Learning the Best Pooling Strategy for Visual Semantic Embedding

Jiacheng Chen[1*] Hexiang Hu[2*] Hao Wu[1] Yuning Jiang[3] Changhu Wang[1]

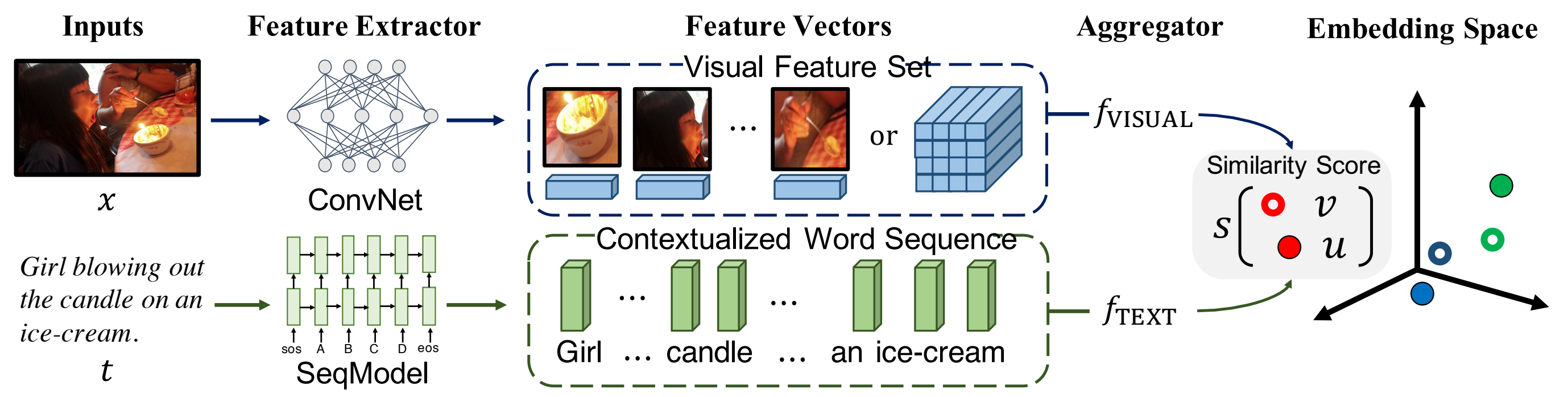
[1]ByteDance AI Lab [2] University of Southern California [3]Alibaba Inc

VIRTUAL JUNE 19-25

Code and pre-trained models available at:
https://vse-infty.github.io/

Background & Motivation

Visual Semantic Embedding (VSE)

