Graph Optimal Transport for Cross-Domain Alignment

Liqun Chen[1] **Zhe Gan**[2] **Yu Cheng**[2] **Linjie Li**[2] **Lawrence Carin**[1] **Jingjing Liu**[2]

Abstract

Cross-domain alignment between two sets of entities (*e.g.*, objects in an image, words in a sentence) is fundamental to both computer vision and natural language processing. Existing methods mainly focus on designing advanced attention mechanisms to simulate soft alignment, with no training signals to *explicitly* encourage alignment. The learned attention matrices are also dense and lacks interpretability. We propose Graph Optimal Transport (GOT), a principled framework that germinates from recent advances in Optimal Transport (OT). In GOT, cross-domain alignment is formulated as a graph matching problem, by representing entities into a dynamically-constructed graph. Two types of OT distances are considered: (i) Wasserstein distance (WD) for node (entity) matching; and (ii) Gromov-Wasserstein distance (GWD) for edge (structure) matching. Both WD and GWD can be incorporated into existing neural network models, effectively acting as a drop-in regularizer. The inferred transport plan also yields *sparse* and *self-normalized* alignment, enhancing the interpretability of the learned model. Experiments show consistent outperformance of GOT over baselines across a wide range of tasks, including image-text retrieval, visual question answering, image captioning, machine translation, and text summarization.

1. Introduction

Cross-domain Alignment (CDA), which aims to associate related entities across different domains, plays a central role in a wide range of deep learning tasks, such as image-text retrieval (Karpathy & Fei-Fei, 2015; Lee et al., 2018), visual question answering (VQA) (Malinowski & Fritz, 2014; An-

tol et al., 2015), and machine translation (Bahdanau et al., 2015; Vaswani et al., 2017). Considering VQA as an example, in order to understand the contexts in the image and the question, a model needs to interpret the latent alignment between regions in the input image and words in the question. Specifically, a good model should: (i) identify entities of interest in both the image (*e.g.*, objects/regions) and the question (*e.g.*, words/phrases), (ii) quantify both intra-domain (within the image or sentence) and cross-domain relations between these entities, and then (iii) design good metrics for measuring the quality of cross-domain alignment drawn from these relations, in order to optimize towards better results.

CDA is particularly challenging as it constitutes a *weakly supervised learning* task. That is, only paired spaces of entity are given (*e.g.*, an image paired with a question), while the ground-truth relations between these entities are not provided (*e.g.*, no supervision signal for a "dog" region in an image aligning with the word "dog" in the question). State-of-the-art methods principally focus on designing advanced attention mechanisms to simulate soft alignment (Bahdanau et al., 2015; Xu et al., 2015; Yang et al., 2016b;a; Vaswani et al., 2017). For example, Lee et al. (2018); Kim et al. (2018); Yu et al. (2019) have shown that learned co-attention can model dense interactions between entities and infer cross-domain latent alignments for vision-and-language tasks. Graph attention has also been applied to relational reasoning for image captioning (Yao et al., 2018) and VQA (Li et al., 2019a), such as graph attention network (GAT) (Veličković et al., 2018) for capturing relations between entities in a graph via masked attention, and graph matching network (GMN) (Li et al., 2019b) for graph alignment via cross-graph soft attention. However, conventional attention mechanisms are guided by task-specific losses, with no training signal to *explicitly* encourage alignment. And the learned attention matrices are often dense and uninterpretable, thus inducing less effective relational inference.

We address whether there is a more principled approach to scalable discovery of cross-domain relations. To explore this, we present Graph Optimal Transport (GOT), a new framework for cross-domain alignment that leverages recent advances in Optimal Transport (OT). OT-based learning aims to optimize for distribution matching via minimizing the cost of transporting one distribution to another. We ex-

tend this to CDA (here a domain can be language, images, videos, etc.). The transport plan is thus redefined as transporting the distribution of embeddings from one domain (*e.g.*, language) to another (*e.g.*, images). By minimizing the cost of the learned transport plan, we explicitly minimize the embedding distance between the domains, *i.e.*, optimizing towards better cross-domain alignment.

Specifically, we convert entities (*e.g.*, objects, words) in each domain (*e.g.*, image, sentence) into a graph, where each entity is represented by a feature vector, and the graph representations are recurrently updated via graph propagation. Cross-domain alignment can then be formulated into a graph matching problem, and be addressed by calculating matching scores based on graph distance. In our GOT framework, we utilize two types of OT distance: (*i*) Wasserstein distance (WD) (Peyré et al., 2019) is applied to node (entity) matching, and (*ii*) Gromov-Wasserstein distance (GWD) (Peyré et al., 2016) is adopted for edge (structure) matching. WD only measures the distance between node embeddings across domains, without considering topological information encoded in the graphs. GWD, on the other hand, compares graph structures by measuring the distance between a pair of nodes within each graph. When fused together, the two distances allow the proposed GOT framework to effectively take into account both node and edge information for better graph matching.

The main contributions of this work are summarized as follows. (*i*) We propose Graph Optimal Transport (GOT), a new framework that tackles cross-domain alignment by adopting Optimal Transport for graph matching. (*ii*) GOT is compatible with existing neural network models, acting as an effective drop-in regularizer to the original objective.

(*iii*) To demonstrate the versatile generalization ability of the proposed approach, we conduct extensive experiments on five diverse tasks: image-text retrieval, visual question answering, image captioning, machine translation, and text summarization. Results show that GOT provides consistent performance enhancement over strong baselines across all the tasks.

2. Graph Optimal Transport Framework

We first introduce the problem formulation of Cross-domain Alignment in Sec. 2.1, then present the proposed Graph Optimal Transport (GOT) framework in Secs. 2.2- 2.4.

2.1. Problem Formulation

Assume we have two sets of entities from two different domains (denoted as $\mathbb{D}_x$ and $\mathbb{D}_y$). For each set, every entity is represented by a feature vector, *i.e.*, $\tilde{\mathbf{X}} = \{\tilde{\boldsymbol{x}}_i\}_{i=1}^n$ and $\tilde{\mathbf{Y}} = \{\tilde{\boldsymbol{y}}_j\}_{j=1}^m$, where n and m are the number of entities in each domain, respectively. The scope of this paper mainly

focuses on tasks involving images and text, thus entities here correspond to objects in an image or words in a sentence. An image can be represented as a set of detected objects, each associated with a feature vector (*e.g.*, from a pre-trained Faster RCNN (Anderson et al., 2018)). With a word embedding layer, a sentence can be represented as a sequence of word feature vectors.

