GEET: GATED EXPLICIT-MOTION EXTRACTION TEMPO-SPATIAL ADAPTER

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot fine-grained (FS-FG) action recognition requires distinguishing visually similar actions that differ primarily in subtle temporal dynamics from only a few labeled examples. Existing foundation-model adaptations often rely on broad spatio-temporal updates, introducing excessive adaptation capacity for this setting. We propose **GEET** (**G**ated **E**xplicit-motion **E**xtraction **T**empo-spatial Adapter), a parameter-efficient adapter for a frozen CLIP backbone that prioritizes temporal specialization while limiting spatial adaptation. GEET separates appearance evolution from explicit frame-to-frame motion and combines them through gated temporal fusion, with lightweight spatial refinement grounding the explicit motion signal in locally coherent spatial evidence. This focused design requires only $\sim$3.8M tunable parameters and video input alone, GEET achieves state-of-the-art FS-FG recognition across diverse benchmarks with substantially lower latency and higher throughput, while remaining robust to diverse visual corruptions and temporal disruptions. Source code is attached in the suppl. material.

1 INTRODUCTION

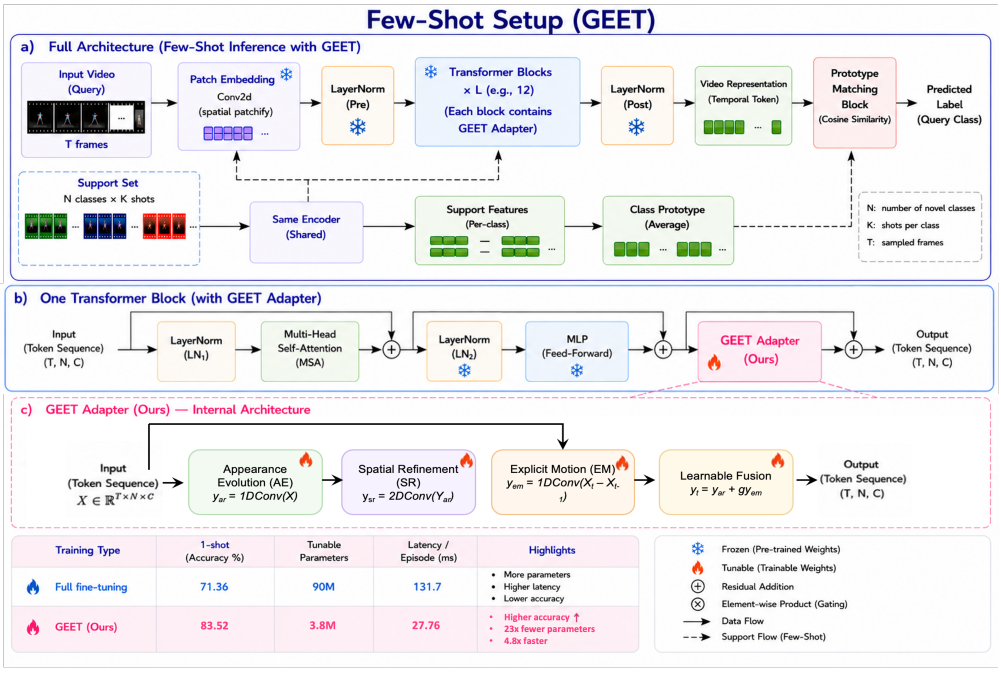

Figure 1: **Overview of the GEET and few-shot setup**. Support and query videos are encoded by a shared frozen CLIP backbone, and query labels are predicted via prototype matching. Compared to full fine-tuning Wang et al. (2024), GEET updates only a small set of parameters, improving accuracy while reducing latency and demonstrating robustness.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Few-shot action recognition (FS-AR) aims to recognize novel action classes from only a few labeled videos (Snell et al., 2017; Thatipelli et al., 2022). The problem is particularly challenging for *few-shot fine-grained* (FS-FG) action recognition (Tang et al., 2023), where classes often share similar actors, objects, and scenes but differ in brief and subtle temporal patterns (Shao et al., 2020; Li et al., 2018; Xu et al., 2022). For example, fine-grained sports actions can be distinguished by small differences in motion order, timing, or phase, while only a few labeled examples are available. Consequently, effective FS-FG recognition requires capturing discriminative temporal changes under limited supervision rather than relying primarily on static appearance.