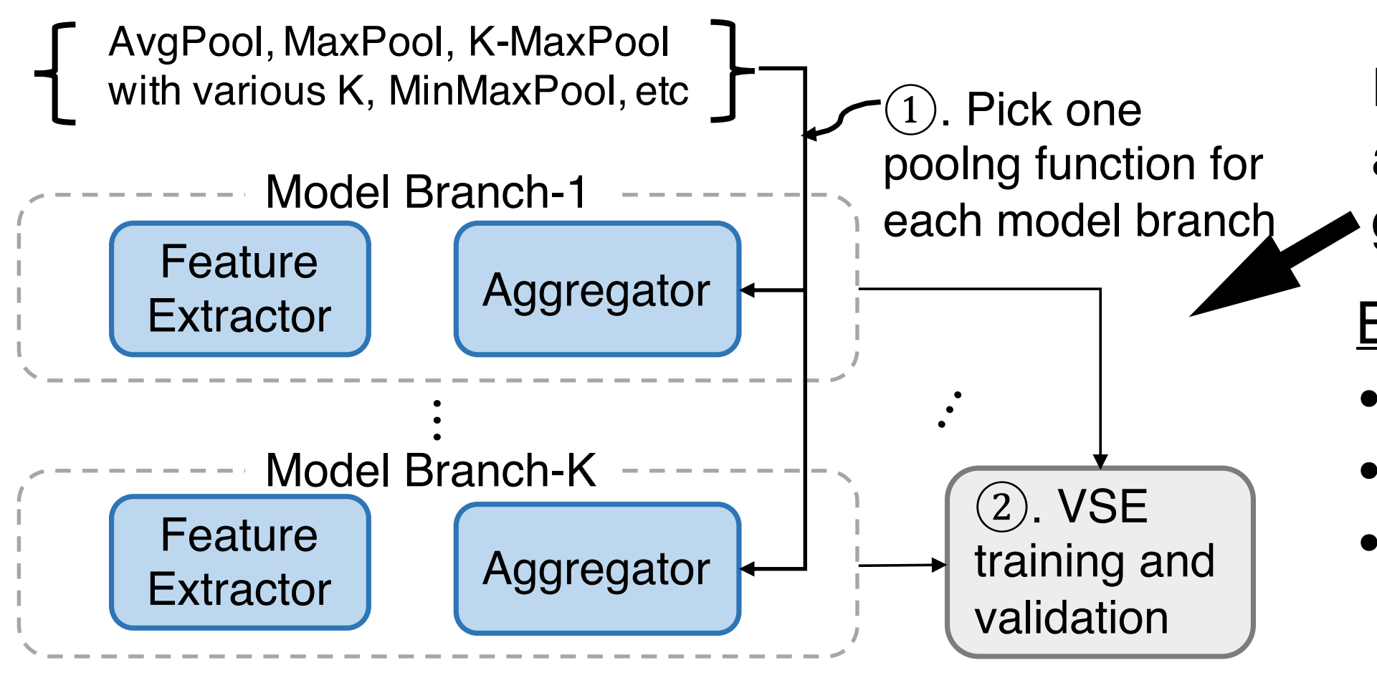


- **Multiple encoders**
- **Shared embedding space**
- **Contrastive learning**

- A classic multimodal learning framework based on dual encoder (Frome *et al.* NeurIPS 2013)
- Common use case: cross-modal retrieval (Kiros *et al.* NeurIPS 2014, Faghri *et al.* BMVC 2018), also used for large-scale visual pre-training recently (Jia *et al.* 2021)
- A series of recent works improved the **visual aggregator** for better feature contextualization (Li *et al.* ICCV 2019, Wehramann *et al.* ICCV 2019, Wang *et al.* ECCV 2020, etc.)

An Inspiring Empirical Finding

Manually-selected pooling function can be surprisingly effective as the feature aggregator


AvgPool, MaxPool, K-MaxPool with various K, MinMaxPool, etc
①. Pick one poolng function for each model branch
Model Branch-1
Feature Extractor
Aggregator
Model Branch-K
Feature Extractor
Aggregator
②. VSE training and validation

Aggregator	#Param	Region T → I	Region I → T	Grid T → I	Grid I → T
AvgPool	0	54.0	68.5	58.9	72.4
Seq2Seq	6.3M	58.5	69.9	61.5	73.3
SelfAttn	3.2M	56.2	70.2	60.3	73.0
GCN+AvgPool	4.2M	54.9	69.0	59.5	71.8
GCN+Seq2Seq	23.1M	**60.7**	72.5	59.5	71.1
Best Pooling Function	0	**60.7**	**74.5**	**61.6**	**76.3**

Repeat ① and ②, so that for the given data modalities and feature extractors, we manually **search for the best global pooling functions** as the feature aggregators

Experimental Setups:

- Image-text retrieval on MS-COCO 5-fold 1K benchmark
- Evaluated with Recall@1
- Fix the text feature backbone, and find the **optimal visual pooling function** for different choices of visual feature extractors
- **Region**: RoI features from a pre-trained object detector (Anderson *et al.* CVPR 2018)
- **Grid**: Standard feature maps from a ConvNet (Jiang *et al.* CVPR 2020)

Search results:

Region feature: MaxPool

Grid feature: K-MaxPool with K=20

Observations:

- The best-selected pooling function can be both **simple and effective**
- The best pooling function **varies when data modality and type of feature extractor changes**
- The search requires **repetitive experiments** -- costly and tedious for multiple modalities and feature extractors
- The search becomes harder when features have **variable lengths** (text, video, etc)

