Artificial Intelligence

Incorporating Hierarchy into Text Encoder: a Contrastive Learning Approach for Hierarchical Text Classification

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, Houfeng Wang*
Key Laboratory of Computational Linguistics, Peking University, MOE, China
{wangzh9969, wangpeiyi9979}@gmail.com
{hlz, sunx5, wanghf}@pku.edu.cn

Code and dataset are available at https://github.com/wzh9969/contrastive-htc

——ACL2022

gesis Leibniz-Institut für Sozialwissenschaften

Forschungszentrum · Research Center L3S

Reported by Yabo Yin

1.Introduction

2.Method

3.Experiments

Introduction

1. The text representation interacts with constant hierarchy representation and thus the interaction seems redundant and less effective.

2. Although such methods can capture the hierarchical information, recent researches demonstrate that encoding the holistic label structure directly by a structure encoder can further improve performance

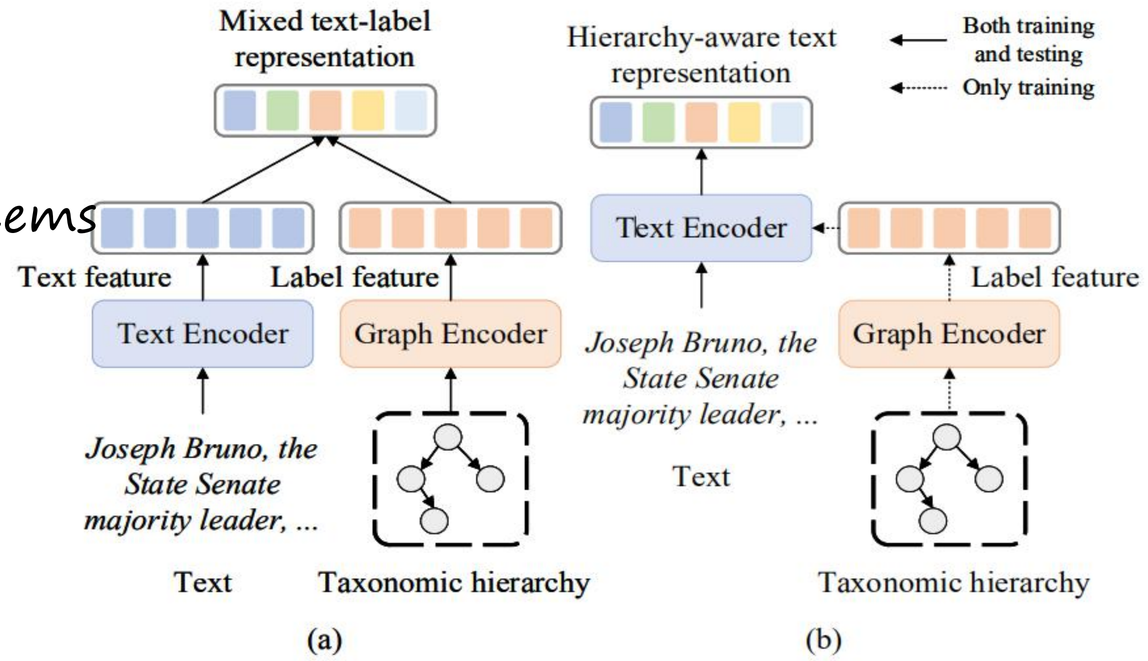


Figure 1: Two ways of introducing hierarchy information. (a) Previous work model text and labels separately and find a mixed representation. (b) Our method incorporating hierarchy information into text encoder for a hierarchy-aware text representation.