Conventional FS-AR methods typically rely on video backbones and task-specific matching or temporal modeling modules trained for the target recognition setting (Cao et al., 2020; Thatipelli et al., 2022; Tang et al., 2023). While effective, learning such representations under limited supervision is challenging, particularly for FS-FG recognition where the discriminative temporal cues are subtle. This motivates leveraging foundation models pretrained on large-scale data: their transferable representations provide a stronger starting point, allowing few-shot learning to focus on adapting existing visual features rather than learning the representation from limited labeled videos.

Foundation models such as CLIP (Radford et al., 2021) provide strong transferable visual representations, motivating their adaptation to few-shot video recognition. Recent approaches span full or parameter-efficient CLIP adaptation (Wang et al., 2024; Pei et al., 2025; Cao et al., 2026), trajectory-based representations (Kumar et al., 2024; 2025), and methods that exploit additional modalities such as text or pose (Zhang et al., 2024; Xing et al., 2026). However, FS-FG recognition requires learning subtle temporal differences from very limited supervision, making heavy adaptation, additional preprocessing, or auxiliary modalities less desirable. We therefore consider a practical setting that seeks high recognition accuracy with few tunable parameters, efficient computation, and the video modality alone.

Research gap. Despite recent progress, existing approaches do not jointly address the practical requirements of FS-FG recognition. Conventional methods often learn task-specific video representations from limited data, while recent foundation-model approaches may introduce substantial adaptation capacity, additional modalities, or preprocessing. An effective FS-FG model should instead leverage strong pretrained visual priors, adapt them with few parameters and low computational overhead, and operate from video alone. Moreover, practical deployment requires maintaining reliable recognition when visual quality is degraded or temporal structure is disrupted.

These considerations lead to set of questions: *Can strong foundation-model priors be efficiently adapted to capture subtle temporal differences in few-shot fine-grained actions using few tunable parameters and video alone? And can such adaptation maintain reliable recognition under visual and temporal degradations relevant to practical deployment?*

To address these questions, we propose **GEET** (**G**ated **E**xplicit-motion **E**xtraction **T**empo-spatial Adapter), a parameter-efficient adapter for a frozen CLIP backbone. GEET models temporal information through two complementary pathways: an appearance-evolution branch that captures local feature evolution through depthwise temporal filtering, and an explicit-motion branch that applies the same filtering to first-order frame-to-frame feature differences. A learnable scalar gate controls the contribution of explicit motion, while a lightweight spatial refinement provides local adaptation after temporal modeling. These components are designed to operate jointly rather than independently: explicit motion is applied on top of appearance evolution and spatial refinement, which provide the temporal context and local spatial evidence that ground the difference signal. Operating within a compact bottleneck and keeping CLIP frozen enables temporal specialization with few trainable parameters and low computational overhead.

With only $\sim$3.8M tunable parameters, GEET outperforms recent few-shot methods such as D^2ST and Task-Adapter++, improving 1-shot accuracy over D^2ST by 7.1–10.9 points on the main fine-grained benchmarks while providing $\sim 4\times$ lower latency and higher throughput. GEET demonstrates noticeable robustness to visual corruptions and temporal disruptions, outperforming recent few-shot methods by 5.8–11.7 points in mean corrupted accuracy. Our contributions are:

- **Temporal-first adaptation.** We show that FS-FG recognition benefits from concentrating foundation-model adaptation on temporal dynamics rather than broadly adapting spatio-temporal features.

