Not All Layers Are Equal in Image-to-Video Transfer

Thinesh Thiyakesan Ponbagavathi[1,2] **Constantin Seibold**[3] **Alina Roitberg**[4]
[1]University of Stuttgart
[2]Intelligent Assistive Systems Lab, University of Hildesheim
[3]University Hospital Heidelberg, Diagnostic and Interventional Radiology
[4] Multimodal Machine Learning and Assistive Computing Lab, University of Münster

Abstract

Parameter-efficient image-to-video transfer adapts frozen image foundation models to video via lightweight trainable adapters. Existing methods insert the same adapter at every transformer layer, implicitly assuming all layers require equal temporal modeling. A layer-wise sensitivity analysis across four video-native model families and two pretraining paradigms reveals that this assumption is wrong: temporal sensitivity increases with depth, and the pretraining paradigm determines its growth profile. Vision-only models exhibit a gradual rise, while vision-language models show early suppression followed by late recovery.

This finding motivates STRAP, a training-free adapter placement rule that derives a spatial-to-temporal transition boundary from a paradigm-matched video model. Spatial adapters are assigned before this boundary and temporal adapters after it, without modifying the adapter architecture itself. On four fine-grained action recognition benchmarks, STRAP improves accuracy by up to 11.8% over uniform placement, reduces trainable parameters by 29 to 48%, and lowers inference time by up to 10%. On standard benchmarks, it matches accuracy with fewer parameters. Code is available at https://github.com/th-nesh/STRAP.

1 Introduction

Adapting frozen image foundation models to video with lightweight temporal modules has become a prominent strategy for video understanding [1–5], avoiding the substantial data and compute costs of video-native pretraining [6, 7]. Yet while considerable effort has gone into designing these temporal adapters, the question of where to place them has received comparatively little attention: existing methods [1–5] find, through ablations comparing presence versus absence of a single temporal adapter type, that uniform placement performs best within this search space. This leaves open whether placing different adapter types based on layer-wise behavior can outperform uniform placement of temporal adapters.

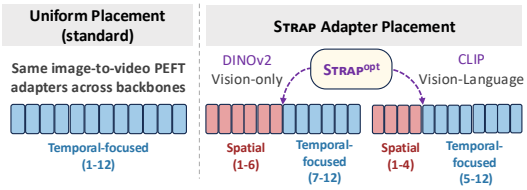


Figure 1: Existing image-to-video adapters place the same temporal adapter across all layers. STRAP instead uses a sensitivity-derived, backbone-specific boundary to place spatial adapters in early layers and temporal adapters in later layers.

In video-native transformers, there is no reason to expect temporal processing to distribute uniformly across depth. If it does not, then uniform adapter placement [1–5] in image-to-video transfer may be misallocating capacity, and might thus be adding temporal modules where they are unnecessary and potentially underserving layers where temporal reasoning concentrates.

We investigate this by analyzing how temporal sensitivity develops across layers in video-native transformers spanning vision-only [6, 8] and vision-language [9, 7] pretraining paradigms. For each

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

model and layer, we apply complementary spatial and temporal perturbations to the input and measure the relative dominance of temporal over spatial sensitivity in the resulting representations. This yields a depth-wise temporal sensitivity ratio that reveals two findings. First, temporal sensitivity is not uniform: it increases with depth across all model families. Second, the shape of this increase depends on the pretraining paradigm. Vision-only models exhibit a gradual rise across depth, whereas vision-language models show an early suppression of temporal sensitivity in the first layers, followed by a recovery in deeper layers.

Building on this analysis, we propose STRAP (**S**ensi**T**ivity-guided Laye**R**-wise **A**dapter **P**lacement), a temporal sensitivity-guided method for placing adapters in image-to-video transfer. STRAP uses a frozen video-native model of the same pretraining paradigm as the image backbone to estimate where temporal sensitivity changes most rapidly. This point defines a backbone-specific transition boundary. In the image model, spatial adapters are placed before the boundary, and temporal adapters after it. Importantly, STRAP does not introduce a new adapter architecture: it keeps existing adapters unchanged and replaces only their uniform layer assignment with a training-free, sensitivity-guided placement rule.

Our contributions can be summarized as follows:

(1) A layer-wise temporal sensitivity analysis across four video-native model families and two pretraining paradigms, revealing that temporal sensitivity increases with depth and follows paradigm-specific transition profiles.

(2) STRAP, a training-free placement rule that uses a paradigm-matched video model to determine where spatial and temporal adapters should be assigned in an image backbone.

(3) Evaluation on four fine-grained and two standard action recognition benchmarks. STRAP improves fine-grained recognition, where temporal order is critical, while matching standard-benchmark performance, in both cases with fewer trainable parameters than uniform placement.

2 Related Work

Parameter-efficient image-to-video adaptation. Parameter-efficient fine-tuning (PEFT) adapts frozen image foundation models to video via lightweight trainable modules while preserving the pretrained backbone. Existing approaches differ in how temporal information is injected. Adapter-based methods, including ST-Adapter [1], AIM [2], M2-CLIP [5], and Omni-CLIP [10], model temporal dependencies via convolutional, temporal difference, or attention-based modules. Extensions such as Frame2Freq [3] capture motion in the frequency domain, while DualPath [4] separates spatial and temporal processing into parallel branches with late fusion. Prompt-based methods, such as VitaCLIP [11], adapt CLIP via multimodal prompt learning using learnable context tokens. Fully fine-tuned approaches, including STAN [12], BIKE [13], and ILA [14], improve temporal reasoning via auxiliary networks and cross-modal alignment, but operate outside the parameter-efficient regime that STRAP targets. Despite their architectural diversity, these methods share a common limitation: temporal adaptation is applied uniformly across layers, implicitly assuming all layers require equal temporal modelling. STRAP challenges this assumption by deriving backbone-specific adapter placement from layer-wise temporal sensitivity profiles of video-native models.

Layer-wise representation analysis. Prior work has performed representation analysis across depth within specific video architectures: VTCD [15] discovers object-tracking concepts in later layers of video models, while [16] traces temporal understanding in video-language models to order-sensitive attention pathways. DiffTrack [17] analyzes temporal correspondence in video diffusion transformers, while related work uses internal representations for controlled generation and motion consistency [18, 19]. These works do not quantify how temporal sensitivity evolves relative to spatial sensitivity across depth, or how this profile varies across pretraining paradigms. STRAP uses this depth-wise sensitivity profile as a training-free signal for adapter placement in image-to-video PEFT.

Sensitivity-guided adaptation. AdaLoRA [20] adaptively allocates LoRA rank budgets across weight matrices using importance scores estimated during fine-tuning. Related work applies layer-wise sensitivity signals to quantization [21] and layer-adaptive temporal correction in long video generation [22]. These methods target specific downstream tasks and rely on task-specific signals. In contrast, STRAP derives a layer-wise sensitivity profile from a frozen, paradigm-matched video model and uses it as a training-free signal for adapter placement across designs, backbones, and datasets — without labels, gradients, or architectural modification.

3 Where does Temporal Sensitivity Reside?

To gain an intuition about optimal adapter placements in image-to-video models, we analyze how temporal sensitivity is distributed across depth in video-native transformers. The resulting profile can then guide adapter placement in image-to-video transfer. The analysis pipeline is summarized in Fig. 2.

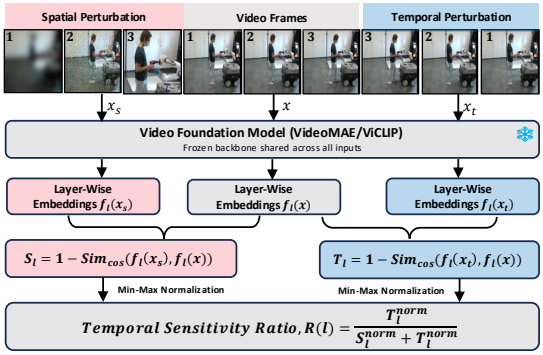


Figure 2: **Sensitivity Metric Pipeline.** Spatially and temporally perturbed versions of a video clip are passed through a frozen video-native model together with the original clip. Layer-wise cosine distances between mean-pooled patch-token embeddings measure each layer's response to spatial and temporal perturbations, yielding the Temporal Sensitivity Ratio $R(l)$.

3.1 Temporal Sensitivity Analysis

Perturbation Design: To probe how the layer-wise representation responds to spatial versus temporal structure, we apply two complementary perturbations to a clip x and compare the resulting embeddings at every transformer block: A spatial perturbation x_s (per-frame Gaussian blur, crop-resize, color jitter) disrupts appearance while preserving frame order and a temporal perturbation x_t (frame reversal, block shuffle, frame drop) disrupts ordering and continuity while preserving the appearance of the retained frames.

While these two perturbations cannot be assumed to be equivalent in terms of semantic severity, we use them as *probes of relative depth-wise responsiveness*, not as absolute measurements of spatial or temporal information.

