Progressive Spatio-Temporal Prototype Matching for Text-Video Retrieval (Supplementary Material)

Pandeng Li[1], Chen-Wei Xie[2], Liming Zhao[2], Hongtao Xie[1], Jiannan Ge[1],
Yun Zheng[2], Deli Zhao[2], Yongdong Zhang[1]
[1] University of Science and Technology of China, [2]DAMO Academy, Alibaba Group
{lpd, gejn}@mail.ustc.edu.cn, {htxie, zhyd73}@ustc.edu.cn
{eniac.xcw, lingchen.zlm, zhengyun.zy}@alibaba-inc.com, zhaodeli@gmail.com

This supplementary material provides more details of the Progressive Spatio-Temporal Prototype Matching (ProST) framework: 1) the time efficiency of ProST; 2) the pseudo code of ProST; 3) more experiment results; 4) limitation.

1. Time Efficiency of ProST

Training time. We train our model and TS2-Net [9] on the Pytorch framework [12]. Tab. 1 shows the training time of ProST. Compared to TS2-Net, ProST reduces training time by about 23.4% and 31.2% on MSRVTT-9k [16] and DiDeMo [6]. Because ProST does not need the token selection module in TS2-Net, which may take up additional training time. More importantly, our similarity calculation within the training batch may be much faster than TS2-Net, which can be seen from the testing time experiment.

Testing time. The testing efficiency is crucial to evaluate the retrieval system. We test all models with one NVIDIA Tesla A100 GPU. Tab. 1 shows the testing time cost of ProST and TS2-Net. In the feature extraction stage, TS2-Net and ProST have comparable time performance. In the similarity search stage, ProST has only two time-consuming matrix multiplication calculations, and does not require frame-level weight prediction. Therefore, ProST reduces the search time by 7-8 times compared to TS2-Net on MSRVTT-9k and DiDeMo.

2. Pseudo Code of ProST

Algorithm 1 provides the pseudo-code of Progressive Spatio-Temporal Prototype Matching in a PyTorch-like style. We decompose the vanilla matching process into two spatio-temporal complementary parts: 1) Object-Phrase Prototype Matching aligns the visual object prototypes and text phrase prototypes generated by Spatial Prototype Generation to emphasize **fine-grained spatial information**; 2) Event-Sentence Prototype Matching exploits event prototypes progressively generated by Temporal Prototype Generation to learn **dynamic semantic alignment**, which explores intrinsic one-to-many video-text relations.

3. More Experiment Results

Experiments on YouCook2 [18]. We choose YouCook2 for performance evaluation, which has rich spatio-temporal details. Tab.2 shows that ProST outperforms recent methods [10], especially in R@5 (23.3→30.2). This proves that spatio-temporal matching leads to more growth on datasets with rich spatio-temporal details.

Post-processing results. The hubness phenomenon [14] is that some points are the nearest neighbors of most points in high-dimensional embedding space, which is harmful for the retrieval performance. To deal with this problem, CAMoE [3] and QB-Norm [1] utilize inverted softmax and query-bank normalization with dynamic inverted softmax for the post-processing of the similarity score matrix, respectively. In Tab. 3, we compare the results with the basic

Table 1. The training and testing time of TS2-Net [9] and ProST on MSRVTT-9k and DiDeMo.

Method	R@1 (Text → Video)		Training time		Testing time			
					Feature extraction		Similarity search	
	MSRVTT-9k	DiDeMo	MSRVTT-9k	DiDeMo	MSRVTT-9k	DiDeMo	MSRVTT-9k	DiDeMo
TS2-Net [9]	47.0	41.8	8.1h	1.6h	47.42s	134.67s	3.54s	3.91s
ProST	48.2	44.9	6.2h	1.1h	47.88s	134.22s	0.48s	0.49s

