Leveraging Temporal Contextualization for Video Action Recognition
———Supplementary Material———

We provide additional experimental analyses and details in the following order:

- Appendix A: Fine-tuning with the Kinetics-400 pretrained model
- Appendix B: More ablation study on VP
- Appendix C: Scalability with ViT-L/14
- Appendix D: Temporal subset analysis
- Appendix E: Impact of weight-space ensembling
- Appendix F: More visualizations of context tokens and attentions
- Appendix G: Datasets and implementation details

A Fine-tuning with the Kinetics-400 Pretrained Model

Table 11: Comparison with state-of-the-arts on few-shot action recognition using Kinetics-400 pretrained model. All the models are first pretrained on Kinetics-400 and subsequently fine-tuned on each dataset.

	HMDB-51				UCF-101				SSv2				All
Method	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=16	Avg.
ActionCLIP [15]	54.3	56.2	59.3	66.1	76.7	80.4	87.6	91.8	4.8	6.9	9.1	12.3	50.5
A5 [6]	46.7	50.4	61.3	65.8	76.3	84.4	90.7	93.0	4.5	6.7	7.2	9.5	49.7
X-CLIP [11]	60.5	66.8	69.3	71.7	89.0	91.4	94.7	96.3	6.6	7.8	9.9	13.7	56.5
ViFi-CLIP [12]	63.0	65.1	69.6	72.0	91.0	93.7	95.0	96.4	6.7	7.9	10.2	13.5	57.0
TC-CLIP (Ours)	**65.3**	**68.5**	**71.4**	**73.0**	**94.1**	**95.6**	**96.6**	**97.3**	**8.7**	**10.1**	**12.1**	**15.2**	**59.0**

Table 12: Comparison with state-of-the-arts on base-to-novel generalization using Kinetics-400 pretrained model. All the models are first pretrained on Kinetics-400 and subsequently fine-tuned on each dataset.

	HMDB-51			UCF-101			SSv2			All (Avg.)		
Method	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
ActionCLIP [15]	69.0	57.2	62.6	85.6	75.3	80.1	8.1	8.7	8.4	54.2	47.1	50.4
A5 [6]	70.4	51.7	59.6	95.8	71.0	81.6	12.9	5.7	7.9	59.7	42.8	49.9
X-CLIP [11]	75.8	52.0	61.7	95.4	74.0	83.4	14.2	11.0	12.4	61.8	45.7	52.5
ViFi-CLIP [12]	77.1	54.9	64.1	95.9	74.1	83.6	15.8	11.5	13.3	62.9	46.8	53.7
TC-CLIP (Ours)	**79.4**	**58.3**	**67.2**	**97.5**	**84.5**	**90.5**	**19.6**	**15.6**	**17.4**	**65.5**	**52.8**	**58.5**

Tables 11 and 12 present the comparison results using the K-400 pretrained model on the few-shot and base-to-novel settings. All the models are first pretrained on the Kinetics-400 dataset and subsequently fine-tuned on each dataset. TC-CLIP demonstrates superior performance over all other methods by significant margins. Particularly in the base-to-novel setup, TC-CLIP outperforms ViFi-CLIP [12] with notable gaps of 3.5%p, 6%p, and 4.8%p in the base, novel, and harmonic mean (HM) on average, respectively.

Table 13: Video-conditional Prompting (VP) ablation. We report K-shot training results where the top-1 accuracy in each dataset is averaged over $K = 2, 4, 8, 16$. Default settings are marked in gray.

(a) Number of prompt vectors.

Case	HMDB	UCF	SSv2	All
2	62.0	90.1	9.4	53.8
4	**63.9**	**90.7**	**9.8**	**54.8**
8	63.7	90.4	**9.8**	54.7

(b) Vision input token selection.

Case	HMDB	UCF	SSv2	All
CLS tokens from all frames	63.3	90.3	**9.8**	54.4
GAP tokens from all frames	63.7	90.5	**9.8**	54.7
Context tokens	**63.9**	**90.7**	**9.8**	**54.8**

(c) Input layer selection.

L_{text}	L_{vision}	HMDB	UCF	SSv2	All
1	1	62.3	89.7	9.1	53.7
1	12	62.0	89.9	9.3	53.7
6	6	62.8	89.5	9.7	54.0
6	12	62.5	90.2	9.7	54.1
12	12	**63.9**	**90.7**	**9.8**	**54.8**

(d) Layer and prompt initialization.

