PiVoT: Proactive Video Templates for Enhancing Video Task Performance

Supplementary Material

Changed	From→To	TSN [5] (%)↑		MViTv2 [4] (%)↑	
		Top-1	Top-5	Top-1	Top-5
PiVoT	-	**51.37**	**78.71**	**70.08**	**93.13**
LoRA	Backbone→Head	24.21	53.30	35.11	63.09
Template	Frame depend→Universal	32.88	61.31	57.70	85.14
	Learn→Fixed	26.19	52.01	60.12	86.11

Table 1. Ablation study of LoRA and template learning for PiVoT.

1. Additional Experiments.

LoRa in backbone vs head. The ablation study shown in Tab. 1 evaluates the effectiveness of LoRA when applied to different parts of the baseline detector. When the LoRA layers are moved from the backbone to the head, the performance of both detectors, TSN and MViTv2, significantly decreases. Specifically, for TSN, the top-1 accuracy drops to 24.21% and the top-5 accuracy to 53.30%, while MViTv2 experiences a decline to 35.11% and 63.09% for top-1 and top-5 accuracy, respectively. This indicates that the backbone plays a critical role in extracting meaningful spatial-temporal features in video detection tasks, and that adapting LoRA to the head limits its capacity to effectively leverage these features. These results highlight that LoRA's effectiveness depends heavily on its application to critical regions of the model, particularly the backbone in this case, where it can better capture the temporal dynamics and spatial features necessary for improving action recognition.

Template Learning. Template learning plays a pivotal role in PiVoT, providing universal adaptability and improving detector performance. When the framework transitions from frame-dependent to universal templates, substantial accuracy degradation is observed. For TSN, universal templates achieve a top-1 accuracy of 32.88% compared to 51.37% for frame-dependent templates, with a similar trend in top-5 accuracy (61.31% versus 78.71%). A similar degradation is observed in MViTv2, where universal templates yield degraded performance. This demonstrates that while universal templates aim to generalize across frames, they cannot match the level of frame-specific optimization provided by PiVoT's default template approach.

Further, replacing learned templates with fixed templates also degrades performance. For TSN, top-1 accuracy falls to 26.19% and top-5 accuracy to 52.01%, while for MViTv2, top-1 accuracy decreases to 60.12% and top-5 accuracy to 86.11%. These results emphasize the necessity of dynamic, learned templates in PiVoT, as fixed templates fail to adapt to the variations in temporal dynamics and action-specific nuances inherent in video sequences. Overall, the original template learning mechanism in PiVoT proves to be critical for achieving superior performance compared to alternative template designs.

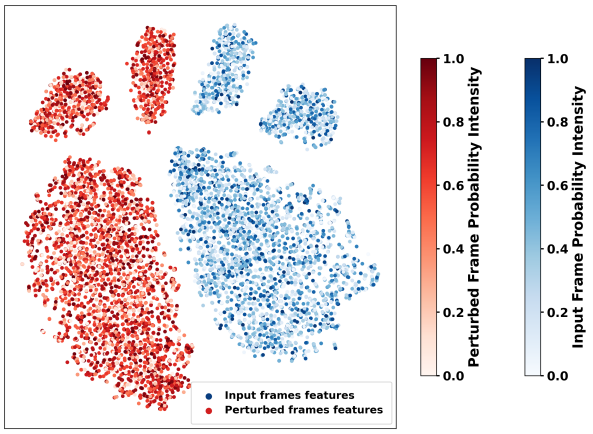

Figure 1. Backbone feature distribution with color intensity varied by detector head logits confidence. Lighter color means detector is less confident and vice -versa.

2. Template Analysis.

Fig. 1 demonstrates the backbone feature distribution of input frames and perturbed frames when provided separately to the respective trained TSM detector. Perturbed frames, created by adding input frames with the generated template, exhibit a distinct separation in feature space compared to the original input frames. The color-intensity variation, corresponding to the detector logits, indicates higher confidence in perturbed frames. This indicates that the template enhances the model's ability to extract discriminative features, resulting in more confident detector predictions. The addition of templates aligns features more closely with task requirements, demonstrating the utility of template-based enhancements for video-based tasks.

We further analyze the template enhancement in the t-SNE plots in Fig. 2, which demonstrate that, at the frame level, the input and perturbed frames exhibit minimal differences in distribution, indicating that the addition of the template does not significantly alter the original frame content. However, when viewed at the feature level in Fig. 1, there is a marked distinction between the input and perturbed frame feature distributions. This highlights that, while the template perturbation is subtle at the pixel level, it has a significant impact on the feature representations extracted by the detector.