Algorithm 1: Pseudo code of Progressive Spatio-Temporal Prototype Matching in a PyTorch-like style.

```
# obj_p:  object prototypes           phr_p:  phrase prototypes
# eve_p:  event prototypes            sen_p:  sentence prototype
def matching(self, obj_p, phr_p, eve_p, sen_p):
   # normalize representation
   obj_p = F.normalize(obj_p, p=2, dim=-1)  # B × L × N_o × D
   phr_p = F.normalize(phr_p, p=2, dim=-1)  # B × N_p × D
   eve_p = F.normalize(eve_p, p=2, dim=-1)  # B × N_e × D
   sen_p = F.normalize(sen_p, p=2, dim=-1)  # B × D

   # Object-Phrase Prototype Matching
   op_logits = torch.einsum("apd,blod->ablpo", [phr_p, obj_p]) # B × B × L ×
   N_p × N_o
   op_logits = op_logits.max(3)[0] # B × B × L × N_o
   op_logits = op_logits.max(2)[0] # B × B × N_o
   op_logits = op_logits.sum(2) / self.obj_num # B × B

   # Event-Sentence Prototype Matching
   es_logits = torch.einsum("ad,bed->abe", [sen_p, eve_p]) # B × B × N_e
   es_logits = es_logits.max(2)[0] # B × B

return op_logits, es_logits
```

Table 2. Text-to-Video retrieval results on the YouCook2 dataset.

Method	Text → Video				
	R@1 ↑	R@5 ↑	R@10↑	MdR ↓	MnR ↓
TACo [17]	4.9	14.7	22.0	68.0	-
COOT [4]	5.9	16.7	24.8	49.7	-
CLIP4Clip [4]	8.3	23.3	33.4	26.0	134.3
TS2-Net [4]	10.2	29.1	39.0	18.0	120.4
ProST [4]	11.4	30.2	41.7	17.0	116.8

Table 3. Text-to-Video R@1 results with the post-process methods. $*$ refers the inverted softmax [3] and ‡ refers our Text-Video Hungarian (TVH) post-processing strategy.

Method	Text → Video			
	MSRVTT-9k	DiDeMo	VATEX	LSMDC
QB-Norm [1]	47.2	43.5	58.8	-
TS2-Net [9]	47.0	41.8	59.1	23.0
TS2-Net*	49.6	47.0	60.2	23.8
TS2-Net‡	51.3	48.8	67.4	23.6
ProST	48.2	44.9	60.6	24.1
ProST*	49.9	48.2	61.4	24.5
ProST‡	**52.4**	**52.1**	**69.1**	**24.6**

Table 4. The ablation study on MSRVTT-9k to investigate the configuration of the layer number N_{fl} of the frame decoder and the layer number N_{el} of the event decoder.

$\{N_{fl}, N_{el}\}$	Text → Video				
	R@1 ↑	R@5 ↑	R@10↑	MdR ↓	MnR ↓
{1, 1}	46.3	72.8	82.0	**2.0**	13.2
{1, 2}	47.0	73.1	82.9	**2.0**	13.0
{1, 3}	47.4	73.5	82.6	**2.0**	12.8
{2, 1}	47.7	73.7	83.0	**2.0**	12.6
{2, 2}	**48.2**	74.6	**83.4**	**2.0**	12.4
{2, 3}	48.0	**74.8**	83.3	**2.0**	**12.3**
{3, 1}	47.1	73.3	83.1	**2.0**	12.8
{3, 2}	48.1	74.4	83.2	**2.0**	12.4
{3, 3}	47.8	74.0	82.9	**2.0**	12.8

inverted softmax. Note that in the previous experiments of the main manuscript, we did not use any post-processing techniques to ensure fairness. Then, we introduce **a very simple post-processing strategy (TVH)** on the text-video retrieval task for the first time. The post-processing of the similarity score matrix is defined as a bipartite maximum matching problem, which can be solved by the Hungarian algorithm [8]. The TVH strategy enables each text query to find the corresponding unique video and the total similarity score is maximized.