A deep neural network $f_{\boldsymbol{\theta}}(\cdot)$ can be designed to take both $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ as initial inputs, and generate contextualized representations:

$$\mathbf{X}, \mathbf{Y} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}), \qquad (1)$$

where $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^n$, $\mathbf{Y} = \{\boldsymbol{y}_j\}_{j=1}^m$, and advanced attention mechanisms (Bahdanau et al., 2015; Vaswani et al., 2017) can be applied to $f_{\boldsymbol{\theta}}(\cdot)$ to simulate soft alignment. The final supervision signal $\boldsymbol{l}$ is then used to learn $\boldsymbol{\theta}$, *i.e.*, the training objective is defined as:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{sup}}(\mathbf{X}, \mathbf{Y}, \boldsymbol{l}). \qquad (2)$$

Several instantiations for different tasks are summarized as follows: (*i*) *Image-text Retrieval*. $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are image and text features, respectively. $\boldsymbol{l}$ is the binary label, indicating whether the input image and sentence are paired or not. Here $f_{\boldsymbol{\theta}}(\cdot)$ can be the SCAN model (Lee et al., 2018), and $\mathcal{L}_{\text{sup}}(\cdot)$ corresponds to ranking loss (Faghri et al., 2018; Chechik et al., 2010). (*ii*) *VQA*. Here $\boldsymbol{l}$ denotes the ground-truth answer, $f_{\boldsymbol{\theta}}(\cdot)$ can be BUTD or BAN model (Anderson et al., 2018; Kim et al., 2018), $\mathcal{L}_{\text{sup}}(\cdot)$ is cross-entropy loss. (*iii*) *Machine Translation*. $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are textual features from the source and target sentences, respectively. Here $f_{\boldsymbol{\theta}}(\cdot)$ can be an encoder-decoder Transformer model (Vaswani et al., 2017), and $\mathcal{L}_{\text{sup}}(\cdot)$ corresponds to cross-entropy loss that models the conditional distribution of $p(\mathbf{Y}|\mathbf{X})$, and here $\boldsymbol{l}$ is not needed. To simplify subsequent discussions, all the tasks are abstracted into $f_{\boldsymbol{\theta}}(\cdot)$ and $\mathcal{L}_{\text{sup}}(\cdot)$.

In most previous work, the learned attention can be interpreted as a soft alignment between $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$. However, only the final supervision signal $\mathcal{L}_{\text{sup}}(\cdot)$ is used for model training, thus lacking an objective *explicitly* encouraging cross-domain alignment. To enforce alignment and cast a regularizing effect on model training, we propose a new objective for Cross-domain Alignment:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{sup}}(\mathbf{X}, \mathbf{Y}, \boldsymbol{l}) + \alpha \cdot \mathcal{L}_{\text{CDA}}(\mathbf{X}, \mathbf{Y}), \qquad (3)$$

where $\mathcal{L}_{\text{CDA}}(\cdot)$ is a regularization term that encourages alignments *explicitly*, and α is a hyper-parameter that balances the two terms. Through gradient back-propagation, the learned $\boldsymbol{\theta}$ supports more effective relational inference. In Section 2.4 we describe $\mathcal{L}_{\text{CDA}}(\cdot)$ in detail.

2.2. Dynamic Graph Construction

Image and text data inherently contain rich sequential/spatial structures. By representing them as graphs and performing
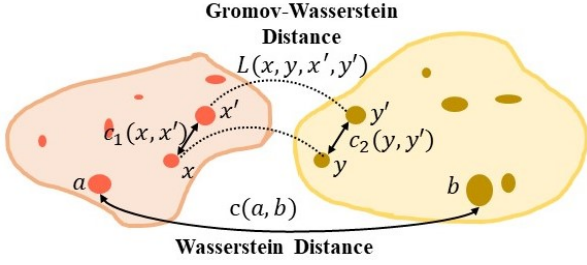
Figure 1. Illustration of the Wasserstein Distance (WD) and the Gromov-Wasserstein Distance (GWD) used for node and structure matching, respectively. WD: $c(\boldsymbol{a}, \boldsymbol{b})$ is calculated between node $\boldsymbol{a}$ and $\boldsymbol{b}$ across two domains; GWD: $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}')$ is calculated between edge $c_1(\boldsymbol{x}, \boldsymbol{x}')$ and $c_2(\boldsymbol{y}, \boldsymbol{y}')$. See Sec. 2.3 for details.

graph alignment, not only cross-domain relations can be modeled, but also intra-domain relations are exploited (*e.g.*, semantic/spatial relations among detected objects in an image (Li et al., 2019a)).

Given $\mathbf{X}$, we aim to construct a graph $\mathcal{G}_x(\mathbb{V}_x, \mathcal{E}_x)$, where each node $i \in \mathbb{V}_x$ is represented by a feature vector $\boldsymbol{x}_i$. To add edges $\mathcal{E}_x$, we first calculate the similarity between a pair of entities inside a graph: $\mathbf{C}_x = \{\cos(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j} \in \mathbb{R}^{n \times n}$. Further, we define $\mathbf{C}_x = \max(\mathbf{C}_x - \tau, 0)$, where τ is a threshold hyper-parameter for the graph cost matrix. Empirically, τ is set to 0.1. If $[\mathbf{C}_x]_{ij} > 0$, an edge is added between node i and j. Given $\mathbf{Y}$, another graph $\mathcal{G}_y(\mathbb{V}_y, \mathcal{E}_y)$ can be similarly constructed. Since both $\mathbf{X}$ and $\mathbf{Y}$ are evolving through the update of parameters $\boldsymbol{\theta}$ during training, this graph construction process is considered "*dynamic*". By representing the entities in both domains as graphs, cross-domain alignment is naturally formulated into a graph matching problem.

In our proposed framework, we use Optimal Transport (OT) for graph matching, where a transport plan $\mathbf{T} \in \mathbb{R}^{n \times m}$ is learned to optimize the alignment between $\mathbf{X}$ and $\mathbf{Y}$. OT possesses several idiosyncratic characteristics that make it a good choice for solving CDA problem. (*i*) *Self-normalization*: all the elements of $\mathbf{T}^*$ sum to 1 (Peyré et al., 2019). (*ii*) *Sparsity*: when solved exactly, OT yields a sparse solution $\mathbf{T}^*$ containing $(2r - 1)$ non-zero elements at most, where $r = \max(n, m)$, leading to a more interpretable and robust alignment (De Goes et al., 2011). (*iii*) *Efficiency*: compared with conventional linear programming solvers, our solution can be readily obtained using iterative procedures that only require matrix-vector products (Xie et al., 2018), hence readily applicable to large deep neural networks.

2.3. Optimal Transport Distances

As illustrated in Figure 1, two types of OT distance are adopted for our graph matching: Wasserstein distance for

Algorithm 1 Computing Wasserstein Distance.

1: **Input:** $\{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{y}_j\}_{j=1}^n, \beta$
2: $\boldsymbol{\sigma} = \frac{1}{n}\mathbf{1_n}, \mathbf{T}^{(1)} = \mathbf{11}^\top$
3: $\mathbf{C}_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j), \mathbf{A}_{ij} = e^{-\frac{\mathbf{C}_{ij}}{\beta}}$
4: **for** $t = 1, 2, 3 \ldots$ **do**
5:   $\mathbf{Q} = \mathbf{A} \odot \mathbf{T}^{(t)}$ // $\odot$ is Hadamard product
6:   **for** $k = 1, 2, 3, \ldots K$ **do**
7:     $\boldsymbol{\delta} = \frac{1}{n\mathbf{Q}\boldsymbol{\sigma}}, \boldsymbol{\sigma} = \frac{1}{n\mathbf{Q}^\top \boldsymbol{\delta}}$
8:   **end for**
9:   $\mathbf{T}^{(t+1)} = \text{diag}(\boldsymbol{\delta})\mathbf{Q}\text{diag}(\boldsymbol{\sigma})$
10: **end for**
11: $\mathcal{D}_{wd} = \langle \mathbf{C}^\top, \mathbf{T} \rangle$
12: Return $\mathbf{T}, \mathcal{D}_w$ // $\langle \cdot, \cdot \rangle$ is the Frobenius dot-product

Algorithm 2 Computing Gromov-Wasserstein Distance.

