GRAPH OPTIMAL TRANSPORT FOR CROSS-DOMAIN ALIGNMENT

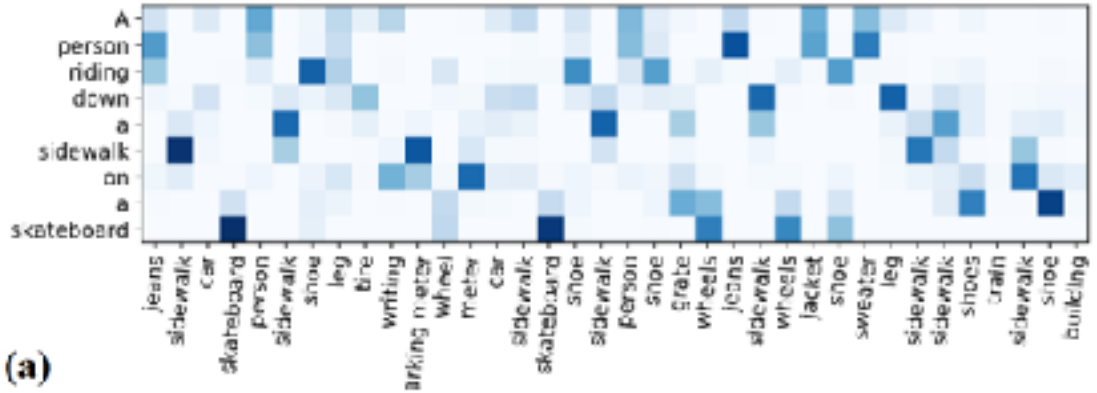
Liqun Chen 1 Zhe Gan 2 Yu Cheng 2 Linjie Li 2 Lawrence Carin 1 Jingjing Liu 2

Xuehan Chen
06/28/2021

OUTLINE

- Cross-domain alignment
- Graph construction:
  - Image Object detection: Faster R-CNN
  - Word: Bi-GRU
- Graph matching(Optimal Transport Distances)
  - Wasserstein Distance
  - Gromov-Wasserstein Distance
  - Graph Matching via OT Distances
- Experimental Results
  - Image-text retrieval, VQA

GRAPH CONSTRUCTION



(a)

- Image features
  - Bottom-up features:$X \in R^{n \times s}$
  - n : number of bounding boxes, s: feature dim
- Text features
  - Bi-LSTM/GRU for feature extraction: $Y \in R^{m \times s}$
  - m : number of word tokens, s: feature dim

