Modular Graph Transformer Networks for Multi-Label Image Classification

Hoang D.Nguyen, Xuan-Son Vu, Duc-Trong Le

@AAAI 2021

Outline

- Multi-Label Image Classification
- Motivating Example
- Modular Graph Transformer Network (MGTN)
- Experiments

Multi-Label Image Classification

Predict

Person, Chair, Umbrella, Car

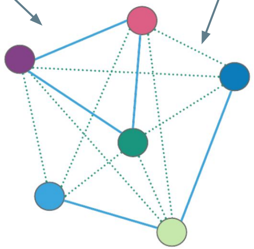
Motivating Example (1)

person, chair, umbrella, car

person, chair, dining table, cup

person, umbrella

dining table, cup, bowl



Semantic Graph between Labels

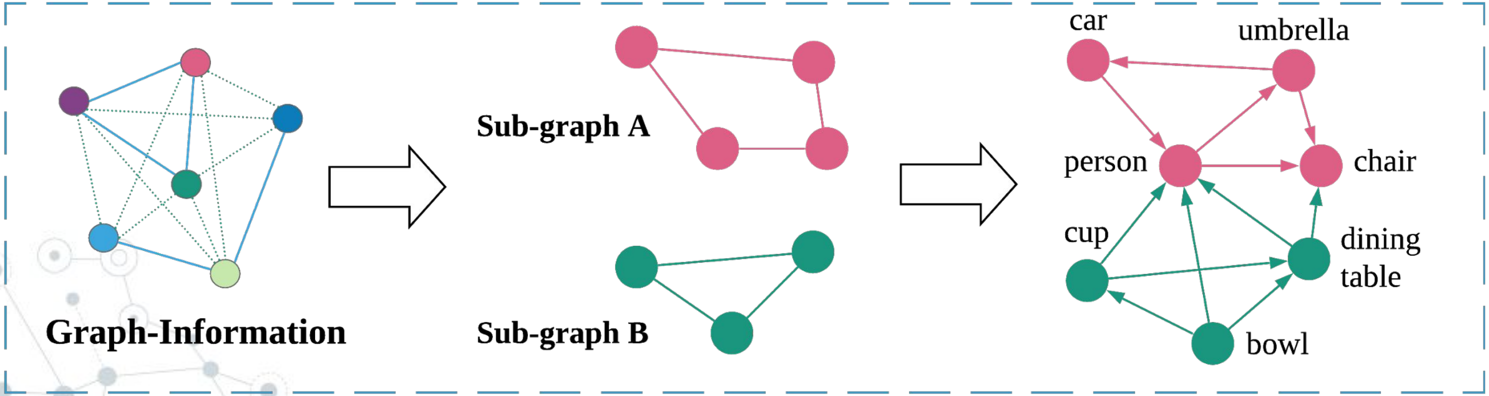
Motivating Example (2)

