Supplemental Material for Efficient Transfer Learning for Video-language Foundation Models

A. Limitation and Broader Impact

Limitation. Although our method achieves excellent results across various video recognition settings, there are still areas for improvement. For example, MSTA connects the gradients of visual and textual features. Despite using gradient scaling to prevent instability, we believe that the backpropagation performed under this setup is not optimal. Therefore, exploring better ways to train this type of module could further enhance MSTA's performance. On the other hand, using spatiotemporal descriptions generated by LLMs for constraints may lead to performance degradation due to the lower quality of these descriptions. In future work, we will further investigate how to leverage LLMs to improve the efficient transfer of video-language models, aiming to explore the generality and versatility of our approach.

Broader Impact. The adaptability of foundational models to various downstream tasks has become a major focus in the current field of machine learning [3, 5, 7]. We believe that retaining the existing capabilities of foundational models while integrating new knowledge is an important issue worth exploring in depth. We hope this research provides new insights and approaches for the broader and long-term application of foundational models. Our work focuses on video recognition tasks, which have widespread applications in real-world scenarios such as surveillance. However, before these technologies can be deployed, it is crucial to carefully address potential issues related to privacy protection and rights.

B. More Implementation Details

B.1. Datasets

Kinetics-400 [1] is a large-scale video dataset containing approximately 240,000 training videos and 20,000 validation videos across 400 human action categories, with an average video length of 10 seconds. Due to its high quality, it has become one of the most popular benchmarks for video recognition.

Kinetics-600 [2] is an extension of Kinetics-400, with approximately 392,000 training videos, 30,000 validation videos, and 60,000 test videos in 600 human action categories. It introduces 220 new action categories not present in Kinetics-400. We evaluate the zero-shot performance on these 220 new categories and use three splits provided by previous work [4]. The test set is used for evaluation, and we report the average performance across the three splits.

UCF-101 [8]is an action recognition dataset containing 13,320 videos in 101 action categories, collected from YouTube. It has three official splits for training and validation data.

HMDB-51 [6] contains 7,000 videos across 51 action categories, collected from movie clips and web videos. The dataset has three official splits, with 3,570 training videos and 1,530 validation videos in each split.

Something-Something V2 [4] is a temporally demanding dataset that requires fine-grained temporal understanding. It includes 220,000 videos across 174 action categories.

C. Additional Ablations

Impact of different descriptions. We investigate the impact of spatiotemporal descriptions on the performance of our proposed method. The results presented in Table S1 show that each description complements the others, highlighting the importance of both spatiotemporal descriptions for recognition tasks. Additionally, we observe that the temporal descriptions have a more pronounced effect on performance compared to the spatiotemporal descriptions.

Table S1. Study on different descriptions.

Description	Base	Novel	H
w/o DES	68.2	53.2	59.77
w/o Spatio	68.4	53.4	59.97
w/o Temporal	**68.8**	53.3	60.06
Full	68.6	**53.5**	**60.12**

Cost Analysis. We analyze the additional costs of our method during training and inference in Table S2. Latency is measured in our baseline training setup, while throughput is evaluated using the largest batch size that can be processed without exceeding the memory limit of a single NVIDIA A100-80G. Our pipeline incurs only an additional 0.1× training time and results in a throughput reduction of approximately 5%, which is acceptable considering the performance gains achieved.

Table S2. Additional cost analysis of our method, we report step latency during training, and throughput (TP) during inference. We refer to Top-1 as zero-shot accuracy on Kinetics-600.

Description	Top-1 (%)	Latency (s)	TP (video/s)
ViCLIP [9]	71.5	**0.45** (1.0x)	**69.3**(1.0x)
Ours	**74.5**	0.51 (1.1x)	65.7 (0.95x)

D. Pseudo-code

The pseudo-code implementation of MSTA is provided below, along with a concise definition of MSTA and its insertion method.

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In CVPR, pages 4724–4733, 2017. 1

[2] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics600. arXiv preprint arXiv:1808.01340, 2018. 1

[3] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In NeurIPS, 2022. 1

[4] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The ”something something” video database for learning and evaluating visual common sense. In ICCV, pages 5843–5851, 2017. 1

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In ICLR, 2022. 1

[6] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In ICCV, pages 2556–2563, 2011. 1

[7] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. In NeurIPS, 2022. 1

[8] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012. 1

[9] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In ICLR, 2024. 2

Algorithm 1 Implementation of Text Adapter & Shared Adapter in PyTorch-like style

```
def _build_adapter(d_model, n_layers, l_start, l_end, mid_dim, dropout_rate=0.1):
    adapter = [None] * (n_layers + 1)
    for i in range(l_start, l_end+1):
        if mid_dim == d_model:
            adapter[i] = nn.Sequential(
                nn.Linear(d_model, mid_dim),
                nn.ReLU()
                )
        else:
            adapter[i] = nn.Sequential(OrderedDict([
                (”down”, nn.Sequential(
                nn.Linear(d_model, mid_dim),
                nn.ReLU()
                )),
                (”dropout”, nn.Dropout(p=dropout_rate)),
                (”up”, nn.Linear(mid_dim, d_model))
                ]))
    adapter = nn.ModuleList([a for a in adapter])
    return adapter
```

Algorithm 2 Implementation of Visual Adapter in PyTorch-like style

```
def _build_visual_adapter(d_model, n_layers, l_start, l_end, mid_dim, dropout_rate=0.1):
    adapter = [None] * (n_layers + 1)
        for i in range(l_start, l_end+1):
            adapter[i] = nn.Sequential(OrderedDict([
            (”down”, nn.Sequential(
            nn.Linear(d_model, mid_dim),
            nn.ReLU()
            )),
            (”dropout”, nn.Dropout(p=dropout_rate)),
            (”up”, nn.Linear(mid_dim, d_model)),
            (”temporal_up”, nn.Conv3d(mid_dim, d_model, kernel_size=(3,1,1), stride=(1, 1, 1),
            padding=”same”, groups=mid_dim, dilation=(1, 1, 1))),
            ]))
    adapter = nn.ModuleList([a for a in adapter])
    return adapter
```