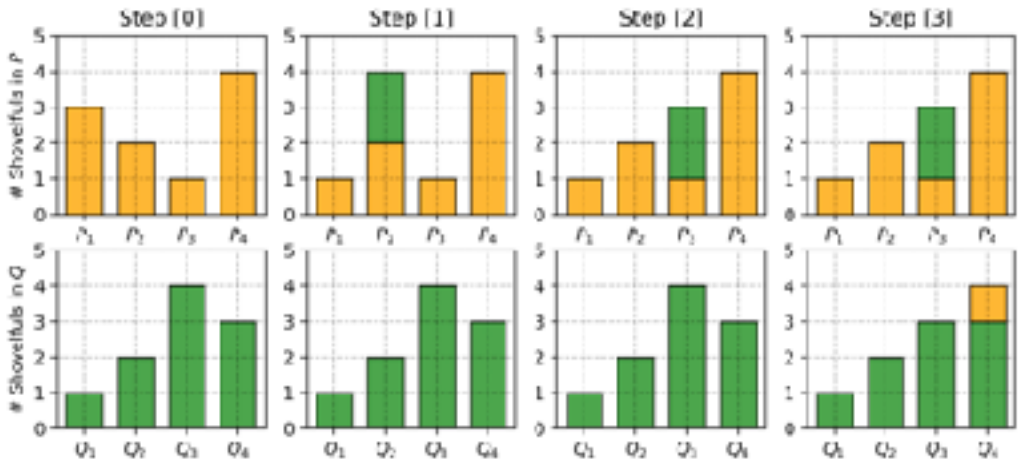
DENOTATION

- Entity feature vector from 2 different domain: $\tilde{\mathbf{X}} = \left\{\tilde{\boldsymbol{x}}_i\right\}_{i=1}^{n}$, $\tilde{\mathbf{Y}} = \left\{\tilde{\boldsymbol{y}}_j\right\}_{j=1}^{m}$
- $\mathbf{X}, \mathbf{Y} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$
- $\mathscr{L}(\boldsymbol{\theta}) = \mathscr{L}_{\mathbf{sup}}(\mathbf{X}, \mathbf{Y}, l)$
- $\mathscr{L}(\boldsymbol{\theta}) = \mathscr{L}_{\mathbf{sup}}(\mathbf{X}, \mathbf{Y}, l) + \alpha \cdot \mathscr{L}_{\mathbf{CDA}}(\mathbf{X}, \mathbf{Y})$

- Probability distribution: $\boldsymbol{\mu} \in \mathbf{P}(\mathbb{X}), \boldsymbol{\nu} \in \mathbf{P}(\mathbb{Y})$, $\boldsymbol{\mu} = \sum_{i=1}^{n} \mathrm{u}_i \delta_{\boldsymbol{x}_i}, \boldsymbol{\nu} = \sum_{j=1}^{m} \mathrm{v}_j \delta_{\boldsymbol{y}_j}$
- $\boldsymbol{\Pi}(\boldsymbol{\mu}, \boldsymbol{\nu})$: all the joint distributions $\gamma(\boldsymbol{x}, \boldsymbol{y})$ with marginals $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\nu}(\boldsymbol{y})$.
- Weight vector: $\mathbf{u} = \left\{\mathrm{u}_i\right\}_{i=1}^{n} \in \Delta_n$, $\mathbf{v} = \left\{\mathrm{v}_i\right\}_{i=1}^{m} \in \Delta_m$
- $\sum_{i=1}^{n} \mathrm{u}_i = \sum_{j=1}^{m} \mathrm{v}_j = 1$

WASSERSTRIN DISTANCE

- Wasserstein Distance is a measure of the distance between two probability distributions.(Earth Mover's distance)
  - eg. the amount of dirt moved x the moving distance
- In order to change P to look like Q
  - First move 2 shovelfuls from P1 to P2 => (P1,Q1) match up.
  - Then move 2 shovelfuls from P2 to P3 => (P2,Q2) match up.
  - Finally move 1 shovelfuls from Q3 to Q4 => (P3,Q3) and (P4,Q4) match up.
  - $\delta i+1=\delta i+Pi-Qi$
  - $\delta 0=0$
  - $\delta 1=0+3-1=2$
  - $\delta 2=2+2-2=2$
  - $\delta 3=2+1-4=-1$
  - $\delta 4=-1+4-3=0$
  - Earth Mover's distance is $W=\sum|\delta i|=5$



WASSERSTRIN DISTANCE

- Apply optimal transport

- Wasserstrin distance
  - Node matching

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

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

  - T is the transport plan
    - calculated by Sinkhorn algorithm
  - x_i, y_j are two instances/nodes
  - C is the pair-wise cost matrix:
  - $C_{ij} = \mathrm{d}\left(x_i, y_j\right)$
    - d: eg. Cosine similarity distance