This distinction underscores the templates' effectiveness in enhancing task-relevant features. By subtly modifying the input frames, the templates guide the detector's feature space towards better alignment with the underlying action-

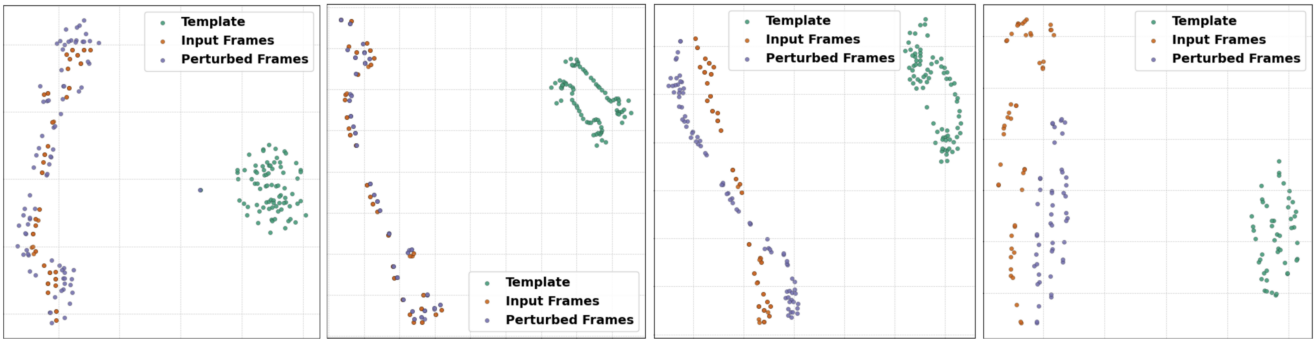

Figure 2. tSNE plot for all four detectors for input frames, perturbed frames, and estimated templates.

specific semantics. This leads to enriched feature representations that improve the detector's performance without compromising the natural temporal and spatial consistency of the input frames. In essence, the templates act as an implicit augmentation mechanism, creating a more expressive and discriminative feature space for accurate action recognition and detection.

3. Limitations

The proposed PiVoT wrapper, while effective in enhancing video-based detectors, has certain limitations that need discussion. First, the method requires training the wrapper specifically with each architecture, limiting its potential as a true plug-and-play solution. Developing a training-free implementation could significantly improve its ease of adoption across diverse models. Second, while performance gains are consistently observed across tasks and datasets, their magnitude cannot be guaranteed. This variability stems from differences in architecture and dataset characteristics, which may affect the wrapper's effectiveness. Another limitation lies in the templates' visibility and influence. Currently, the templates have substantial freedom to enhance task-specific performance, but this poses challenges when perturbed videos need to be publicly shared or used outside controlled environments. Making the templates imperceptible would allow broader adoption to detectors, which might not have our wrapper installed. This means that if the invisible templates are already embedded in the video, the need for having PiVoT on every copy of the detector will be eliminated. We leave all these useful directions for our future work.

4. Potential Social Impact

Video analysis tasks have diverse applications in the health industry, sports, entertainment, and surveillance, where accuracy is critical. For instance, in healthcare, home fall detection systems rely on accurate video-based monitoring to ensure timely assistance for patients. Similarly, in sports and entertainment, analyzing player movements with precision enhances performance evaluation and strategy development. Surveillance systems, which often operate in real-time, require high accuracy to detect anomalies effectively.

PiVoT, a template-based approach, offers a practical solution by significantly improving the accuracy of video-based detectors without substantially increasing system size or complexity. This efficiency ensures that existing systems can be enhanced with minimal computational overhead, making the solution scalable for deployment in resource-constrained environments. By enabling high performance without large architectural modifications, the technique broadens accessibility and applicability, addressing the growing demand for robust, efficient, and accurate video analysis across diverse real-world scenarios.

5. Implementation details

We provide implementation details for our methods, focusing on the architecture of our framework, the detector details, and where LoRA layers are applied in the backbone.