1: **Input:** $\{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{y}_j\}_{j=1}^n$, probability vectors $\boldsymbol{p}, \boldsymbol{q}$
2: Compute intra-domain similarities:
3:   $[\mathbf{C}_x]_{ij} = \cos(\boldsymbol{x}_i, \boldsymbol{x}_j), [\mathbf{C}_y]_{ij} = \cos(\boldsymbol{y}_i, \boldsymbol{y}_j),$
4: Compute cross-domain similarities:
5:   $\mathbf{C}_{xy} = \mathbf{C}_x^2 \boldsymbol{p} \mathbf{1_m}^\top + \mathbf{C}_y \boldsymbol{q} (\mathbf{C}_y^2)^\top$
6: **for** $t = 1, 2, 3 \ldots$ **do**
7:   // Compute the pseudo-cost matrix
8:   $\mathcal{L} = \mathbf{C}_{xy} - 2\mathbf{C}_x \mathbf{T} \mathbf{C}_y^\top$
9:   Apply Algorithm 1 to solve transport plan $\mathbf{T}$
10: **end for**
11: $\mathcal{D}_{gw} = \langle \mathcal{L}^\top, \mathbf{T} \rangle$
12: Return $\mathbf{T}, \mathcal{D}_{gw}$

node matching, and Gromov-Wasserstein distance for edge matching.

Wasserstein Distance Wasserstein distance (WD) is commonly used for matching two distributions (*e.g.*, two sets of node embeddings). In our setting, discrete WD can be used as a solver for network flow and bipartite matching (Luise et al., 2018). The definition of WD is described as follows.

Definition 2.1. Let $\boldsymbol{\mu} \in \mathbf{P}(\mathbb{X}), \boldsymbol{\nu} \in \mathbf{P}(\mathbb{Y})$ denote two discrete distributions, formulated as $\boldsymbol{\mu} = \sum_{i=1}^n \mathbf{u}_i \delta_{\boldsymbol{x}_i}$ and $\boldsymbol{\nu} = \sum_{j=1}^m \mathbf{v}_j \delta_{\boldsymbol{y}_j}$, with $\delta_{\boldsymbol{x}}$ as the Dirac function centered on $\boldsymbol{x}$. $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$ denotes all the joint distributions $\boldsymbol{\gamma}(\boldsymbol{x}, \boldsymbol{y})$, with marginals $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\nu}(\boldsymbol{y})$. The weight vectors $\mathbf{u} = \{\mathbf{u}_i\}_{i=1}^n \in \Delta_n$ and $\mathbf{v} = \{\mathbf{v}_i\}_{i=1}^m \in \Delta_m$ belong to the n- and m-dimensional simplex, respectively (*i.e.*, $\sum_{i=1}^n \mathbf{u}_i = \sum_{j=1}^m \mathbf{v}_j = 1$), where both $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are probability distributions. The Wasserstein distance between the two discrete distributions $\boldsymbol{\mu}, \boldsymbol{\nu}$ is defined as:

$$\mathcal{D}_w(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\boldsymbol{\gamma} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \boldsymbol{\gamma}} [c(\boldsymbol{x}, \boldsymbol{y})]$$

$$= \min_{\mathbf{T} \in \Pi(\mathbf{u}, \mathbf{v})} \sum_{i=1}^n \sum_{j=1}^m \mathbf{T}_{ij} \cdot c(\boldsymbol{x}_i, \boldsymbol{y}_j), \quad (4)$$

where $\Pi(\mathbf{u}, \mathbf{v}) = \{\mathbf{T} \in \mathbb{R}_+^{n \times m} | \mathbf{T}\mathbf{1}_m = \mathbf{u}, \mathbf{T}^\top \mathbf{1}_n = \mathbf{v}\}$, $\mathbf{1}_n$ denotes an n-dimensional all-one vector, and $c(\boldsymbol{x}_i, \boldsymbol{y}_j)$ is the cost function evaluating the distance between $\boldsymbol{x}_i$ and
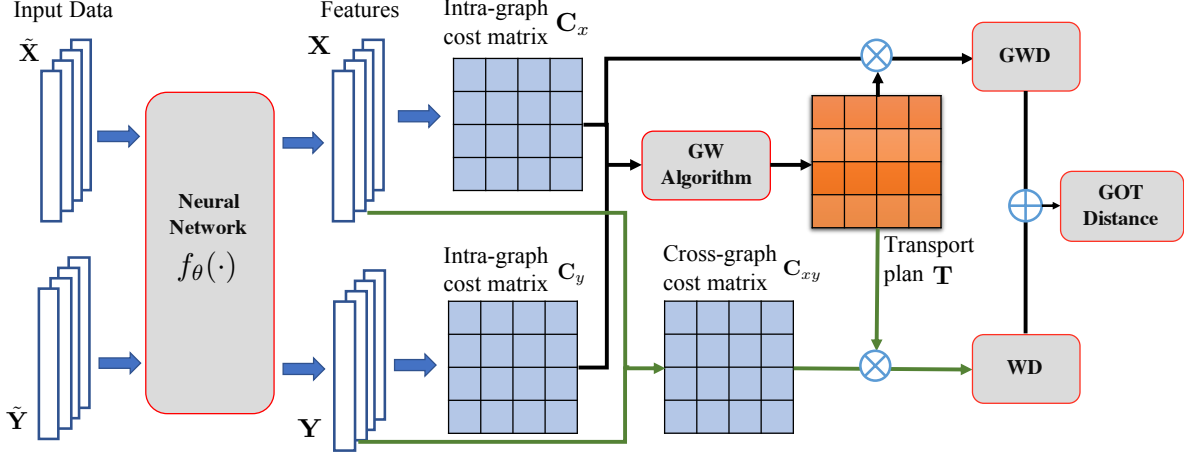
Figure 2. Schematic computation graph of the Graph Optimal Transport (GOT) distance used for cross-domain alignment. WD is short for Wasserstein Distance, and GWD is short for Gromov-Wasserstein Distance. See Sec. 2.1 and 2.4 for details.

$\boldsymbol{y}_j$. For example, the cosine distance $c(\boldsymbol{x}_i, \boldsymbol{y}_j) = 1 - \frac{\boldsymbol{x}_i^\top \boldsymbol{y}_j}{||\boldsymbol{x}_i||_2 ||\boldsymbol{y}_j||_2}$ is a popular choice. The matrix $\mathbf{T}$ is denoted as the transport plan, where $\mathbf{T}_{ij}$ represents the amount of mass shifted from u_i to v_j.