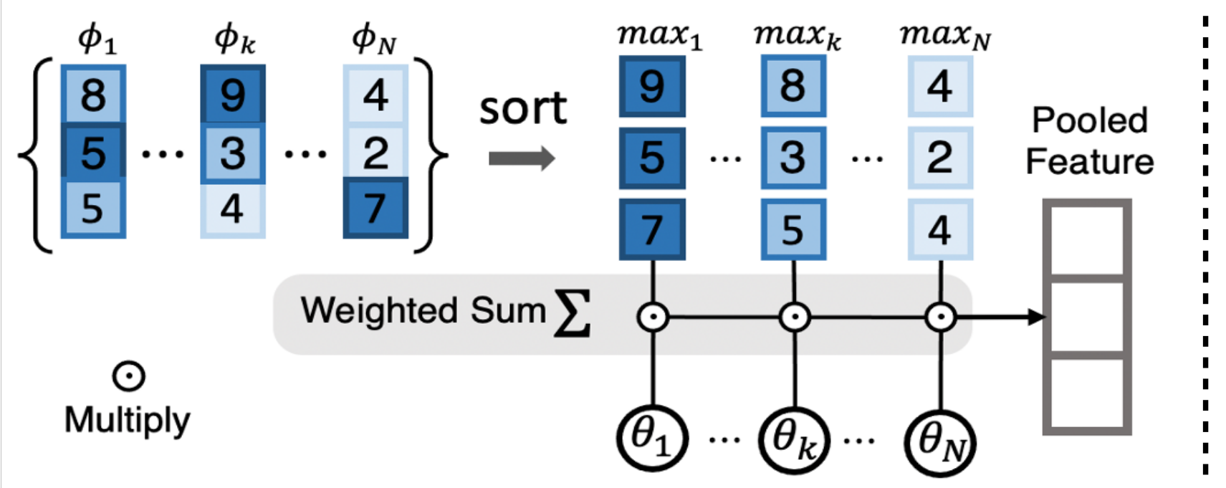
Therefore, we want a general/universal pooling operator that can:

- generalize over various pooling functions
- be trained to approximate the proper pooling strategy based on the data modality (e.g., image, text, video) and feature extractor (e.g., object detector or standard ConvNet for image, LSTM or BERT for sentence)
- naturally handle features with variable lengths

Generalized Pooling Operator (GPO)

1. Define the global pooling as a weighted sum over sorted features

- Features: $\{\phi_n\}_{n=1}^{N}$
- Pooling Coefficients: θ
- Sort: sort each feature dimension separately