person, chair, umbrella, car

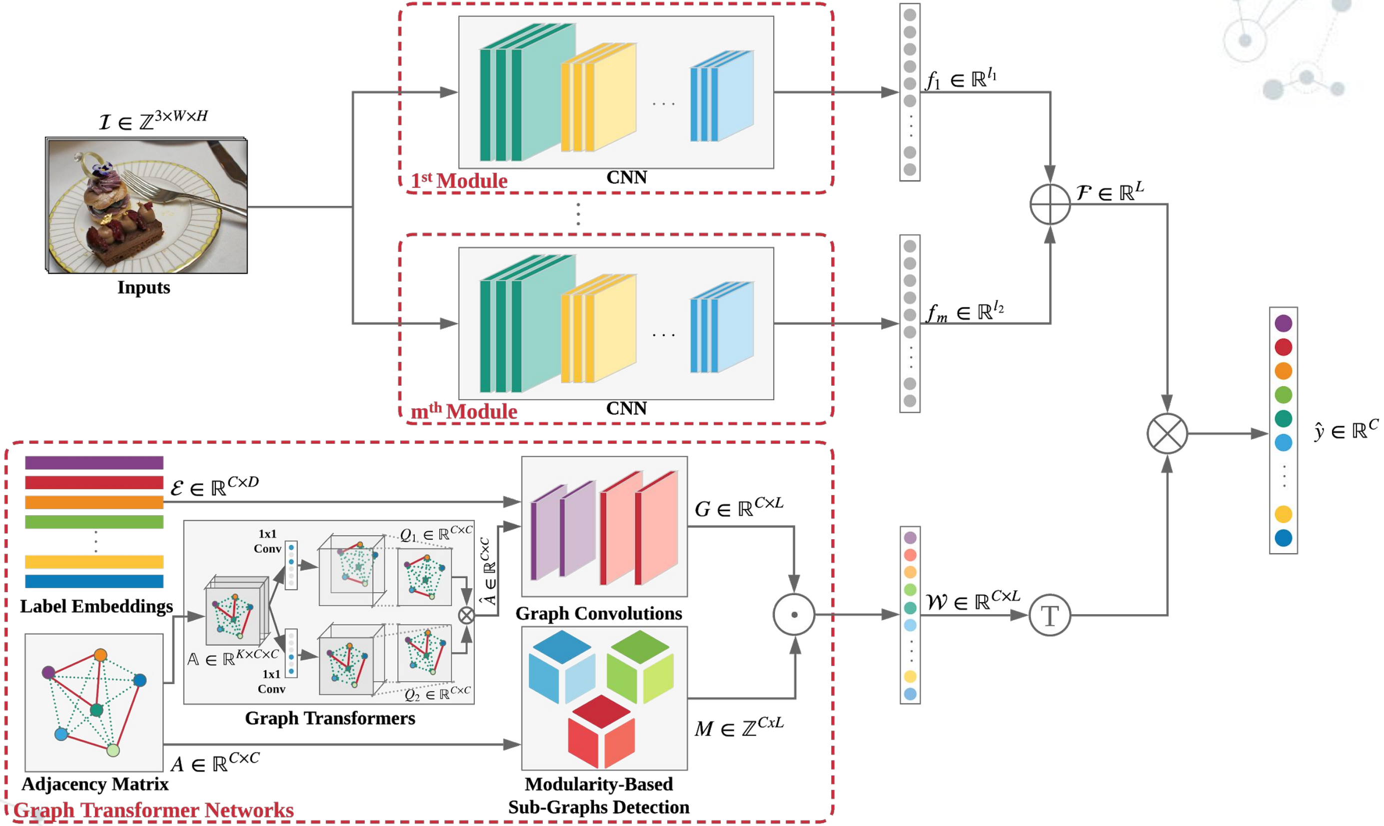
person, chair, dining table, cup

person, umbrella

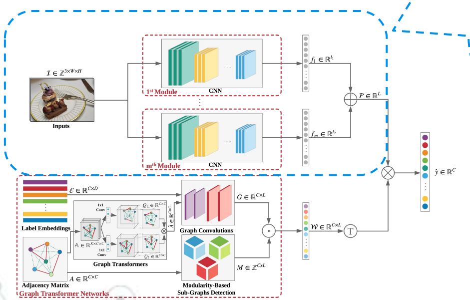
dining table, cup, bowl

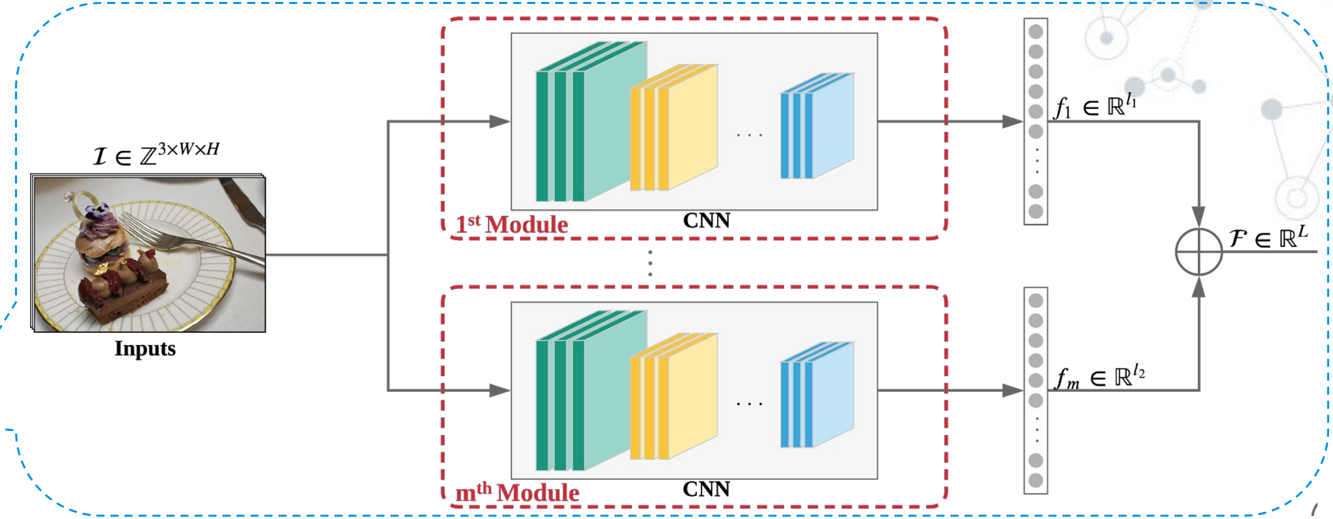


Modular Graph Transformer Network (MGTN)



