaces. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pp. 200–214, 2022.

Alvarez-Melis, D. and Jaakkola, T. Gromov-wasserstein alignment of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1881–1890, 2018.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Asadulaev, A., Korotin, A., Egiazarian, V., Mokrov, P., and Burnaev, E. Neural optimal transport with general cost functionals. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gIiz7tBtYZ.

Assel, H. V., Vincent-Cuaz, C., Courty, N., Flamary, R., Frossard, P., and Vayer, T. Distributional reduction: Unifying dimensionality reduction and clustering with gromov-wasserstein, 2024. URL https://arxiv.org/abs/2402.02239.

Backhoff-Veraguas, J., Beiglböck, M., and Pammer, G. Existence, duality, and cyclical monotonicity for weak transport costs. *Calculus of Variations and Partial Differential Equations*, 58(6):203, 2019.

Beier, F., Beinert, R., and Steidl, G. On a linear gromov–wasserstein distance. *IEEE Transactions on Image Processing*, 31:7292–7305, 2022.

Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L., and Liu, J. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pp. 1542–1553. PMLR, 2020.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., and Wei, F. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. 2021.

Choi, J., Choi, J., and Kang, M. Generative modeling through the semi-dual formulation of unbalanced optimal transport. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=7WQt1J13ex.

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

Chowdhury, S. and Needham, T. Generalized spectral clustering via gromov-wasserstein learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 712–720. PMLR, 2021.

Chung, J. S., Nagrani, A., and Zisserman, A. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. Word translation without parallel data. *CoRR*, abs/1710.04087, 2017. URL http://arxiv.org/abs/1710.04087.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Cuturi, M., Meng-Papaxanthos, L., Tian, Y., Bunne, C., Davis, G., and Teboul, O. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

Daniels, G., Maunu, T., and Hand, P. Score-based generative neural networks for large-scale optimal transport. *Advances in Neural Information Processing Systems*, 34, 2021.

Delon, J., Desolneux, A., and Salmona, A. Gromov–wasserstein distances between gaussian distributions. *Journal of Applied Probability*, 59(4):1178–1198, 2022.

Demetci, P., Santorella, R., Sandstede, B., Noble, W. S., and Singh, R. Scot: single-cell multi-omics alignment with optimal transport. *Journal of computational biology*, 29(1):3–18, 2022.

Demetci, P., Huy Tran, Q., Redko, I., and Singh, R. Breaking isometric ties and introducing priors in Gromov-Wasserstein distances. In Dasgupta, S., Mandt, S., and Li, Y. (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 298–306. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/demetci24a.html.

Dumont, T., Lacombe, T., and Vialard, F.-X. On the existence of monge maps for the gromov–wasserstein problem. *Foundations of Computational Mathematics*, pp. 1–48, 2024.

Fan, J., Liu, S., Ma, S., Zhou, H.-M., and Chen, Y. Neural monge map estimation and its applications. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=2mZSlQscj3. Featured Certification.

Feydy, J., Séjourné, T., Vialard, F.-X., ichi Amari, S., Trouvé, A., and Peyré, G. Interpolating between optimal transport and mmd using sinkhorn divergences, 2018. URL https://arxiv.org/abs/1810.08278.

Flamary, R., Févotte, C., Courty, N., and Emiya, V. Optimal spectral transportation with application to music transcription. *Advances in Neural Information Processing Systems*, 29, 2016.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Frigo, O., Sabater, N., Demoulin, V., and Hellier, P. Optimal transportation for example-guided color transfer. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part III 12*, pp. 655–670. Springer, 2015.

Gengembre, N., Le Blouch, O., and Gendrot, C. Disentangling prosody and timbre embeddings via voice conversion. In *Interspeech 2024*, pp. 2765–2769, 2024. doi: 10.21437/Interspeech.2024-207.

Gozlan, N., Roberto, C., Samson, P.-M., and Tetali, P. Kantorovich duality for general transport costs and applications. *Journal of Functional Analysis*, 273(11):3327–3405, 2017.

Grave, E., Joulin, A., and Berthet, Q. Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1880–1890. PMLR, 2019.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Gushchin, N., Kolesov, A., Korotin, A., Vetrov, D. P., and Burnaev, E. Entropic neural optimal transport via diffusion processes. *Advances in Neural Information Processing Systems*, 36, 2024.

Heinzerling, B. and Strube, M. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T. (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

Hur, Y., Guo, W., and Liang, T. Reversible Gromov-Monge Sampler for Simulation-Based Inference. *arXiv e-prints*, art. arXiv:2109.14090, September 2021. doi: 10.48550/arXiv.2109.14090.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

Kallenberg, O. *Foundations of modern probability*, volume 2. Springer, 1997.

Klein, D., Uscidda, T., Theis, F., and Cuturi, M. Generative entropic neural optimal transport to map within and across spaces. *arXiv preprint arXiv:2310.09254*, 2023.

Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. Wasserstein-2 generative networks. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=bEoxzW_EXsa.

Korotin, A., Li, L., Genevay, A., Solomon, J. M., Filippov, A., and Burnaev, E. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34, 2021b.

Korotin, A., Li, L., Solomon, J., and Burnaev, E. Continuous wasserstein-2 barycenter estimation without minimax optimization. In *International Conference on Learning Representations*, 2021c. URL https://openreview.net/forum?id=3tFAs5E-Pe.

Korotin, A., Selikhanovych, D., and Burnaev, E. Kernel neural optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=Zuc_MHtUma4.

Korotin, A., Selikhanovych, D., and Burnaev, E. Neural optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=d8CBRlWNkqH.

Lezcano-Casado, M. Trivializations for gradient-based optimization on manifolds. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 9154–9164, 2019.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Luecken, M. D., Burkhardt, D. B., Cannoodt, R., Lance, C., Agrawal, A., Aliee, H., Chen, A. T., Deconinck, L., Detweiler, A. M., Granados, A. A., Huynh, S., Isacco, L., Kim, Y. J., Klein, D., KUMAR, B. D., Kuppasani, S., Lickert, H., McGeever, A., Mekonen, H., Melgarejo, J. C., Morri, M., Müller, M., Neff, N., Paul, S., Rieck, B., Schneider, K., Steelman, S., Sterr, M., Treacy, D. J., Tong, A., Villani, A.-C., Wang, G., Yan, J., Zhang, C., Pisco, A. O., Krishnaswamy, S., Theis, F. J., and Bloom, J. M. A sandbox for prediction and integration of DNA, RNA, and proteins in single cells. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=gN35BGa1Rt.

Luo, Y., Zhang, S.-Y., Zheng, W.-L., and Lu, B.-L. WGAN domain adaptation for eeg-based emotion recognition. In *International Conference on Neural Information Processing*, pp. 275–286. Springer, 2018.

Makkuva, A., Taghvaei, A., Oh, S., and Lee, J. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681. PMLR, 2020.

Mazelet, S., Flamary, R., and Thirion, B. Unsupervised learning for optimal transport plan prediction between unbalanced graphs, 2025. URL https://arxiv.org/abs/2506.12025.

Memoli, F. On the use of Gromov-Hausdorff Distances for Shape Comparison. In Botsch, M., Pajarola, R., Chen, B., and Zwicker, M. (eds.), *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007. ISBN 978-3-905673-51-7. doi: /10.2312/SPBG/SPBG07/081-090.

Mémoli, F. Spectral gromov-wasserstein distances for shape matching. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 256–263. IEEE, 2009.

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

Mémoli, F. and Needham, T. Distance distributions and inverse problems for metric measure spaces. *Studies in Applied Mathematics*, 149(4):943–1001, 2022.

Mémoli, F. and Needham, T. Comparison results for gromov–wasserstein and gromov–monge distances. *ESAIM: Control, Optimisation and Calculus of Variations*, 30:78, 2024.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge?, 2018. URL https://arxiv.org/abs/1801.04406.

Mokrov, P., Korotin, A., Kolesov, A., Gushchin, N., and Burnaev, E. Energy-guided entropic neural optimal transport. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=d6tUsZeVs7.

Montesuma, E. F., Mboula, F. N., and Souloumiac, A. Recent advances in optimal transport for machine learning. *arXiv preprint arXiv:2306.16156*, 2023.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Persiianov, M., Chen, J., Mokrov, P., Tyurin, A., Burnaev, E., and Korotin, A. Learning of population dynamics: Inverse optimization meets JKO scheme. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tVJIKd6CLF.

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

Peyré, G., Cuturi, M., et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Pooladian, A.-A. and Niles-Weed, J. Entropic estimation of optimal transport maps, 2024.