- **Parameter-efficient GEET adapter.** We introduce GEET, a compact video-only adapter that models temporal evolution and explicit motion within a low-dimensional bottleneck, while restricting spatial adaptation to lightweight refinement, requiring only $\sim$3.8M tunable parameters.
- GEET achieves state-of-the-art FS-FG recognition with substantially lower inference latency and higher throughput, while maintaining strong robustness against diverse visual corruptions and temporal disruptions.

2 Related Work

We organize the related work into three areas most relevant to our work:

Vision–language pretraining for video actions. CLIP (Radford et al., 2021) provides open-vocabulary visual representations that have been extended to video by ActionCLIP (Xu et al., 2021), X-CLIP (Ni et al., 2022), ViFi-CLIP (Rasheed et al., 2023), and CAST (Lee et al., 2023) through video–text alignment, cross-frame attention, temporal alignment, and spatial–temporal cross-attention. These rely on full backbone fine-tuning or heavy video-specific modules, which is inefficient for fine-grained actions with rapid, subtle motion.

Parameter-efficient video adaptation. Following lightweight adapters in NLP (Houlsby et al., 2019), ST-Adapter (Pan et al., 2022), DUALPATH (Park et al., 2023), D^{2}ST (Pei et al., 2025), and AIM (Yang et al., 2023) adapt ViT backbones through spatial–temporal modules built from multi-branch or attention-based designs, while Task-Adapter (Cao et al., 2024) and Task-Adapter++ (Cao et al., 2026) add task-specific and order-aware alignment. All jointly model spatial appearance and temporal structure. We instead target fine-grained recognition, where short-duration motion is the main discriminative signal, through focused temporal adaptation without additional spatial modeling.

Few-shot action recognition. Classical FSAR methods such as ProtoNet (Snell et al., 2017), OTAM (Cao et al., 2020), TRX (Perrett et al., 2021), PAL (Zhu et al., 2021), STRM (Thatipelli et al., 2022), and HyRSM (Wang et al., 2022b) learn global similarity metrics or temporal alignments, while VPD (Hong et al., 2021) distills pose supervision into RGB features. Such cues are less suited to actions separated by brief motion evolution rather than coarse appearance. CLIP-based methods exploit stronger semantic priors: CLIP-FSAR (Wang et al., 2024) fine-tunes the full backbone with temporal transformers, and MA-FSAR (Xing et al., 2026) freezes it but adds multi-modal and meta-learning components. Others refine matching or temporal structure: TEAM (Lee et al., 2025) removes explicit temporal alignment, MPL (Yang et al., 2026) learns match-guided prototypes, TSAM (Li et al., 2025) models frame order as a sequence, HMP-Net (Chen et al., 2025) builds hierarchical meta-prototypes, and EMP-Net (Wu et al., 2024) reduces cost via multi-level post reasoning. TATs (Kumar et al., 2024) and Trokens (Kumar et al., 2025) instead represent motion as trajectory tokens, which is effective on fine-grained data but requires an offline extraction stage and depends on tracker quality.

In contrast, GEET retains a frozen CLIP visual backbone and introduces a compact temporal-first adapter that explicitly separates appearance evolution from frame-to-frame motion, followed by minimal spatial refinement. This design avoids attention-heavy modules, auxiliary modalities, and offline preprocessing, while requiring only a small number of tunable parameters.

3 GEET Adapter

We propose GEET, a parameter-efficient adapter for temporal adaptation of a frozen CLIP ViT backbone (Fig. 1). Fine-grained actions often share similar actors, objects, and scenes, while differing in subtle motion patterns and their temporal evolution. This distinction becomes particularly challenging in the few-shot setting, where only a few labeled examples are available to learn such class-specific motion cues. GEET therefore prioritizes temporal modeling while keeping the adaptation parameter-efficient to reduce overfitting under limited supervision. In particular, it separates slowly varying appearance evolution from explicit frame-to-frame motion, while using only minimal spatial refinement. GEET is inserted into each transformer block as a residual adapter: it is applied to the block output after the MLP sub-layer and its update is added back residually (Fig. 1b),

while the CLIP backbone remains frozen. Let B denote the batch size, T the number of input frames, $H \times W$ the spatial grid of patch tokens, and C the channel dimension. Given patch tokens $\mathbf{X}_p \in \mathbb{R}^{HW\times(BT)\times C}$ (excluding the class token, which the adapter leaves unchanged), we first rearrange the frame-wise tokens into their spatio-temporal organization by permuting and reshaping the batch, frame, and spatial axes:

$$\mathbf{V} = \mathrm{rearrange}(\mathbf{X}_p) \in \mathbb{R}^{B\times C\times T\times H\times W}. \tag{1}$$