$\mathcal{D}_w(\boldsymbol{\mu}, \boldsymbol{\nu})$ defines an optimal transport distance that measures the discrepancy between each pair of samples across the two domains. In our graph matching, this is a natural choice for node (entity) matching.

Gromov-Wasserstein Distance   Instead of directly calculating distances between two sets of nodes as in WD, Gromov-Wasserstein distance (GWD) (Peyré et al., 2016; Chowdhury & Mémoli, 2019) can be used to calculate distances between pairs of nodes within each domain, as well as measuring how these distances compare to those in the counterpart domain. GWD in the discrete matching setting can be formulated as follows.

Definition 2.2. Following the same notation as in Definition 2.1, Gromov-Wasserstein distance between $\boldsymbol{\mu}, \boldsymbol{\nu}$ is defined as:

$$\mathcal{D}_{gw}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\boldsymbol{\gamma} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}) \sim \boldsymbol{\gamma}, (\boldsymbol{x}',\boldsymbol{y}') \sim \boldsymbol{\gamma}} \left[L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}') \right]$$
$$= \min_{\hat{\mathbf{T}} \in \Pi(\mathbf{u}, \mathbf{v})} \sum_{i,i',j,j'} \hat{\mathbf{T}}_{ij} \hat{\mathbf{T}}_{i'j'} L(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}'_i, \boldsymbol{y}'_j), \quad (5)$$

where $L(\cdot)$ is the cost function evaluating the intra-graph structural similarity between two pairs of nodes $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$ and $(\boldsymbol{y}_j, \boldsymbol{y}'_j)$, *i.e.*, $L(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}'_i, \boldsymbol{y}'_j) = ||c_1(\boldsymbol{x}_i, \boldsymbol{x}'_i) - c_2(\boldsymbol{y}_i, \boldsymbol{y}'_i)||$, where $c_i, i \in [1, 2]$ are functions that evaluate node similarity within the same graph (*e.g.*, the cosine similarity).

Similar to WD, in the GWD setting, $c_1(\boldsymbol{x}_i, \boldsymbol{x}'_i)$ and $c_2(\boldsymbol{y}_i, \boldsymbol{y}'_i)$ (corresponding to the edges) can be viewed as

two nodes in the dual graphs (Van Lint et al., 2001), where edges are projected into nodes. The learned matrix $\hat{\mathbf{T}}$ now becomes a transport plan that helps aligning the edges in different graphs. Note that, the same c_1 and c_2 are also used for graph construction in Sec. 2.2.

2.4. Graph Matching via OT Distances

Though GWD is capable of capturing edge similarity between graphs, it cannot be directly applied to graph alignment, since only the similarity between $c_1(\boldsymbol{x}_i, \boldsymbol{x}'_i)$ and $c_2(\boldsymbol{y}_i, \boldsymbol{y}'_i)$ is considered, without taking into account node representations. For example, the word pair ("*boy*", "*girl*") has similar cosine similarity as the pair ("*football*", "*basketball*"), but the semantic meanings of the two pairs are completely different, and should not be matched.

On the other hand, WD can match nodes in different graphs, but fails to capture the similarity between edges. If there are duplicated entities represented by different nodes in the same graph, WD will treat them as identical and ignore their neighboring relations. For example, given a sentence "*there is a red book on the blue desk*" paired with an image containing several desks and books in different colors, it is difficult to correctly identity which book in the image the sentence is referring to, without understanding the relations among the objects in the image.

To best couple WD and GWD and unify these two distances in a mutually-beneficial way, we propose a transport plan $\mathbf{T}$ shared by both WD and GWD. Compared with naively employing two different transport plans, we observe that this joint plan works better (see Table 8), and faster, since we only need to solve $\mathbf{T}$ once (instead of twice). Intuitively, with a shared transport plan, WD and GWD can enhance each other effectively, as $\mathbf{T}$ utilizes both node and edge

information simultaneously. Formally, the proposed GOT distance is defined as:

$$\mathcal{D}_{got}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T} \in \Pi(\mathbf{u},\mathbf{v})} \sum_{i,i',j,j'} \mathbf{T}_{ij} \Big(\lambda c(\boldsymbol{x}_i, \boldsymbol{y}_j)$$
$$+ (1-\lambda)\mathbf{T}_{i'j'}\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}_i', \boldsymbol{y}_j') \Big). \quad (6)$$

We apply the Sinkhorn algorithm (Cuturi, 2013; Cuturi & Peyré, 2017) to solve WD (4) with an entropic regularizer (Benamou et al., 2015):

$$\min_{\mathbf{T} \in \Pi(\mathbf{u},\mathbf{v})} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{T}_{ij} c(\boldsymbol{x}_i, \boldsymbol{y}_j) + \beta H(\mathbf{T}), \quad (7)$$

where $H(\mathbf{T}) = \sum_{i,j} \mathbf{T}_{ij} \log \mathbf{T}_{ij}$, and β is the hyper-parameter controlling the importance of the entropy term. Details are provided in Algorithm 1. The solver for GWD can be readily developed based on Algorithm 1, where $\boldsymbol{p}, \boldsymbol{q}$ are defined as uniform distributions (as shown in Algorithm 2), following Alvarez-Melis & Jaakkola (2018). With the help of the Sinkhorn algorithm, GOT can be efficiently implemented in popular deep learning libraries, such as PyTorch and TensorFlow.

To obtain a unified solver for the GOT distance, we define the unified cost function as:

$$L_{\text{unified}} = \lambda c(\boldsymbol{x}, \boldsymbol{y}) + (1-\lambda)L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}'), \quad (8)$$

where λ is the hyper-parameter for controlling the importance of different cost functions. Instead of using projected gradient descent or conjugated gradient descent as in Xu et al. (2019b;a); Vayer et al. (2018), we can approximate the transport plan $\mathbf{T}$ by adding back L_{unified} in Algorithm 2, so that Line 9 in Algorithm 2 helps solve $\mathbf{T}$ for WD and GWD at the same time, effectively matching both nodes and edges simultaneously. The solver for calculating the GOT distance is illustrated in Figure 2, and the detailed algorithm is summarized in Algorithm 3. The calculated GOT distance is used as the cross-domain alignment loss $\mathcal{L}_{\text{CDA}}(\mathbf{X}, \mathbf{Y})$ in (3), as a regularizer to update parameters $\boldsymbol{\theta}$.

3. Related Work

Optimal Transport Wasserstein distance (WD), a.k.a. Earth Mover's distance, has been widely applied to machine learning tasks. In computer vision, Rubner et al. (1998) uses WD to discover the structure of color distribution for image search. In natural language processing, WD has been applied to document retrieval (Kusner et al., 2015) and sequence-to-sequence learning (Chen et al., 2019a). There are also studies adopting WD in Generative Adversarial Network (GAN) (Goodfellow et al., 2014; Salimans et al., 2018; Chen et al., 2018; Mroueh et al., 2018; Zhang et al., 2020) to alleviate the mode-collapse issue. Recently, it has also

---

Algorithm 3 Computing GOT Distance.