Redko, I., Courty, N., Flamary, R., and Tuia, D. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on artificial intelligence and statistics*, pp. 849–858. PMLR, 2019.

Rout, L., Korotin, A., and Burnaev, E. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=5JdLZg346Lw.

Ryu, J., Lopez, R., Bunne, C., and Regev, A. Cross-modality matching and prediction of perturbation responses with labeled gromov-wasserstein optimal transport, 2024.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. Improving GANs using optimal transport. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkQkBnJAb.

Santambrogio, F. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Springer International Publishing, 2015. ISBN 9783319208282. URL https://books.google.ru/books?id=UOHHCgAAQBAJ.

Scetbon, M., Peyré, G., and Cuturi, M. Linear-time gromov wasserstein distances using low rank couplings and costs. In *International Conference on Machine Learning*, pp. 19347–19365. PMLR, 2022.

Sebbouh, O., Cuturi, M., and Peyré, G. Structured transforms across spaces with cost-regularized optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pp. 586–594. PMLR, 2024.

Séjourné, T., Vialard, F.-X., and Peyré, G. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. *arXiv preprint arXiv:2009.04266*, 2020.

Sotiropoulou, A. and Alvarez-Melis, D. Strongly isomorphic neural optimal transport across incomparable spaces. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024. URL https://openreview.net/forum?id=uRySKj54Pg.

Titouan, V., Courty, N., Tavenard, R., and Flamary, R. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pp. 6275–6284. PMLR, 2019a.

Titouan, V., Flamary, R., Courty, N., Tavenard, R., and Chapel, L. Sliced gromov-wasserstein. *Advances in Neural Information Processing Systems*, 32, 2019b.

Uscidda, T. and Cuturi, M. The monge gap: A regularizer to learn all transport maps. In *International Conference on Machine Learning*, pp. 34709–34733. PMLR, 2023.

Uscidda, T., Eyring, L., Roth, K., Theis, F. J., Akata, Z., and marco cuturi. Disentangled representation learning through geometry preservation with the gromov-monge gap. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. URL https://openreview.net/forum?id=5LWt9sdYSo.

Vayer, T. A contribution to optimal transport on incomparable spaces. *arXiv preprint arXiv:2011.04447*, 2020.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Vincent-Cuaz, C., Vayer, T., Flamary, R., Corneli, M., and Courty, N. Online graph dictionary learning. In *International conference on machine learning*, pp. 10564–10574. PMLR, 2021.

Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. Semi-relaxed gromov-wasserstein divergence and applications on graphs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RShaMexjc-x.

Wang, R. and Zhang, Z. Quadratic-form optimal transport, 2025. URL https://arxiv.org/abs/2501.04658.

Wang, T. and Goldfeld, Z. Neural entropic gromov-wasserstein alignment. *arXiv preprint arXiv:2312.07397*, 2023.

Xu, H., Luo, D., and Carin, L. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32, 2019.

Xu, H., Luo, D., Carin, L., and Zha, H. Learning graphons via structured gromov-wasserstein barycenters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10505–10513, 2021.

Zhang, Z., Mroueh, Y., Goldfeld, Z., and Sriperumbudur, B. K. Cycle consistent probability divergences across different spaces. In *International Conference on Artificial Intelligence and Statistics*, 2021. URL https://api.semanticscholar.org/CorpusID:244478145.

Zhang, Z., Goldfeld, Z., Mroueh, Y., and Sriperumbudur, B. K. Gromov–wasserstein distances: Entropic regularization, duality and sample complexity. *The Annals of Statistics*, 52(4):1616–1645, 2024.

Contents

A Proofs of theorems and lemmas **17**

B Toy sanity-check experiment **18**

C Metrics and embeddings **18**

C.1 Metrics 18

C.2 Embeddings 20

D Extended experiments **21**

D.1 Marginal metrics for main text experiments 21

D.2 Trained on small dataset, evaluated w.r.t. *whole* data space 23

D.3 Experiments on statistically related batches for continuous solvers 23

D.4 GloVe 24

D.5 BPEmb 25

D.6 Biological dataset 26

E Solvers' implementation details **26**

E.1 NeuralGW. 27

F Broader discussions **28**

F.1 Discrete GW solvers under low statistical relatedness data scenario 28

G NeuralGW initialization strategy **29**

A Proofs of theorems and lemmas

Proof of Lemma 5.1. First, we recall the dual formulation of (1), see, e.g., (Fan et al., 2023):

$$\mathrm{OT}(\mathbb{P},\mathbb{Q}) \stackrel{\text{def}}{=} \max_{f}\left[\min_{T}\int \big(c(\mathbf{x},T(\mathbf{x})) - f(T(\mathbf{x}))\big)d\mathbb{P}(\mathbf{x}) + \int f(\mathbf{y})d\mathbb{Q}(\mathbf{y})\right], \tag{8}$$

respectively. Note that the existence of a solution (f^*, T^*) of (8) is due to (Fan et al., 2023, Theorem 2). Our proof starts with (5). For each $\mathbf{P}$, we rewrite the inner optimization over $\mathbf{P}$ using (8) for the cost $c(\mathbf{x},\mathbf{y}) = -\langle \mathbf{P}\mathbf{x},\mathbf{y}\rangle$:

$$\begin{aligned}
\mathrm{Const}\,(\mathbb{P},\mathbb{Q}) - \mathrm{innerGW}_2^2(\mathbb{P},\mathbb{Q}) = \max_{\mathbf{P}\in F_{d_x,d_y}} \max_{\pi\in\Pi(\mathbb{P},\mathbb{Q})} \int\limits_{\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}} \langle \mathbf{P}\mathbf{x},\mathbf{y}\rangle_{d_y} d\pi(\mathbf{x},\mathbf{y}) = \\
- \min_{\mathbf{P}\in F_{d_x,d_y}} \left[\min_{\pi\in\Pi(\mathbb{P},\mathbb{Q})} \int\limits_{\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}} -\langle \mathbf{P}\mathbf{x},\mathbf{y}\rangle_{d_y} d\pi(\mathbf{x},\mathbf{y})\right] = \\
- \min_{\mathbf{P}\in F_{d_x,d_y}} \left[\max_{f} \int\limits_{\mathbb{R}^{d_x}} f(\mathbf{y})d\mathbb{Q}(\mathbf{y}) + \min_{T\colon \mathbb{R}^{d_x}\to\mathbb{R}^{d_y}} \int\limits_{\mathbb{R}^{d_x}} -\langle \mathbf{P}\mathbf{x},T(\mathbf{x})\rangle_{d_y} - f(T(\mathbf{x}))d\mathbb{P}(\mathbf{x})\right] = \\
- \min_{\mathbf{P}\in F_{d_x,d_y}} \max_{f}\min_{T} \mathcal{L}(\mathbf{P},f,T).
\end{aligned}$$

Proof of Theorem 5.2. We expand $\mathcal{L}(\mathbf{P}^*, f^*, T^*)$ and use the fact that T^* is an optimal GW map.

$$\mathcal{L}(\mathbf{P}^*, f^*, T^*) = \int\limits_{\mathbb{R}^{d_y}} f^*(\mathbf{y})d\mathbb{Q}(\mathbf{y}) - \int\limits_{\mathbb{R}^{d_x}} \left[\langle \mathbf{P}^*\mathbf{x}, T^*(\mathbf{x})\rangle_{d_y} + f^*\big(T^*(\mathbf{x})\big)\right] d\mathbb{P}(\mathbf{x}). \tag{9}$$

Since $T^*_\sharp\mathbb{P} = \mathbb{Q}$, by the change of variables formula we get:

$$\int\limits_{\mathbb{R}^{d_x}} f^*\big(T^*(\mathbf{x})\big)d\mathbb{P}(\mathbf{x}) = \int\limits_{\mathbb{R}^{d_y}} f^*(\mathbf{y})d\mathbb{Q}(\mathbf{y}).$$

Plugging this into (9), we get:

$$\mathcal{L}(\mathbf{P}^*, f^*, T^*) = -\int\limits_{\mathbb{R}^{d_x}} \langle \mathbf{P}^*\mathbf{x}, T^*(\mathbf{x})\rangle_{d_y} d\mathbb{P}(\mathbf{x}).$$

Since T^* is an optimal GW map, the induced coupling $\pi^* = [\mathrm{id}_{\mathbb{R}^{d_x}}, T^*]_\sharp\mathbb{P}$ is GW-optimal. By the equivalence between the innerGW and max-correlation formulations, and the choice of $\mathbf{P}^*$ in Theorem 5.2, the pair $(\mathbf{P}^*, \pi^*)$ maximizes the correlation objective. Consequently, $\mathbf{P}^*$ solves the outer minimization in (6), and π^* is also optimal for the fixed matrix $\mathbf{P}^*$.