This representation exposes the temporal axis explicitly, allowing GEET to operate across frames while preserving the spatial correspondence of patch tokens.

Bottleneck residual adaptation. Direct adaptation in the full C-dimensional feature space introduces unnecessary trainable capacity in the few-shot setting. We therefore perform the temporal operations in a lower-dimensional bottleneck. Specifically, an FC-down projection $\phi_\downarrow$ reduces the channel dimension to $C_\alpha = \lfloor \alpha C \rfloor$:

$$\mathbf{U} = \phi_\downarrow(\mathbf{V}), \qquad \phi_\downarrow : \mathbb{R}^C \to \mathbb{R}^{C_\alpha}. \tag{2}$$

The projection is applied independently at every temporal and spatial location; hence, it changes only the channel dimension and does not mix information across frames or patches. After temporal and spatial adaptation, denoted collectively by $\mathcal{G}(\cdot)$, an FC-up projection $\phi_\uparrow$ maps the bottleneck features back to the original CLIP dimension. The resulting update is added residually to the input:

$$\hat{\mathbf{V}} = \mathbf{V} + \phi_\uparrow(\mathcal{G}(\mathbf{U})), \qquad \phi_\uparrow : \mathbb{R}^{C_\alpha} \to \mathbb{R}^C. \tag{3}$$

Thus, GEET learns a low-dimensional correction to the frozen representation rather than replacing it. This residual formulation preserves the original CLIP feature pathway while allowing task-specific temporal information to be introduced through the adapter.

Temporal feature decomposition. Fine-grained actions can differ through short temporal changes despite having similar actors, objects, and scenes. We therefore model temporal information through two complementary branches: **appearance evolution (AE)**, which operates on the feature sequence itself, and **explicit motion (EM)**, which operates on changes between consecutive frames. For appearance evolution, we apply a depthwise one-dimensional convolution along the temporal dimension:

$$\mathbf{A} = \mathrm{DWConv}_t(\mathbf{U}). \tag{4}$$

Here, DWConv_t processes each bottleneck channel independently along time at every spatial location. Consequently, $\mathbf{A}$ captures local temporal evolution without introducing spatial or cross-channel mixing. To make frame-to-frame changes explicit, we compute the first-order temporal difference

$$\Delta\mathbf{U}_t = \mathbf{U}_t - \mathbf{U}_{t-1}, \qquad \Delta\mathbf{U}_1 = \mathbf{0}, \tag{5}$$

and apply a second depthwise temporal convolution:

$$\mathbf{M} = \mathrm{DWConv}_t(\Delta\mathbf{U}). \tag{6}$$

The differencing operation suppresses components that remain unchanged between adjacent frames and emphasizes local feature transitions. $\mathbf{M}$ therefore provides an explicit representation of short-term motion changes, complementary to the temporal context retained in $\mathbf{A}$.

Gated temporal fusion. The two temporal branches are combined using a learnable scalar gate g:

$$\mathbf{Z} = \mathbf{A} + g\,\mathbf{M}. \tag{7}$$

We initialize $g = 0$, so the motion branch contributes no residual at the beginning of optimization. During training, g is learned jointly with the adapter parameters and controls the contribution of explicit motion features. Using a scalar gate introduces only one additional parameter per adapter and provides a simple mechanism for balancing the two temporal representations. The resulting decomposition can also be interpreted from a temporal filtering perspective. The appearance branch processes the original feature trajectory, whereas the motion branch processes its first-order difference. The former retains local temporal context, while the latter emphasizes temporal changes. Their gated combination therefore exposes both feature evolution and explicit motion transitions to the subsequent refinement stage.