θ **of AvgPool** $\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}$

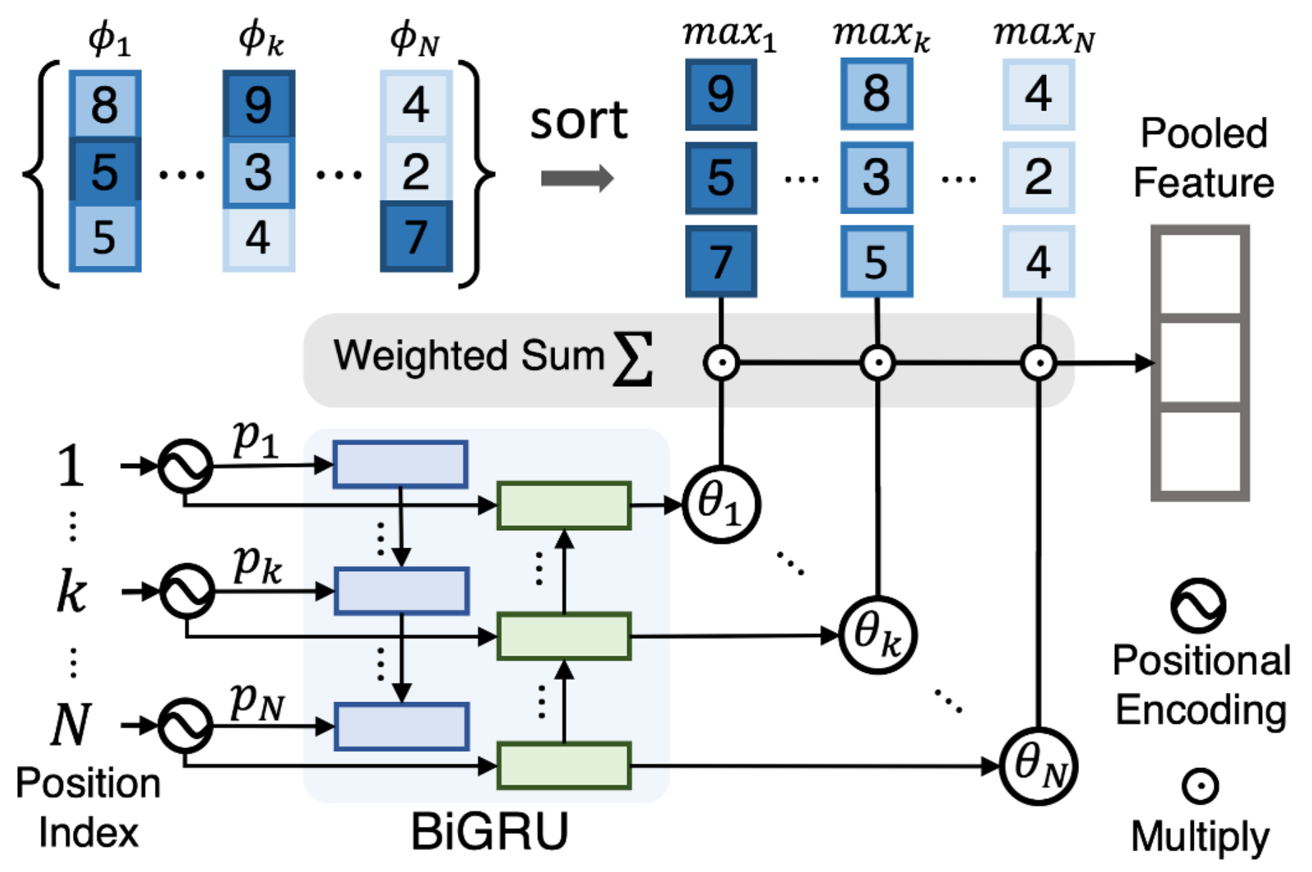
θ **of MaxPool** $1, 0, 0, \dots, 0$

θ **of K-MaxPool** $\frac{1}{K}, \dots, \frac{1}{K}, 0, \dots$

2. Generate the pooling coefficients

- Instead of making θ a trainable vector, we need a coefficients generator: θ=g(N), to handle variable lengths
- Parametrize g(·) with a sequence model, which can be LSTM, GRU, Transformer or other model architectures

3. The Final Implementation

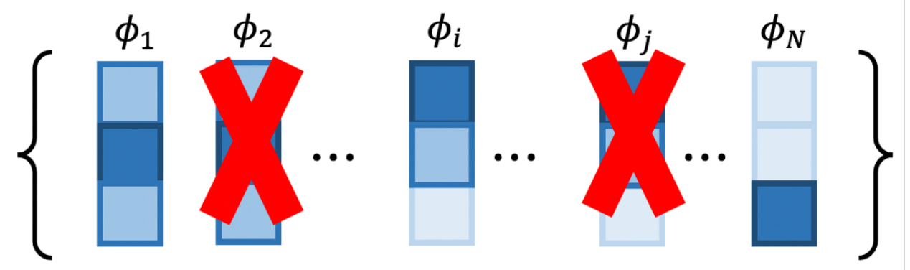


BiGRU: simple and parameter-efficient, we set the number of hidden dimension to be 32 to make minimize the computational overhead

Trigonometric Positional Encoding:

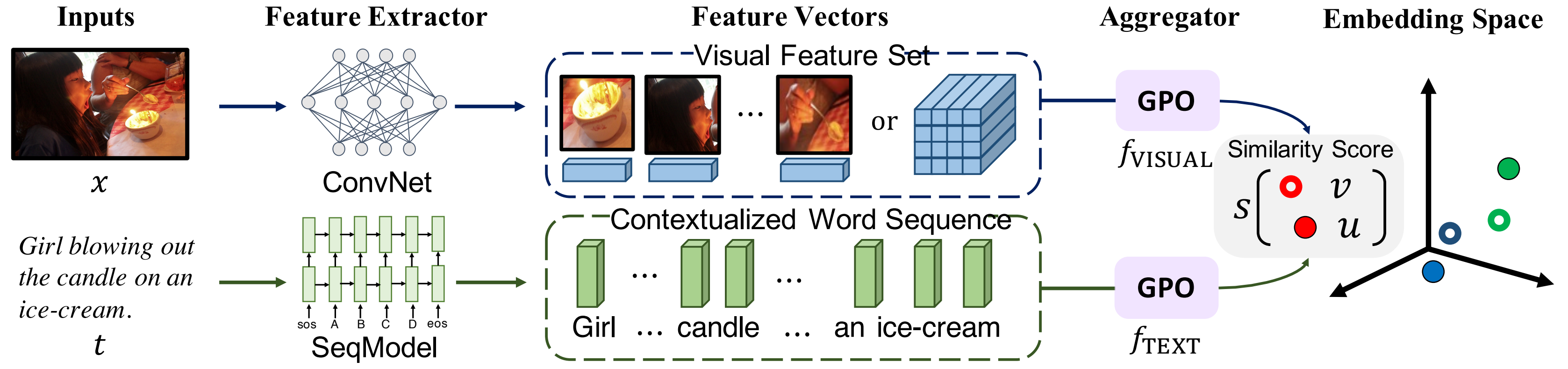
- preserve prior information such as relative distance, ordinal relation
- Better generalization to unseen length