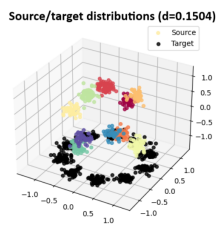
Finally, since f^* is an optimal dual potential for the corresponding OT problem, by (9) we have:

$$\begin{aligned}
-\mathcal{L}(\mathbf{P}^*, f^*, T^*) &= \int\limits_{\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}} \langle \mathbf{P}^*\mathbf{x},\mathbf{y}\rangle_{d_y} d\pi^*(\mathbf{x},\mathbf{y}) \\
&= \max_{\pi\in\Pi(\mathbb{P},\mathbb{Q})} \int\limits_{\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}} \langle \mathbf{P}^*\mathbf{x},\mathbf{y}\rangle_{d_y} d\pi(\mathbf{x},\mathbf{y}) \\
&= -\min_{T\colon \mathbb{R}^{d_x}\to\mathbb{R}^{d_y}} \mathcal{L}(\mathbf{P}^*, f^*, T),
\end{aligned}$$

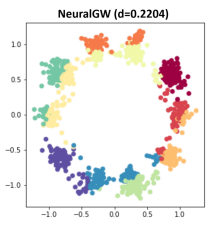
which completes the proof.

B Toy sanity-check experiment

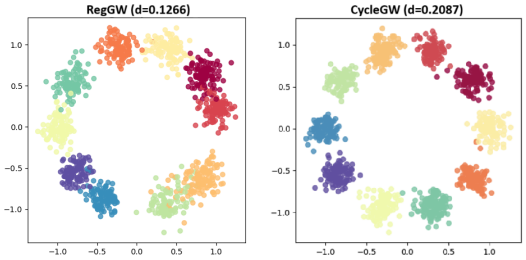
To illustrate the capabilities of the solvers and as a necessary sanity check for the implementations, we propose a toy experiment. In this setup, the source distribution is a mixture of Gaussians in $\mathbb{R}^3$ and the target is also a mixture of Gaussians but in $\mathbb{R}^2$, see Figure 5a. By choosing this experiment on incomparable spaces, we ensure the solvers are actually capable of dealing with a real Gromov-Wasserstein problem. The obtained results for the baselines can be found in Figure 5d and 5c, Figure 5b shows the result for our method, NeuralGW. As we can see, for all methods a component of the source distribution is mostly mapped to a component/neighbouring components of the target distribution, indicating the correct GW alignment. We also report the distortion d obtained after the methods were trained, the ground truth value for Figure 5a was computed using the discrete GW solver from the POT library.



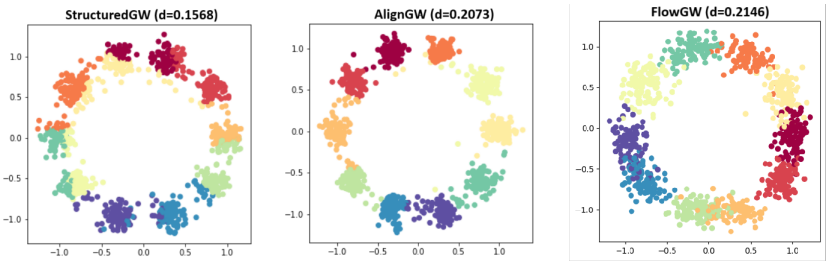
(a) Source, target distributions.



(b) NeuralGW solver (ours)



(c) Batched solvers (RegGW and CycleGW)



(d) Baseline solvers (StructuredGW, AlignGW, FlowGW).

Figure 5: Learned GWOT map T by different solvers; Toy (3D→2D) experiment.

C Metrics and embeddings

C.1 Metrics

This subsection explains how our evaluation metrics are computed. In the main text, we report what we refer to as *matching metrics*. These include Top-k accuracy, cosine similarity, and the Fraction of Samples Closer Than the True Match (FOSCTTM). Additionally, in Appendix D, we report the *marginal metrics*, Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), Sinkhorn divergence (Feydy et al., 2018), Bures-Wasserstein unexplained variance percentage (B$\mathbb{W}_2^2$-UVP) (Korotin et al., 2021a) and distortion. All the metrics (with exception of FOSCTTM) are computed on normalized data, therefore, $\tilde{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}\|_2$ where $\|\cdot\|_2$ denotes the Euclidean vector norm. Below we provide a concise explanation about how they are computed as well as the motivation to pick them.

Matching metrics

These metrics aim to provide an intuitive assessment of the quality of the predicted alignments by measuring how well the recovered correspondences match the ground truth. In order to compute them, it is necessary to know the actual ground-truth pairs.

- **Top k-accuracy** (↑). Let $\mathbf{x}_{\text{test}} \in \mathbb{R}^{m \times d_x}$ be the m test samples from the source domain and $\mathbf{y}_{\text{pred}} \in \mathbb{R}^{m \times d_y}$ their predicted counterparts. For each $\hat{y}_j$ in $\mathbf{y}_{\text{pred}}$, we use `faiss` to retrieve the k nearest neighbors from the *reference* spaces and denote their indices by $\{c_{j,1}, \ldots, c_{j,k}\}$. Given the

ground-truth labels $\{l_1, \ldots, l_m\}$ for the test samples, the Top-k accuracy is defined as

$$\text{Top } k = \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}\{\, l_j \in \{c_{j,1}, \ldots, c_{j,k}\}\,\}.$$

The size of the *reference* space may vary depending on the specific experiment, as discussed in the main text. This metric is particularly useful due to its intuitive interpretability. Additionally, evaluating performance across different values of k in Top-k accuracy offers insight into how well the solver reconstructs the target distribution while preserving its internal structure.

- **Fraction of Samples Closed Than the True Match (FOSCTTM).** (↓) For each ground-truth pair (x_i, y_i), let $\hat{y}_i = T(x_i)$ be the transported point and let $d(\cdot, \cdot)$ denote the Euclidean distance. We compute the proportion of target samples y_j that satisfy $d(\hat{y}_i, y_j) < d(\hat{y}_i, y_i)$. Formally,

$$\text{FOSCTTM}(i) = \frac{1}{N} \left| \{ j : d(\hat{y}_i, y_j) < d(\hat{y}_i, y_i) \} \right|.$$

The overall metric is the average $\text{FOSCTTM} = \frac{1}{N} \sum_{i=1}^{N} \text{FOSCTTM}(i)$. A perfect alignment yields $\text{FOSCTTM} = 0$. We note that this metric is insensitive to the choice of **reference** space, as it depends solely on distances to true paired samples.

- **Cosine similarity** (↑). It is computed between the predicted vector and the optimal pair vector in the target *reference* space.

Marginal metrics

We additionally report several metrics to ensure that the learned transport not only aligns individual samples but also accurately matches the overall distributions. The metrics considered are the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), the Sinkhorn divergence (Feydy et al., 2018), and the Bures-Wasserstein unexplained variance percentage (Korotin et al., 2021a). The metrics reported are computed for the testing data only. These metrics are computed between the empirical pushforward of the source distribution, $T_\sharp \hat{\mathbb{P}} \stackrel{\text{def}}{=} \frac{1}{N_x} \sum_{i=1}^{N_x} \delta(T(\mathbf{x}_i)) \approx \hat{\mathbb{Q}}$ (i.e., the output of the learned transport map or coupling) and the actual target empirical distribution, $\hat{\mathbb{Q}}$, to assess how well the learned mapping aligns the two distributions. However, these metrics do not directly reflect the quality of the solver itself, as they capture only distribution-level alignment and not, for example, matching or correspondence-based fidelity.

- **Maximum Mean Discrepancy (MMD)** (↓). It measures how different two distributions are by comparing their average representations in a high-dimensional feature space defined by a kernel. We denote $\mathbf{z} = T(\mathbf{x})$, therefore $\mathbf{z} \sim T_\sharp \mathbb{P}$. In our setup we used a Gaussian RBF kernel:

$$\text{MMD}^2(T_\sharp \mathbb{P}, \mathbb{Q}) = \mathbb{E}_{\mathbf{z}, \mathbf{z}' \sim T_\sharp \mathbb{P}}[k(\tilde{\mathbf{z}}, \tilde{\mathbf{z}}')] + \mathbb{E}_{\mathbf{y}, \mathbf{y}' \sim \mathbb{Q}}[k(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}')] - 2\mathbb{E}_{\mathbf{z} \sim T_\sharp \mathbb{P}, \mathbf{y} \sim \mathbb{Q}}[k(\tilde{\mathbf{z}}, \tilde{\mathbf{y}})],$$

where $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$, we set $\sigma = 10$.