Layer init.	Prompt init.	HMDB	UCF	SSv2	All
CLIP weight	"`a photo of a`"	**63.9**	**90.7**	9.8	**54.8**
Random init.	"`a photo of a`"	63.5	90.5	**9.9**	54.6
CLIP weight	Random init.	62.5	90.2	9.5	54.1

Table 14: Computational cost–performance trade-off of VP design. In the case of vision-text late-fusion design, class name embeddings are pre-computed. Models are evaluated on the zero-shot setting without the weight ensemble. Costs are measured using a single A6000 GPU.

Case	K-600 Top-1	GFLOPs Vision	GFLOPs Text	GFLOPs Cross-modal	Latency (s) Training	Latency (s) Inference
Vision-text late-fusion	70.9	301	2.96	0.06	0.54 (1.00×)	0.042 (1.00×)
Video-conditional Prompting (VP)	72.7 (+1.8)	301	2.96	0.06	0.58 (1.07×)	0.045 (1.07×)

B More Ablation Study on VP

Table 13 examines the design choice of VP in TC-CLIP on the few-shot setting.

Number of prompt vectors. Increasing the number of prompt vectors does not necessarily improve performance. 4 prompt vectors are employed by default.

Vision token selection. Using context tokens in VP yields better results than employing `[CLS]` tokens or global average pooled (GAP) tokens from all frames. This demonstrates that proper contextualization of vision features is essential to transfer the video information to the text side.

Input layer selection. We vary the layer indices of the text and vision inputs $\{L_{\text{text}}, L_{\text{vision}}\}$ in the VP module $f_{\theta_{\text{VP}}}(\mathbf{p}^{L_{\text{text}}-1}, \mathbf{s}_{\text{proj}}^{L_{\text{vision}}})$. We observe that conditional prompting at the early stage ($L_{\text{text}} = 1$) does not generalize well, regardless of the vision layer index. The early-stage prompting design is hard to generalize in a full fine-tuning scenario, possibly because CLIP was initially trained in a vision-text late-alignment fashion. Consequently, we choose the late-stage prompting by adopting the last layers for both modalities.

Layer and prompt initialization. We initialize the VP module's weight using the weight from the last layer of the CLIP text encoder because random initialization often results in unstable training results in the few-shot scenario. Simi-

Table 15: Comparison with state-of-the-arts on zero-shot action recognition using ViT-L/14. All the models are trained on Kinetics-400 and directly evaluated on other datasets. † denotes that the results are reproduced with our implementation. The best results are in **bold-faced** numbers, and the second-best ones are underlined.

Method	WE	HMDB-51	UCF-101	K600 (Top-1)	K600 (Top-5)	All (Top-1)
ViFi-CLIP [12]†		55.6 ± 0.5	86.1 ± 0.8	77.8 ± 0.9	95.6 ± 0.2	73.2
TC-CLIP (Ours)		**56.1** ± 0.3	**86.9** ± 0.9	**80.1** ± 0.7	**96.5** ± 0.1	**74.4**
ViFi-CLIP [12]†	✓	55.8 ± 0.7	88.1 ± 1.3	81.1 ± 0.7	96.7 ± 0.1	75.0
Open-VCLIP [17]	✓	**59.0** ± 0.6	87.6 ± 1.2	81.1 ± 0.8	96.3 ± 0.3	75.9
TC-CLIP (Ours)	✓	57.1 ± 0.7	**88.9** ± 0.9	**83.1** ± 0.7	**97.3** ± 0.1	**76.4**

Table 16: Temporal subset analysis using the temporal subset [13] on Kinetics-400 and SSv2. Gains over ViFi-CLIP are indicated in green.

	K-400 fully-supervised		SSv2 16-shot	
Method	All	Temporal	All	Temporal
ViFi-CLIP [12]	83.9	87.8	12.4	25.9
TC-CLIP (Ours)	85.2 (+1.3)	89.2 (+1.4)	14.0 (+1.6)	29.9 (+4.0)

larly, it is beneficial to initialize the learnable prompt vectors using the prompt template "`a photo of a`" following several prompt tuning methods [9, 18].

Computational cost analysis. Although VP requires the instance-conditional computation of text embeddings, the added cost is minor. As in Table 14, for a pair of video and text inputs, the GFLOPs required by the text encoder cost only about 1% of those needed by the vision encoder. Given these minimal text-related costs, VP adds only an extra 0.07× in latency compared to the vision-text late-fusion design using pre-computed text embeddings. Considering the observed performance gain, this is an acceptable trade-off.