Sensitivity Computation: A frozen video-native foundation model then processes x, x_s, and x_t. At each block $l \in \{1, \ldots, L\}$, we extract mean-pooled patch-token embeddings $f_l(\cdot)$ and define

$$S_l = 1 - Sim_{cos}\big(f_l(x_s), f_l(x)\big) \quad \text{and} \quad T_l = 1 - Sim_{cos}\big(f_l(x_t), f_l(x)\big) \tag{1}$$

as its spatial and temporal sensitivity, respectively. We average S_l and T_l over clips, then independently min-max normalize (indicated by $*^{norm}$) each response sequence across layers for every model-dataset pair to prevent the raw perturbation magnitude from dominating the ratio while preserving the depth-wise profile induced by each perturbation type. Inspired by relative sensitivity indices in variance-based analysis [23], we then define *Temporal Sensitivity Ratio* of a block as $R(l) = T_l^{\text{norm}}/(T_l^{\text{norm}} + S_l^{\text{norm}})$, $R(l) \in [0, 1]$ measures relative temporal dominance: values near 0 indicate stronger response to spatial perturbations, while values near 1 indicate stronger response to temporal perturbations. The trajectory of $R(l)$ across depth thus indicates where representations transition from appearance-sensitive to temporally-sensitive.

Analysis Setup: We compute $R(l)$ for four video-native model families spanning vision-only and vision-language pretraining: VideoMAE [6] (masked reconstruction), V-JEPA 2.1 [8] (predictive latent), ViCLIP [9] (contrastive vision–language), and InternVideo2 [7] (a hybrid). For each, we evaluate all available Small/Base/Large variants on 500 clips per model–dataset pair. We average the normalized $R(l)$ profiles across six benchmarks to capture a global depth-wise sensitivity pattern for each proxy model, covering fine-grained, standard, and human-robot interaction recognition.

3.2 *Not All Layers Are Equal* in processing temporal information

The top row of Fig. 3 shows how $R(l)$ behaves with increasing normalized network depth for the four different models. $R(l)$ displays a clear trend across all settings as early layers respond more strongly to appearance perturbations, whereas deeper layers become increasingly responsive to temporal-order perturbations, indicating that temporal sensitivity is not uniformly distributed across depth.

Finding 1: *The temporal sensitivity of a video network increases with depth.*

While $R(l)$ displays the highest values towards the final layers of a network, its growth profile differs across model families. Vision-only models (VideoMAE and V-JEPA) exhibit a gradual increase of

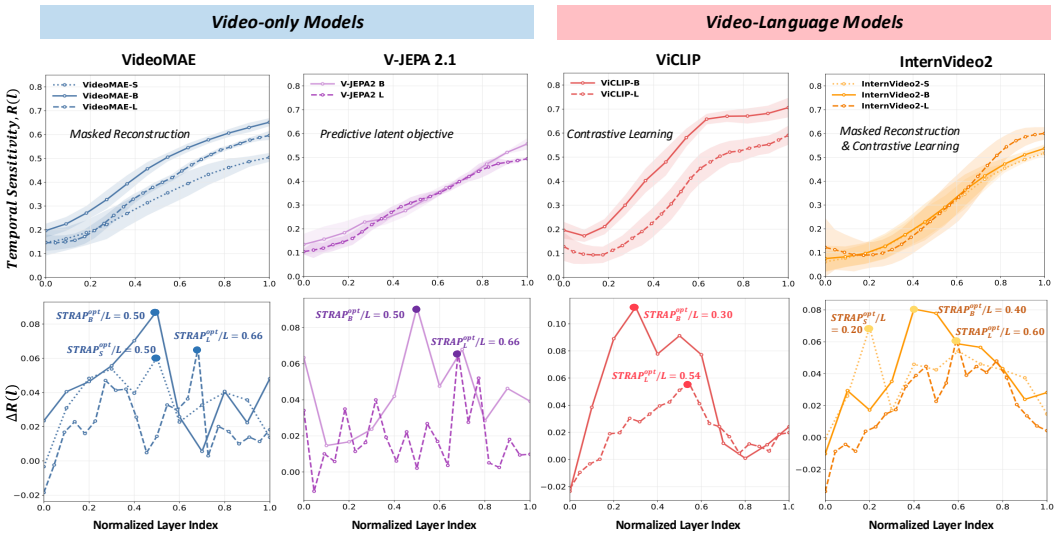


Figure 3: **Temporal sensitivity profiles are depth-dependent and paradigm-determined.** $R(l)$ increases with depth across all models (Finding 1). Vision-only models show gradual emergence; vision-language models exhibit early suppression followed by a late-layer rise (Finding 2), consistent with the influence of cross-modal alignment. Profiles are stable across scales and datasets.

$R(l)$ across depth. In contrast, the vision-language models such as ViCLIP show early suppression, where $R(l)$ decreases in the first few layers before recovering later. This trend can also be observed partially in the hybrid case of InternVideo2: despite its masked-reconstruction component, it retains the early suppression characteristic of vision-language pretraining, but recovers with a smoother, more gradual rise than ViCLIP. Within each paradigm, the profile shapes of $R(l)$ behave similarly across available size variants albeit the absolute sensitivity may vary. The growth profile of $R(l)$ is therefore a signature of the pretraining paradigm and is preserved across model scales.

Finding 2: *The depth-wise growth of $R(l)$ depends on the pretraining paradigm.*

To verify that our findings generalize, we extend the sensitivity analysis to two additional video-only models (VideoMAEv2 [24] and V-JEPA1 [25]) and two video-language models (UMT [26] and InternVideo [27]). Both findings hold across these models; see Appendix A.3 for details.

3.3 Design Implications

From these two findings, we can derive design constraints for image-to-video adapters. Finding 1 implies that uniform placement is suboptimal: distributing temporal modules evenly across depth allocates capacity to layers whose representations are predominantly appearance-driven. Finding 2 further implies that no fixed depth-wise rule can transfer across backbones; instead, the proxy used to determine placement must match the pretraining paradigm of the target image model.

4 STRAP: Sensitivity-guided Layer-wise Adapter Placement

Sec. 3 shows that temporal sensitivity emerges non-uniformly across depth and follows a paradigm-dependent profile. STRAP (**S**ensi**T**ivity-guided Laye**R**-wise **A**dapter **P**lacement) converts this profile into a layer-wise adapter placement rule.

4.1 Preliminaries

Backbone and adapters. We consider a transformer backbone with L layers processing a video $\mathbf{x} = \{x_1, \ldots, x_T\}$. Let $h_l \in \mathbb{R}^{B\times T\times N\times D}$ denote the token representation at layer l, where B is the batch size, T the number of frames, N the number of spatial tokens per frame, and D the feature dimension. Adapters are inserted into transformer blocks through residual updates,

$$h_l' = h_l + A_l(h_l), \tag{2}$$

where $A_l(\cdot)$ is a lightweight trainable module and the image-pretrained backbone remains frozen.

Spatial adapters. Spatial adapters operate independently on each frame and do not introduce cross-frame interactions. Following image-adaptation designs such as AdaptFormer [28], we reshape h_l from (B, T, N, D) to $(B{\cdot}T, N, D)$ and apply a bottleneck transformation frame-wise:

$$A_l^{\text{spatial}}(h_l)[t] = f(h_l[t]W_{\text{down}})W_{\text{up}}, \quad \forall t \in \{1, \ldots, T\} \tag{3}$$

where $W_{\text{down}} \in \mathbb{R}^{D\times r}$, $W_{\text{up}} \in \mathbb{R}^{r\times D}$, $r \ll D$, and f is a non-linearity. These adapters refine per-frame spatial representations while preserving temporal independence.

Temporal adapters. Temporal adapters introduce cross-frame interactions into an image-pretrained backbone. We write them generically as

$$A_l^{\text{temporal}}(h_l) = \mathcal{T}_l(h_l W_{\text{down}})\, W_{\text{up}} \tag{4}$$

where $\mathcal{T}_l$ is a lightweight temporal operator. Different image-to-video PEFT methods instantiate $\mathcal{T}_l$ differently: ST-Adapter [1] uses temporal depth-wise convolutions, Frame2Freq [3] applies multi-scale Fourier-domain and temporal convolutions, and M2-CLIP [5] combines temporal convolution, spatial convolution, and temporal difference modeling. STRAP keeps these temporal adapter designs unchanged and modifies only their layer-wise placement.

4.2 Sensitivity-guided Placement

Since the shape of $R(l)$ itself is paradigm-dependent, no fixed layer index applies across backbones and we require a layer-level signal that is comparable across paradigms. We instead consider the discrete derivative

$$\Delta R(l) = R(l+1) - R(l), \tag{5}$$

which captures the local rate at which temporal sensitivity emerges across depth. As shown in the bottom row of Fig. 3, $\Delta R(l)$ admits a clearly identifiable maximum across all model families and scales, despite the differing shapes of $R(l)$ itself.