- **Sinkhorn Divergence** (↓). It quantifies the discrepancy between two probability distributions, $T_\sharp \mathbb{P}$ and $\mathbb{Q}$ in our case.

$$S_\varepsilon(T_\sharp \mathbb{P}, \mathbb{Q}) = \text{OT}_\varepsilon(T_\sharp \mathbb{P}, \mathbb{Q}) - \frac{1}{2}\text{OT}_\varepsilon(T_\sharp \mathbb{P}, T_\sharp \mathbb{P}) - \frac{1}{2}\text{OT}_\varepsilon(\mathbb{Q}, \mathbb{Q}),$$

where OT_ε is the entropy-regularized optimal transport cost with regularization strength $\varepsilon > 0$. In our computations we set $\varepsilon = 0.01$.

- **Bures-Wasserstein unexplained variance percentage (B$\mathbb{W}_2^2$-UVP)** (↓). The B$\mathbb{W}_2^2$-UVP metric measures how much of the target distribution's covariance is not captured by the pushforward of the source distribution.

$$\text{B}\mathbb{W}_2^2\text{-UVP}\,(T\sharp\mathbb{P}, \mathbb{Q}) = \frac{100\%}{\frac{1}{2}\,\text{Var}(\mathbb{Q})}\,\text{B}\mathbb{W}_2^2\,(T\sharp\mathbb{P}, \mathbb{Q})$$

where $\text{B}\mathbb{W}_2^2\,(T\sharp\mathbb{P}, \mathbb{Q})$ is the Bures–Wasserstein metric between the Gaussian approximations of the distributions $T\sharp\mathbb{P}$ and $\mathbb{Q}$.

- **Distortion (↓).** This metric is related to the Gromov-Wasserstein cost introduced in (3).

$$\mathrm{D}_2^2(\mathbb{P}, T) \stackrel{\text{def}}{=} \int_{\mathbb{R}^{d_x} \times \mathbb{R}^{d_x}} (c_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') - c_{\mathcal{Y}}(T(\mathbf{x}), T(\mathbf{x}')))^2 d\mathbb{P}(\mathbf{x}) d\mathbb{P}(\mathbf{x}').$$

  In practice we compute it for the test samples according to the following expression:

$$\mathrm{D}(\tilde{\mathbf{x}}, T) \stackrel{\text{def}}{=} \frac{1}{n^2} \sum_{i,j=1}^{n} \left(c_{\mathcal{X}}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) - c_{\mathcal{Y}}(T(\tilde{\mathbf{x}}_i), T(\tilde{\mathbf{x}}_j))\right)^2.$$

  As this metric depends on the type of cost functions, we set $c_{\mathcal{X}} = c_{\mathcal{Y}} = c$ to be the cosine distance, $c(\tilde{\cdot}, \tilde{\cdot}') = 1 - \text{cos-sim}(\tilde{\cdot}, \tilde{\cdot}')$. This choice of c, along with the normalization of input vectors to unit norm, ensures consistency across methods. In particular, approaches such as StructuredGW and NeuralGW employ the inner product as their intra-domain cost function, whereas others directly use the cosine distance. By normalizing and adopting a cosine-based cost, we unify the representation and make the comparisons between methods more meaningful.

C.2 Embeddings

Here we provide details about dataset construction and the embeddings.

GloVe embeddings and datasets

The GloVe embeddings of words are generated using the GloVe algorithm (Pennington et al., 2014). Its main advantage is that the embedded vectors capture semantic relationships and exhibit linear substructures in the vector space. This allows meaningful computation of distances and alignments, which is central to the GWOT framework. The authors provide access to the GloVe embeddings of four datasets: Wikipedia, Gigaword, Common Crawl, and Twitter. We focus solely on the GloVe embeddings for the Twitter corpus. As we mentioned in the main text, we randomly take 400K samples from a total of 1.2 million, we use the pre-trained embeddings provided by the authors.

Byte-Pair embeddings and datasets

The original MUSE embeddings were obtained using fastText (Joulin et al., 2016); however, most methods considered in this paper struggle to align them effectively. Although AlignGW (Alvarez-Melis & Jaakkola, 2018) reports positive results on these embeddings, its evaluation relies on cross-domain similarity local scaling (CSLS) (Conneau et al., 2017). CSLS is commonly used to improve multilingual alignment, but introduces a corrective bias that may overestimate the intrinsic alignment capabilities of the solver. Therefore, we consider an alternative approach for embedding the MUSE vocabularies.

We construct new embeddings using the bpemb library[6], which provides pretrained subword embeddings in different dimensions. We select the largest available vocabulary (200K) to maximize coverage of the MUSE dictionaries, excluding words without corresponding embeddings. We consider English and Spanish, retaining only words with unique translations to ensure that the resulting datasets satisfy our definition of statistical relatedness in Section §4.1. Further details on BPEmb can be found in the original paper.

Following this procedure, we construct BPEmb datasets for the MUSE (English and Spanish) and Twitter vocabularies. Although filtering reduces the number of samples, the resulting datasets remain sufficiently large for training continuous methods.

WNTA embeddings and datasets

The VoxCeleb2 dataset (Chung et al., 2018) consists of over one million utterances from more than 6,000 speakers, making it one of the largest publicly available collections for speaker recognition research. For feature extraction, we adopt a pretrained WavLM model (Chen et al., 2021), a large-scale self-supervised architecture that jointly learns masked speech prediction and denoising. Following **?**, we use the WNTA (Weighted Non-Timbral Aggregation) variant of this model to produce embeddings that capture non-timbral voice traits (prosody, accent, etc.). The model is available on HuggingFace[7] and comprises a pretrained WavLM model followed by a custom top-layer block that performs statistics pooling and dimensionality reduction. We generate embeddings of two different dimensions, 128 and 64, and denote them as VoxCeleb2-WNTA. We consider only the first 600K samples.

[6] https://github.com/bheinzerling/bpemb/tree/master

[7] `https://huggingface.co/Orange/Speaker-wavLM-pro`

D Extended experiments

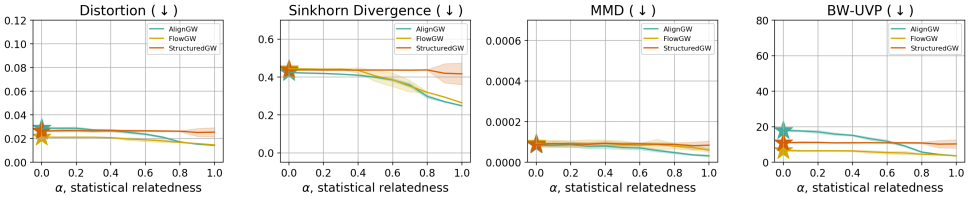
Overview of the conducted experiments. To help the reader navigating over all our considered experiments, We provide Table 1 summarizing the full list of experiments in our paper.

Dataset name	Embedding	Total samples	Train/test size	Source → Target	Evaluated on
Twitter	GloVe	400K	Baseline solvers 6000/2048	100→50	8048 samples Figure 3
					400K samples Figure 8a
				50→25	400K samples Figure 10b
				50→100	400K samples Figure 10a
				25→50	400K samples Figure 10c
Twitter	GloVe	400K	Continuous solvers 360K/2048	100→50	400K samples Figure 4a
				50→25	400K samples Figure 11b
				50→100	400K samples Figure 11a
MUSE	Byte-Pair	90K	Baseline solvers 6000/2048	100(English) → 50(English)	90K samples Figure 8b
			Continuous solvers 88K/2048	100(English) → 50(English)	90K samples Figure 4b
MUSE	Byte-Pair	60K	Baseline solvers 6000/2048	100(English) → 100(Spanish)	60K samples Table 3
			Continuous solvers 58K/2048	100(English) → 100(Spanish)	
VoxCeleb2	WavLM-WNTA	600K	Baseline solvers 6000/2048	128→64	8048 samples Figure 3c
					600K samples Figure 8
			Continuous solvers 598K/2048	128→64	600K samples Figure 4c

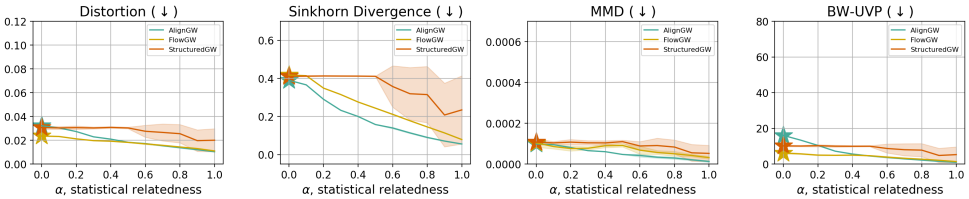
Table 1: Summary of experiments

D.1 Marginal metrics for main text experiments

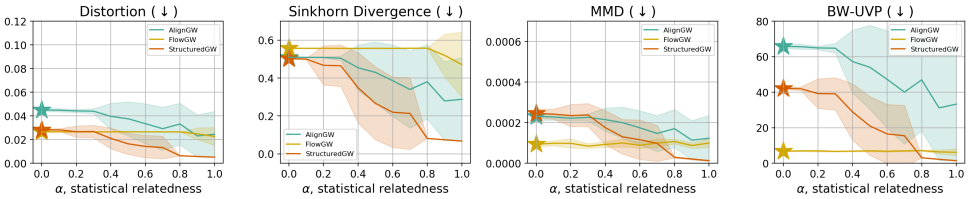
Here we report marginal metrics for our main experiments in Figures 3 and 4. The marginal metrics for the baselines can be found in Figure 6 and in Figure 7 for continuous solvers.