Local spatial refinement (SR). After temporal fusion, we apply a lightweight spatial operation to capture local relationships between neighboring patches. For each frame independently, a depthwise 3×3 convolution is applied over the spatial dimensions and added residually:

$$\mathbf{Z}' = \mathbf{Z} + \mathrm{DWConv}_s(\mathbf{Z}), \qquad \mathcal{G}(\mathbf{U}) := \mathbf{Z}'. \tag{8}$$

Because DWConv_s is depthwise, it operates independently on each channel and introduces no additional cross-channel mixing. Its 3×3 receptive field restricts spatial adaptation to local neighborhoods. Placing this operation after temporal fusion further ensures that spatial refinement acts on the temporally adapted representation rather than directly on the original appearance features. Finally, $\mathbf{Z}'$ is projected back to C dimensions through Eq. 3 and added to the frozen CLIP features. GEET therefore follows a compact sequence of operations: channel reduction, appearance and motion modeling along time, gated temporal fusion, local spatial refinement, and residual projection to the backbone representation. All convolutions inside the adapter are depthwise, while channel interaction is confined to the two bottleneck projections. This structure provides explicit temporal modeling with limited trainable capacity, which is particularly suitable for few-shot fine-grained action recognition.

Grounding explicit motion as a design choice. The three components are therefore not independent additions, but form a unified design: *appearance evolution* (AE) establishes the temporal context, *spatial refinement* (SR) consolidates it into locally coherent frame-wise evidence, and *explicit motion* (EM) complements both by emphasizing first-order temporal changes within the temporally and spatially refined representation. EM is thus designed to operate jointly with AE and SR, and its contribution is expected to emerge from their complementary interaction rather than as a standalone pathway.

Pseudocode for episodic training/inference with GEET is in Sec. H (Algorithms 1–2) in Appendix.

4 Experiments

Benchmarks. We evaluate GEET on standard few-shot fine-grained action benchmarks: Diving48 (Li et al., 2018), FineGym99/288 (Shao et al., 2020), Toyota SmartHome (Das et al., 2019), MPII Cooking 2 (Rohrbach et al., 2016), and HAA500 (Chung et al., 2021). For Diving48 and FineGym, we adopt the class splits from M^{3}Net (Tang et al., 2023) for fair comparison; for the rest, we construct episodic splits with a 70%/10%/20% train/val/test partition.

Train/test protocol. We follow the episodic 5-way meta-learning protocol of M^{3}Net (Tang et al., 2023), training for 60,000 episodes on Diving48/FineGym. Each episode has a support and query set; we freeze CLIP and optimize only the adapter on the query loss, using batch size 8, standard video augmentations, and NVIDIA H100 GPUs, with 8 frames per video via segment-based sampling. At test time, we evaluate 10,000 episodes under 1/3/5-shot settings and report Top-1 accuracy with single-view inference. Implementation details are in Sec. G in Appendix.

5 Results and Discussion

5.1 GEET establishes state-of-the-art with notable gains

GEET consistently improves few-shot fine-grained recognition across *Diving48*, *FineGym99*, and *FineGym288* in Table 1, with particularly strong gains in the challenging 1-shot setting on the CLIP ViT-B/16 backbone.

Trokens, the closest competitor, uses a stronger DINOv2 backbone and offline trajectory extraction, requiring additional preprocessing, storage, and dependence on tracker quality. In contrast, GEET operates directly on video with a frozen CLIP ViT-B/16 and few tunable parameters, while its gains extend beyond sports-centric benchmarks to *Toyota SmartHome*, *HAA500*, and *MPII Cooking 2* (Table 2), demonstrating consistent performance across diverse fine-grained action domains.