4. Better Generalization



During training, we randomly drop 20% of the inputs vectors to the pooling operator to perturb the size of the feature set

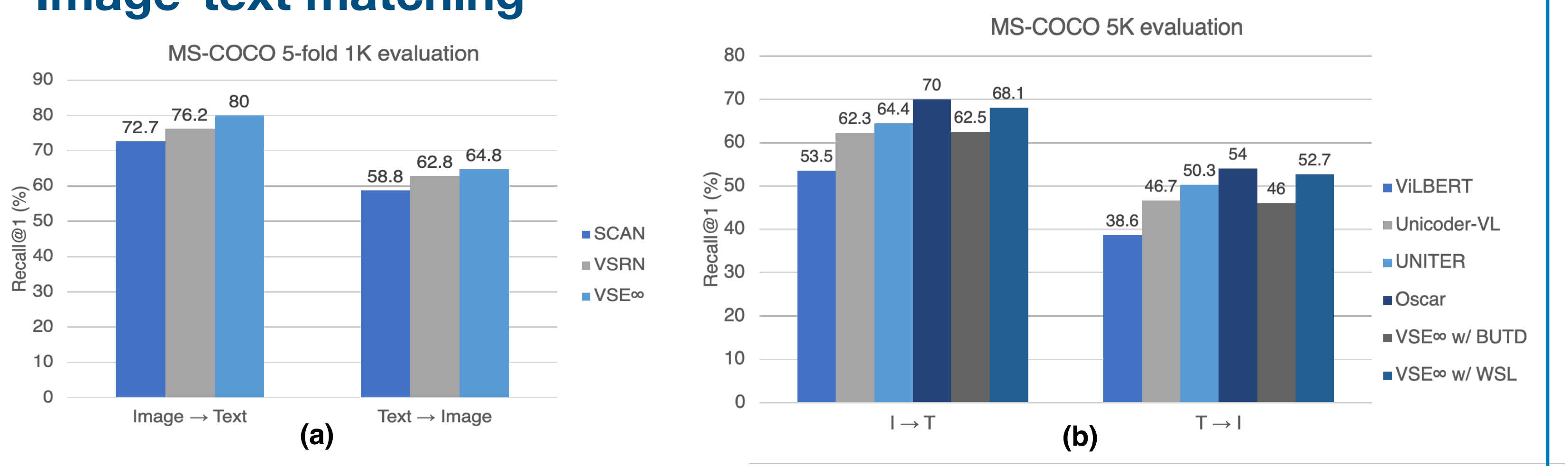
Build up VSE∞ with GPO

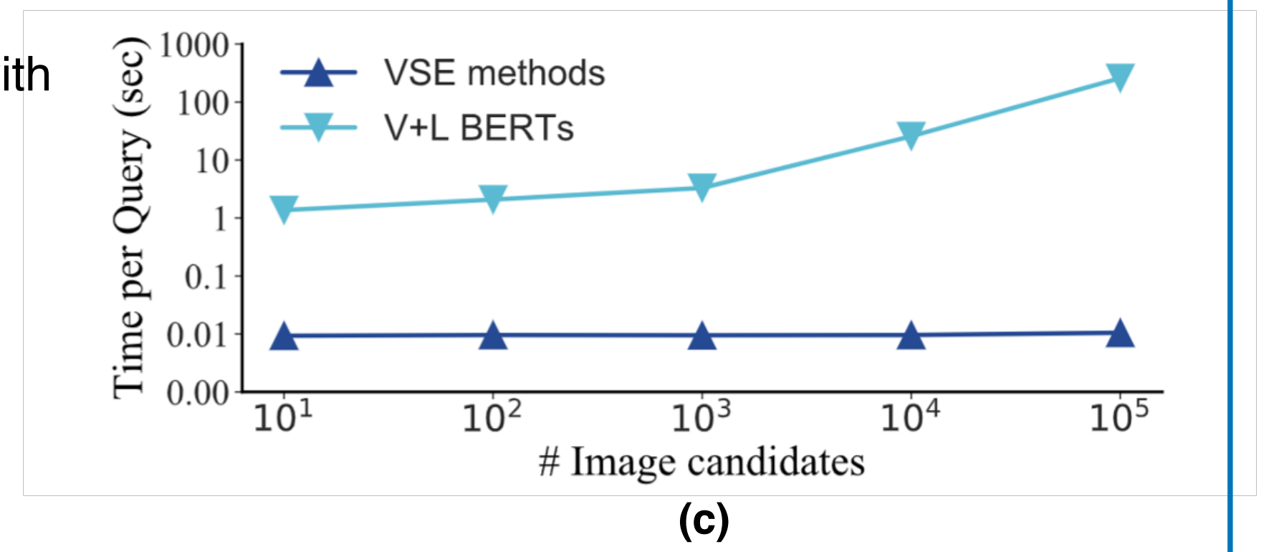


- Follow the standard VSE++ training framework to train the model (Faghri et al. 2017);
- Using the GPO as the default plug-and-play feature aggregator for all model branches