Method

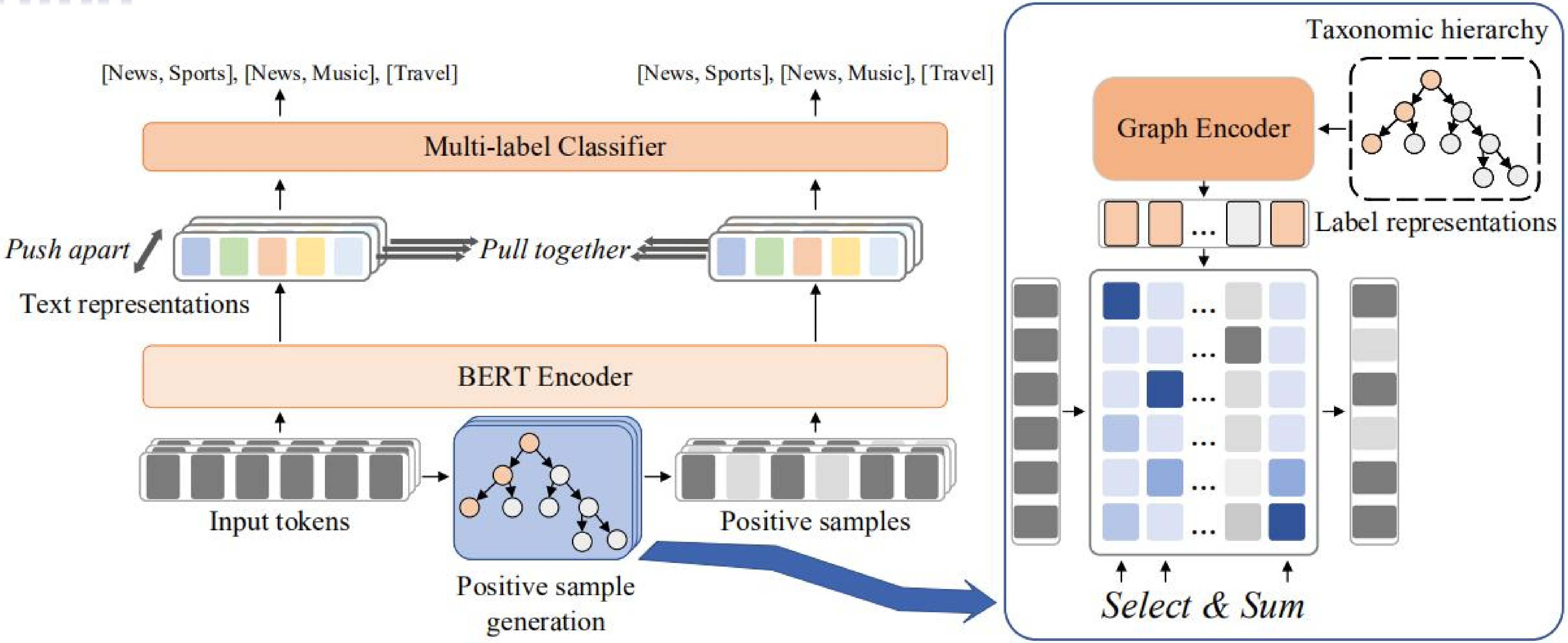


Figure 2: An overview of HGCLR under a batch of 3. HGCLR adopts a contrastive learning framework to regularize BERT representations. We construct positive samples by masking unimportant tokens under the guidance of hierarchy and labels. By pulling together and pushing apart representations, the hierarchy information can be injected into the BERT encoder.

Method

Problem Definition

$x = \{x_1, x_2, ..., x_n\}$, subset y of label set Y

Y are predefined a Directed Acyclic Graph (DAG) $G = (Y, E)$.

Text Encoder

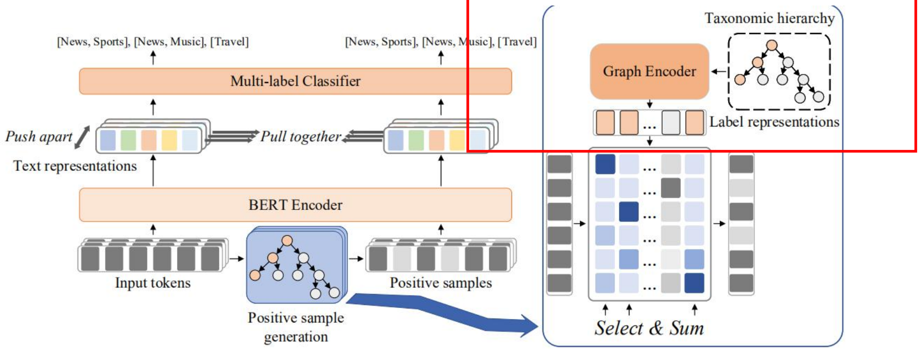
$$x = \{[\text{CLS}], x_1, x_2, ..., x_{n-2}, [\text{SEP}]\} \quad (1)$$