- where

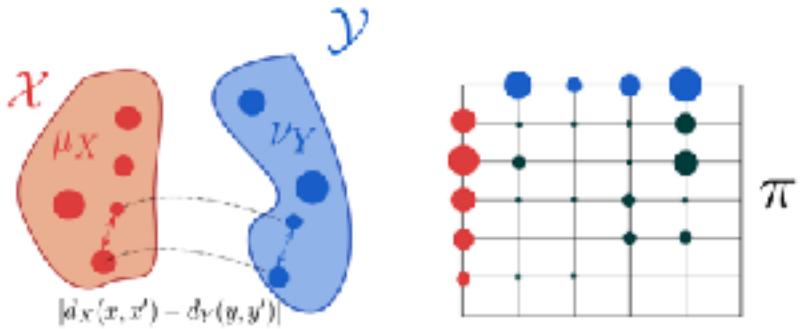
$$\Pi(\mathbf{u}, \mathbf{v}) = \left\{\mathbf{T} \in \mathbb{R}_{+}^{n \times m} \mid \mathbf{T}\mathbf{1}_m = \mathbf{u}, \mathbf{T}^{\top}\mathbf{1}_n = \mathbf{v}\right\}$$

- $\mathbf{1}_n$: n-dimensional all-one vector

$$c\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right) = 1 - \frac{\boldsymbol{x}_i^{\top} \boldsymbol{y}_j}{\| \boldsymbol{x}_i \|_2 \| \boldsymbol{y}_j \|_2}$$

GROMOV-WASSERSTEIN DISTANCE

- Gromov-Wasserstein Distance:
  - Structure matching
  - $\mathscr{D}_{gw}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\boldsymbol{\gamma} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \boldsymbol{\gamma}, (\boldsymbol{x}', \boldsymbol{y}') \sim \gamma} \left[L\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}'\right) \right]$
  - $= \min_{\hat{\mathbf{T}} \in \Pi(\mathbf{u}, \mathbf{v})} \sum_{i,i',j,j'} \hat{\mathbf{T}}_{ij} \hat{\mathbf{T}}_{i'j'} L\left(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}'_i, \boldsymbol{y}'_j\right)$
  - $L\left(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}'_i, \boldsymbol{y}'_j\right) = \left\| c_1\left(\boldsymbol{x}_i, \boldsymbol{x}'_i\right) - c_2\left(\boldsymbol{y}_i, \boldsymbol{y}'_i\right) \right\|$



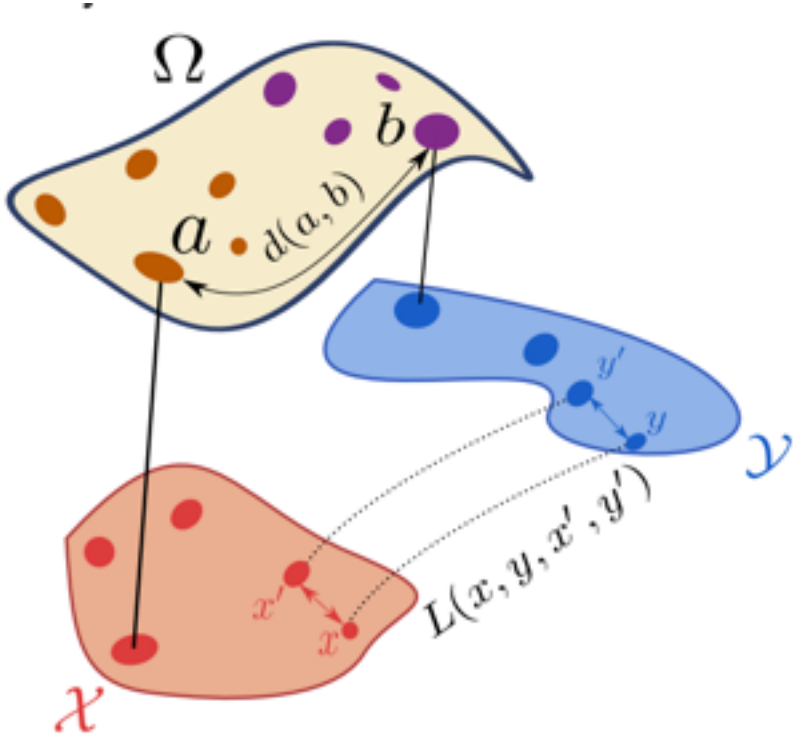
  - $L(\cdot)$: intra-graph structural similarity between two pairs of nodes
  - C: node similarity within the same graph