1: **Input:** $\{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{y}_j\}_{j=1}^m$, hyper-parameter λ
2: Compute intra-domain similarities:
3:    $[\mathbf{C}_x]_{ij} = \cos(\boldsymbol{x}_i, \boldsymbol{x}_j), [\mathbf{C}_y]_{ij} = \cos(\boldsymbol{y}_i, \boldsymbol{y}_j),$
4: $\boldsymbol{x}_i' = g_1(\boldsymbol{x}_i), \boldsymbol{y}_j' = g_2(\boldsymbol{y}_j) // g_1, g_2$ denote two MLPs
5: Compute cross-domain similarities:
6:    $\mathbf{C}_{ij} = \cos(\boldsymbol{x}_i', \boldsymbol{y}_j')$
7: **if** $\mathbf{T}$ is shared: **then**
8:    Update $\mathcal{L}$ in Algorithm 2 (Line 8) with:
9:        $\mathcal{L}_{\text{unified}} = \lambda \mathbf{C} + (1-\lambda)\mathcal{L}$
10:   Plug in $\mathcal{L}_{\text{unified}}$ back to Algorithm 2 and solve new $\mathbf{T}$
11:   Compute $\mathcal{D}_{got}$
12: **else**
13:   Apply Algorithm 1 to obtain $\mathcal{D}_w$
14:   Apply Algorithm 2 to obtain $\mathcal{D}_{gw}$
15:   $\mathcal{D}_{got} = \lambda \mathcal{D}_w + (1-\lambda)\mathcal{D}_{gw}$
16: **end if**
17: Return $\mathcal{D}_{got}$

---

been used for vision-and-language pre-training to encourage word-region alignment (Chen et al., 2019b). Besides WD, Gromov-Wasserstein distance (Peyré et al., 2016) has been proposed for distributional metric matching and applied to unsupervised machine translation (Alvarez-Melis & Jaakkola, 2018).

There are different ways to solve the OT distance, such as linear programming. However, this solver is not differentiable, thus it cannot be applied in deep learning frameworks. Recently, WGAN (Arjovsky et al., 2017) proposes to approximate the dual form of WD by imposing a 1-Lipschitz constraint on the discriminator. Note that the duality used for WGAN is restricted to the W-1 distance, $i.e., \| \cdot \|$. The Sinkhorn algorithm was first proposed in Cuturi (2013) as a solver for calculating an entropic regularized OT distance. Thanks to the Envelop Theorem (Cuturi & Peyré, 2017), the Sinkhorn algorithm can be efficiently calculated and readily applied to neural networks. More recently, Vayer et al. (2018) proposed the fused GWD for graph matching. Our proposed GOT framework enjoys the benefits of both Sinkhorn algorithm and fused GWD: it is (i) capable of capturing more structured information via marrying both WD and GWD; and (ii) scalable to large datasets and trainable with deep neural networks.

Graph Neural Network Neural networks operating on graph data was first introduced in Gori et al. (2005) using recurrent neural networks. Later, Duvenaud et al. (2015) proposed a convolutional neural network over graphs for classification tasks. However, these methods suffer from scalability issues, because they need to learn node-degree-specific weight matrices for large graphs. To alleviate this issue, Kipf & Welling (2016) proposed to use a single weight matrix per layer in the neural network, which is capable of handling varying node degrees through an appropriate normalization of the adjacency matrix of the data. To fur-

Method	Sentence Retrieval			Image Retrieval			
	R@1	R@5	R@10	R@1	R@5	R@10	Rsum
VSE++ (ResNet) (Faghri et al., 2018)	52.9	–	87.2	39.6	–	79.5	–
DPC (ResNet) (Zheng et al., 2020)	55.6	81.9	89.5	39.1	69.2	80.9	416.2
DAN (ResNet) (Nam et al., 2017)	55.0	81.8	89.0	39.4	69.2	79.1	413.5
SCO (ResNet) (Huang et al., 2018)	55.5	82.0	89.3	41.1	70.5	80.1	418.5
SCAN (Faster R-CNN, ResNet) (Lee et al., 2018)	67.7	88.9	94.0	44.0	74.2	82.6	452.2
Ours (Faster R-CNN, ResNet):							
SCAN + WD	70.9	92.3	95.2	49.7	78.2	86.0	472.3
SCAN + GWD	69.5	91.2	95.2	48.8	78.1	85.8	468.6
SCAN + GOT	**70.9**	**92.8**	**95.5**	**50.7**	**78.7**	**86.2**	**474.8**
VSE++ (ResNet) (Faghri et al., 2018)	41.3	–	81.2	30.3	–	72.4	–
DPC (ResNet) (Zheng et al., 2020)	41.2	70.5	81.1	25.3	53.4	66.4	337.9
GXN (ResNet) (Gu et al., 2018)	42.0	–	84.7	31.7	–	74.6	–
SCO (ResNet) (Huang et al., 2018)	42.8	72.3	83.0	33.1	62.9	75.5	369.6
SCAN (Faster R-CNN, ResNet)(Lee et al., 2018)	46.4	77.4	87.2	34.4	63.7	75.7	384.8
Ours (Faster R-CNN, ResNet):							
SCAN + WD	50.2	80.1	89.5	37.9	66.8	78.1	402.6
SCAN + GWD	47.2	78.3	87.5	34.9	64.4	76.3	388.6
SCAN + GOT	**50.5**	**80.2**	**89.8**	**38.1**	**66.8**	**78.5**	**403.9**

Table 1. Results on image-text retrieval evaluated on Recall@K (R@K). Upper panel: Flickr30K; lower panel: COCO.

ther improve the classification accuracy, the graph attention network (GAT) (Veličković et al., 2018) was proposed by using a learned weight matrix instead of the adjacency matrix, with masked attention to aggregate node neighborhood information.

Recently, the graph neural network has been extended to other tasks beyond classification. Li et al. (2019b) proposed graph matching network (GMN) for learning similarities between graphs. Similar to GAT, masked attention is applied to aggregate information from each node within a graph, and cross-graph information is further exploited via soft attention. Task-specific losses are then used to guide model training. In this setting, an adjacency matrix can be directly obtained from the data and soft attention is used to induce alignment. In contrast, our GOT framework does not rely on explicit graph structures in the data, and uses OT for graph alignment.

4. Experiments

To validate the effectiveness of the proposed GOT framework, we evaluate performance on a selection of diverse tasks. We first consider vision-and-language understanding, including: (i) image-text retrieval, and (ii) visual question answering. We further consider text generation tasks, including: (iii) image captioning, (iv) machine translation, and (v) abstractive text summarization.

4.1. Vision-and-Language Tasks

Image-Text Retrieval For image-text retrieval task, we use pre-trained Faster R-CNN (Ren et al., 2015) to extract bottom-up-attention features (Anderson et al., 2018) as the

image representation. A set of 36 features is created for each image, each feature represented by a 2048-dimensional vector. For captions, a bi-directional GRU (Schuster & Paliwal, 1997; Bahdanau et al., 2015) is used to obtain textual features.