We denote the layer at which this peak occurs as STRAP optimum

$$\text{STRAP}^{\text{opt}} = \arg\max_l \Delta R(l). \tag{6}$$

We interpret $\text{STRAP}^{\text{opt}}$ as the depth at which the network most rapidly shifts from appearance-dominated to temporally-dominated representation and use it as the boundary between spatial and temporal adaptation. Layers before this transition are relatively more appearance-responsive, while layers after it become more responsive to temporal-order perturbations. This criterion is analogous to transition or knee-point detection in structured curves [29, 30].

Placement rule. Given $\text{STRAP}^{\text{opt}}$, the adapter types are assigned as:

$$\text{adapter}(l) = \begin{cases} \text{spatial} & \text{if } l \le \text{STRAP}^{\text{opt}} \\ \text{temporal} & \text{if } l > \text{STRAP}^{\text{opt}} \end{cases} \tag{7}$$

Early layers retain frame-wise spatial adaptation, while later layers receive temporal adapters where the proxy model shows stronger temporal responsiveness.

Backbone matching. Each image backbone is paired with a video-native proxy of matching pretraining paradigm and scale. For example, this matches: DINOv2 with VideoMAE, yielding $\text{STRAP}^{\text{opt}} = 6$ and a 6S/6T split, and CLIP with ViCLIP, yielding $\text{STRAP}^{\text{opt}} = 4$ and a 4S/8T split. Sweeping the spatial-temporal boundary across configurations confirms that the sensitivity-derived boundary is empirically optimal across datasets (Fig. 4).

Adapter design. STRAP is adapter-agnostic: it determines *where* adapters are placed, not *what* adapters are used. We retain the original temporal adapter designs of each base method, ST-Adapter [1], Frame2Freq [3], and M2-CLIP [5], and replace only their uniform layer assignment with sensitivity-guided placement. Spatial adapters use a lightweight bottleneck adapter following AdaptFormer [28]. Prior methods including AIM [2] and DualPath [4] combine spatial and temporal modules but apply the same pattern uniformly across depth.

Table 1: **Fine-grained action recognition.** STRAP improves accuracy (up to +11.8%) while reducing trainable parameters by up to 48% and inference time by up to 10%. Gains are largest on temporally demanding datasets (D&A, IKEA, HRI) and consistent across methods and backbones.

Method	Strategy	Params (M)	Inf. Time (ms)†	Top-1 Accuracy (%) ↑ Diving	Drive&Act	IKEA	HRI-30
Fully Fine-tuned Frameworks							
ORViT [31]	-	160	-	88.0	-	-	-
VideoSwin-B [32]	-	88	-	-	72.9	72.6	-
UniformerV2 [33]	-	115	-	-	76.7	-	-
SlowOnly [34]	-	32	-	-	-	-	86.6
DINOv2-B/14 backbone (6S/6T, derived from VideoMAE)							
ST-Adapter [1]	Uniform	7.3	54.5	88.6	75.2	70.5	85.5
	STRAP	**4.2** (↓42%)	**50.8** (↓6.8%)	**88.9** (+0.3)	**81.9** (+6.7)	**77.6** (+7.1)	**89.2** (+3.7)
Frame2Freq [3]	Uniform	7.3	57.7	91.7	81.5	76.4	89.8
	STRAP	**4.2** (↓42%)	**51.7** (↓10.4%)	**92.5** (+0.8)	**83.5** (+2.0)	**79.8** (+3.4)	**91.2** (+1.4)
M2-CLIP [5]	Uniform	14.8	64.6	90.0	77.7	74.9	80.5
	STRAP	**7.7** (↓48%)	**57.8** (↓10.4%)	**90.3** (+0.3)	**82.9** (+5.2)	**79.6** (+4.7)	**92.3** (+11.8)
CLIP-B/16 backbone (4S/8T, derived from ViCLIP)							
ST-Adapter [1]	Uniform	7.3	46.7	90.4	72.6	70.2	81.1
	STRAP	**5.2** (↓29%)	**44.2** (↓5.5%)	**90.5** (+0.1)	**80.8** (+8.2)	**73.3** (+3.1)	**86.2** (+5.1)
Frame2Freq [3]	Uniform	7.3	49.6	**92.2**	77.4	**78.0**	85.7
	STRAP	**5.2** (↓29%)	**47.2** (↓4.9%)	91.1 (-1.1)	**81.5** (+4.1)	76.5 (-1.5)	**86.3** (+0.6)
M2-CLIP [5]	Uniform	14.8	52.3	89.3	77.2	71.9	85.9
	STRAP	**9.9** (↓33%)	**48.8** (↓6.7%)	**90.3** (+1.0)	**82.9** (+5.7)	**76.2** (+4.3)	**93.1** (+7.2)

†Inference time per sample (ms) measured on a single NVIDIA A100 GPU, averaged over 100 runs.

5 Experiments

5.1 Experimental Setup

Unless otherwise, all experiments use frozen DINOv2-B/14 [35] and CLIP-B/16 [36] as the image-pretrained backbones, which we adapt via PEFT. We train only the adapter modules and a linear classifier, keeping the image-pretrained backbone fixed. Backbone-to-proxy matching and the resulting placement boundaries follow Sec. 4.2. We compare STRAP against conventional uniform adapter placement under identical training and evaluation settings, using 16-frame inputs at 224×224 resolution (32 frames for SSv2 and 8 frames for Kinetics) and AdamW for 60 epochs. Additional implementation details are provided in the Appendix A.1.

5.2 Main Results: Fine-Grained Action Recognition

Datasets. We evaluate the main setting on four fine-grained action recognition benchmarks: Diving48 [37] (diving competitions), Drive&Act [38] (driver behavior analysis), IKEA-ASM [39] (furniture assembly), and HRI-30 [34] (fine-grained human-robot interaction), where subtle motion and temporal ordering are class-discriminative alongside appearance.

Results. We first compare STRAP to uniform adapter placement, our main baseline, for three image-to-video PEFT adapters: ST-Adapter, Frame2Freq, and M2-CLIP. Results are reported for both DINOv2 and CLIP backbones. We also include published fully fine-tuned video models in gray for context, although these methods are outside the parameter-efficient setting targeted by this work.

Table 1 shows that STRAP achieves state-of-the-art results on all four fine-grained benchmarks among the compared methods. Against uniform placement, STRAP improves accuracy in most settings while reducing trainable parameters by 29–48% and inference time by 5–10%. The largest gains appear on Drive&Act, IKEA-ASM,

Table 2: **Results for symmetric and non-symmetric actions.** *Sym* and *N-S* denote accuracy on symmetric and non-symmetric action subsets. Our method has the largest gains for symmetric actions (e.g., *opening* vs. *closing*), which require distinguishing temporal order and direction, precisely the structure targeted by late-layer temporal adapters.

Method	Strat.	Drive&Act Sym	Drive&Act N-S	Drive&Act All	IKEA-ASM Sym	IKEA-ASM N-S	IKEA-ASM All	HRI-30 Sym	HRI-30 N-S	HRI-30 All
ST-Adapter	Uni.	66.4	83.4	75.2	68.5	73.9	70.5	81.5	95.3	85.5
	STRAP	**78.0**	**85.5**	**81.9**	**80.3**	**76.2**	**77.6**	**85.4**	**96.8**	**89.2**
Frame2Freq	Uni.	77.1	85.5	81.5	81.0	74.2	76.4	85.6	**98.9**	89.8
	STRAP	**79.6**	**87.1**	**83.5**	**86.8**	**76.4**	**79.8**	**88.2**	97.2	**91.2**
M2-CLIP	Uni.	68.4	85.0	77.7	75.3	74.6	74.9	78.4	84.6	80.5
	STRAP	**78.8**	**86.6**	**82.9**	**83.7**	**77.6**	**79.6**	**89.3**	**98.2**	**92.3**

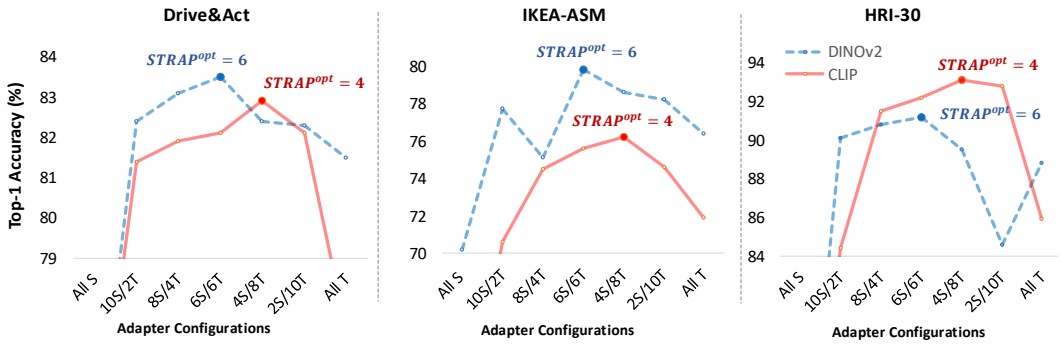


Figure 4: Ablation of spatial-temporal adapter allocation across layers. We vary the number of spatial (S) and temporal (T) adapters under varying depth.

and HRI-30: with DINOv2 and ST-Adapter, accuracy improves by +6.7%, +7.1%, and +3.7%, respectively. Gains on Diving48 are smaller. With CLIP, STRAP improves ST-Adapter and M2-CLIP on all datasets, while Frame2Freq shows small drops on Diving48 and IKEA-ASM. These results suggest that layer-wise spatial-temporal placement is useful for fine-grained video understanding with fewer trainable parameters needed.