GRAPH CONSTRUCTION

- Graph X: $\mathcal{G}_x\left(\mathbb{V}_x, \mathcal{E}_x\right)$,
  - Node $i \in \mathbb{V}_x$, feature vector $\boldsymbol{x}_i$.
  - Edges $\mathcal{E}_x$: calculate the similarity between a pair of entities inside a graph
- Image graph
  - Dot-product/cosine distance between objects within the image
  - $\mathbf{C}_x = \left\{ \cos\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \right\}_{i,j} \in \mathbb{R}^{n \times n}$
- Text graph
  - $\mathbf{C}_y = \left\{ \cos\left(\boldsymbol{y}_i, \boldsymbol{y}_j\right) \right\}_{i,j} \in \mathbb{R}^{m \times m}$
- Graph Pruning: sparse graph representation
  - $\mathbf{C}_x = \max\left(\mathbf{C}_x - \tau, 0\right)$, If $\left[\mathbf{C}_x\right]_{ij} > 0$, an edge is added between node i and j.

GOT(GRAPH-BASED OPTIMAL TRANSPORT)

- GOT(Graph-based Optimal transport)
- Wasserstrin distance
  - Node matching
- Gromov-Wasserstein Distance:
  - Structure matching
- Combining WD, GWD for better feature alignment
- $$\mathscr{D}_{got}(\boldsymbol{\mu},\boldsymbol{\nu}) = \min_{\mathbf{T}\in\Pi(\mathbf{u},\mathbf{v})} \sum_{i,i',j,j'} \mathbf{T}_{ij} \left(\lambda c\left(\boldsymbol{x}_i,\boldsymbol{y}_j\right) \right.$$
- $$\left. +(1-\lambda)\mathbf{T}_{i'j'}\mathscr{L}\left(\boldsymbol{x}_i,\boldsymbol{y}_j,\boldsymbol{x}_i',\boldsymbol{y}_j'\right) \right)$$
- $L_{GOT} = \lambda_1 L_W + \lambda_2 L_{GW}$
- $L_{\text{unified}} = \lambda c(\boldsymbol{x},\boldsymbol{y}) + (1-\lambda)L\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{x}',\boldsymbol{y}'\right)$



- To solve WD with an entropic regularizer
- $\min_{\mathbf{T}\in\Pi(\mathbf{u},\mathbf{v})} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{T}_{ij} c\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right) - \beta H(\mathbf{T})$
- $H(\mathbf{T}) = -\sum_{i,j} \mathbf{T}_{ij} \log \mathbf{T}_{ij}$

- Gibbs kernel for Sinkhorn:
- $P_\gamma \overset{\text{def}}{=} \underset{P\in U(a,b)}{\operatorname{argmin}} \langle P, M_{XY}\rangle - \gamma E(P) \qquad \boldsymbol{u}\in\mathbb{R}_+^n, \boldsymbol{v}\in\mathbb{R}_+^m$
- $P_\gamma = \operatorname{diag}(u) K \operatorname{diag}(v), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}$
- $\mathbf{u}\odot(\mathbf{K}\mathbf{v}) = \mathbf{a} \quad \text{and} \quad \mathbf{v}\odot\left(\mathbf{K}^{\mathrm{T}}\mathbf{u}\right) = \mathbf{b}$
- $\mathbf{u}^{(\ell+1)} \overset{\text{def.}}{=} \dfrac{\mathbf{a}}{\mathbf{K}\mathbf{v}^{(\ell)}} \quad \text{and} \quad \mathbf{v}^{(\ell+1)} \overset{\text{def.}}{=} \dfrac{\mathbf{b}}{\mathbf{K}^{\mathrm{T}}\mathbf{u}^{(\ell+1)}}$

Algorithm 1 Computing Wasserstein Distance.