Architecture Details. We employ a 3D attention-based U-Net network inspired by [1, 3, 6] to estimate templates from the input frames. We chose the 3D U-Net for its ability to capture fine-grained spatio-temporal features while preserving resolution-critical for learning effective templates. Its skip connections and hierarchical structure yield semantically meaningful perturbations for AR and STAD. LoRA layers are integrated into the detector's backbone, as described afterward. The entire framework is trained end-to-end, with the detector initialized using a pretrained model.

Detector Details. For all detectors, we use the default configuration files provided by the MMACTION2 toolbox [2]. Below are the names of the config files used for our experiments:

1. TSN: tsn_imagenet-pretrained-r50_8xb32-1x1x8-50e_sthv2-rgb and tsn_imagenet-pretrained-r50_8xb32-1x1x8-100e_kinetics400-rgb
2. TSM: tsm_imagenet-pretrained-r50_8xb16-1x1x8-50e_sthv2-rgb and tsm_imagenet-pretrained-r50_8xb16-1x1x8-50e_kinetics400-rgb

3. MViTv2: mvit-small-p244_k400-pre_16xb16-u16-100e_sthv2-rgb and mvit-small-p244_32xb16-16x4x1-200e_kinetics400-rgb
4. SlowFast: slowfast_kinetics400-pretrained-r50_8xb16-4x16x1-20e_ava21-rgb

The models are first trained on the respective dataset to reproduce the reported performance. This trained model is then further used with our PiVoT wrapper.

However, for VideoMAE, we use checkpoints from the official VideoMAE AR repo and VideoMAE STAD repo. We adopt the ViT-S backbone with default training schedules for Kinetics-400 and Something-Something-V2, and the ViT-S backbone with Kinetics-400 pretraining for AVA 2.2. The pretrained model performance is first reproduced on the respective dataset and then wrapped with PiVoT.

LoRA Application We use different backbones for different detectors. TSN and TSM use ResNet-50; SlowFast uses 3D ResNet-50; and MViTv2 and VideoMAE both use a multi-scale ViT backbone.

In this ResNet-50 backbone, LoRA is selectively applied to the convolutional layers in residual layers 3 and 4. Specifically, LoRA is integrated into the BasicBlock and Bottleneck modules for these layers. For the BasicBlock, LoRA is applied at the first and second convolutional layers (conv1 and conv2), adapting the channel dimensions through low-rank matrices. Similarly, in the Bottleneck module, LoRA is applied at each of the three convolutional layers (conv1, conv2, and conv3), modifying the input or output channels to enhance feature adaptation. By limiting LoRA to these deeper layers, the model focuses on refining high-level feature representations without overburdening earlier network stages.

In the 3D ResNet-50 SlowFast LoRA network, LoRA is applied selectively to specific 3D convolutional layers within both the slow and fast pathways. Specifically, LoRA is applied to the conv1 layer and multiple convolutional layers across all ResNet stages (layer1, layer2, layer3, and layer4). This includes the main convolutional layers within each block, such as conv1, conv2, and conv3 in the bottleneck layers. This selective application focuses on enhancing the representational capacity of key layers without modifying the entire model.

Finally, for MViTv2 and VideoMAE, LoRA is applied within the multi-scale attention mechanism to the query and key projections. Specifically, LoRA introduces two low-rank projection layers for the input features, reducing their dimensionality to a smaller rank r. These reduced representations are then projected back to the original dimensions using corresponding projection layers before being added to the standard query and key projections. This enables LoRA to enhance the attention mechanism's adaptability while minimizing additional parameter overhead. These LoRA adaptations are applied at each attention block across all transformer layers of the MViT model, making them integral to improving the model's representational capacity and flexibility.

References

[1] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, 2016. 2

[2] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 2

[3] Mobarakol Islam, VS Vibashan, V Jeya Maria Jose, Navodini Wijethilake, Uppal Utkarsh, and Hongliang Ren. Brain tumor segmentation and survival prediction using 3d attention unet. In *MICCAI-W*, 2020. 2

[4] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 1

[5] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 1

[6] Adrian Wolny, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, Amaya Vilches Barro, Marion Louveaux, Christian Wenzl, Sören Strauss, David Wilson-Sánchez, Rena Lymbouridou, Susanne S Steigleder, Constantin Pape, Alberto Bailoni, Salva Duran-Nebreda, George W Bassel, Jan U Lohmann, Miltos Tsiantis, Fred A Hamprecht, Kay Schneitz, Alexis Maizel, and Anna Kreshuk. Accurate and versatile 3d segmentation of plant tissues at cellular resolution. *eLife*, 2020. 2