$$H = \text{BERT}(\text{x}) \quad (2)$$

$$H \in \mathbb{R}^{n \times d_h}$$

$$h_x = h_{[\text{CLS}]}$$

Method

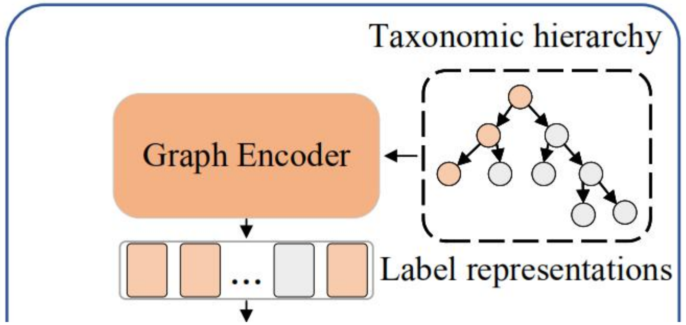


Graph Encoder

$$f_i = \text{label\_emb}(y_i) + \text{name\_emb}(y_i). \quad (3) \qquad F \in \mathbb{R}^{k \times d_h}$$

$$A_{ij}^G = \frac{(f_i W_Q^G)(f_j W_K^G)^T}{\sqrt{d_h}} + c_{ij} + b_{\phi(y_i, y_j)} \quad (4)$$

$$W_Q^G \in \mathbb{R}^{d_h \times d_h}$$
$$W_K^G \in \mathbb{R}^{d_h \times d_h}.$$



where $c_{ij} = \frac{1}{D} \sum_{n=1}^{D} w_{e_n}$ and $D = \phi(y_i, y_j)$

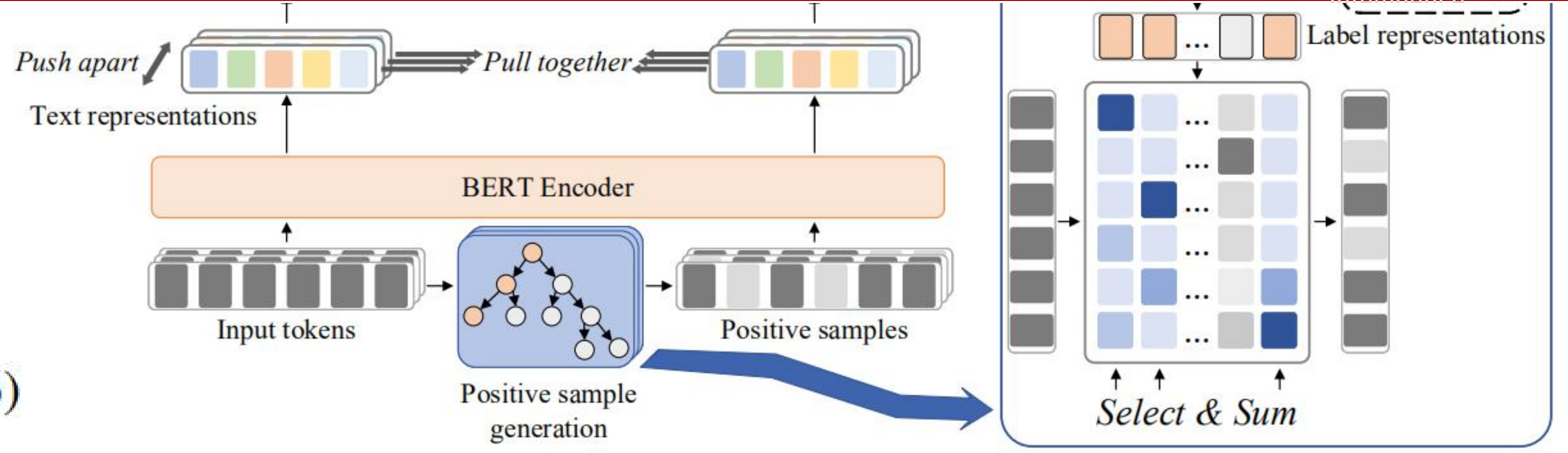
node y_i and y_j, one and only one path $(e_1, e_2, ..., e_D)$ $\quad w_{e_i} \in \mathbb{R}^1$

$\phi(y_i, y_j)$ denotes the distance between two nodes y_i and y_j

$b_{\phi(y_i, y_j)}$ is a learnable scalar indexed by $\phi(y_i, y_j)$

$$L = \text{LayerNorm}(\text{softmax}(A^G)V + F) \quad (5)$$

Method



Positive Sample Generation

$$\{e_1, e_2, ..., e_n\} = \text{BERT\_emb}(x) \qquad (6)$$

$$q_i = e_i W_Q, k_j = l_j W_K, A_{ij} = \frac{q_i k_j^T}{\sqrt{d_h}} \qquad (7)$$

$$P_{ij} = \text{gumbel\_softmax}(A_{i1}, A_{i2}, ..., A_{ik})_j \qquad (8)$$

$$P_i = \sum_{j \in y} P_{ij} \qquad (9)$$

the positive sample $\hat{x}$ is constructed as:

$$\hat{x} = \{x_i \text{ if } P_i > \gamma \text{ else } \mathbf{0}\} \qquad (10)$$

$$\hat{H} = \text{BERT}(\hat{x}) \qquad (11)$$

Method

positive pairs $(h_i, \hat{h}_i)$.

a batch of N there are $2(N-1)$ negative pairs.