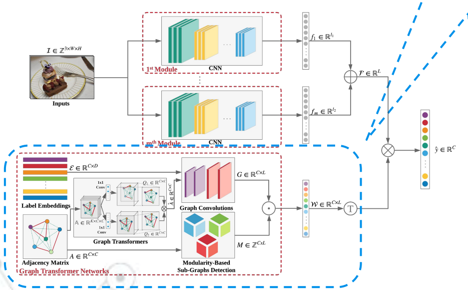
Convolutional Neural Networks

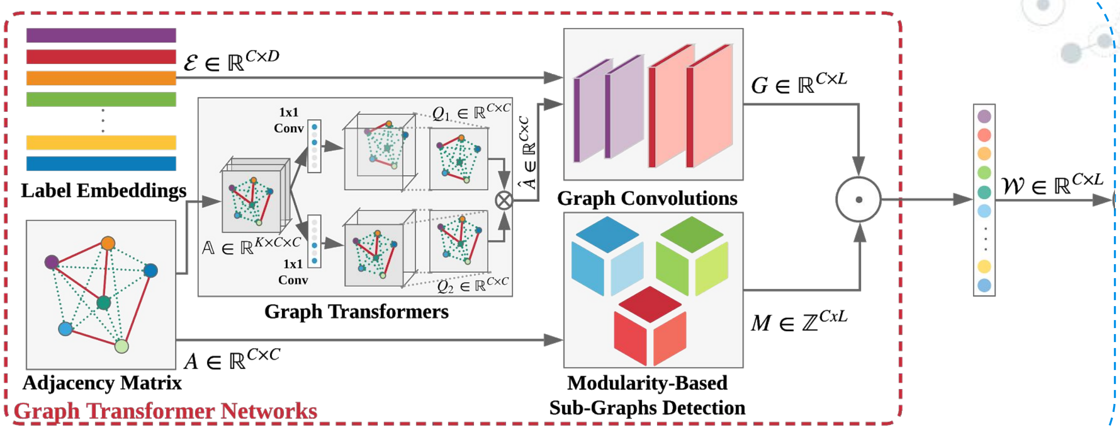




Idea: Exploit multiple modules of CNNs to infer the image-label representation $\mathcal{F} \in \mathbb{R}^{L}$ for each image input $\mathcal{I} \in \mathbb{Z}^{3\times W\times H}$