To investigate where STRAP helps most, we also use the split into symmetric and non-symmetric classes [40, 3]. Symmetric actions, such as *opening* vs. *closing* or *taking* vs. *putting*, share similar movements but differ in temporal direction. They are therefore hard to resolve from appearance alone and require modeling how the action unfolds. Table 2 shows a clear trend: STRAP brings the largest gains on these temporally ambiguous classes. For example, on IKEA-ASM, symmetric-action accuracy improves by +11.8%, +5.8%, and +8.4% depending on the adapter type. These gains are usually larger than those on non-symmetric actions, suggesting that STRAP allocates temporal capacity where temporal direction matters most.

5.3 Does Placement Matter Beyond Parameter Count?

Placement direction. We next test whether the order of spatial and temporal adapters matters (Table 3, upper section). Reversing the layout, i.e., placing temporal adapters in early layers and spatial adapters in later layers, strongly degrades performance. Compared to the original order, accuracy drops by -11.1% on Drive&Act, -40.7% on IKEA-ASM, and -15.7% on HRI-30. This shows that the gain is not due to mixing both adapter types alone, and the spatial-to-temporal direction is critical.

Component contribution. In Table 3 (middle section), we isolate spatial and temporal adapters using Frame2Freq with DINOv2 and the STRAP boundary $\text{STRAP}^{\text{opt}} = 6$. Spatial-only adaptation is weak, especially on HRI-30. In contrast, using temporal adapters only in the later layers already outperforms uniform temporal placement, despite using fewer parameters. Our strategy using both performs best, indicating that early spatial adaptation remains useful, but temporal adaptation is better concentrated in later layers.

Matched-capacity comparison. In theory, the improved performance could be simply because our method uses fewer trainable parameters, thereby reducing overfitting. We test this in Table 3 (bottom section), by comparing against a width-adjusted uniform baseline with the same parameter count as STRAP. This matched-capacity baseline performs worse, with drops of -13.3 points on Drive&Act, -3.7 points on IKEA-ASM, and -3.5 points on HRI-30. Thus, the gains are not explained by parameter count alone, but by the layer-wise placement of spatial and temporal adapters.

5.4 Analyzing the Spatial-Temporal Adapter Boundary

We now investigate how the spatial-temporal adapter boundary affects performance. In Figure 4, we vary the spatial/temporal adapter split while keeping the number of adapted layers fixed for DINOv2 and CLIP. Balanced allocations generally outperform spatial- or temporal-heavy configurations, with the best results lying near the sensitivity-derived boundaries: $\text{STRAP}^{\text{opt}} = 6$ for DINOv2 and $\text{STRAP}^{\text{opt}} = 4$ for CLIP. Using the boundary induced by the opposite proxy, i.e., $\text{STRAP}^{\text{opt}} = 4$ for DINOv2 or $\text{STRAP}^{\text{opt}} = 6$ for CLIP, moves the allocation away from the empirical peak. This acts as

Table 3: **Adapter placement ablation.** We isolate: (*i*) component contribution - does each adapter type help? (*ii*) placement direction - does spatial-before-temporal matter? (*iii*) capacity control - is the gain from placement or parameter count? Results reported using Frame2Freq Adapter and DINOv2 (STRAP$^{\text{opt}} = 6$).

Configuration	Params	D&A	IKEA	HRI
Placement direction				
Reversed (T→S)	4.2	72.4	39.1	75.5
STRAP (S→T)	**4.2**	**83.5**	**79.8**	**91.2**
Component				
Spatial only (layers 1-6)	0.6	71.7	66.6	51.1
Temporal only (layers 7-12)	3.6	81.5	78.0	90.1
STRAP (Ours, 6S+6T)	**4.2**	**83.5**	**79.8**	**91.2**
Capacity control (matched budget)				
Uniform Temporal	7.3	81.5	76.4	89.8
Uniform Temporal (adjusted width)	4.2	70.2	76.1	87.7
STRAP (Ours)	**4.2**	**83.5**	**79.8**	**91.2**

Table 4: **Placement, interleaving and boundary-rule ablations.** Accuracy (%) on IKEA-ASM and HRI-30. STRAP performs best in every column, with the fewest parameters on DINOv2.

		Frame2Freq		M2-CLIP	
Configuration	**Params**	**IKEA**	**HRI**	**IKEA**	**HRI**
Placement vs. capacity (DINOv2)					
Uniform (12T)	7.3	76.4	89.8	74.9	80.5
12S+6T	4.8	78.1	88.2	78.4	90.3
12S/12T	8.5	41.1	72.9	78.6	91.2
STRAP (6S/6T)	**4.2**	**79.8**	**91.2**	**79.6**	**92.3**
Interleaving (DINOv2)					
Interleaved (7S/7T)	4.9	77.8	88.4	78.6	90.2
STRAP (6S/6T)	**4.2**	**79.8**	**91.2**	**79.6**	**92.3**
Boundary rule (CLIP)					
Midpoint (6S/6T)	4.2	74.2	83.4	75.6	92.2
Threshold = 0.5 (6S/6T)	4.2	74.2	83.4	75.6	92.2
STRAP (4S/8T)	**5.2**	**76.5**	**86.3**	**76.2**	**93.1**

a proxy-mismatch diagnostic and suggests that the transition point is paradigm-specific rather than directly transferable across backbones. Thus, STRAP provides a training-free boundary selection rule without labeled validation or split-specific tuning, while remaining a practical guide rather than a universal optimum. Full sweeps across all adapter types and both backbones are provided in Appendix A.4.

Placement, not capacity, drives the gains. We test whether STRAP's gains simply reflect added capacity (Table 4). STRAP outperforms both spatial adapters in every layer with temporal adapters in the last six (12S+6T) and both types in every layer (12S/12T), despite the latter nearly doubling the parameters, so its gains come from placement. 12S/12T also collapses Frame2Freq whereas 12S+6T does not, suggesting that placing both types together in early layers causes instability. Interleaving the two types around the boundary to simulate a softer boundary (7S/7T) also underperforms. Finally, alternative boundary rules fall short: on CLIP, the midpoint and a fixed 0.5 threshold both yield 6S/6T, trailing STRAP by 0.6-2.9 points; on DINOv2, the midpoint matches the STRAP boundary, while the threshold yields 7S/5T, which also trails it.

5.5 Generalization to Other Backbones and Model Sizes

We also test whether the STRAP$^{\text{opt}}$ boundary is applicable beyond the DINOv2-B and CLIP-B backbones used in the main experiments. Table 5 considers two extensions: larger model scales, using DINOv2-L and CLIP-L, and a different vision-only image-pretrained backbone, MAE-B [41]. Across these settings, our method consistently outperforms the standard uniform adapter placement, with gains ranging from $+0.4\%$ on IKEA-ASM with DINOv2-L to $+14.5\%$ with MAE-B. At the same time, it reduces trainable parameters by 42–55%. This suggests that the vision-only boundary is not specific to the initial models. To further understand the generalization capability of STRAP to other video understanding tasks, we perform experiments on temporal action detection (TAD) (Appendix A.7). On THUMOS14 with VideoMAE, STRAP also improves over uniform placement with 22% fewer trainable parameters.

Table 5: **Generalization of STRAP.** *Model size*: boundaries shift deeper with scale but remain identifiable and effective. *Backbone*: VideoMAE-derived STRAP$^{\text{opt}} = 6$ transfers well to MAE-B. Results reported with ST-Adapter.

Configuration	Strategy	Params	IKEA	HRI
Generalization to Model Size				
DINOv2				
ViT-B (STRAP$^{\text{opt}}$=6)	Uniform	7.3	70.5	85.5
	STRAP	**4.2**	**77.6**	**89.2**
ViT-L (STRAP$^{\text{opt}}$=16)	Uniform	19.0	80.1	90.3
	STRAP	**8.5**	**80.5**	**91.8**
CLIP				
ViT-B (STRAP$^{\text{opt}}$=4)	Uniform	7.3	70.2	81.1
	STRAP	**5.2**	**73.3**	**86.2**
ViT-L (STRAP$^{\text{opt}}$=13)	Uniform	19.0	75.6	88.2
	STRAP	**10.4**	**76.7**	**89.4**
Generalization to Backbone				
MAE-B (STRAP$^{\text{opt}}$=6)	Uniform	7.3	43.2	70.5
	STRAP	**4.2**	**57.7**	**74.5**

5.6 Generalization to Standard Action Recognition

Datasets. Beyond fine-grained action recognition, we ask whether STRAP remains useful when the temporal signal is less central, and we evaluate coarse-grained video classification on Kinetics-400 [42] and SSv2 [43]. Kinetics-400 [42] contains ∼240k YouTube clips and 400 broad categories. SSv2 [43] contains ∼220k crowd-sourced videos of predefined everyday hand-object interactions, where temporal reasoning is needed to distinguish visually similar action templates, such as pushing something left vs. right. Together, they cover temporally dependent and more appearance-dominated regimes of video understanding.