5.2 GEET achieves the accuracy–compute efficiency–unimodality objective

The accuracy–compute efficiency–unimodality objective asks whether a few-shot video model can improve recognition while remaining compact, fast, and single-stream (video only). We therefore

Table 1: **Few-shot fine-grained action recognition on Diving48, FineGym99, and FineGym288** (%) under 5-way 1/5-shot settings. Best results are in bold. † *denotes results reproduced in the paper. "*" denotes multi-modal methods incorporating additional text input and a text encoder.* ϕ *denotes additional pre-processing. "§" denotes tunable parameters changes depending on the number of classes and the component* $(3.68M - 3.89M)$.

Method	Venue	Tunable Params (M)	Diving48		FineGym99		FineGym288	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
TSN+Cosine (Wang et al., 2016)	ECCV'16	25.5	29.72	38.81	36.63	42.76	36.68	44.82
ProtoNet (Snell et al., 2017)	NeurIPS'17	26.8	59.80	77.69	62.99	77.99	58.98	74.87
OTAM (Cao et al., 2020)	CVPR'20	23.5	41.64	49.78	48.93	55.61	47.58	55.62
PAL (Zhu et al., 2021)	BMVC'21	26.9	50.40	61.17	58.85	68.64	58.32	65.77
TRX (Perrett et al., 2021)	CVPR'21	47.1	62.53	82.01	66.55	83.86	64.03	81.05
STRM (Thatipelli et al., 2022)	CVPR'22	30.8	62.29	81.03	68.09	84.33	64.81	80.64
HyRSM (Wang et al., 2022b)	CVPR'22	65.6	61.04	81.74	67.09	82.92	62.86	78.81
MoLo (Wang et al., 2023)	CVPR'23	89.6	-	-	73.3	84.8	-	-
TATs (Kumar et al., 2024)	ECCV'24	10.8	-	-	81.8	87.9	-	-
M^{3}Net (Tang et al., 2023)	MM'23	30.0	72.35	85.44	72.70	86.71	68.66	83.05
DINOv2 Backbone								
Trokens (Kumar et al., 2025)$^{\phi}$	ICCV'25	17.4	67.20†	81.30†	**84.40**	89.90	-	-
CLIP ViT-B/16 Backbone								
D^{2}ST (Pei et al., 2025)†	ICCV'25	7.3	73.29	83.85	76.81	87.55	70.33	82.08
EMP-Net (Wu et al., 2024)†*	ECCV'24	7.5	71.00	69.20	-	-	-	-
Task-Adapter (Cao et al., 2024)†*	MM'24	7.2	68.04	75.09	64.48	70.54	-	-
Task-Adapter++ (Cao et al., 2026)†*	PR'26	8.0	46.45	64.17	60.82	68.12	63.07	72.76
TEAM (Lee et al., 2025)†	CVPR'25	87.0	38.50	54.90	-	-	-	-
MPL (Yang et al., 2026)†	CVPR'26	14.0	39.29	48.70	-	-	-	-
GEET (**AE**)	–	3.68 §	79.36	88.48	80.55	89.60	79.16	89.90
GEET (**AE** + **SR**)	–	3.72 §	81.15	89.77	81.90	90.90	80.52	89.84
GEET (**AE** + **SR** + **EM**)	–	3.80 §	**83.52** ±0.16	**90.24** ±0.14	83.88 ±0.19	**91.62** ±0.13	**81.18** ±0.17	**90.16** ±0.18

Table 2: **Few-shot fine-grained action recognition on additional datasets** (%)

Method	ToyotaSH		HAA500		MPII Cooking 2	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Task-Adapter++ (Cao et al., 2026)†	62.39	76.30	97.05	99.10	59.85	70.00
D^{2}ST (Pei et al., 2025)†	60.25	73.18	96.52	98.78	57.23	71.16
GEET (**AE**)	65.10	78.03	96.60	98.83	60.03	72.50
GEET (**AE** + **SR**)	64.14	77.08	96.71	98.96	60.27	71.94
GEET (**AE** + **SR** + **EM**)	**66.34**	**78.96**	**97.18**	**99.13**	**62.08**	**73.88**

compare episodic inference efficiency between GEET and D^{2}ST under an identical few-shot protocol on a single NVIDIA H100 80GB HBM3 GPU using FP16. FLOPs are measured with `fvcore`, and latency with CUDA events over timed blocks of B=8 consecutive meta-forwards, each episode using a 5-way 1-shot setting with T=8 frames (40 support and 40 query frame-rows). Ablations on adapter depth, frame count, and shot number are Sec. A in Appendix.