Graph Transformer Networks

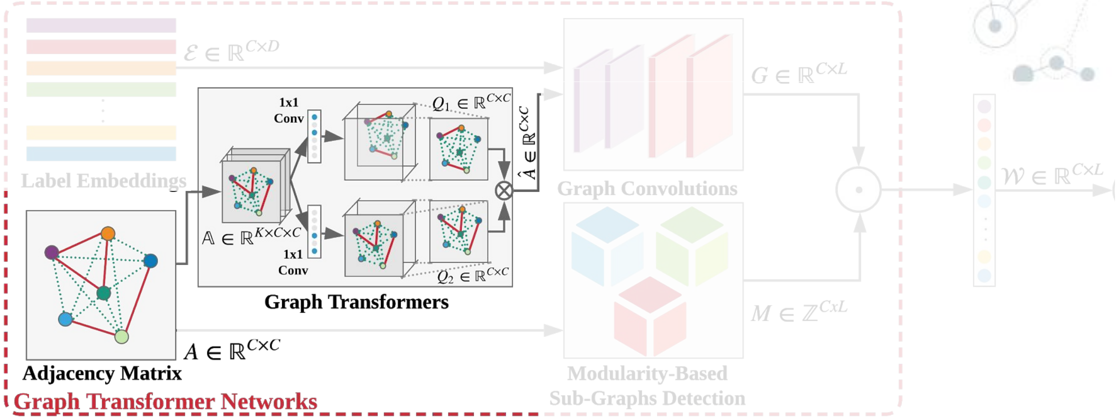




Idea: Exploit semantic information ($\mathcal{E} \in \mathbb{R}^{C\times D}$ & topological information ($A \in \mathbb{R}^{C\times C}$) between the set of C labels

Graph Transformers

Idea: Dynamically select important topological connections of the label graph.



Adjacency Tensor

$$\mathbb{A}_{kij} = \begin{cases} 1 & \text{if } P_{ij} \in [t_{k-1}, t_k), i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbb{A}_1 \equiv I$

$$P_{ij} = \varrho * A_{ij} / d_i$$

$$d_i = \sum_k A_{i,k}$$

$$t_k \in [0, 1], k \in \{1, \ldots, K\}$$

Two 1x1 convolutions $Q_1, Q_2 \in \mathbb{R}^{C \times C}$

(Yun et al. 2019)

$$Q_1 = \phi(\mathbb{A}, \text{softmax}(W_\phi^1))$$

$$Q_2 = \phi(\mathbb{A}, \text{softmax}(W_\phi^2))$$

$$\hat{A} = \eta(Q_1 Q_2 + I)$$

$$W_\phi^1, W_\phi^2 \in \mathbb{R}^{1 \times 1 \times K}$$

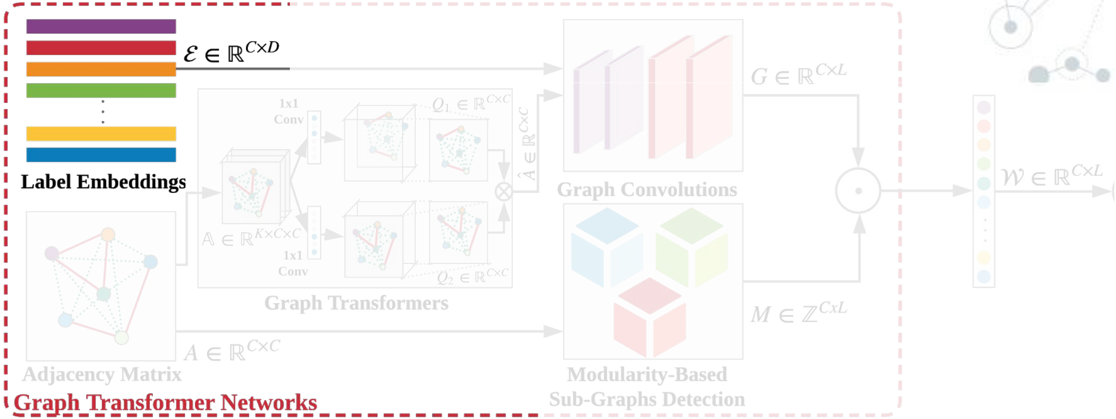
$$\hat{A} \in \mathbb{R}^{C \times C}$$

where

$$\eta(A) = d^{\frac{-1}{2}} A d^{\frac{-1}{2}}$$

Eigenvector-based Embedding Transformation

Idea: Exploit semantic information from label embeddings



Pre-Trained Embeddings:

1) **Char2Vec** (Kim et al. 2015): character-level, D = 300

2) **BERT_Base** (Devlin et al. 2018): word-level, averaging the last layer, D = 768