Experiments

Image-text matching



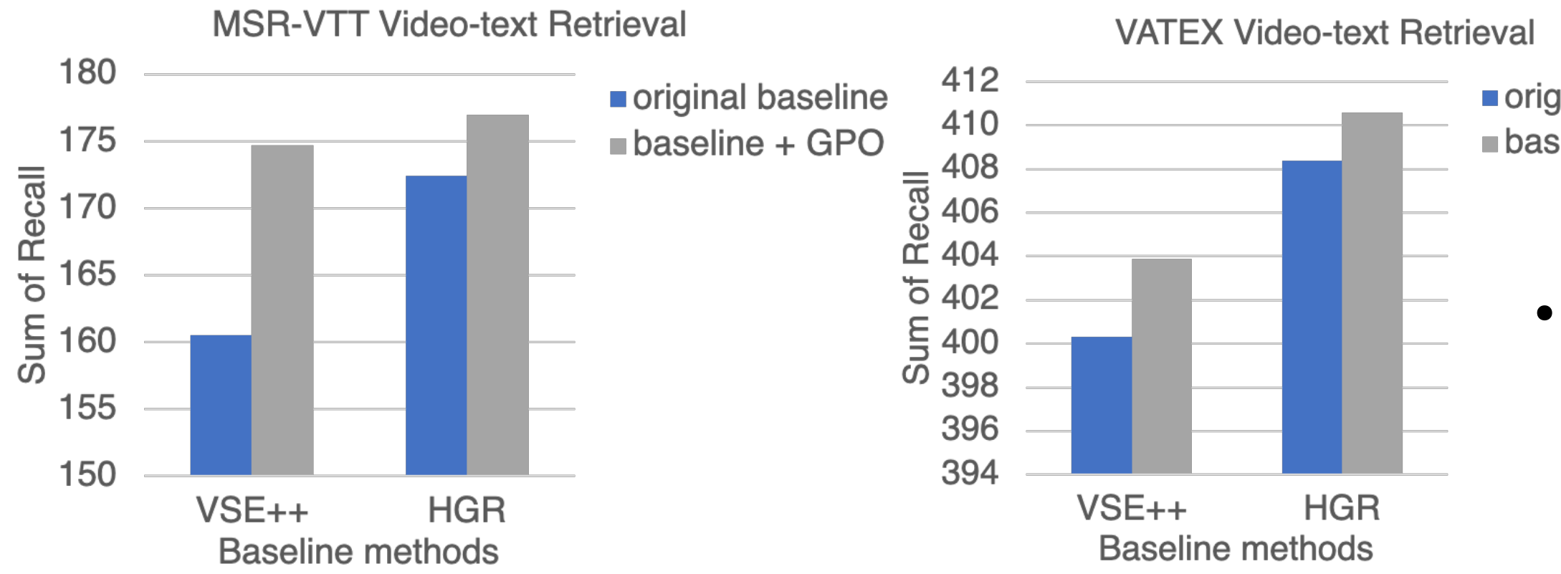


(a). Compare with VSE-based method or method with simple cross-modal interaction

(b). Compare with methods based on Vision-Language BERT

(c). Efficiency simulation on image retrieval task for VSE-based and V+L-BERT-based methods