We evaluate our model on the Flickr30K (Plummer et al., 2015) and COCO (Lin et al., 2014) datasets. Flickr30K contains 31,000 images, with five human-annotated captions per image. We follow previous work (Karpathy & Fei-Fei, 2015; Faghri et al., 2018) for the data split: 29,000, 1,000 and 1,000 images are used for training, validation and test, respectively. COCO contains 123,287 images, each image also accompanied with five captions. We follow the data split in Faghri et al. (2018), where 113,287, 5,000 and 5,000 images are used for training, validation and test, respectively.

We measure the performance of image retrieval and sentence retrieval on Recall at K (R@K) (Karpathy & Fei-Fei, 2015), defined as the percentage of queries retrieving the correct images/sentences within the top K highest-ranked results. In our experiment, $K = \{1, 5, 10\}$, and Rsum (Huang et al., 2017) (summation over all R@K) is used to evaluate the overall performance. Results are summarized in Table 1. Both WD and GWD can boost the performance of the SCAN model, while WD achieves a larger margin than GWD. This indicates that when used alone, GWD may not be a good metric for graph alignment. When combining the two distances together, GOT achieves the best performance.

Figure 3 provides visualization on the learned transport plan in GOT and the learned attention matrix in SCAN. Both serve as a proxy to lend insights into the learned alignment. As shown, the attention matrix from SCAN is much denser
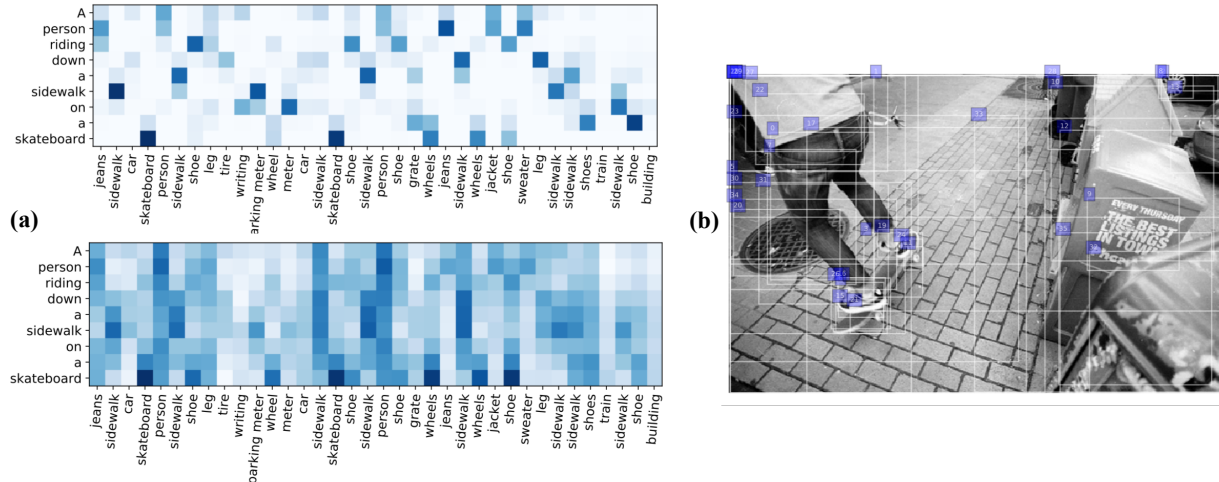
Figure 3. (a) A comparison of the inferred transport plan from GOT (top chart) and the learned attention matrix from SCAN (bottom chart). Both serve as a lens to visualize cross-domain alignment. The horizontal axis represents image regions, and the vertical axis represents word tokens. (b) The original image.

Model	BAN	BAN+GWD	BAN+WD	BAN+GOT
Score	66.00	66.21	66.26	**66.44**

Table 2. Results (accuracy) on VQA 2.0 validation set, using BAN (Kim et al., 2018) as baseline.

Model	BUTD	BAN-1	BAN-2	BAN-4	BAN-8
w/o GOT	63.37	65.37	65.61	65.81	66.00
w/ GOT	**65.01**	**65.68**	**65.88**	**66.10**	**66.44**

Table 3. Results (accuracy) of applying GOT to BUTD (Anderson et al., 2018) and BAN-m (Kim et al., 2018) on VQA 2.0. m denotes the number of glimpses.

and noisier than the transport plan inferred by GOT. This shows our model can better discover cross-domain relations between image-text pairs, since the inferred transport plan is more interpretable and has less ambiguity. For example, both the words "*sidewalk*" and "*skateboard*" match the corresponding image regions very well.

Because of the Envelope Theorem (Cuturi & Peyré, 2017), GOT needs to be calculated only during the forward phase of model training. Therefore, it does not introduce much extra computation time. For example, when using the same machine for image-text retrieval experiments, SCAN required 6hr 34min for training and SCAN+GOT 6hr 57min.

Visual Question Answering We also consider the VQA 2.0 dataset (Goyal et al., 2017), which contains human-annotated QA pairs on COCO images (Lin et al., 2014). For each image, an average of 3 questions are collected, with 10 candidate answers per question. The most frequent answer from the annotators is selected as the correct answer. Following previous work (Kim et al., 2018), we take the answers that appear more than 9 times in the training set as

candidate answers, which results in 3129 candidates. Classification accuracy is used as the evaluation metric, defined as $\min(1, \frac{\text{\# humans provided ans.}}{3})$.

The BAN model (Kim et al., 2018) is used as baseline, with the original codebase used for fair comparison. Results are summarized in Table 2. Both WD and GWD improve the BAN model on the validation set, and GOT achieves further performance lift.

We also investigate whether different architecture designs affect the performance gain. We consider BUTD (Anderson et al., 2018) as an additional baseline, and apply different number of glimpses m to the BAN model, denoted as BAN-m. Results are summarized in Table 3, with the following observations: (i) When the number of parameters in the tested model is small, such as BUTD, the improvement brought by GOT is more significant. (ii) BAN-4, a simpler model than BAN-8, when combined with GOT, can outperform BAN-8 without using GOT (66.10 v.s. 66.00). (iii) For complex models such as BAN-8 that might have limited space for improvement, GOT is still able to achieve performance gain.