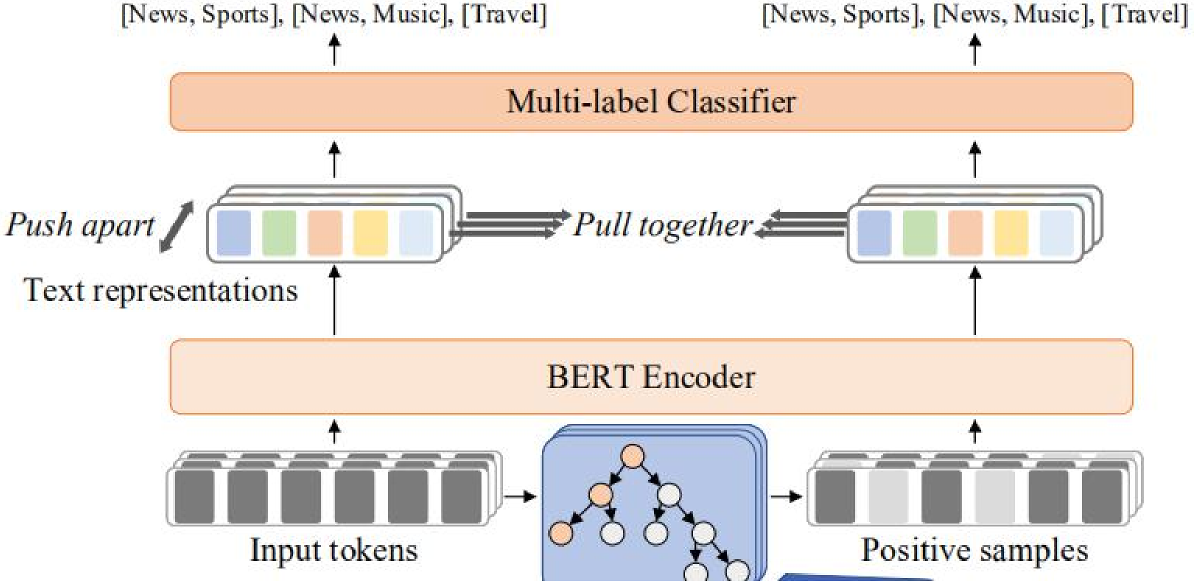
$$c_i = W_2 \mathrm{ReLU}(W_1 h_i)$$
$$\hat{c}_i = W_2 \mathrm{ReLU}(W_1 \hat{h}_i) \qquad (12)$$

$$\mathbf{Z} = \{z \in \{c_i\} \cup \{\hat{c}_i\}\}$$

$$L_m^{con} = -\log \frac{\exp(\mathrm{sim}(z_m, \mu(z_m))/\tau)}{\sum_{i=1, i\neq m}^{2N} \exp(\mathrm{sim}(z_m, z_i)/\tau)} \qquad (13)$$

$$\mu(z_m) = \begin{cases} c_i, & \text{if } z_m = \hat{c}_i \\ \hat{c}_i, & \text{if } z_m = c_i \end{cases} \qquad (14)$$

$$L^{con} = \frac{1}{2N} \sum_{m=1}^{2N} L_m^{con} \qquad (15)$$



$$p_{ij} = \mathrm{sigmoid}(W_c h_i + b_c)_j \qquad (16)$$

$$L_{ij}^C = -y_{ij}\log(p_{ij}) - (1-y_{ij})\log(1-p_{ij}) \qquad (17)$$

$$L^C = \sum_{i=1}^{N}\sum_{j=1}^{k} L_{ij}^C \qquad (18)$$

$$L = L^C + \hat{L^C} + \lambda L^{con} \qquad (19)$$

Experiments

Dataset	$\|Y\|$	Depth	Avg($\|y_i\|$)	Train	Dev	Test
WOS	141	2	2.0	30,070	7,518	9,397
NYT	166	8	7.6	23,345	5,834	7,292
RCV1-V2	103	4	3.24	20,833	2,316	781,265

Table 1: Data Statistics. $|Y|$ is the number of classes. Depth is the maximum level of hierarchy. Avg($|y_i|$) is the average number of classes per sample.