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

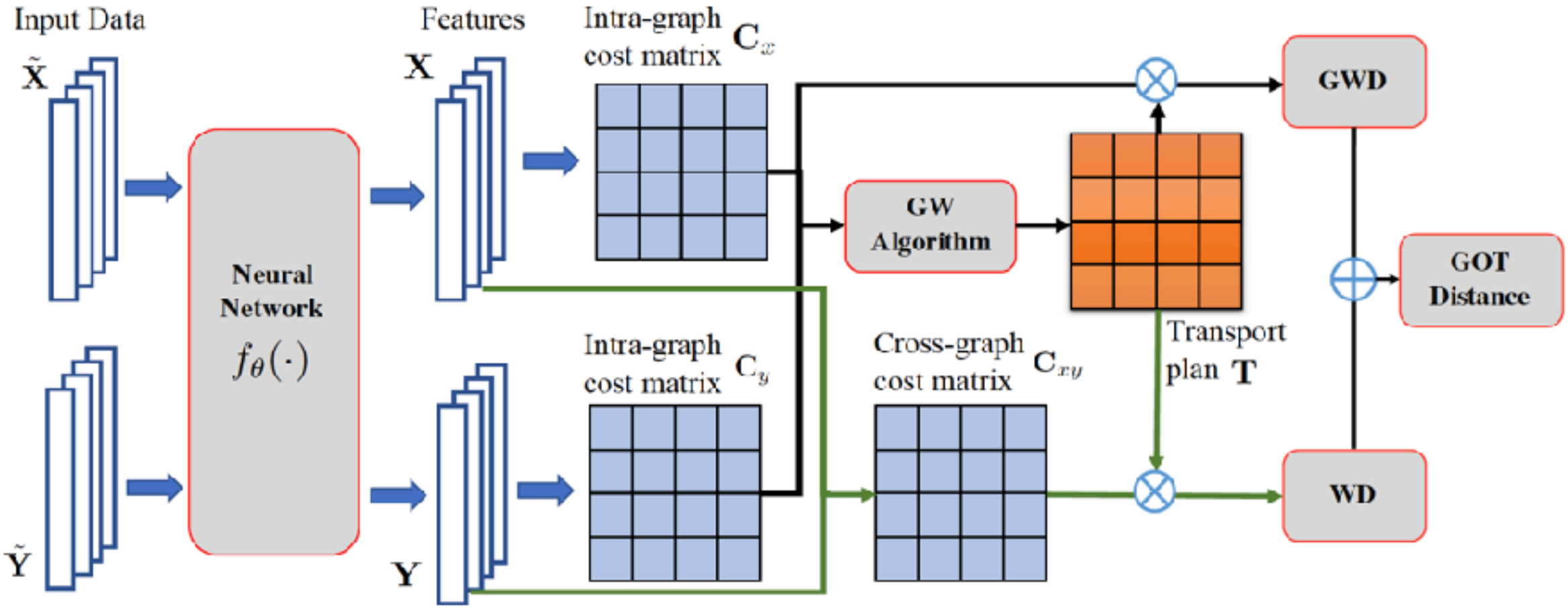
ALGORITHM2

Algorithm 2 Computing Gromov-Wasserstein Distance.

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

- $\mathscr{D}_{gw}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\boldsymbol{\gamma} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}) \sim \boldsymbol{\gamma}, (\boldsymbol{x}', \boldsymbol{y}') \sim \gamma} \left[L\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}'\right) \right]$
  $= \min_{\hat{\mathbf{T}} \in \Pi(\mathbf{u}, \mathbf{v})} \sum_{i,i',j,j'} \hat{\mathbf{T}}_{ij} \hat{\mathbf{T}}_{i'j'} L\left(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}_i', \boldsymbol{y}_j'\right)$
- $L\left(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}_i', \boldsymbol{y}_j'\right) = \left\| c_1\left(\boldsymbol{x}_i, \boldsymbol{x}_i'\right) - c_2\left(\boldsymbol{y}_i, \boldsymbol{y}_i'\right) \right\|$

SCHEMATIC COMPUTATION GRAPH OF GOT



ALGORITHM3

Algorithm 3 Computing GOT Distance.