Tab. 3 shows that TVH outperforms the existing inverted softmax, especially on DiDeMo and VATEX. Our ProST‡ also achieves better results, reaching 52.4%, 52.1%, 69.1%, 24.6% R@1 on MSRVTT-9k, DiDeMo, VATEX and LSMDC. In particular, ProST‡ has the largest improvement on VATEX, while the improvement on LSMDC is moderate. This may be due to the high difficulty of the LSMDC dataset, resulting in a large deviation between the current similarity ranking and the correct ranking. It is dif-

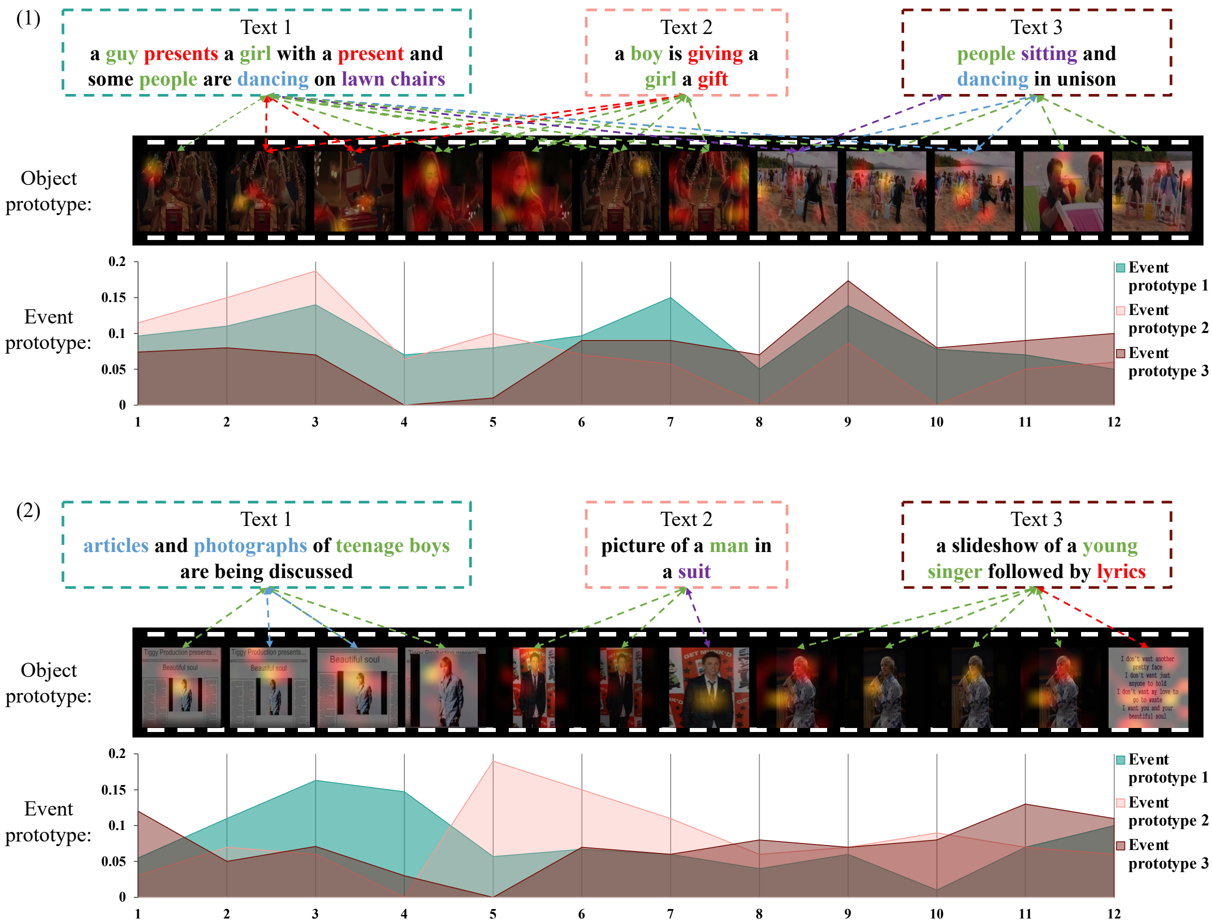

Figure 1. More visualization results of the object and event prototypes. We sample 12 frames in the video and object prototypes are shown as highlighted response regions in the frame. Then, we show cross-attention event weights in a line graph. Best viewed in color.