(a) Twitter-GloVe dataset (Source:100 → Target:50)

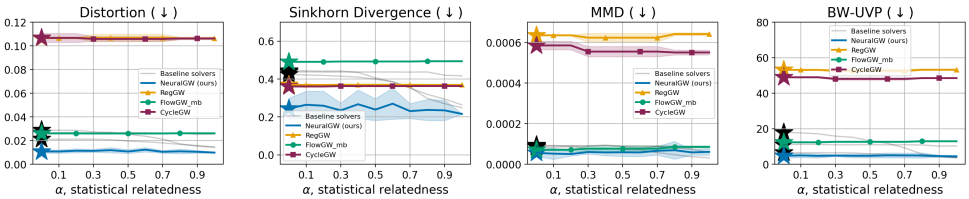


(b) MUSE-BPEmb (English) dataset (Source:100 → Target:50)

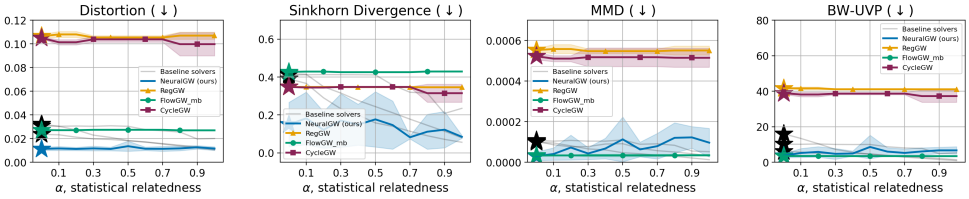


(c) VoxCeleb2-WNTA dataset (Source:128 → Target:64)

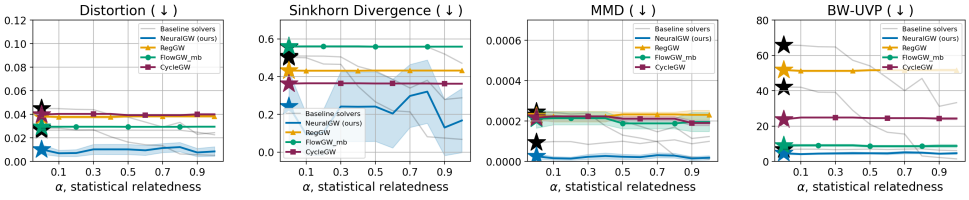
Figure 6: Marginal metrics of baseline GWOT solvers for **Twitter-GloVe** (6a), **MUSE-BPEmb (English)** (6b) and **VoxCeleb2-WNTA** (6c) at different levels of statistical relatedness α. Solvers were trained with $N_{\text{train}}/2 = 3000$ samples from spaces of 400K, 90K and 600K, respectively. The plot shows results on a 2048-sample test subset. Accuracy was computed on the *reference* space of $N_{\text{train}}+N_{\text{test}}=8048$ samples.



(a) Twitter-GloVe dataset (Source:100 → Target:50)



(b) MUSE-BPEmb (English) dataset (Source:100 → Target:50)



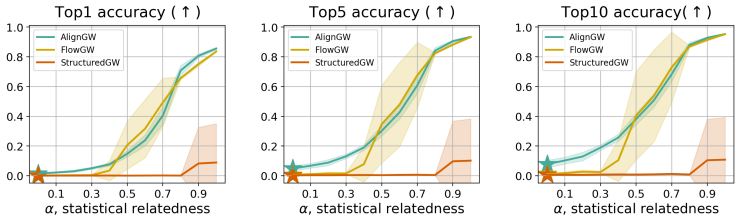
(c) VoxCeleb2-WNTA dataset (Source:128 → Target:64)

Figure 7: Marginal metrics for the GWOT solvers in the large-scale **Twitter-GloVe** (7a), **MUSE-BPEmb (English)** (7b) and **VoxCeleb2-WNTA** (7c) embeddings at different levels of statistical relatedness α. The solvers were trained with $N_{\text{train}}/2 = 198$K, $N_{\text{train}}/2 = 44$K and $N_{\text{train}}/2 = 288$K samples, respectively. The plot shows results on a 2048-sample test subset. Accuracy was computed on the *whole* data space of $N_{\text{train}}+N_{\text{test}}$ of 400K, 90K and 600K samples, respectively.

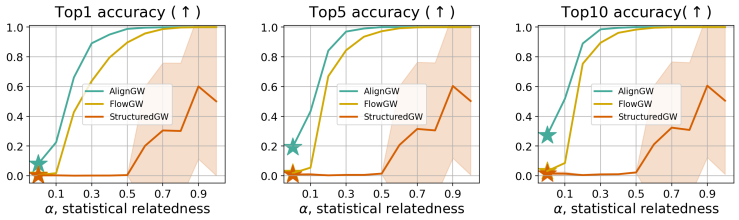
Conclusion. We observe that marginal metrics correlate moderately with matching metrics, indicating that most methods align the marginal distributions to some extent. However, these metrics alone are insufficient to evaluate solver performance, as they ignore the structural alignment central to Gromov-Wasserstein objectives. While useful for assessing how target-like the transported samples are, marginal metrics should be considered complementary. Notably, our method, NeuralGW, consistently outperforms baselines in both structural and marginal alignment, and is the only approach capable of accurately matching marginals across all levels of statistical relation between source and target domains.

D.2 Trained on small dataset, evaluated w.r.t. *whole* data space

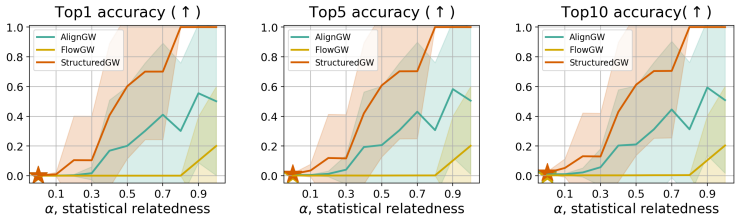
The accuracy evaluation involves identifying the k-nearest neighbours for a given vector within a target vector space. As discussed in Section §4.2, this space was limited to a small subset of $N_{train} + N_{test}$ samples for the experiments included in the main text, Figure 3. This is justified since the methods were trained on a similar number of samples. However, evaluating accuracy across the entire data space offer insights into the models' ability to capture the structures of the probability distributions. Therefore, in Figure 8 we show the Top-k accuracies w.r.t. the *whole* data space.



(a) Twitter-GloVe dataset (Source:100 → Target:50)



(b) MUSE-BPEmb (English) dataset (Source:100 → Target:50)



(c) VoxCeleb2-WNTA (Source:128 → Target:64)

Figure 8: Performance of the baseline GWOT solvers for **Twitter-GloVe** (8a), **MUSE-BPEmb (English)** (8b) and VoxCeleb2-WNTA (8c) at different levels of statistical relatedness α for the $100 \rightarrow 50$ setup. Solvers were trained with $N_{\text{train}}/2 = 3000$ samples from spaces of 400K and 90K, respectively; the plot shows results on a 2048-sample test subset. Accuracy metrics were computed over the *whole* data space of 400K and 90K samples, respectively.

Conclusion. The performance of the baseline solvers drops significantly when computing the accuracies on the *whole* data space. This decline is expected, as these methods are trained using a limited subset of the data, which restricts their ability to generalize effectively to unseen samples.