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

- $L_{GOT} = \lambda_1 L_W + \lambda_2 L_{GW}$
- $L_{\text{unified}} = \lambda c(\boldsymbol{x}, \boldsymbol{y}) + (1-\lambda) L\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}'\right)$
- $\mathscr{D}_{got}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T}\in\Pi(\mathbf{u},\mathbf{v})} \sum_{i,i',j,j'} \mathbf{T}_{ij} \left(\lambda c\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right) + (1-\lambda)\mathbf{T}_{i'j'} \mathscr{L}\left(\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{x}'_i, \boldsymbol{y}'_j\right) \right)$

SCAN MODEL

$$s_{ij} = \frac{v_i^T e_j}{\|v_i\| \|e_j\|}, i \in [1,k], j \in [1,n]$$

$$a_i^t = \sum_{j=1}^{n} \alpha_{ij} e_j \qquad \alpha_{ij} = \frac{\exp(\lambda_1 \bar{s}_{ij})}{\sum_{j=1}^{n} \exp(\lambda_1 \bar{s}_{ij})}$$

$$R(v_i, a_i^t) = \frac{v_i^T a_i^t}{\|v_i\| \|a_i^t\|}$$

$$S_{AVG}(I,T) = \frac{\sum_{i=1}^{k} R(v_i, a_i^t)}{k}$$

$$l(I,T) = \sum_{\hat{T}} [\alpha - S(I,T) + S(I,\hat{T})]_+ + \sum_{\hat{I}} [\alpha - S(I,T) + S(\hat{I},T)]_+$$

EXPERIMENTS

- **Image-Text Retrieval**

- Image representation: pre-trained Faster R-CNN (36,2048)
- Textual features: bi-directional GRU
- Dataset: Flickr30K (Plummer et al.,2015), COCO (Lin et al., 2014)
- Model: SCAN
- Triple loss: $L_{\text{triple}}\left(x_i, y_i, y_j\right) = \max\left\{0, \gamma + \text{sim}\left(x_i, y_i\right) - \text{sim}\left(x_i, y_j\right)\right\}$
- $\left(x_i, y_i\right)$ is a positive image-text pair
- y_j is the most negative sample for x_i
- γ : hyper-parameter
- sim: function~ evaluates the similarity of image and text
- Final objective: $L_{\text{final}} = L_{\text{triple}} + L_{GOT}$