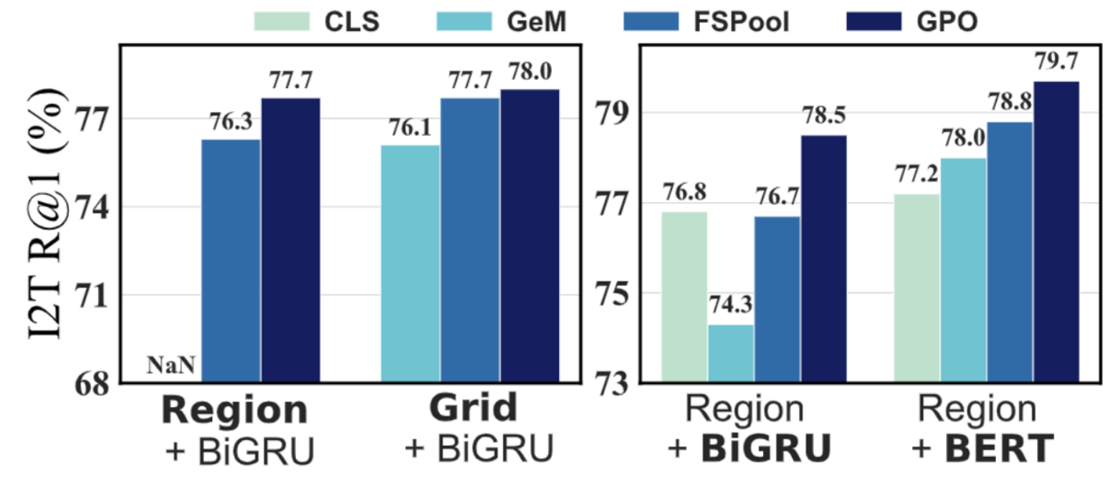
Video-text matching



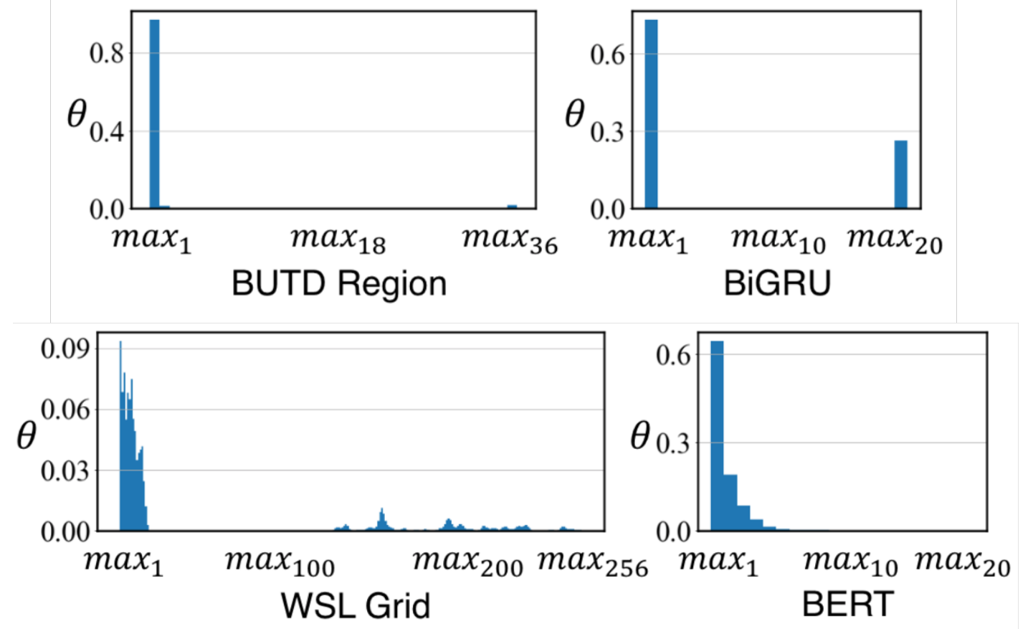
MSR-VTT: Xu *et al.* ICCV 2016
VATEX: Wang *et al.* ICCV 2019
VSE++: Faghri *et al.* BMVC 2018
HGR: Chen *et al.* CVPR 2020