D.3 Experiments on statistically related batches for continuous solvers

As mentioned at the end of Subsection §5.2, we also report the performance of the continuous solvers when trained on statistically related batches under different levels of statistical relatedness α. Figure 9 presents the results for one of our main experimental settings—Twitter-GloVe, from dimension 100 to 50—originally introduced in Figure 4a. The total number of training samples remains fixed at $N_{\text{train}}/2 = 200$K, and we use batches of 500 samples across all methods. Further details on the hyperparameters are provided in Appendix E. We omit the MUSE-BPEmb case, as we expect qualitatively similar results.

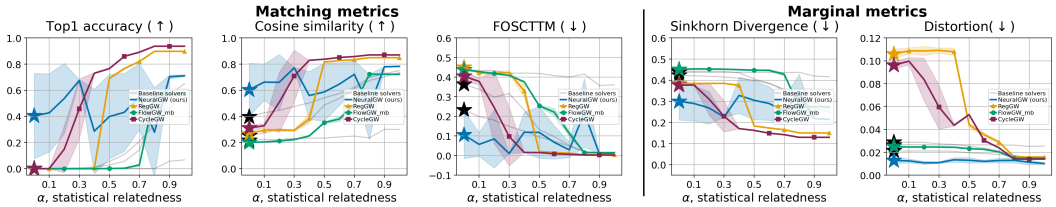


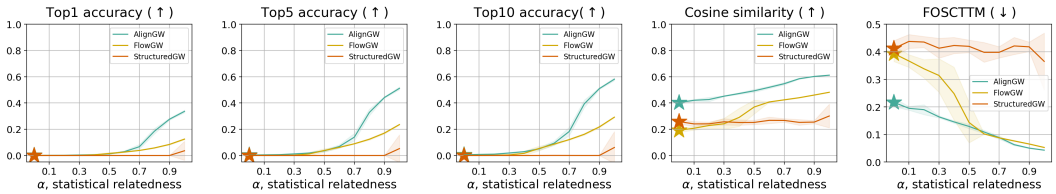
Figure 9: Matching metrics for continuous methods trained on **statistically related batches** on the Twitter-GloVe dataset (Source:100 → Target:50). $N_{train} = 400$K, $n_{batch} = 500$ and the Top-k accuracy are computed on the *whole* vector space.

Conclusion. As mentioned in the main text, it is important to clarify that this setup is both unfair and unrealistic, as it requires providing statistically related batches to the solvers. We observe that this tweak significantly boosts performance compared to the fair setup, which confirms the correct implementation of the solvers and their proper functionality under specific, but artificial, conditions.

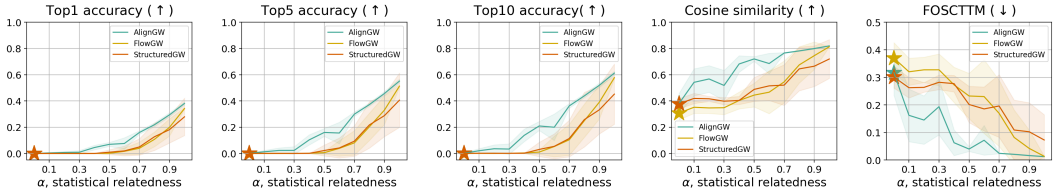
D.4 GloVe

Experiments on additional dimensionalities for baseline and continuous solvers

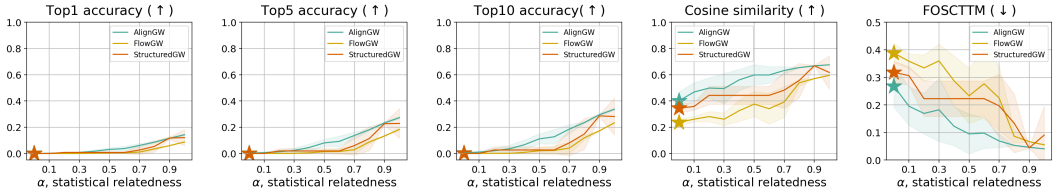
Figures 10 and 11 show the performance of the baseline and continuous methods, respectively, across additional dimensional configurations of the GloVe embedding. Specifically, we consider the transformations $50 \to 100$, $50 \to 25$, and $25 \to 50$. Accuracies are reported over the entire *whole* data space of 400K samples. The parameters, training and test samples are the same as in the original experiments. For the continuous solvers we exclude the $25 \to 50$ setup as it produces a result similar to $50 \to 100$ due to the inability of the continuous solvers to transform low-to-high dimensional manifolds.



(a) Source: 50 → Target:100



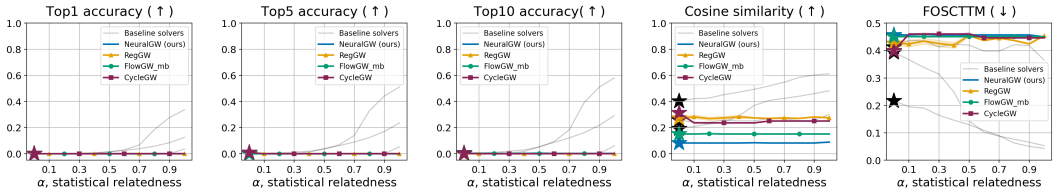
(b) Source: 50 → Target:25



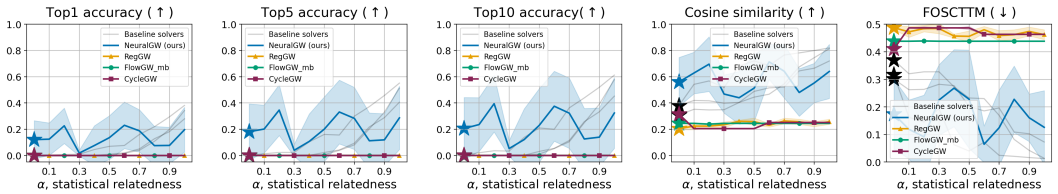
(c) Source: 25 → Target:50

Figure 10: Performance of baseline GWOT solvers for **Twitter-GloVe** at different statistical relatedness levels in additional setups. Accuracies computed on the *whole* data space of 400K samples.

Conclusion. The performance of both baseline and continuous solvers drops significantly when the embedding dimensionality is reduced. This is expected, as lower-dimensional vectors may fail to



(a) Source: 50 → Target:100



(b) Source: 50 → Target:25

Figure 11: Performance of the continuous GWOT solvers for the **Twitter-GloVe** embeddings at different levels of statistical relatedness α in additional dimensional setups.

adequately capture the structural features of the space. Conversely, none of the continuous solvers, including our NeuralGW, are able to effectively handle low-to-high dimensional setups.

Additional experiments for NeuralGW for high-to-low dimension in small dataset ($\mathbf{N_{train} = 6K}$). Here we consider the same setup used for the baseline methods in §4.2 to show how a neural approach performs when it is trained using a limited amount of samples, $N_{train} = 6$K, see Table 2.

Dimensions	Stat. Relatedness	Top 1	Top 5	Top 10	Cosine similarity	FOSCTTM
100→50	$\alpha = 0.2$	0.000	0.000	0.003	0.100	0.462
	$\alpha = 0.5$	0.000	0.001	0.004	0.107	0.470
	$\alpha = 1.0$	0.000	0.003	0.005	0.117	0.447
50→25	$\alpha = 0.2$	0.000	0.001	0.003	0.165	0.441
	$\alpha = 0.5$	0.001	0.004	0.007	0.176	0.423
	$\alpha = 1.0$	0.000	0.000	0.004	0.174	0.435
50→100	$\alpha = 0.2$	0.000	0.002	0.003	0.086	0.455
	$\alpha = 0.5$	0.000	0.003	0.005	0.084	0.459
	$\alpha = 1.0$	0.000	0.002	0.003	0.089	0.450
25→50	$\alpha = 0.2$	0.000	0.002	0.005	0.113	0.440
	$\alpha = 0.5$	0.000	0.000	0.001	0.095	0.464
	$\alpha = 1.0$	0.001	0.003	0.005	0.124	0.436

Table 2: Performance of the NeuralGW solver for the **Twitter-GloVe** embeddings at different levels of statistical relatedness α in **all** setups. The solver was trained with $N_{train}/2 = 3000$ samples from a whole space of 400K, this plot shows results for a testing subset of 2048 samples, the metrics were computed considering the *whole* space of 400K samples.

Conclusion.

NeuralGW is unable to model the inner structures when the number of training samples is small (6K). In general, adversarial algorithms like those used to train NeuralGW require plenty of data to obtain meaningful results.