Table 6: **Standard (non-fine-grained) video classification results.** STRAP reduces trainable parameters while slightly improving accuracy across most settings.

		SSv2		K400	
Method	**Strategy**	**Params (M)**	**Top-1 (%)**	**Params (M)**	**Top-1 (%)**
Fully Fine-tuned Frameworks					
Video SWIN [32]	-	88.1	69.6	88.1	80.6
ViT-B (finetuned)	-	86.1	69.3	86.1	81.0
ILA [14]	-	-	66.8	-	81.6
STAN-self [12]	-	-	69.5	-	84.2
DINOv2-B/14 backbone (6S/6T, derived from VideoMAE)					
ST-Adapter [1]	Uniform	14.3	71.4	7.3	**81.6**
	STRAP	**7.8**	**71.8**	**4.2**	81.5
Frame2Freq [3]	Uniform	14.3	**71.9**	7.3	81.1
	STRAP	**7.8**	71.6	**4.2**	**81.3**
M2-CLIP [5]	Uniform	28.6	71.2	14.8	81.8
	STRAP	**14.9**	**71.6**	**7.7**	**81.9**
CLIP-B/16 backbone (4S/8T, derived from ViCLIP)					
ST-Adapter [1]	Uniform	14.3	69.5	7.3	82.0
	STRAP	**10.4**	**70.1**	**5.2**	**82.2**
Frame2Freq [3]	Uniform	14.3	**70.4**	7.3	82.1
	STRAP	**10.4**	70.2	**5.2**	**82.2**
M2-CLIP [5]	Uniform	28.6	69.1	14.8	81.5
	STRAP	**19.9**	**70.0**	**9.9**	**82.3**

Results. Table 6 shows that across adapters and backbones, STRAP reduces the number of trainable parameters substantially, by up to 48%, while matching or slightly improving accuracy in most settings. The improvements are naturally smaller than on fine-grained benchmarks, where temporal adaptation is more critical. Another possible explanation is that datasets such as Kinetics-400 contain many web videos in which object, scene, and appearance cues are already well covered by image-pretrained representations, making extensive video-specific adaptation less critical than in fine-grained settings that depend more strongly on subtle temporal structure. However, the consistent parameter reduction with stable accuracy suggests that optimizing adapter placement is a simple and robust strategy to at least boost the efficiency of the mechanism in standard video classification.

6 Discussion

Why should the temporal sensitivity of video-pretrained models transfer to image-to-video PEFT? We do not assume that representations transfer from video to image models. Instead, STRAP uses a weaker structural prior derived from our analysis in Section 3: temporal sensitivity emerges non-uniformly across depth, and its emergence profile depends on the pretraining paradigm. Concretely, we derive the boundary $\text{STRAP}^{\text{opt}}$ from the temporal sensitivity profile of a frozen, paradigm-matched video model: a vision-only proxy for DINOv2 and a vision-language proxy for CLIP. The empirical results support this: the derived boundaries perform well in the allocation sweep (Fig. 4), improve over uniform placement across adapters and datasets (Table 1), and hold across additional backbones and model scales (Table 5). This suggests that STRAP is not simply a balanced-split heuristic: the boundary is mid-depth for DINOv2-B, earlier for CLIP-B, and shifts with scale (Table 5), reflecting paradigm- and scale-specific sensitivity profiles without validation data.

Is the boundary a principled choice or a tuned hyperparameter? The boundary is not tuned on downstream validation data. It is derived from the peak change in the temporal sensitivity ratio of a paradigm-matched video-pretrained reference model, as defined in Section 3. The allocation sweep in Fig. 4 shows that these sensitivity-derived boundaries are best or near-best among spatial-temporal splits for both DINOv2 and CLIP. We therefore view $\text{STRAP}^{\text{opt}}$ as a training-free placement rule, while acknowledging that the exact best split can vary slightly across datasets.

When does STRAP help for recognition quality, and when is it mainly an efficiency gain? STRAP yields the largest accuracy gains on fine-grained video tasks (Table 1), where subtle temporal differences are often class-discriminative. This is not inherited from existing image-to-video adapters: prior adapters (with the exception of Frame2Freq [3]) struggle to consistently improve fine-grained activity recognition [3]. STRAP improves this setting by changing *where* the adapters are placed, rather than introducing a new temporal adapter. The symmetric-action analysis in Table 2 supports this: improvements are largest for visually similar classes that differ mainly in temporal direction, suggesting that existing adapters are limited not only by their design but also by where temporal

capacity is placed. On standard benchmarks such as Kinetics-400 [42], where appearance cues are often sufficient, STRAP mainly acts as an efficiency mechanism, reducing trainable parameters without sacrificing accuracy (Table 6). Thus, STRAP is most beneficial for accuracy when temporal structure is discriminative, and mainly improves efficiency when appearance cues dominate.

7 Limitations

STRAP rests on the assumption that the depth-wise temporal sensitivity profile of a video-native model is a meaningful prior for adapter placement in an image-pretrained backbone of the same pretraining paradigm. While the empirical results support this, a theoretical basis for why paradigm matching is sufficient or when it might fail is missing. The perturbation-based sensitivity metric mixes spatial and temporal perturbations, which differ in semantic severity, and although min-max normalization reduces scale differences, the absolute values of $R(l)$ should be interpreted as relative depth-wise trends rather than calibrated measurements. Finally, STRAP assumes a single hard boundary between spatial and temporal adapters. A soft or graduated transition could capture finer structure in the sensitivity profile, particularly for architectures where the transition may not be as sharp.

8 Conclusion

This work studied an overlooked question in image-to-video transfer: *do all layers in an image-to-video model need the same temporal adaptation?* The answer is *no*. Temporal sensitivity increases with depth, and the profile of that increase is shaped by how the model was pretrained. STRAP uses this observation as a placement rule: spatial adapters go in early layers, temporal adapters in later ones, with the boundary read off a frozen video-native proxy. No new adapter architecture is needed, and no labels or gradients are involved in determining the placement. The practical payoff is clearest on fine-grained benchmarks, where temporal order separates otherwise similar classes. On standard benchmarks, the accuracy gains are smaller, but the parameter and compute savings are consistent. Whether similar transitions appear in video generation or cross-modal settings is a natural next question. More generally, existing work on image-to-video PEFT has focused almost entirely on designing better adapters. The results suggest that where those adapters are placed matters too, and that this direction has been largely unexplored.

Acknowledgments and Disclosure of Funding

This research was partially supported by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy – EXC 2120/1 – 390831618 and the Baden-Württemberg Stiftung (Elite Postdoc Program, PRIDRIVE project). This research was also supported by institutional funding of the University of Stuttgart, the University of Hildesheim and the University of Münster. The authors also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Thinesh Thiyakesan Ponbagavathi. The authors also gratefully acknowledge the computing time provided on the high- performance computer HoreKa by the National High-Performance Computing Center at KIT. This center is jointly supported by the Federal Ministry of Education and Research and the Ministry of Science, Research and the Arts of Baden- Württemberg, as part of the National High-Performance Computing (NHR) joint funding program (https://www.nhr-verein.de/en/our-partners). HoreKa is partly funded by the German Research Foundation (DFG).

References

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

[2] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition, 2023.

[3] Thinesh Thiyakesan Ponbagavathi, Constantin Seibold, and Alina Roitberg. Frame2freq: Spectral adapters for fine-grained video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24073–24083, June 2026.

[4] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path adaptation from image to video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2203–2213, 2023.

[5] Mengmeng Wang, Jiazheng Xing, Boyuan Jiang, Jun Chen, Jianbiao Mei, Xingxing Zuo, Guang Dai, Jingdong Wang, and Yong Liu. A multimodal, multi-task adapting framework for video action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5517–5525, 2024.

[6] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 2022.

[7] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Ziang Yan, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling foundation models for multimodal video understanding, 2024.

[8] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mahmoud Assran, Koustuv Sinha, Michael Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

[9] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *The Twelfth International Conference on Learning Representations*, 2023.

[10] Mushui Liu, Bozheng Li, and Yunlong Yu. *OmniCLIP: Adapting CLIP for Video Recognition with Spatial-Temporal Omni-Scale Feature Learning*. 2024.

[11] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Vita-clip: Video and text adaptive clip via multimodal prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23034–23044, 2023.

[12] Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H. Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring, 2023.

[13] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models, 2023.