- Apply GPO to existing video-text matching methods for frame-level feature aggregation and text feature aggregation

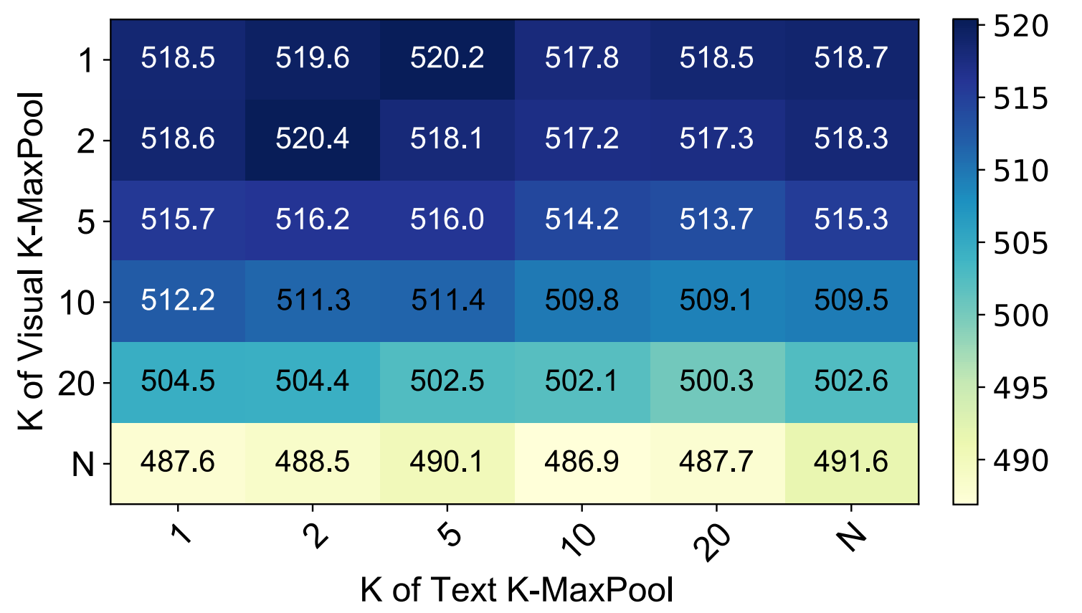
More Analyses



(a). Left: different visual aggregators, AvgPool as text aggregator for BiGRU; **Right**: different text aggregators, GPO as the visual aggregator for Region feature



(b). Visualizing pooling coefficients



K of Visual K-MaxPool \ K of Text K-MaxPool	1	2	5	10	20	N
1	518.5	519.6	520.2	517.8	518.5	518.7
2	518.6	520.4	518.1	517.2	517.3	518.3
5	515.7	516.2	516.0	514.2	513.7	515.3
10	512.2	511.3	511.4	509.8	509.1	509.5
20	504.5	504.4	502.5	502.1	500.3	502.6
N	487.6	488.5	490.1	486.9	487.7	491.6

Sum of recall of VSE∞ : 520.8

(c). Compare VSE∞ with the results of manual grid search (features: BUTD Region for image, BiGRU for text)

(a). Compare GPO with **other learnable pooling operators** from literature, on various combinations of image and text feature extractors. GeM (Radenovic *et al.* 2017), FSPool (Zhang *et al.* 2020).

(b).Visualize the **learned pooling coefficients** on different combinations of feature extractors. The results are consistent with the initial empirical finding

(c). The manual grid search over two features: the search complexity is O(N*N), N is the number of possible pooling functions. By using GPO as the default feature aggregator, **the manual search with repetitive experiments can be effectively eliminated**