C Scalability with ViT-L/14

Table 15 shows the zero-shot performance comparison using CLIP ViT-L/14 as a backbone. In the case of using WE, our model outperforms ViFi-CLIP [12] and Open-VCLIP [17] by 1.4%p and 0.5%p on average, respectively.

D Temporal Subset Analysis

We adopt the temporal subset analysis suggested in [13] to further analyze the temporal modeling ability of trained models. The temporal subset consists of several action classes that require more temporal information to recognize them, *i.e.*, the classes of videos that cannot be recognized by human annotators after randomly shuffling the frames. As shown in Table 16, TC-CLIP's gains over ViFi-CLIP [12] on the temporal subsets are more substantial than the gains when evaluated on the full validation splits, demonstrating the superiority in handling temporal information.

E Impact of Weight-space Ensembling

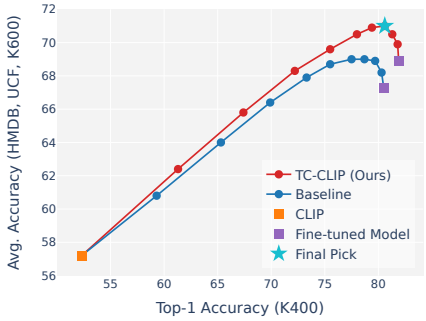

Fig. 8: Weight averaging ablation.

In Fig. 8, we evaluate the effectiveness of weight ensembling by varying the ensemble ratio w from 0 to 1 with a step size of 0.1. Specifically, the backbone weights of both vision and text encoders are linearly interpolated between CLIP and fine-tuned model, *i.e.*, $\theta_w = (1-w)\cdot\theta_{\text{CLIP}} + w\cdot\theta_{\text{fine-tuned}}$. The y-axis shows the average accuracy on the zero-shot video datasets, and the x-axis means the accuracy on the fine-tuning dataset K-400. Our model achieves a better trade-off than the baseline as our curve is always on top of the baseline's curve. This demonstrates that our model takes more advantages from weight ensembling. We choose $w = 0.7$ as our final ensemble ratio.

F More Visualizations of Context Tokens and Attentions

Context token visualization. Fig. 9 visualizes the seed tokens and context tokens from the last layer of the vision encoder in TC-CLIP. The seed tokens mainly consist of patch tokens from the most informative regions in each frame, often corresponding to the foreground, such as a person, animals, hands, and objects. To visualize each context token, we colorize its corresponding source token positions using the average color of the input image patches of that region. It is noteworthy that a single context token (highlighted in red) successfully tracks and summarizes a specific object or part throughout the entire video.

Class token attention visualization. Fig. 10 visualizes the attention maps of TC-CLIP compared to ViFi-CLIP [12] using `[CLS]` token as a query in each frame. As shown in Fig. 10(a)–(b), during the action of throwing or shooting objects, TC-CLIP tends to focus more on dynamically moving parts such as hands and arms. Furthermore, as in Fig. 10(c)–(d), TC-CLIP highlights multiple objects simultaneously based on inter-object relationships. During actions like "swinging baseball bat," TC-CLIP focuses on both the bat and the baseball being struck, whereas ViFi-CLIP only highlights salient areas in individual frames. Fig. 10(e)–(f) also shows TC-CLIP's consistent attention towards objects with deformations across frames, which is more striking than ViFi-CLIP's.

Patch token attention visualization. Fig. 11 shows the attention maps of TC-CLIP compared to other temporal modeling approaches [11,16,17] by using a patch token as a query. To visualize the attention map from TC, we assign attention values of context tokens to their corresponding source patch token positions. In both examples, the token interactions of cross-frame attention [11, 16] and temporal window expansion [17] cannot reach the frames far from the