[14] Shuyuan Tu, Qi Dai, Zuxuan Wu, Zhi-Qi Cheng, Han Hu, and Yu-Gang Jiang. Implicit temporal modeling with learnable alignment for video recognition, 2023.

[15] Matthew Kowal, Achal Dave, Rares Ambrus, Adrien Gaidon, Konstantinos G. Derpanis, and Pavel Tokmakov. Understanding video transformers via universal concept discovery. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10946–10956, 2024.

[16] Yumeng Shi, Quanyu Long, Yin Wu, and Wenya Wang. Causality matters: How temporal information emerges in video language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(11):9006–9014, Mar. 2026.

[17] Jisu Nam, Soowon Son, Dahyun Chung, Jiyoung Kim, Siyoon Jin, Junhwa Hur, and Seungryong Kim. Emergent temporal correspondences from video diffusion transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[18] Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[19] Hyeonho Jeong, Chun-Hao P. Huang, Jong Chul Ye, Niloy J. Mitra, and Duygu Ceylan. Track4Gen: Teaching Video Diffusion Models to Track Points Improves Video Generation . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7276–7287, Los Alamitos, CA, USA, June 2025. IEEE Computer Society.

[20] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] Tianxing Wu, Zheng Chen, Cirou Xu, Bowen Chai, Yong Guo, Yutong Liu, Linghe Kong, and Yulun Zhang. Lsgquant: Layer-sensitivity guided quantization for one-step diffusion real-world video super-resolution. *arXiv preprint arXiv:2602.03182*, 2026.

[22] Jiahao Tian, Chenxi Song, Wei Cheng, and Chi Zhang. Free-lunch long video generation via layer-adaptive o.o.d correction, 2026.

[23] Andrea Saltelli. Sensitivity analysis for importance assessment. *Risk analysis : an official publication of the Society for Risk Analysis*, 22:579–90, 07 2002.

[24] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14549–14560, June 2023.

[25] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. Featured Certification.

[26] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked Teacher: Towards Training-Efficient Video Foundation Models . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19891–19903. IEEE Computer Society, October 2023.

[27] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[28] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022.

[29] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011.

[30] Gilles Raîche, Theodore A. Walls, David Magis, Martin Riopel, and Jean-Guy Blais. Non-graphical solutions for cattell's scree test. *Methodology*, 9(1):23–29, 2013.

[31] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 3148–3159, 2022.

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[33] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer, 2022.

[34] Francesco Iodice, Elena De Momi, and Arash Ajoudani. Hri30: An action recognition dataset for industrial human-robot interaction. In *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022.

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[37] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[38] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[39] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021.

[40] Thinesh Thiyakesan Ponbagavathi and Alina Roitberg. Order matters: On parameter-efficient image-to-video probing for recognizing nearly symmetric actions. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17675–17682, 2026.

[41] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[42] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[43] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, 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, 2017.

[44] Wenjie Pei, Qizhong Tan, Guangming Lu, Jiandong Tian, and Jun Yu. D2st-adapter: Disentangled-and-deformable spatio-temporal adapter for few-shot action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11317–11326, October 2025.

[45] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.

[46] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.

[47] Pulkit Kumar, Shuaiyi Huang, Matthew Walmer, Sai Saketh Rambhatla, and Abhinav Shrivastava. Trokens: Semantic-aware relational trajectory tokens for few-shot action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13544–13556, October 2025.

[48] Pulkit Kumar, Namitha Padmanabhan, Luke Luo, Sai Saketh Rambhatla, and Abhinav Shrivastava. Trajectory-aligned space-time tokens for few-shot action recognition. In *European Conference on Computer Vision*, pages 474–493. Springer, 2024.

[49] Shuming Liu, Chen Zhao, Fatimah Zohra, Mattia Soldan, Alejandro Pardo, Mengmeng Xu, Lama Alssum, Merey Ramazanova, Juan León Alcázar, Anthony Cioppa, Silvio Giancola, Carlos Hinojosa, and Bernard Ghanem. Opentad: A unified framework and comprehensive study of temporal action detection. *arXiv preprint arXiv:2502.20361*, 2025.

[50] Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. End-to-end temporal action detection with 1b parameters across 1000 frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18591–18601, June 2024.

A Technical appendix

This appendix provides additional implementation details and analyses supporting the main paper. We first describe the training and evaluation setup in Section A.1, then validate the sensitivity metric through leave-one-out perturbation analysis in Section A.2. We extend our sensitivity analysis to four more Video backbones in Section A.3. Section A.4 reports full spatial–temporal boundary sweeps across adapter families and backbones. Finally, Sections A.5 and A.6 present few-shot results and an extended symmetric/non-symmetric action breakdown with CLIP-based models. To understand generalization to other video understanding tasks, A.7 presents Temporal Action Detection (TAD) results.

A.1 Implementation Details

Backbones and adapter placement. Unless stated otherwise, all experiments use frozen DINOv2-B/14 and CLIP-B/16 as image-pretrained backbones. Adapter modules are inserted into each transformer block following the placement convention of the respective base method: ST-Adapter [1], Frame2Freq [3], and M2-CLIP [5]. STRAP modifies only the layer-wise assignment of spatial and temporal adapters. Temporal adapters use a bottleneck dimension of 384 following the base methods, while spatial adapters use a bottleneck dimension of 64 following AdaptFormer [28]. Only the adapter modules and a linear classifier are trained; the image-pretrained backbone remains frozen throughout.

Optimization. All models are trained with AdamW for 60 epochs, using weight decay 0.001 and label smoothing 0.1. We use the learning rates specified by respective adapters for CLIP and a standard learning rate of 1×10^{-3} for DINOv2 backbone. We apply linear warmup for the first 5 epochs on Diving48, Kinetics-400, and SSv2, and for the first 2 epochs on Drive&Act, IKEA-ASM, and HRI-30. The batch size is 8 per GPU.

Hardware. Training uses 4 NVIDIA A100 GPUs for Drive&Act, IKEA-ASM, and HRI-30, and 16 NVIDIA A100 GPUs for Diving48, Kinetics-400, and SSv2.

Data preprocessing and sampling. All models use inputs at 224×224 resolution. We use 16 frames for Drive&Act, IKEA-ASM, and HRI-30, 8 frames for Kinetics-400, and 32 frames for SSv2 and Diving48. At inference, we use three temporal clips for Drive&Act, IKEA-ASM, HRI-30, and Kinetics-400, and three spatial crops with uniform sampling for Diving48, Kinetics-400, and SSv2. Official dataset splits are used throughout.

Few-shot evaluation. Few-shot action recognition follows the standard N-way K-shot episodic protocol of D2-ST Adapter [44], including the number of evaluation episodes per setting.

Sensitivity analysis. The temporal sensitivity profile $R(l)$ is computed using 500 unlabeled clips per model–dataset pair, sampled from the training split. We use 6 datasets: Kinetics [42], SSv2[43], Diving48 [37], Drive&Act [38], IKEA-ASM [39] and HRI-30 [34]. For each proxy model, we compute $R(l)$ independently for each dataset, normalize the resulting profile per model–dataset pair, and average the profiles across all six evaluation datasets. We then compute $\Delta R(l)$ on the averaged profile and select STRAP$^{\text{opt}}$ as the layer with the maximum increase. This yields a single backbone-specific boundary per proxy model, which is applied uniformly across all downstream tasks without per-dataset tuning. The cost of deriving $R(l)$ is equivalent to standard inference over the sampled clips for the corresponding proxy model and is incurred only once; it does not affect training or evaluation.

A.2 Perturbation Robustness Analysis

We analyze the sensitivity of $R(l)$ to the perturbation set using a leave-one-out analysis. For spatial perturbations, we remove random crop, spatial blur, and color jitter in turn; for temporal perturbations, we remove frame shuffle, frame reversal, and frame drop in turn. Figures 5 and 6 show the resulting profiles for VideoMAE and ViCLIP.

Across all variants, the depth-wise profile of $R(l)$ remains stable. For VideoMAE, the gradual increase toward later layers is preserved under all perturbation subsets. For ViCLIP, the characteristic early suppression followed by late recovery remains visible regardless of which perturbation is removed. The overlaid profiles show tight clustering across leave-one-out variants, indicating that the observed spatial–temporal transition is not driven by a single perturbation choice.