3) **RoBERTa** (Liu et al. 2019): word-level, averaging the last layer, D = 768

Eigenvector-based transformation for pre-trained embedding E

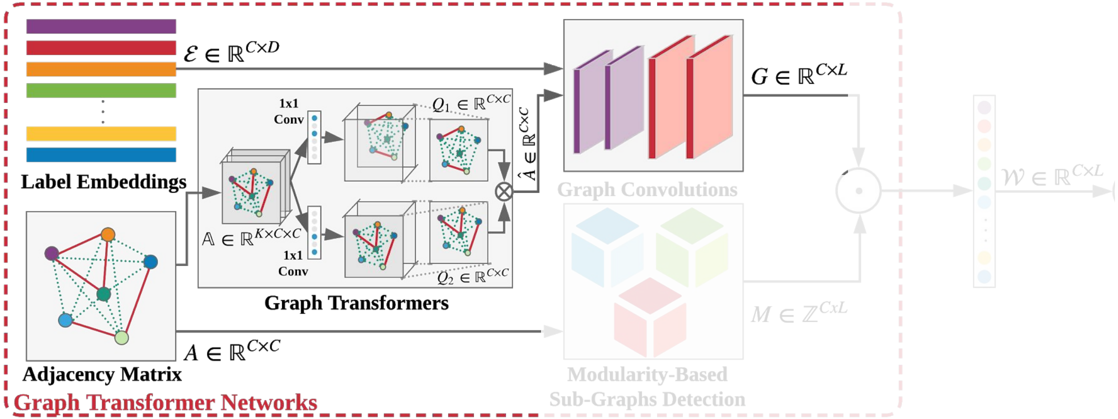
$$\mathcal{E} = E \cdot \mathcal{C}^T$$

where $\mathcal{C}_i$ the eigenvector centrality of the label *i*-th

$$\mathcal{C}_i = \frac{1}{\lambda} \sum_k a_{k,i}\, \mathcal{C}_k$$

$\lambda \neq 0$ is the largest eigenvalue

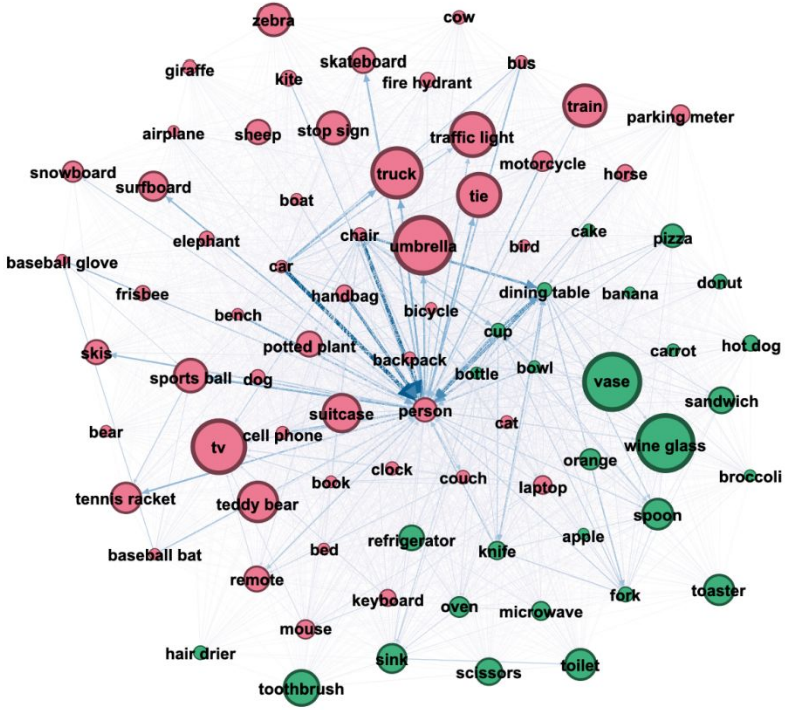
Graph Convolutional Network



Idea: Jointly exploit label-level word embedding and topological information via graph convolutional networks (GCN, Kipf et al. 2017)

$$G = \mathrm{GCN}(\mathcal{E}, \hat{A}) \qquad G \in \mathbb{R}^{C \times L}$$

Modularity-based Sub-Graphs Detection



MS-COCO

The Clauset-Newman-Moore agglomeration algorithm (Clauset et al. 2004)

Modularity-based Enhancement

Idea: Exploit different highly inter-connected sets of objects among sub-graphs

The modularity of a graph :