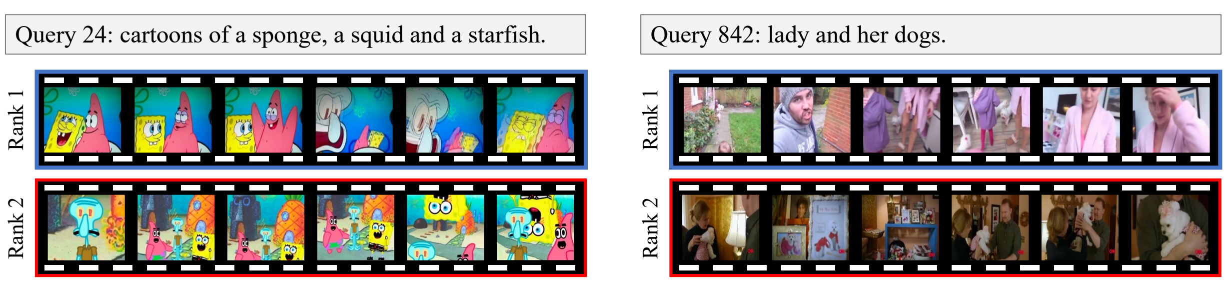

Figure 2. Some failure text-video retrieval examples. We rank the retrieval results based on their similarity scores. Red box: the correctly retrieved groundtruth video. Blue box: the incorrectly retrieved video by our model.

ficult to improve with simple post-processing.

Ablation study. We conduct experiments on various decoder layer configurations in Tab. 4. When the number of decoder layers is configured as $\{1, 1\}$, the effect of the model is poor. Other configurations result in good performance. This may be because a single-layer transformer is not enough to model complex spatio-temporal relations.

More visualization examples. As shown in Fig. 1, we show more visualization results of object prototypes and event prototypes. This further illustrates that ProST can achieve good spatial local alignments and temporal dynamic event semantic alignments.

Fig. 2 displays two cases where our model fails to rank the groundtruth video at the top. Nevertheless, we argue that ProST may have retrieved the more relevant video in these failure cases. For instance, for query 24, we retrieved the cartoon about sponges, octopuses, and starfish at rank 1. However, this case is judged as the retrieval failure. For query 842, the text description "dogs" refers to more than one dog, and the videos we searched indeed match the text description. However, the groundtruth video has only one identical dog. We think that ProST may have the potential to achieve better results after improving these non-discriminative text descriptions.

4. Limitation

Similar to existing text-video retrieval methods [9, 5, 15, 2], our method is suitable for fine-grained ranking rather than large-scale ranking. To pursue large-scale retrieval, we can use the existing global embedding methods [10, 13] combined with indexing algorithms [7, 11] for text-video matching in the coarse ranking phase. Then, we utilize ProST to perform further spatio-temporal matching on the coarsely ranked Top-N instances in the fine-grained ranking phase.

References

[1] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation. In *CVPR*, 2022. 1, 2

[2] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, 2020. 4

[3] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 1, 2

[4] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. In *NeurIPS*, 2020. 2

[5] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *CVPR*, 2022. 4

[6] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1

[7] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 2011. 4

[8] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 2

[9] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. Ts2-net: Token shift and selection transformer for text-video retrieval. In *ECCV*, 2022. 1, 2, 4

[10] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 2022. 1, 4

[11] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *TPAMI*, 2020. 4

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 1

[13] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021. 4

[14] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *JMLR*, 2010. 1

[15] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In *CVPR*, 2021. 4

[16] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1

[17] Jianwei Yang et al. Taco: Token-aware cascade contrastive learning for video-text alignment. In *ICCV*, 2021. 2

[18] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 1