Experiments

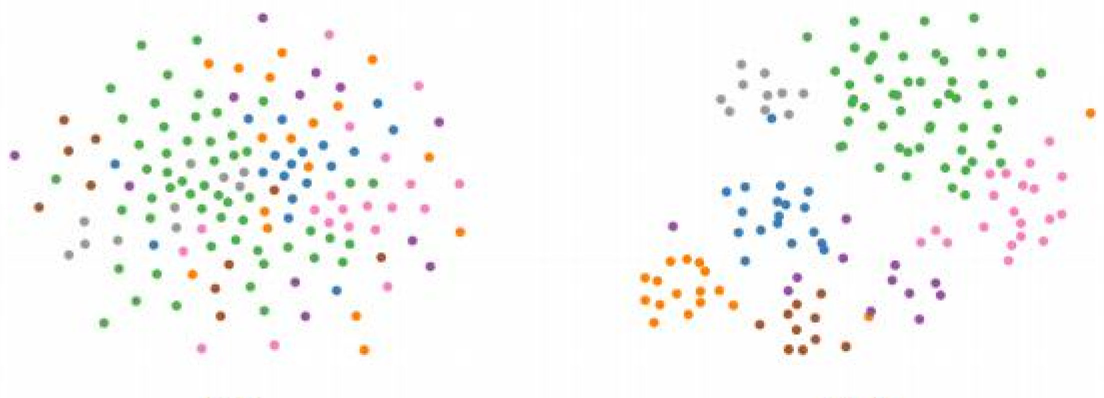
Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Hierarchy-Aware Models						
TextRCNN (Zhou et al., 2020)	83.55	76.99	70.83	56.18	81.57	59.25
HiAGM (Zhou et al., 2020)	85.82	80.28	74.97	60.83	83.96	63.35
HTCInfoMax (Deng et al., 2021)	85.58	80.05	-	-	83.51	62.71
HiMatch (Chen et al., 2021)	86.20	80.53	-	-	84.73	64.11
Pretrained Language Models						
BERT (Our implement)	85.63	79.07	78.24	65.62	85.65	67.02
BERT (Chen et al., 2021)	86.26	80.58	-	-	86.26	67.35
BERT+HiAGM (Our implement)	86.04	80.19	78.64	66.76	85.58	67.93
BERT+HTCInfoMax (Our implement)	86.30	79.97	78.75	67.31	85.53	67.09
BERT+HiMatch (Chen et al., 2021)	86.70	81.06	-	-	86.33	**68.66**
HGCLR	**87.11**	**81.20**	**78.86**	**67.96**	**86.49**	68.31

Table 2: Experimental results of our proposed model on several datasets. For a fair comparison, we implement some baseline with BERT encoder. We cannot reproduce the BERT results reported in Chen et al. (2021) so that we also report the results of our version of BERT.

Experiments

Ablation Models	Micro-F1	Macro-F1
BERT	85.75	79.36
HGCLR	**87.46**	**81.52**
-*r.p.* GCN	87.06	80.63
-*r.p.* GAT	87.18	81.45
-*r.m.* graph encoder	86.67	80.11
-*r.m.* contrastive loss	86.72	80.97

Table 3: Performance when replace or remove some components of HGCLR on the development set of WOS. *r.p.* stands for *replace* and *r.m.* stands for *remove*. We remove the contrastive loss by setting $\lambda = 0$.



(a) (b)

Figure 3: T-SNE visualization of the label representations on WOS dataset. Dots with same color are labels with a same father. (a) BERT model. (b) Our approach.

Experiments

Variants of Graphormer	Micro-F1	Macro-F1
Base architecture	**87.46**	**81.52**
-w/o name embedding	86.40	80.40
-w/o spatial encoding	86.88	80.42
-w/o edge encoding	87.25	80.54

Table 4: Performance with variants of Graphormer on development set of WOS. We remove name embedding, spatial encoding, and edge encoding respectively. “w/o” stands for “without”.

Generation Strategy	Micro-F1	Macro-F1
Hierarchy-guided	**87.46**	**81.52**
Dropout	86.94	79.91
Random masking	87.19	81.16
Adversarial attack	86.67	80.24

Table 5: Impact of different positive example generation techniques on the development set of WOS. Hierarchy-guided is the proposed method. We control the valid tokens in positive samples roughly the same for random methods. We select FGSM as the attack algorithm following Pan et al. (2021).

Thank you!