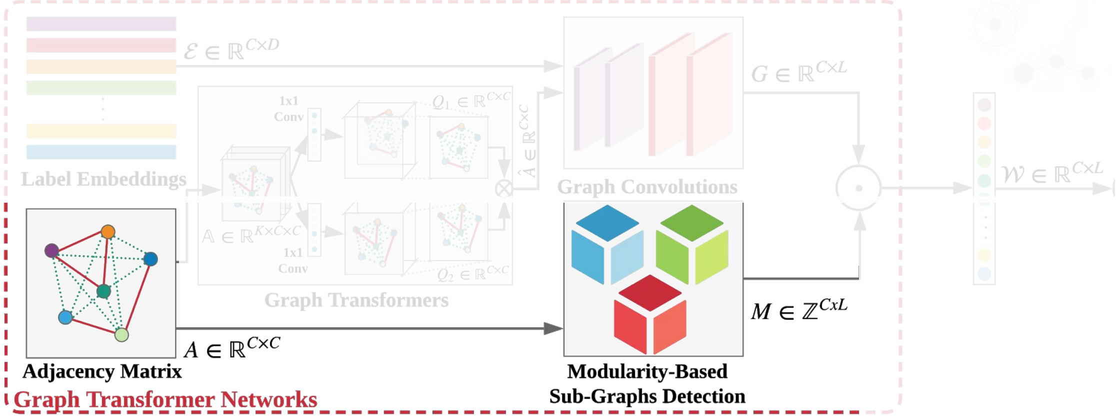
$$\mathfrak{Q} = \frac{1}{2\mathfrak{m}} \sum_{i,j} \left[A_{i,j} - \frac{\mathscr{d}_i \mathscr{d}_j}{2\mathfrak{m}} \right] \delta(c_i, c_j)$$

where

$$\mathfrak{m} = \frac{1}{2} \sum_{i,j} A_{i,j}$$

$$\mathscr{d}_i = \sum_k A_{i,k}$$

$\delta(u, v)$ is 1 if $u = v$ otherwise 0

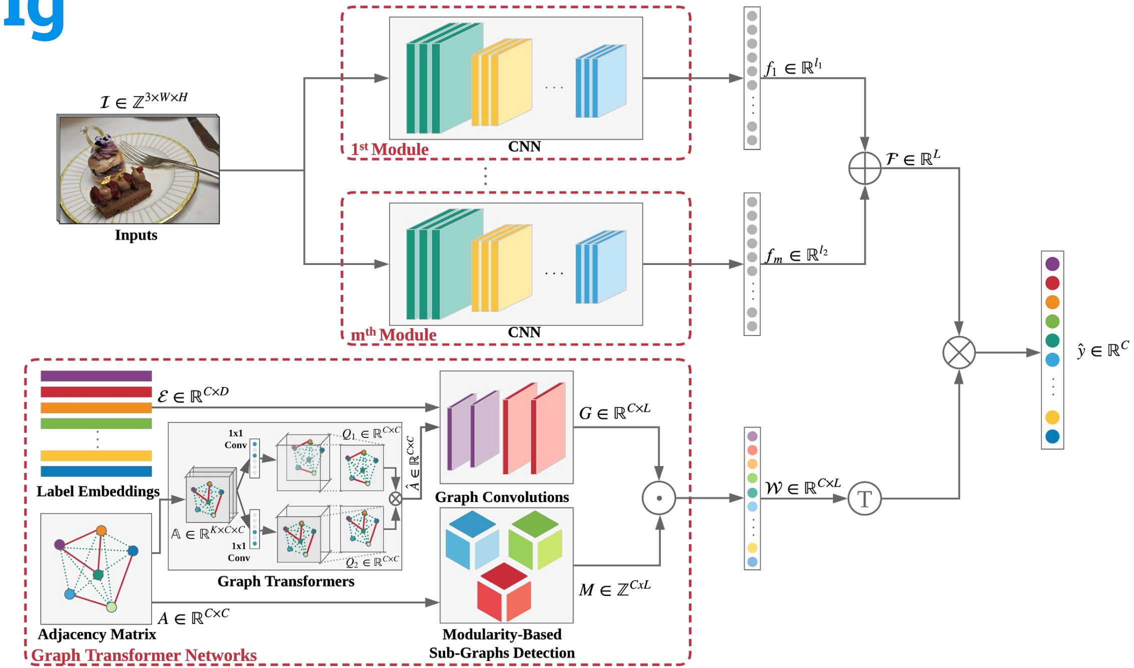


Suppose that ***m*** sub-graphs are discovered. Multiple CNNs with configurable backbones are employed using a control tensor M with a threshold τ :

$$M_{iv} = \begin{cases} \tau & \text{if } S_i = p \text{ and } v \in f_p \\ \frac{1-\tau}{m-1} & \text{otherwise} \end{cases}$$

where S is the sub-graph assignment, $f_p \in \mathbb{R}^{l_p}$ is the image-level representation