4.2. Text Generation Tasks

Image Captioning We conduct experiments on image captioning using the same COCO dataset. The same bottom-up-attention features (Anderson et al., 2018) used in image-text retrieval are adopted here. The text decoder is one-layer LSTM with 256 hidden units. The word embedding dimension is set to 256. Results are summarized in Table 4. A similar performance gain is introduced by GOT. The relative performance boost from WD to GOT over CIDEr score is: $\frac{\text{GOT}-\text{WD}}{\text{WD}-\text{MLE}} = \frac{109.2-107.9}{107.9-106.3} = 81.25\%$. This attributes

Method	CIDEr	BLEU-4	BLUE-3	BLEU-2	BLEU-1	ROUGE	METEOR
Soft Attention (Xu et al., 2015)	-	24.3	34.4	49.2	70.7	-	23.9
Hard Attention (Xu et al., 2015)	-	25.0	35.7	50.4	71.8	-	23.0
Show & Tell (Vinyals et al., 2015)	85.5	27.7	-	-	-	-	23.7
ATT-FCN (You et al., 2016)	-	30.4	40.2	53.7	70.9	-	24.3
SCN-LSTM (Gan et al., 2017)	101.2	33.0	43.3	56.6	72.8	-	25.7
Adaptive Attention (Lu et al., 2017)	108.5	33.2	43.9	58.0	74.2	-	26.6
MLE	106.3	34.3	45.3	59.3	75.6	55.2	26.2
MLE + WD	107.9	34.8	46.1	60.1	76.2	55.6	26.5
MLE + GWD	106.6	33.3	45.2	59.1	75.7	55.0	25.9
MLE + GOT	**109.2**	**35.1**	**46.5**	**60.3**	**77.0**	**56.2**	**26.7**

Table 4. Results of image captioning on the COCO dataset.

Model	EN-VI uncased	EN-VI cased	EN-DE uncased	EN-DE cased
Transformer (Vaswani et al., 2017)	29.25 ± 0.18	28.46 ± 0.17	25.60 ± 0.07	25.12 ± 0.12
Transformer + WD	29.49 ± 0.10	28.68 ± 0.14	25.83 ± 0.12	25.30 ± 0.11
Transformer + GWD	28.65 ± 0.14	28.34 ± 0.16	25.42 ± 0.17	24.82 ± 0.15
Transformer + GOT	$\mathbf{29.92 \pm 0.11}$	$\mathbf{29.09 \pm 0.18}$	$\mathbf{26.05 \pm 0.17}$	$\mathbf{25.54 \pm 0.15}$

Table 5. Results of neural machine translation on EN-DE and EN-VI.

Method	ROUGE-1	ROUGE-2	ROUGE-L
ABS+ (Rush et al., 2015)	31.00	12.65	28.34
LSTM (Hu et al., 2018)	36.11	16.39	32.32
LSTM + GWD	36.31	17.32	33.15
LSTM + WD	36.81	17.34	33.34
LSTM + GOT	**37.10**	**17.61**	**33.70**

Table 6. Results of abstractive text summarization on the English Gigawords dataset.

Figure 4. Inferred transport plan for aligning source and output sentences in abstractive summarization.

to the additional GWD introduced in GOT that can help model implicit intra-domain relationships in images and captions, leading to more accurate caption generation.

Machine Translation In machine translation (and abstractive summarization), the word embedding spaces of the source and target sentences are different, which can be considered as different domains. Therefore, GOT can be used to align those words with similar semantic meanings between the source and target sentences for better translation/summarization. We choose two machine translation benchmarks for experiments: (i) English-Vietnamese TED-talks corpus, which contains 133K pairs of sentences from the IWSLT Evaluation Campaign (Cettolo et al., 2015); and (ii) a large-scale English-German parallel corpus with 4.5M pairs of sentences, from the WMT Evaluation Campaign (Vaswani et al., 2017). The Texar codebase (Hu et al., 2018) is used in our experiments.

We apply GOT to the Transformer model (Vaswani et al., 2017) and use BLEU score (Papineni et al., 2002) as the evaluation metric. Results are summarized in Table 5. As also observed in Chen et al. (2019a), using WD can improve the performance of the Transformer for sequence-to-sequence learning. However, if only GWD is used, the test BLEU score drops. Since GWD can only match the edges, it ignores supervision signals from node representations. This
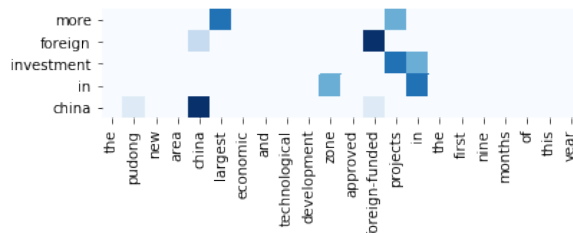
serves as empirical evidence to support our hypothesis that using GWD alone may not be enough to improve performance. However, GWD serves as a complementary method for capturing graph information that might be missed by WD. Therefore, when combining the two together, GOT achieves the best performance. Example translations are provided in Table 7.

Abstractive Summarization We evaluate abstractive summarization on the English Gigawords benchmark (Graff et al., 2003). A basic LSTM model as implemented in Texar (Hu et al., 2018) is used in our experiments. ROUGE-1, -2 and -L scores (Lin, 2004) are reported. Table 6 shows that both GWD and WD can improve the performance of the LSTM. The transport plan for source and output sentences alignment is illustrated in Figure 4. The learned alignment is sparse and interpretable. For instance, the words "*largest*" and "*projects*" in the source sentence matches the words "*more*" and "*investment*" in the output summary very well.

4.3. Ablation study

We conduct additional ablation study on the EN-VI and EN-DE datasets for machine translation.

Reference:	Indias new prime minister, Narendra Modi, is meeting his Japanese counterpart, Shinzo Abe, in Tokyo to discuss economic and security ties, on his first major foreign visit since winning Mays election.
MLE:	India s new prime minister , Narendra Modi , meets his Japanese counterpart , Shinzo Abe , in Tokyo , during his first major foreign visit in May to discuss economic and security relations .
GOT:	India s new prime minister , Narendra Modi , is meeting his Japanese counterpart Shinzo Abe in Tokyo in his first major foreign visit since his election victory in May to discuss economic and security relations.

Reference:	Chinese leaders presented the Sunday ruling as a democratic breakthrough because it gives Hong Kongers a direct vote, but the decision also makes clear that Chinese leaders would retain a firm hold on the process through a nominating committee tightly controlled by Beijing.
MLE:	The Chinese leadership presented the decision of Sunday as a democratic breakthrough , because it gives Hong Kong citizens a direct right to vote , but the decision also makes it clear that the Chinese leadership maintains the **expiration of** a nomination committee closely controlled by Beijing .
GOT:	The Chinese leadership presented the decision on Sunday as a democratic breakthrough , because Hong Kong citizens have a direct electoral right , but the decision also makes it clear that the Chinese leadership remains firmly in hand with a nominating committee controlled by Beijing.

Table 7. Comparison of German-to-English translation examples. For each example, we show the human translation (reference) and the translation from MLE and GOT. We highlight the key-phrase differences between reference and translation outputs in blue and red, and denote the error in translation in bold. In the first example, GOT correctly maintains all the information in *"since winning May's election"* by translating to *"since his election victory in May"*, whereas MLE only generate *"in May"*. In the second example, GOT successfully keeps the information *"Beijing"*, whereas MLE generates wrong words *"expiration of"*.

Model	EN-VI uncased	EN-DE uncased
GOT (shared)	29.92 ± 0.11	26.05 ± 0.18
GOT (unshared)	29.77 ± 0.12	25.89 ± 0.17

Table 8. Ablation study on transport plan in machine translation. Both models were run 5 times with the same hyper-parameter setting.