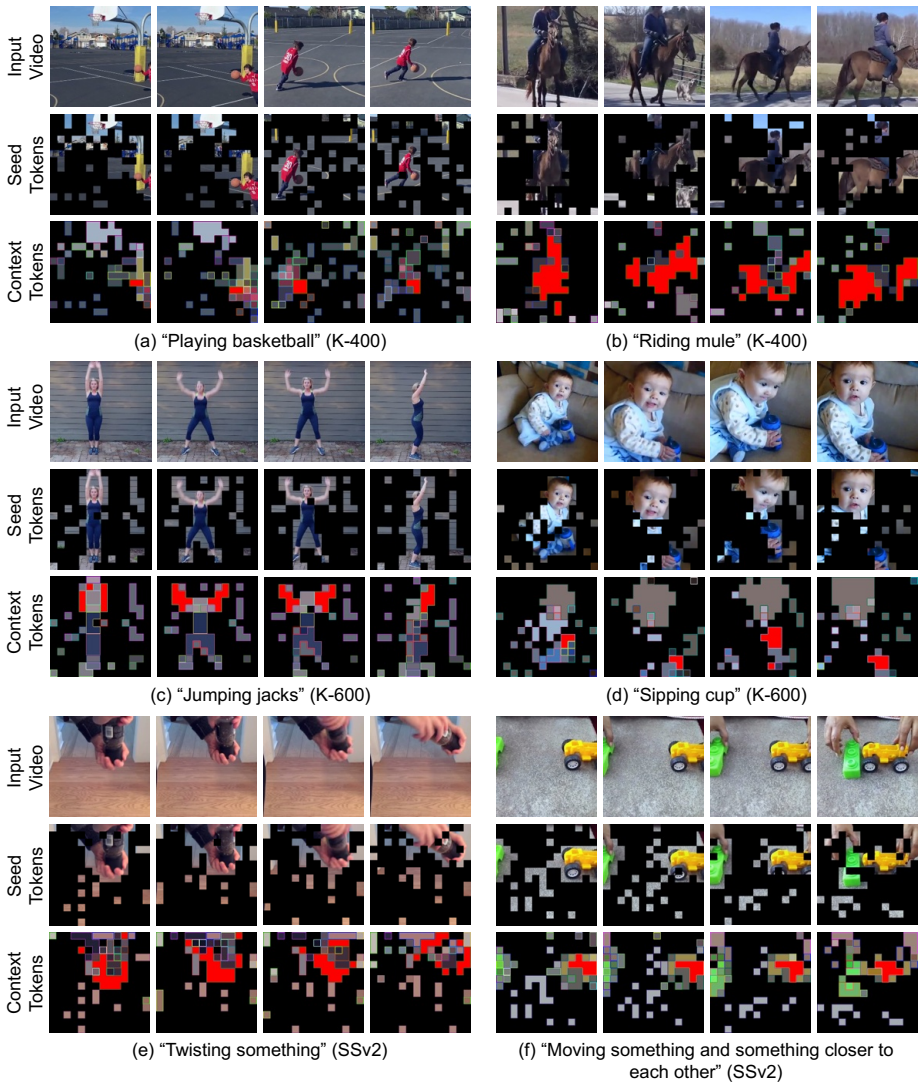

Fig. 9: Context token visualization of TC-CLIP on Kinetics-400, Kinetics-600, and SSv2 datasets. We visualize selected seed tokens and the resulting context tokens in the last layer of the vision encoder. Patch tokens with the same inner and border color are summarized into one context token. Regions highlighted in red represent a specific object or part grouped into a single context token throughout the video.