Learning



$$\mathcal{W} = G \odot M \qquad \hat{y} = \mathcal{W}^T \mathcal{F}$$

$$\mathcal{L} = -\frac{1}{C}\sum_{c=1}^{C} y_c \log(\sigma(\hat{y}_c)) + (1-y_c)\log(1-\sigma(\hat{y}_c))$$

Experimental Setups

- **Dataset**: MS-COCO (81 labels), Fashion550K (66 labels)
- **Evaluation metrics**: mAP, per-class (CP, CR, CF1), overall (OP, OR, OF1)
- **Preprocessing:** Resize images 512x512 to 448x448 with random horizontal flips
- **Implementation**: Dual ResNeXt-50-32x4d backbones, 2-layer GCN, $\tau = 0.999$, adjacency thresholds (MS-COCO = $[0.1, 0.3, 1.0]$, Fashion550K = $[0.2, 0.4, 1.0]$d learning rate decays by a factor of 10 for every 20 epochs.

Experimental Results - MS-COCO

Method	mAP	CP	CR	CF1	OP	OR	OF1
CNN-RNN (Wang et al. 2016)	61.2	-	-	-	-	-	-
SRN (Zhu et al. 2017)	77.1	81.6	65.4	71.2	82.7	69.9	75.8
Baseline(ResNet101) (He et al. 2016)	77.3	80.2	66.7	72.8	83.9	70.8	76.8
Multi-Evidence (Ge, Yang, and Yu 2018)	–	80.4	70.2	74.9	85.2	72.5	78.4
ML-GCN (Chen et al. 2019b)	82.4	84.4	71.4	77.4	85.8	74.5	79.8
ML-GCN (ResNeXt50 with ImageNet)	86.2	85.8	77.3	81.3	86.2	79.7	82.8
A-GCN (Li et al. 2019)	83.1	84.7	72.3	78.0	85.6	75.5	80.3
KSSNet (Wang et al. 2020b)	83.7	84.6	73.2	77.2	87.8	76.2	81.5
SGTN (Our) (Vu et al. 2020)	86.6	77.2	**82.2**	79.6	76.0	**82.6**	79.2
MGTN(Base)	86.9	**89.4**	74.5	81.3	**90.9**	76.3	83.0
MGTN(Final)	**87.0**	86.1	77.9	**81.8**	87.7	79.4	**83.4**

Model Analysis - Embeddings

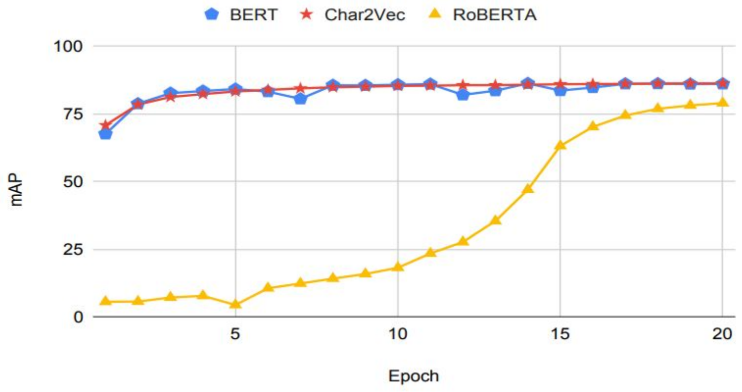


Figure 4: Learning patterns of MGTN with different label embeddings in 20 epochs. The MGTN model with the setting using RoBERTa$_{avg_12}$ label embedding shows a slow learning speed in comparison to others.