D.5 BPEmb

MUSE-BPemb, source: English (100) → target: Spanish (100) We considered two different languages for source and target, the dimension of embedding was equal. The total number of samples was $N = 60$K, $N_{train} = 6$K for baseline solvers and $N_{train} = 57$K for NeuralGW, $N_{test} = 2048$ for both cases. The metrics were computed using the *whole* data space. See Table 3 for the results.

Conclusion. We tested a selection of methods for this experiment, but none of them, whether baseline or continuous approaches, were able to produce successful results in this setup. It is worth noting

Baseline Solvers						
	FlowGW		AlignGW		StructuredGW	
α	Top 10	FOSCTTM	Top 10	FOSCTTM	Top 10	FOSCTTM
0.0	0.0013	0.4222	0.0012	0.4056	0.0000	0.5037
0.5	0.0029	0.3983	0.0022	0.3768	0.0006	0.4541
1.0	0.0267	0.3321	0.0146	0.3085	0.0009	0.4443
Continuous solvers						
	RegGW		FlowGW (mb)		NeuralGW (ours)	
α	Top 10	FOSCTTM	Top 10	FOSCTTM	Top 10 mean (std)	FOSCTTM mean (std)
0.0	0.0001	0.4521	0.0006	0.4767	0.0518 (0.0633)	0.3811 (0.1286)
0.5	0.0009	0.4411	0.0007	0.4770	0.0284 (0.0564)	0.4379 (0.1067)
1.0	0.0013	0.4384	0.0008	0.4772	0.0748 (0.0751)	0.3514 (0.1321)

Table 3: Results for MUSE dataset for English and Spanish as source and target languages, respectively, both are 100-dimensional BP embeddings.

that the MUSE dataset was used in the original AlignGW paper; however, the type of embedding was different, resulting in a higher embedding dimensionality, which may have contributed to better performance in that context.

D.6 Biological dataset

As seen in the previous experiments, the baselines and NeuralGW solvers have their own limitations and drawbacks heavily linked to their nature. However, in spite of their independent performance, they could partially recover the inner geometry of the domains. In this section, we propose a stress test scenario in which the solvers of the conventional GWOT problem yield poor results.

We explore the performance of the baselines solvers and NeuralGW in a biological dataset called bone marrow (Luecken et al., 2021) which is considered in the method that we named FlowGW (Klein et al., 2023). This dataset consists of 6224 samples of two different RNA profiling methods (ATAC+GEX and ADT+GEX), the samples in each domain belong to the same donors, therefore, the real pairs are known. The dimensionality for source and target domains are 38 and 50, respectively. Results can be found in Table 4. All the solvers are tested using 5000 samples for training and the rest for testing.

	FlowGW		AlignGW		StructuredGW		NeuralGW	
α	Top 10	FOSCTTM	Top 10	FOSCTTM	Top 10	FOSCTTM	Top 10	FOSCTTM
0.2	0.004	0.486	0.004	0.488	0.001	0.489	0.003	0.479
0.5	0.004	0.489	0.004	0.483	0.004	0.487	0.003	0.514
1.0	0.004	0.49	0.004	0.484	0.004	0.49	0.003	0.459

Table 4: Results for bone marrow dataset.

Conclusion. In all the cases, the solvers could not properly replicate the inner geometry of the distributions even for totally statistically related setups, this leaded to get metrics corresponding to random guessing, i.e. accuracies close to 0 and FOSCTTM close to 0.5. There is one case of success for a solver trained on this dataset which corresponds to FlowGW (Klein et al., 2023), however, the reported results in their paper were obtained using a fused-GW solver instead of a conventional GW.

Finally, we can state that there is no general solver for the GWOT problem, all the currently available methods struggle when dealing with real world scenarios, i.e. statistically unrelated data, or with real world datasets, i.e. not consistent inner structures.

E Solvers' implementation details

All the experiments were done without any normalization for the source and target vectors and for all the studied methods (baselines and NeuralGW). A total of ten repetitions were performed. It is important to clarify that the parameters listed below are the ones we used to align the embeddings, they may require some tweaks to make them work in the toy setup.

StructuredGW.(Sebbouh et al., 2024) We used the code from the official repository:

`https://github.com/othmanesebbouh/prox_rot_aistats`

As specified in Section §3, the algorithm uses an iterative solver that updates the cost matrix $\mathbf{T}$ by implementing several methods depending on the type of regularization, we only use the exact computation without any regularization. The plan π is also updated every iteration by performing Sinkhorn iterations, we set this number of iterations to 1000. The entropy is set to $\varepsilon = 1e$-4. The total number of iterations is set to 200 or until convergence.

In this implementation, the authors use the Optimal Transport Tools (OTT) library (Cuturi et al., 2022). The computation time per repetition until convergence was 30 minutes in average on a CPU.

AlignGW.(Alvarez-Melis & Jaakkola, 2018) We use the official implementation of the method taken from the repository:

`https://github.com/dmelis/otalign`

We set the entropy term to ε =1e-3 and use the cosine distance to compute the source and target intra-cost matrices $\mathbf{C}^x$ and $\mathbf{C}^y$. We later normalize them by dividing them by their respective means as proposed in the original implementation. The model was trained on a CPU and the average training time was 10 minutes per repeat. The implementation uses the POT library. To get the predicted target from the obtained couplig $\mathbf{T}$, we first compute the indices `idx=np.argmax(T, axis=1)` and then the chosen target samples are `y_pred=y[idx]`. We chose this strategy as it gave better performance. Then, we train a `scikit-learn`'s MLPRegressor on top of this prediction as an inference method for test data, the parameters are: hidden_layer_sizes=256, random_state=1, max_iter=500.

FlowGW. (Klein et al., 2023) We used the implementation provided in the OTT library for the GENOT with slight modifications to adapt it to our pipeline. The hyperparameters for the vector field were as follows: Number of frequencies: 128, layers per block: 8, hidden dimension: 1024, activation function: SiLu, optimizer: AdamW (lr=1e-4). The Gromov-Wasserstein solver was set to work with entropy ε = 1e-3 and using cosine distance to compute the intra-domain matrices.

RegGW. (Uscidda et al., 2024) For the sake of fairness, our implementation of this solver is based on the publicly available implementation for the Monge gap regularizer from the OTT library (Cuturi et al., 2022). To compute the Gromov-Wasserstein distance we used the GW solver from the library and took the entropy regularized cost. The following parameters were used for training: $\varepsilon_{fit} = 0.01$, $\varepsilon_{reg} = 0.001$, $\lambda = 1$. The transport model was parametrized as an MLP with $[512, 256, 256]$ dimensions for the hidden layers, the optimizer learning rate was 1e-4 and a batch size of 256.

CycleGW (Zhang et al., 2021). We used the code provided by the authors in their repository:

`https://github.com/ZhengxinZh/GMMD`

As in the original implementation, we used fully connected neural networks (FCNN) to parametrize f and g, in both cases the network consisted on a single layer with 512 neurons, and trained using the Adam optimizer with a learning rate of $1e$-3 (as suggested in the original paper). Both regularization parameters, λ_x and λ_y in the original paper, were set to 0.1. The multiplier of the distortion term was set to 5e-4. In spite of following the original implementation, it was not possible to make the solver work for our setups beyond the toy experiment.

E.1 NeuralGW.

The innerGW problem in (4) can be optimized using our Algorithm 1.

Every experiment runs for 150 epochs. Each network iteration takes new i.i.d. samples from source and target datasets. f and T are parametrized using multi-layer perceptrons with n_l with width h. Matrix $\mathbf{P}$ is parameterized by a single-layer linear neural network, we use GeoTorch Lezcano-Casado (2019) to constrain each row of its rows to lie on the unit sphere, ensuring row-wise normalization is maintained throughout training. These models are trained for k_f, k_T and k_p iterations, respectively. These parameters were chosen following the inner structure of the min-max-min problem, we optimized the inner transport model T_γ more often than the others, from here we fine-tuned the values to get good metrics. The implementation details can be seen in Table 5. The batch size is 512 for the experiments with 400K, 90K and 600K samples and 64 for the experiments with 6K. Every run takes around to 2 hours running (for large-scale experiments) on a GPU NVIDIA A100. Our code is written in `PyTorch`.