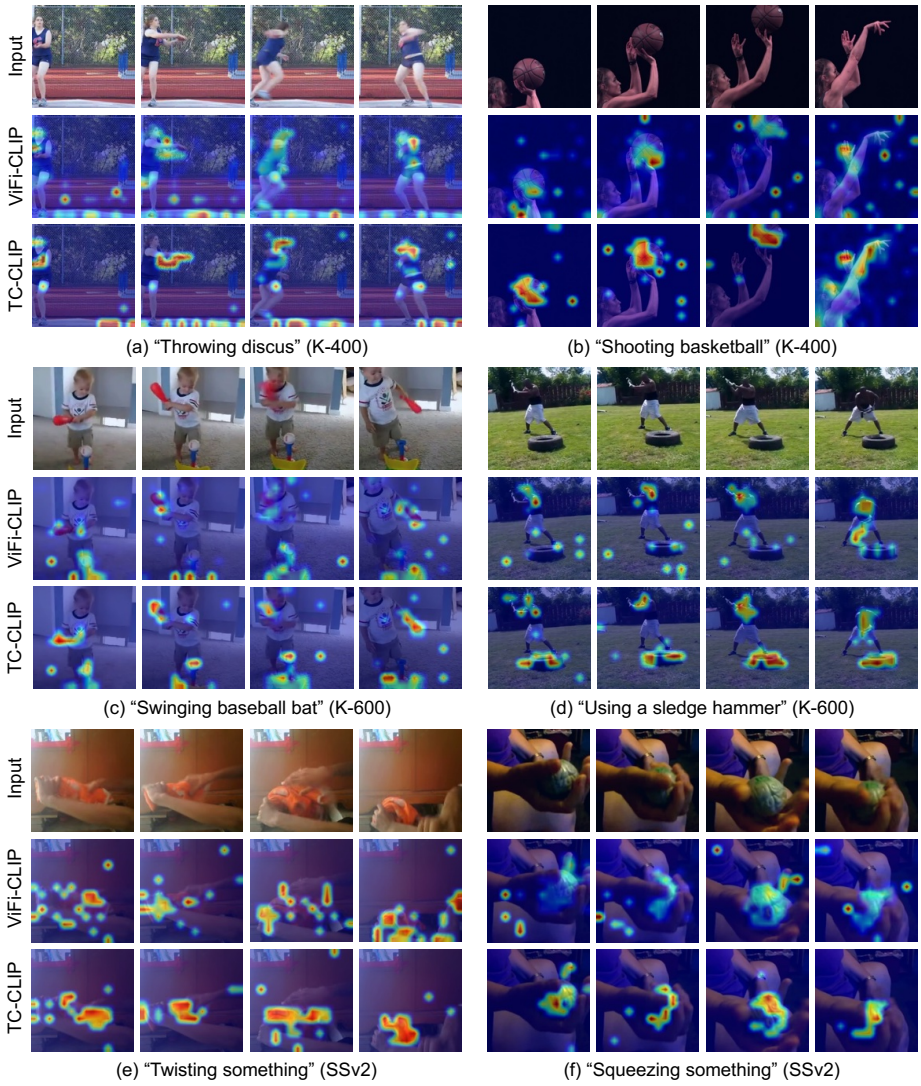

Fig. 10: Attention visualization of TC-CLIP in comparison with ViFi-CLIP [12] on Kinetics-400, Kinetics-600, and SSv2 datasets using `[CLS]` token as a query in each frame. (**a**)–(**b**): TC-CLIP tends to focus more on fast-moving parts such as hands and arms. (**c**)–(**d**): While ViFi-CLIP dominantly attends to the most salient regions, TC-CLIP attends to multiple objects based on inter-object relationships relevant to the occurring actions. (**e**)–(**f**): TC-CLIP consistently attends to the main object with deformations throughout the video.