EXPERIMENTS

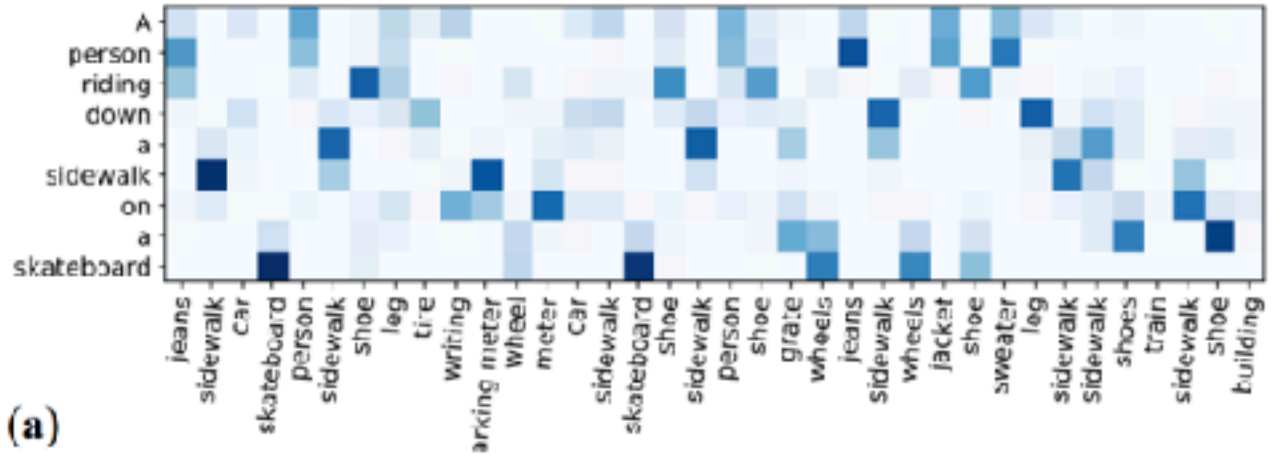
- Flickr30K
- COCO

Method	Sentence Retrieval R@1	R@5	R@10	Image Retrieval R@1	R@5	R@10	Rsum
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**
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**

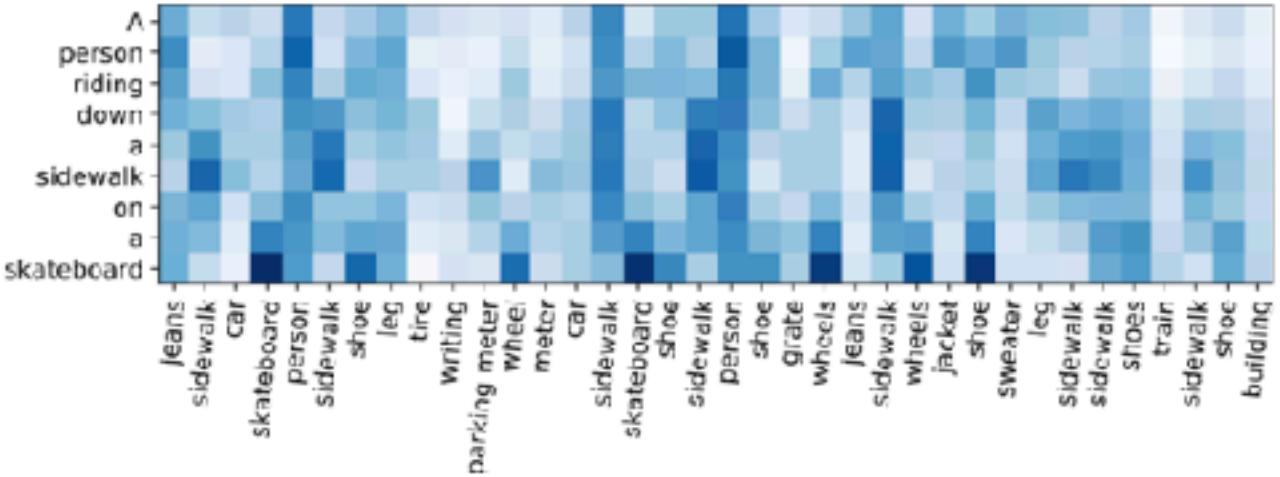
EXPERIMENTS

word tokens

transport plan from GOT



(a)



learned attention matrix from SCAN

(b)

image regions

EXPERIMENTS

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.

- Visual Question Answering
- Dataset: VQA 2.0
  - human annotated QA pairs on COCO images
- Model: BAN, BUTD

- Classification:
  - cross-entropy loss L_{cls}
- Final objective:
  - $L_{\text{final}} = L_{cls} + L_{GOT}$

REFERENCE

- 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.
- Cuturi, M. and Peyr e, G. Computational optimal transport.2017.
- Peyr e, G., Cuturi, M., and Solomon, J. Gromov-wasserstein averaging of kernel and distance matrices. In ICML, 2016.
- Lee, K.-H. et al. Stacked cross attention for image-text matching. In ECCV, 2018.
- Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. Vse++: Improved visual-semantic embeddings. In BMVC, 2018.