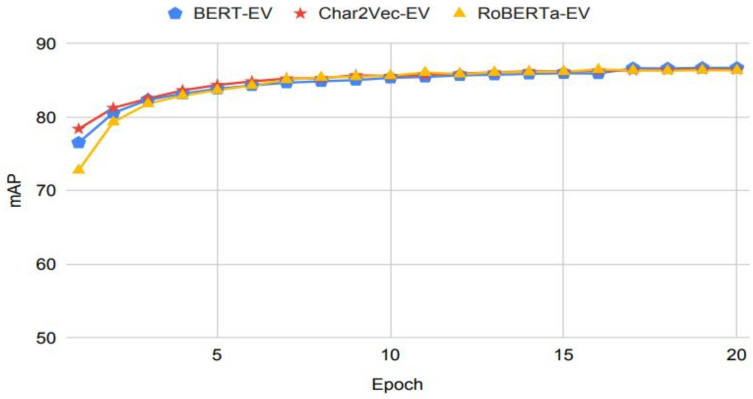
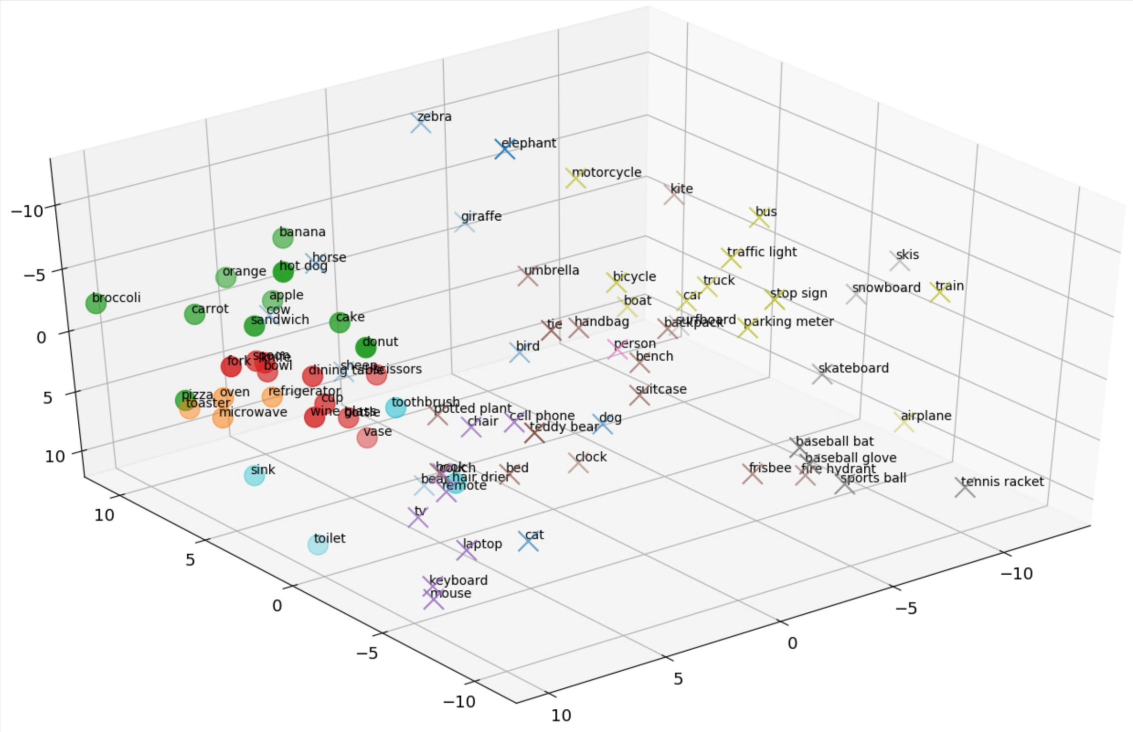


Figure 5: The EV-enhancement for label embedding helps the MGTN's model learn faster, even MGTN with the setting using the RoBERTa$_{avg_12}$ now learns faster. Note: y-axis here is ranged in $[50, 100]$ for visibility.

3D t-SNE Visualization on MS-COCO



Conclusion

- Introduce **Modular Graph Transformer Network (MGTN)** to solve multi-label image classification
  - Employ **multiple CNN backbones** on unfolded **sub-graphs**
  - Exploit **topological and semantic information** via the graph transformer and EV-based embedding transformation.
- **MGTN** shows significant improvements against SOTA methods on MS-COCO and Fashsion550K

Thanks!

Any questions?