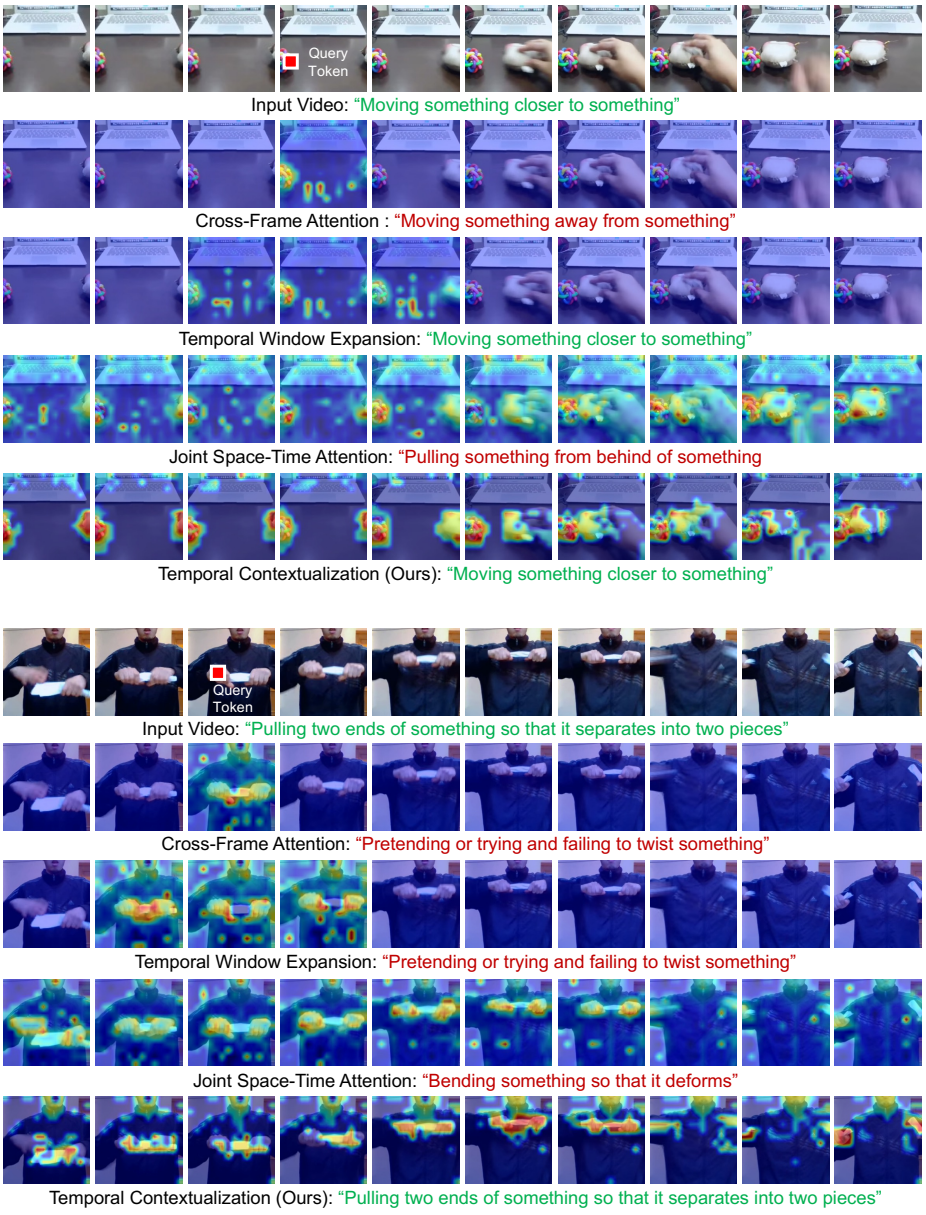

Fig. 11: Attention visualization of TC-CLIP in comparison with various temporal information learning approaches on SSv2 dataset. We visualize the attention map in the last vision encoder layer using a ball (top) and a hand (bottom) as a query (denoted with red boxes). To visualize the attention map from TC, we assign attention values of context tokens to their corresponding source patch token positions. Unlike other approaches, our method successfully highlights informative regions globally over frames.

query position, although the main action actually happens in the latter part of videos. The joint space-time attention model, on the other hand, is capable of global modeling but fails to focus on informative regions. In contrast, TC-CLIP consistently highlights the regions relevant to the query positions (*e.g.*, hands and grabbed objects) throughout the video, leading to more accurate predictions.

G Experimental Setup Details

G.1 Dataset Details

We conduct experiments over 5 action recognition benchmarks: Kinetics-400 [8] & 600 [2], HMDB-51 [10], UCF-101 [14], and Something-Something v2 (SSv2) [4].

Kinetics-400 [8] is a large-scale action recognition dataset with a total of 400 action classes, where its video clips are collected from YouTube and last for about 10 seconds. It contains around 240k training videos and 20k validation videos.

Kinetics-600 [2] is an extension of Kinetics-400 with approximately 480k video clips covering 600 action categories. The videos are divided into 390k for training, 30k for validation, and 60k for testing. We mainly adopt the validation split for zero-shot evaluation.

HMDB-51 [10] dataset includes 6,869 clips divided into 51 action categories. There are three individual splits for training and validation.

UCF-101 [14] is an action recognition dataset collected from YouTube, including 13,320 video clips with 101 action categories. Similar to HMDB-51, the training and test videos have three splits.

SSv2 [4] is a challenging dataset with 174 fine-grained action classes, which are more temporally biased than the other datasets. The standard split consists of 168,913 training videos and 24,777 validation videos.

G.2 Implementation Details

During the bipartite soft matching [1, 7], we start with the seed tokens arranged based on the `[CLS]` token attention values in each frame. These tokens are then divided into two sets by alternating positions. Subsequently, r pairs of tokens with the highest cosine similarity are merged by averaging their features, and the remaining two sets are then concatenated back together. We set r to 100 in practice. This process is repeated iteratively, employing a constant r scheduling for every iteration with an exception in the final iteration to ensure that the number of final context tokens becomes k.