λ	0	0.1	0.3	0.5	0.8	1.0
BLEU	28.65	29.31	29.52	29.65	**29.92**	29.49

Table 9. Ablation study of the hyper-parameter λ on the EN-VI machine translation dataset.

Shared Transport Plan T   As discussed in Sec. 2.4, we use a shared transport plan $\mathbf{T}$ to solve the GOT distance. An alternative is not to share this $\mathbf{T}$ matrix. The comparison results are provided in Table 8. GOT with a shared transport plan achieves better performance than the alternative. Since we only need to run the iterative Sinkhorn algorithm once, it also saves training time than the unshared case.

Hyper-parameter λ   We perform ablation study on the hyper-parameter λ in (6). We select λ from $[0, 1]$ and report results in Table 9. When $\lambda = 0.8$, EN-VI translation performs the best, which indicates that the weight on WD needs to be larger than the weight on GWD, since intuitively node matching is more important than edge matching for machine translation. However, both WD and GWD contribute to GOT achieving the best performance.

5. Conclusions

We propose Graph Optimal Transport, a principled framework for cross-domain alignment. With the Wasserstein and Gromov-Wasserstein distances, both intra-domain and cross-domain relations are captured for better alignment. Empirically, we observe that enforcing alignment can serve as an effective regularizer for model training. Extensive experiments show that the proposed method is a generic framework that can be applied to a wide range of cross-domain tasks. For future work, we plan to apply the proposed framework to self-supervised representation learning.

Acknowledgements

References

Alvarez-Melis, D. and Jaakkola, T. S. Gromov-wasserstein alignment of word embedding spaces. *arXiv:1809.00013*, 2018.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

Antol, S. et al. Vqa: Visual question answering. In *ICCV*,

2015.

Arjovsky, M. et al. Wasserstein generative adversarial networks. In *ICML*, 2017.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 2015.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., and Federico, M. The IWSLT 2015 evaluation campaign. In *International Workshop on Spoken Language Translation*, 2015.

Chechik, G., Sharma, V., Shalit, U., and Bengio, S. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 2010.

Chen, L., Dai, S., Tao, C., Zhang, H., Gan, Z., Shen, D., Zhang, Y., Wang, G., Zhang, R., and Carin, L. Adversarial text generation via feature-mover's distance. In *NeurIPS*, 2018.

Chen, L., Zhang, Y., Zhang, R., Tao, C., Gan, Z., Zhang, H., Li, B., Shen, D., Chen, C., and Carin, L. Improving sequence-to-sequence learning via optimal transport. *arXiv preprint arXiv:1901.06283*, 2019a.

Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019b.

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

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

Cuturi, M. and Peyré, G. Computational optimal transport. 2017.

De Goes, F. et al. An optimal transport approach to robust reconstruction and simplification of 2d shapes. In *Computer Graphics Forum*, 2011.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, 2015.

Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. Vse++: Improved visual-semantic embeddings. In *BMVC*, 2018.

Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., Carin, L., and Deng, L. Semantic compositional networks for visual captioning. In *CVPR*, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NeurIPS*, 2014.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *IEEE International Joint Conference on Neural Networks*, 2005.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

Graff, D., Kong, J., Chen, K., and Maeda, K. English gigaword. *Linguistic Data Consortium, Philadelphia*, 2003.

Gu, J., Cai, J., Joty, S. R., Niu, L., and Wang, G. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*, 2018.

Hu, Z., Shi, H., Yang, Z., Tan, B., Zhao, T., He, J., Wang, W., Yu, X., Qin, L., Wang, D., et al. Texar: A modularized, versatile, and extensible toolkit for text generation. *arXiv preprint arXiv:1809.00794*, 2018.

Huang, Y., Wang, W., and Wang, L. Instance-aware image and sentence matching with selective multimodal lstm. In *CVPR*, 2017.

Huang, Y., Wu, Q., Song, C., and Wang, L. Learning semantic concepts and order for image and sentence matching. In *CVPR*, 2018.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

Kim, J.-H., Jun, J., and Zhang, B.-T. Bilinear attention networks. In *NeurIPS*, 2018.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. From word embeddings to document distances. In *ICML*, 2015.

Lee, K.-H. et al. Stacked cross attention for image-text matching. In *ECCV*, 2018.

Li, L., Gan, Z., Cheng, Y., and Liu, J. Relation-aware graph attention network for visual question answering. In *ICCV*, 2019a.

Li, Y., Gu, C., Dullien, T., Vinyals, O., and Kohli, P. Graph matching networks for learning the similarity of graph structured objects. In *ICML*, 2019b.

Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Lu, J., Xiong, C., Parikh, D., and Socher, R. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

Luise, G., Rudi, A., Pontil, M., and Ciliberto, C. Differential properties of sinkhorn approximation for learning with wasserstein distance. *arXiv:1805.11897*, 2018.

Malinowski, M. and Fritz, M. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NeurIPS*, 2014.

Mroueh, Y., Li, C.-L., Sercu, T., Raj, A., and Cheng, Y. Sobolev GAN. In *ICLR*, 2018.

Nam, H., Ha, J.-W., and Kim, J. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

Peyré, G., Cuturi, M., and Solomon, J. Gromov-wasserstein averaging of kernel and distance matrices. In *ICML*, 2016.

Peyré, G., Cuturi, M., et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 2019.

Plummer, B. A. et al. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Rubner, Y., Tomasi, C., and Guibas, L. J. A metric for distributions with applications to image databases. In *ICCV*, 1998.

Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. Improving GANs using optimal transport. In *ICLR*, 2018.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *Transactions on Signal Processing*, 1997.

Van Lint, J. H., Wilson, R. M., and Wilson, R. M. *A course in combinatorics*. Cambridge university press, 2001.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. Optimal transport for structured data with application on graphs. *arXiv:1805.09114*, 2018.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *ICLR*, 2018.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. In *CVPR*, 2015.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for Wasserstein distance. In *arXiv:1802.04307*, 2018.

Xu, H., Luo, D., and Carin, L. Scalable gromov-wasserstein learning for graph partitioning and matching. In *NeurIPS*, 2019a.

Xu, H., Luo, D., Zha, H., and Carin, L. Gromov-wasserstein learning for graph matching and node embedding. In *ICML*, 2019b.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

Yang, Z., He, X., Gao, J., Deng, L., and Smola, A. Stacked attention networks for image question answering. In *CVPR*, 2016a.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. Hierarchical attention networks for document classification. In *NAACL*, 2016b.

Yao, T., Pan, Y., Li, Y., and Mei, T. Exploring visual relationship for image captioning. In *ECCV*, 2018.

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. Image captioning with semantic attention. In *CVPR*, 2016.

Yu, Z., Yu, J., Cui, Y., Tao, D., and Tian, Q. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019.

Zhang, R., Chen, C., Gan, Z., Wen, Z., Wang, W., and Carin, L. Nested-wasserstein self-imitation learning for sequence generation. *arXiv:2001.06944*, 2020.

Zheng, Z., Zheng, L., Garrett, M., Yang, Y., Xu, M., and Shen, Y.-D. Dual-path convolutional image-text embeddings with instance loss. *TOMM*, 2020.