Algorithm 1 Training algorithm for Neural Gromov-Wasserstein OT

Input: Distributions $\mathbb{P}$ and $\mathbb{Q}$ obtained from samples.
Output: Optimal matrix P_ω, critic f_θ and transport map T_γ.
for $i = 1, 2, 3, \ldots, n_{epochs}$ **do**
  for $i = 1, 2, 3, \ldots, k_P$ **do**
    Sample batch from source and target distributions $X \sim \mathbb{P}$, $Y \sim \mathbb{Q}$.
    Compute P loss $\mathcal{L}_P = -\frac{1}{N}\sum_{n=1}^{N}\langle \mathbf{P}_\omega \mathbf{x}, T_\gamma(\mathbf{x})\rangle$
    Gradient step over ω using $\frac{\partial \mathcal{L}_P}{\partial \omega}$
    for $j = 1, 2, 3, \ldots, k_f$ **do**
      for $k = 1, 2, 3, \ldots, k_T$ **do**
        Sample batch from source and target distributions $X \sim \mathbb{P}$, $Y \sim \mathbb{Q}$.
        Compute mover loss $\mathcal{L}_T = -\frac{1}{N}\sum_{n=1}^{N}\langle \mathbf{P}_\omega \mathbf{x}, T_\gamma(\mathbf{x})\rangle - \frac{1}{N}\sum_{n=1}^{N} f_\theta(T(\mathbf{x}))$
        Gradient step over γ using $\frac{\partial \mathcal{L}_T}{\partial \gamma}$
      Sample batch from source and target distributions $X \sim \mathbb{P}$, $Y \sim \mathbb{Q}$.
      Compute critic loss $\mathcal{L}_f = \frac{1}{N}\sum_{n=1}^{N} f_\theta(T_\gamma(\mathbf{x})) - \frac{1}{N}\sum_{n=1}^{N} f_\theta(\mathbf{y})$
      Gradient step over θ using $\frac{\partial \mathcal{L}_f}{\partial \theta}$

Table 5: Parameters for NeuralGW.

	Model	k	n_l	h	lr
100→50	f	k_f =1	4	1024	1e-4
50→25	T	k_T =5			
25→50	P	k_P =1			
50→25	f	k_f =1	4	256	1e-4
	T	k_T =10			
	P	k_P =1			

To stabilize NeuralGW during training on word and voice embeddings, we incorporate the well-known R_1 gradient penalty Mescheder et al. (2018) into the critic loss $\mathcal{L}_f$, following GAN-based training methods. The resulting objective, minimized with respect to the critic parameters θ, is given by

$$\mathcal{L}_f = \frac{1}{N}\sum_{n=1}^{N} f_\theta(T_\gamma(\mathbf{x}_n)) - \frac{1}{N}\sum_{n=1}^{N} f_\theta(\mathbf{y}_n) + \lambda \frac{1}{N}\sum_{n=1}^{N} \|\nabla_{\mathbf{y}_n} f_\theta(\mathbf{y}_n)\|_2^2 ,$$

where $\lambda = 0.1$ controls the regularization strength. with $\lambda = 0.1$. All the results for NeuralGW on embeddings use this regularizer for training the critic.

F Broader discussions

F.1 Discrete GW solvers under low statistical relatedness data scenario

Our experimental results (Section 4.2 of the main text) testify that the discrete baseline solvers perform unsatisfactory when the level of statistical relatedness α in the data tends to zero. We hypothesis that the main reason behind this behavior is as follows. Small amount of data used for discrete solvers hardly could "catch" the intrinsic geometry of the underlining distribution. When we apply discrete GW solver, the Gromov-Wasserstein mapping is learned between the geome-

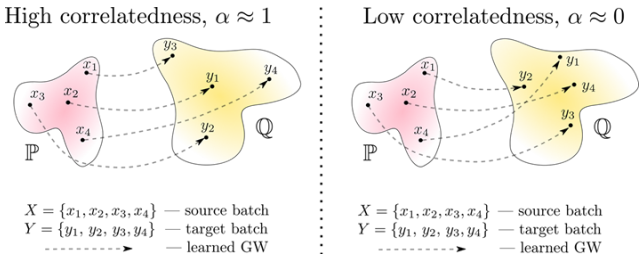


Figure 12: Discrete GW maps fitted under high (left) and low (right) levels of statistical relatedness.

tries induced by sample distributions, not original distributions, see Figure 12 for illustration. These "induced" geometries may be different from the original ones, they may have other symmetries and other properties. Matching them may result in Gromov-Wasserstein map which is quite different from the real GW map. Importantly, our hypothesis above is (indirectly) supported by recently derived sample complexity rates for Gromov-Wasserstein distance (Zhang et al., 2024). The authors consider GWOT problem (3) for quadratic costs $c_{\mathcal{X}}, c_{\mathcal{Y}}$. In this case, they observe that the sample convergence rate of the GW distance between empirical (sample-based) measures to the GW distance between reference measures $(\mathbb{P}, \mathbb{Q})$ is of the order $N^{\frac{-2}{\min d_{\mathcal{X}}, d_{\mathcal{Y}}}}$, where N is the number of samples. Note that this discouraging exponential rate corresponds exactly to statistically unrelated ($\alpha \approx 0$) setup, where the discrete samples from the source and target reference distributions are independent. Besides, the sample complexity of *GW maps*, not distances, may be even worse.

On the other hand, when the level of statistical relatedness is high ($\alpha = 1$), the GW problem is reduced to finding the proper permutation of the data, see Figure 12, left part. The true solution of discrete GW in this case *coincides* with the true underlining GW map. If the learned map properly generalizes to new (unseen) samples, then the resulting performance is expected to be satisfactory. It is the behaviour we observe in our experiments, Section 4.2.

G NeuralGW initialization strategy

To alleviate the stability challenges arising from the min-max-min optimization in NeuralGW, we initialize the learnable matrix $\mathbf{P}_\omega$ using the solution of the InnerGW problem (Equation (4)) between two non-centered Gaussian distributions. We denote this optimal matrix by $\mathbf{P}_\mathbf{g}^*$, which we obtain by combining (Vayer, 2020, Lemma 4.4.2) with results from (Delon et al., 2022, Section 3). Our initialization procedure consists of the following steps:

1. Given the empirical source and target distributions $X = \{\mathbf{x}_i\}_{i=1}^N$ and $Y = \{\mathbf{y}_i\}_{i=1}^N$, we compute $m_x = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i$, $m_y = \frac{1}{N}\sum_{i=1}^N \mathbf{y}_i$, $\Sigma_x = \mathrm{Var}(X)$, and $\Sigma_y = \mathrm{Var}(Y)$.
2. We determine the specific optimal $\mathbf{P}_\mathbf{g}^*$ that corresponds to these empirical means and covariances.
3. In Algorithm 1, we freeze the training of $\mathbf{P}_\omega$ and train only T_γ and f_θ for 15 epochs.
4. After this initial pre-training of T_γ and f_θ, we unfreeze $\mathbf{P}_\omega$ and resume regular training as described in Algorithm 1.

This strategy improves the training stability of NeuralGW. For the Twitter-GloVe and MUSE-BPEmb datasets, it allows us to remove the R_1 regularizer presented in Appendix E.1. However, the R_1 regularizer remains necessary for the VoxCeleb2-WNTA dataset.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper includes comparisons of different Gromov-Wasserstein solvers in continuous time. It also introduces a new solver named NeuralGW that aims to solve the issues of the currently available approaches. This directly correlates to the claims stated in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We clearly state the limitations of the current solvers at the end of Subsection §4.2. We also discuss the limitations of our approach at the end of Subsection §5.2. Our paper includes a deep discussion regarding statistical relatedness in Section §4.1. We also include details regarding number of runs, size of the datasets and setups in Subsection §4.2.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have two main theoretical results intoroduced in Section §5.1, Lemma 5.1 and Theorem 5.2 which proofs are contained in Appendix §A.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the required information to reproduce the results in the paper, the construction of the datasets can be found in Section §4.2 and Appendix C.2. Information about code for baselines and hyperparameters used can be found in Appendix E. The algorithm for our new NeuralGW approach is provided in Appendix E.1, the code will be included in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the datasets we use in the paper are publicly available and we provide links to them in Subsection §4.2. The code of baselines is based on official implementations and links to them are available in Appendix E. Code not included there will be included in the supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Details regarding how the data is obtained and split is included in Subsection §4.2 and the hyperparameters for the models are shared in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our results are conducted over several repeats on different seeds and we report mean and standard deviation values in the main text as well as in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information regarding approximate execution timings in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: There are potential societal consequences of our work, none of which we feel must be specifically highlighted. here

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not provide any high risk pretrained model or data.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We carefully credit the authors of papers and code through the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce a new solver named NeuralGW, the code is included in Appendix E.1 and the code will be included in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not include any type of experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not perform research with human subjects or crowdsourcing.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for writing and rephrasing. Therefore, they do not impact the core methodology or originality of the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.