During the training, we sample 16 frames to form a video clip. During the evaluation, two temporal clips with one spatial crop (2×1 view) per video are sampled to produce a prediction unless otherwise stated. The learnable prompts are initialized with the prompt "`a photo of a`" following [9, 18], and the weight of the VP module is initialized with the weight from the last layer of the CLIP

text encoder. For training recipes, we follow [12] for zero-shot, few-shot, and fully-supervised settings and follow [5] for base-to-novel generalization. By default, we use the AdamW optimizer with momentum betas of (0.9, 0.98) and a weight decay of 0.001. The VP module's initial learning rate is 10× larger than the base learning rate in each setting. Training configurations and evaluation metrics in each protocol are specified below.

Zero-shot action recognition. The models are trained on Kinetics-400 and evaluated on HMDB-51, UCF-101, and Kinetics-600 datasets. For HMDB-51 and UCF-101, we report the average and standard deviation of top-1 accuracy across three official validation splits. In the case of Kinetics-600, we apply the zero-shot evaluation protocol from [3], which exploits 220 categories of Kinetics-600 that do not appear in Kinetics-400. We use the three splits provided by [3], each containing 160 categories. The results include the average top-1 and top-5 accuracy and their respective standard deviations. During the training, the base learning rate is set to 8×10^{-6} and is decayed to 8×10^{-8} following the cosine decay scheduler. The batch size is 256, and the total number of epochs is 10, including 5 linear warmup epochs.

Few-shot action recognition. We adopt the K-shot training splits from [12] that randomly sampled $K = 2, 4, 8, 16$ videos from each class on HMDB-51, UCF-101, and SSv2. The models are evaluated using the first validation split of HMDB-51 and UCF-101 and the full validation split of SSv2. The base learning rate is set to 2×10^{-6} and is decayed to 2×10^{-8}. The batch size is 64, and the total number of epochs is set to 50, starting with 5 linear warmup epochs.

Base-to-novel generalization. We adopt the base and novel splits from [12]. The models are trained on a set of base (seen) classes in a few-shot manner and subsequently evaluated on a set of novel (unseen) classes for four datasets: Kinetics-400, HMDB-51, UCF-101, and SSv2. Each dataset comprises three training splits containing randomly sampled 16 shots of base action categories. We report the average accuracy over three splits. For HMDB-51 and UCF-101, the training and validation consider only their first split, whereas, for Kinetics and SSv2, the models are evaluated on their full validation split. The base learning rate is set to 3.33×10^{-6} and is decayed to 3.33×10^{-8}. The batch size is 64. The number of epochs is 12, including 2 warmup epochs.

Fully-supervised action recognition. The models are trained on Kinetics-400 and evaluated on its complete validation split. The base learning rate is set to 2.2×10^{-5} and is decayed to 2.2×10^{-7} following the cosine decay scheduler. The batch size is 512, and the total epochs is 30 epochs, including 5 linear warmup epochs.

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022) 8
2. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv preprint arXiv:1808.01340 (2018) 8

3. Chen, S., Huang, D.: Elaborative rehearsal for zero-shot action recognition. In: ICCV (2021) 9
4. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV (2017) 8
5. Huang, X., Zhou, H., Yao, K., Han, K.: Froster: Frozen clip is a strong teacher for open-vocabulary action recognition. In: ICLR (2024) 9
6. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: ECCV (2022) 1
7. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing (1990) 8
8. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017) 8
9. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: CVPR (2023) 3, 8
10. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV (2011) 8
11. Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., Ling, H.: Expanding language-image pretrained models for general video recognition. In: ECCV (2022) 1, 4
12. Rasheed, H., Khattak, M.U., Maaz, M., Khan, S., Khan, F.S.: Fine-tuned clip models are efficient video learners. In: CVPR (2023) 1, 3, 4, 6, 9
13. Sevilla-Lara, L., Zha, S., Yan, Z., Goswami, V., Feiszli, M., Torresani, L.: Only time can tell: Discovering temporal data for temporal modeling. In: WACV (2021) 3
14. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012) 8
15. Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021) 1
16. Wasim, S.T., Naseer, M., Khan, S., Khan, F.S., Shah, M.: Vita-clip: Video and text adaptive clip via multimodal prompting. In: CVPR (2023) 4
17. Weng, Z., Yang, X., Li, A., Wu, Z., Jiang, Y.G.: Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In: ICML (2023) 3, 4
18. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR (2022) 3, 8