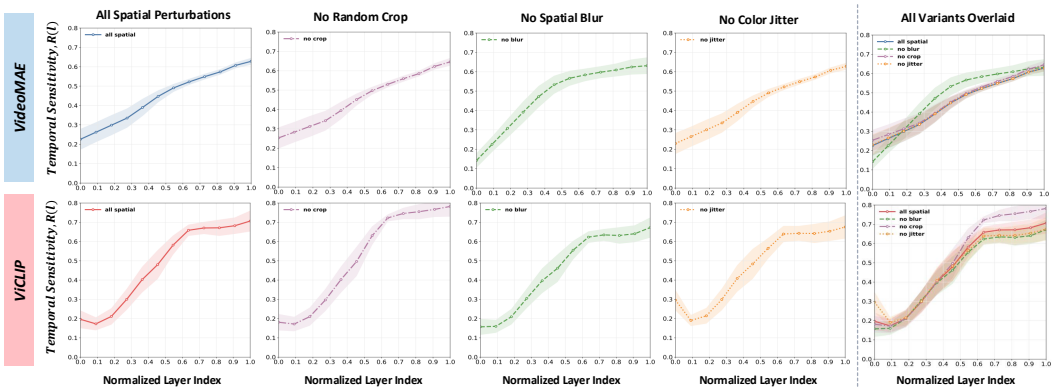


Figure 5: **Impact of spatial perturbation choice on** $R(l)$. Leave-one-out analysis removing each spatial perturbation in turn: random crop, spatial blur, and color jitter. The depth-wise profile and paradigm-specific signatures remain consistent across all variants for both VideoMAE and ViCLIP.

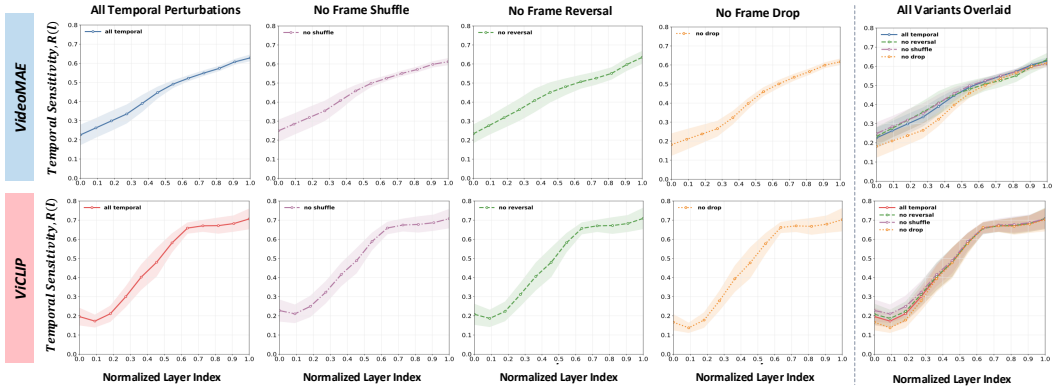


Figure 6: **Impact of temporal perturbation choice on** $R(l)$. Leave-one-out analysis removing each temporal perturbation in turn: frame shuffle, frame reversal, and frame drop. The depth-wise profile and paradigm-specific signatures remain stable across all variants for both VideoMAE and ViCLIP.

A.3 Sensitivity Analysis for Additional Video Backbones

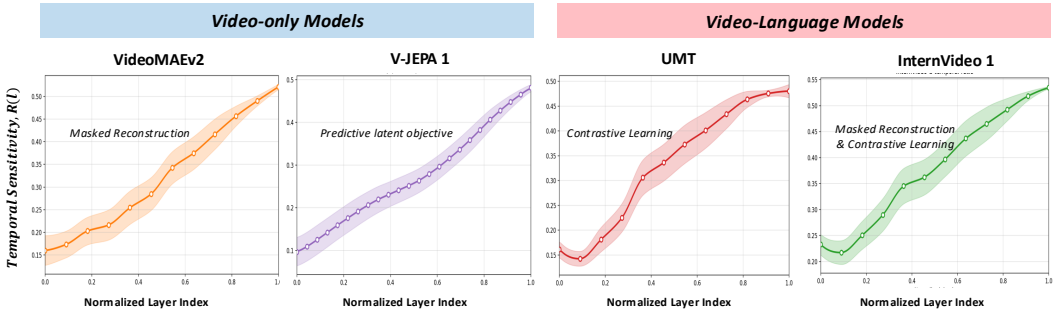


Figure 7: Layer-wise temporal sensitivity $R(l)$ of four additional video foundation models. $R(l)$ increases with depth, and its growth profile depends on the pretraining paradigm.

To check that our findings generalize, we repeat the sensitivity analysis of Sec. 3 on four additional video foundation models: two video-only models, VideoMAEv2 [24] (masked reconstruction) and V-JEPA1 [25] (predictive latent), and two video-language models, UMT [26] (contrastive) and InternVideo 1 [27] (masked reconstruction and contrastive). The setup is unchanged. As Fig. 7 shows,

both findings hold. $R(l)$ increases with depth in all four models (Finding 1). The video-only models rise gradually, whereas the video-language models show an early dip before recovering (Finding 2).

A.4 Full Boundary Sweep Results

To further analyze the effect of the spatial–temporal boundary, we provide full boundary sweeps for all three adapter families: ST-Adapter, Frame2Freq, and M2-CLIP. For each method and backbone, we vary the number of spatial and temporal adapters while keeping the adapter width fixed. This produces configurations ranging from all-spatial adaptation to all-temporal adaptation, with intermediate spatial–temporal splits such as 10S/2T, 8S/4T, 6S/6T, 4S/8T, and 2S/10T. The STRAP-predicted boundary is marked with †.

Across adapter families, the sweep confirms that adapter placement has a strong effect on performance. All-spatial configurations are consistently weak, showing that temporal modeling is necessary for fine-grained video recognition. At the same time, all-temporal placement is also often suboptimal, suggesting that applying temporal modules uniformly across depth can misallocate capacity. Intermediate spatial–temporal splits generally provide the best accuracy–efficiency trade-off.

For DINOv2, the sensitivity-derived 6S/6T split is consistently competitive and often achieves the best or near-best performance across datasets and adapter types. For CLIP, the best region shifts toward a later temporal allocation, with the 4S/8T split performing strongest in most settings. This agrees with the sensitivity analysis: the vision-only proxy predicts a later spatial–temporal transition for DINOv2, while the vision-language proxy predicts an earlier transition for CLIP. Importantly, using the same boundary for both backbones would miss these backbone-specific performance regions.

These sweeps show that STRAP does not rely on a manually tuned downstream split. Instead, the sensitivity-derived boundary provides a strong training-free default that lies close to the empirical performance peak across adapters, backbones, and datasets.

Table 7: Ablation of spatial–temporal adapter allocation across layers for ST-Adapter. We vary the number of spatial (S) and temporal (T) adapters under varying depth (parameter count varies accordingly). Bold indicates the best result per column.

Backbone	Strategy	Params (M)	Drive&Act	IKEA-ASM	HRI-30
DINOv2 (ST-Adapter)	ALL SPATIAL	1.1	73.7	70.2	68.5
	10S/2T	2.2	80.6	75.9	80.6
	8S/4T	3.1	81.1	77.1	87.9
	6S/6T†	4.2	81.9	**77.6**	**89.2**
	4S/8T	5.2	80.9	76.5	88.0
	2S/10T	6.1	**82.6**	75.3	91.1
	ALL TEMPORAL	7.3	75.2	70.5	85.5
CLIP (ST-Adapter)	ALL SPATIAL	1.1	73.4	59.3	66.9
	10S/2T	2.2	79.9	68.8	67.5
	8S/4T	3.1	80.3	71.0	74.8
	6S/6T	4.2	80.1	69.2	79.5
	4S/8T†	5.2	**80.8**	**73.3**	**86.2**
	2S/10T	6.1	80.5	66.8	82.3
	ALL TEMPORAL	7.3	72.6	70.2	81.1

† STRAP-predicted boundary.

A.5 Few-Shot Action Recognition

Datasets. We evaluate few-shot generalization on SSv2-Small [43], HMDB-51 [45], and UCF-101 [46] under 1-shot and 5-shot protocols, following standard few-shot video benchmarks [44, 47]. SSv2-Small emphasizes motion-dependent temporal reasoning, while HMDB-51 and UCF-101 cover broader human actions with varying reliance on temporal structure.

Table 8: Ablation of spatial–temporal adapter allocation across layers for Frame2Freq. We vary the number of spatial (S) and temporal (T) adapters under varying depth (parameter count varies accordingly). Bold indicates the best result per column.

Backbone	Strategy	Params (M)	Drive&Act	IKEA-ASM	HRI-30
DINOv2 (Frame2Freq)	ALL SPATIAL	1.1	73.7	70.2	68.5
	10S/2T	2.2	82.4	77.7	90.1
	8S/4T	3.1	83.1	75.1	90.8
	6S/6T†	4.2	**83.5**	**79.8**	**91.2**
	4S/8T	5.2	82.3	78.6	89.5
	2S/10T	6.1	82.4	78.6	84.6
	ALL TEMPORAL	7.3	81.5	76.4	89.8
CLIP (Frame2Freq)	ALL SPATIAL	1.1	73.4	59.3	66.9
	10S/2T	2.2	80.7	68.4	71.4
	8S/4T	3.1	80.8	72.8	76.4
	6S/6T	4.2	81.0	74.2	83.4
	4S/8T†	5.2	**81.5**	76.5	**86.3**
	2S/10T	6.1	81.4	72.1	82.1
	ALL TEMPORAL	7.3	77.4	**78.0**	85.7

† STRAP-predicted boundary.

Table 9: Ablation of spatial–temporal adapter allocation across layers for M2-CLIP. We vary the number of spatial (S) and temporal (T) adapters under varying depth (parameter count varies accordingly). Bold indicates the best result per column.

Backbone	Strategy	Params (M)	Drive&Act	IKEA-ASM	HRI-30
DINOv2 (M2-CLIP)	ALL SPATIAL	1.1	73.7	70.2	68.5
	10S/2T	3.4	81.8	76.9	88.2
	8S/4T	5.6	82.5	78.7	91.0
	6S/6T†	7.7	**82.9**	**79.6**	**92.3**
	4S/8T	9.9	82.3	79.1	90.2
	2S/10T	12.1	82.5	77.4	91.6
	ALL TEMPORAL	14.8	77.7	74.9	80.5
CLIP (M2-CLIP)	ALL SPATIAL	1.1	73.4	59.3	66.9
	10S/2T	3.4	81.4	70.6	84.4
	8S/4T	5.6	81.9	74.5	91.5
	6S/6T	7.7	82.1	75.6	92.2
	4S/8T†	9.9	**82.9**	**76.2**	**93.1**
	2S/10T	12.1	82.1	74.6	92.8
	ALL TEMPORAL	14.8	77.2	71.9	85.9

† STRAP-predicted boundary.

Results. Table 10 reports Top-1 accuracy for ST-Adapter, Frame2Freq, and M2-CLIP under uniform and STRAP placement using a CLIP backbone. STRAP improves over uniform placement in all 18 few-shot settings, with gains ranging from $+0.2\%$ for Frame2Freq on SSv2-Small and HMDB-51 in the 5-shot setting to $+2.0\%$ for Frame2Freq on HMDB-51 in the 1-shot setting. These consistent gains across adapter designs, datasets, and shot levels suggest that the sensitivity-derived boundary is not a dataset-specific artifact. Instead, it provides a useful placement prior even in low-data regimes, where inefficient or poorly placed adaptation can more easily overfit.

Table 10: **Few-shot action recognition.** STRAP improves over uniform placement in all 18 settings with no degradation, while reducing parameters by 29-33%. State-of-the-art methods are shown for reference; STRAP achieves competitive performance with substantially fewer parameters.

			SSv2-Small		HMDB-51		UCF-101	
Method	**Strategy**	**Params(M)**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
State-of-the-Art Methods								
TATs [48]	-	18.1	47.9	64.4	60.0	77.0	92.0	95.5
Trokens [47]	-	17.4	53.4	68.9	69.8	82.3	94.0	97.9
D2-ST Adapter [44]	-	7.3	55.0	69.3	77.1	88.2	96.4	99.1
Parameter Efficient Fine-tuning Frameworks								
ST-Adapter [1]	Uniform	7.3	53.1	68.0	74.1	87.3	95.7	98.7
	STRAP	**5.2**(↓29%)	**53.6**(+0.5)	**69.1**(+1.1)	**74.9**(+0.8)	**87.8**(+0.5)	**96.0**(+0.3)	**99.2**(+0.5)
Frame2Freq [3]	Uniform	7.3	55.7	71.3	71.6	85.5	93.6	98.6
	STRAP	**5.2**(↓29%)	**56.0**(+0.3)	**71.5**(+0.2)	**73.6**(+2.0)	**85.7**(+0.2)	**95.3**(+1.7)	**99.2**(+0.6)
M2-CLIP [5]	Uniform	14.8	54.1	70.7	73.6	87.4	94.9	99.0
	STRAP	**9.9**(↓33%)	**54.8**(+0.7)	**71.1**(+0.4)	**74.4**(+0.8)	**87.7**(+0.3)	**96.1**(+1.2)	**99.4**(+0.4)

Table 11: **Fine-grained action recognition with symmetric and non-symmetric breakdown.** Sym and N-Sym denote accuracy on symmetric and non-symmetric action subsets, respectively; Top-1 is overall accuracy. STRAP's largest gains concentrate on symmetric actions, which require distinguishing temporal order and direction -precisely the structure targeted by late-layer temporal adapters.

		Drive&Act			IKEA-ASM			HRI-30		
Method	**Strategy**	Sym	N-Sym	Top-1	Sym	N-Sym	Top-1	Sym	N-Sym	Top-1
DINOv2-B/14 backbone (6S/6T, derived from VideoMAE)										
ST-Adapter [1]	Uniform	66.4	83.4	75.2	68.5	73.9	70.5	81.5	95.3	85.5
	STRAP	**78.0**	**85.5**	**81.9**	**80.3**	**76.2**	**77.6**	**85.4**	**96.8**	**89.2**
Frame2Freq [3]	Uniform	77.1	85.5	81.5	81.0	74.2	76.4	85.6	**98.9**	89.8
	STRAP	**79.6**	**87.1**	**83.5**	**86.8**	**76.4**	**79.8**	**88.2**	97.2	**91.2**
M2-CLIP [5]	Uniform	68.4	85.0	77.7	75.3	74.6	74.9	78.4	84.6	80.5
	STRAP	**78.8**	**86.6**	**82.9**	**83.7**	**77.6**	**79.6**	**89.3**	**98.2**	**92.3**
CLIP-B/16 backbone (4S/8T, derived from ViCLIP)										
ST-Adapter [1]	Uniform	55.5	**88.4**	72.6	70.3	70.7	70.4	73.8	95.7	81.1
	STRAP	**73.5**	87.3	**80.8**	**80.0**	**70.8**	**73.3**	**80.0**	**98.5**	**86.2**
Frame2Freq [3]	Uniform	64.4	**88.6**	77.4	**83.9**	**75.1**	**78.0**	79.1	**98.9**	85.7
	STRAP	**75.1**	87.4	**81.5**	83.3	73.8	76.5	**80.4**	98.2	**86.3**
M2-CLIP [5]	Uniform	67.4	85.7	77.2	76.2	72.6	71.9	81.8	95.1	85.9
	STRAP	**76.9**	**88.4**	**82.9**	**82.5**	**73.2**	**76.2**	**90.2**	**98.9**	**93.1**

A.6 Symmetric Action Recognition

Table 11 extends the symmetric and non-symmetric action breakdown from the main paper by including CLIP-B/16 results in addition to DINOv2-B/14.

The trend observed with DINOv2 also holds for CLIP: STRAP produces its largest gains on symmetric action subsets, where classes share similar appearance but differ in temporal order or direction. For CLIP with ST-Adapter, symmetric accuracy improves by +18.0% on Drive&Act (55.5→73.5) and +9.7% on IKEA-ASM (70.3→80.0). For CLIP with M2-CLIP, symmetric accuracy improves by +9.5% on Drive&Act and +6.3% on IKEA-ASM. These gains are generally larger than those on non-symmetric subsets, where appearance cues are more discriminative.

There are a few exceptions. With CLIP and Frame2Freq, STRAP improves symmetric accuracy on Drive&Act and HRI-30, but drops on IKEA-ASM, matching the overall accuracy drop for this

adapter–backbone pair in Table 1. With CLIP and ST-Adapter, non-symmetric accuracy on Drive&Act decreases slightly, suggesting that the placement shift can redistribute capacity toward temporal disambiguation at a small cost to appearance-driven classes.

Overall, the symmetric-action results support the main claim of STRAP: concentrating temporal adapters in later layers improves recognition most when temporal order is class-discriminative.

A.7 Temporal Action Detection on THUMOS14

To test whether the STRAP boundary transfers beyond action recognition, we apply it to temporal action detection on THUMOS14 using the OpenTAD [49] framework with a VideoMAE [6] backbone at the standard 768-frame input. We use the derived boundary directly, without any new sensitivity analysis, and train all three configurations under identical settings. We use a video-pretrained backbone rather than an image backbone, which would have required additional hyperparameter tuning. STRAP achieves a higher average mAP than uniform placement with fewer parameters, and is ahead at 4 of 5 tIoU thresholds. Gains are small, which is consistent with THUMOS14 not being a fine-grained benchmark (Sec. 5.6).

Table 12: **Temporal action detection on THUMOS14** (VideoMAE, OpenTAD). Average mAP and mAP at each tIoU threshold. Best result per accuracy column in bold.

		mAP @ tIoU					
Config	Avg	0.3	0.4	0.5	0.6	0.7	Params
Spatial-only [28]	69.03	84.76	79.93	72.34	61.53	46.62	1.2M
Uniform [50]	70.36	85.58	**81.32**	73.96	61.95	49.00	4.0M
STRAP	**70.75**	**86.06**	80.88	**74.41**	**63.05**	**49.34**	3.1M

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to the end of the introduction section to find the list of contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to section 7

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification:[N/A]

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Refer to Section 5.1 and appendix A.1

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: GitHub link provided in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Refer to Section 5.1 and A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments are too expensive to repeat many times (Appendix A.1).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Refer to A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Only open-source datasets are used and there are no ethics violations

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use open-sourced datasets and model weights with appropriate citations.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used only for editing and formatting purposes.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.