Table 3 shows that GEET achieves higher accuracy with substantially lower latency and higher throughput, despite similar FLOPs per episode. This highlights that FLOPs alone do not capture practical inference efficiency: GEET's lightweight temporal-first adaptation is considerably more efficient at runtime while operating on the video modality alone.

5.3 GEET IS SCALABLE AND APPLICABLE TO THE FULLY-SUPERVISED APPROACH

Backbone scalability. Table 5 swaps CLIP ViT-B/16 for the larger ViT-L/14 (3.80M → 12.79M adapter parameters), with no architectural changes. Accuracy improves consistently across all three

Table 3: **GEET enables compute-efficient few-shot adaptation**, with lower latency and higher throughput. † *denotes results reproduced by us;* ↓ *lower is better and* ↑ *higher is better.*

Method	1-shot Acc. (%) ↑	GFLOPs Episode ↓	Latency Episode (ms) ↓	Throughput (episodes/s) ↑	Effective Compute (TFLOP/s) ↑
D^2ST (Pei et al., 2025) †	73.29	2990.0	110.25	9.07	26.92
Task-Adapter++ (Cao et al., 2026) †	68.04	3335.95	68.54	14.60	48.67
EMP-Net (Wu et al., 2024) †	71.00	2964.24	128.16	7.80	23.12
GEET	**83.52**	**2921.3**	**27.76**	**36.03**	**105.28**
Gain (GEET vs. D^2ST)	↑ +10.23	↓ 2.3%	↓ 4.00×	↑ 3.97×	↑ 3.91×

Table 4: **Fully-supervised results on Diving48.** Top-1 accuracy under the standard fully-supervised protocol, included to show that GEET also applies beyond the few-shot setting. † *denotes results reproduced*

Method	**Venue**	**Tunable Params (M)**	**Frames**	**Top-1**
Non-CLIP-based methods				
ORViT TimeSformer (Herzig et al., 2022)	CVPR'22	160.0	32	88.0
BEVT (Wang et al., 2022a)	CVPR'22	88.1	16	86.7
VideoSwin-B (Liu et al., 2022b)	CVPR'22	88.1	32	81.9
SlowFast R101 (Feichtenhofer et al., 2019)	ICCV'19	62.8	64+16	77.6
TQN (Zhang et al., 2021)	CVPR'21	–	all	81.8
CLIP-based methods ViT-B/16				
BDC-CLIP (Long et al., 2025) †	ICML'25	125.4	16	76.6
PeVL (Zhang et al., 2024)	CVPR'24	28.0	16	84.5
DiST (Qing et al., 2023) †	ICCV'23	19.0	16	76.85
AIM (Yang et al., 2023) †	ICLR'23	11.0	32	88.9
DualPath (Park et al., 2023) †	CVPR'23	10.0	32	88.7
ST Adapter (Pan et al., 2022) †	NeurIPS'22	7.2	16	84.2
TDS-CLIP (Wang et al., 2026) †	TCSVT'26	24.1	16	88.84
GEET (ours)	–	3.80	16	85.8 ± 0.13
GEET (ours)	–	3.80	32	$\mathbf{89.7 \pm 0.15}$

benchmarks and both 1-/5-shot settings, confirming GEET's gains stem from its temporal specialization mechanism and scale gracefully with backbone capacity. **Fully-supervised recognition.** On Diving48 (Table 4), GEET achieves the highest top-1 accuracy among CLIP-based methods, and matches or exceeds all non-CLIP baselines, using only 3.80M tunable parameters, far fewer than any competitor. The same holds on FineGym99/288 (Appendix, Sec. C). GEET's temporal-first adaptation is thus not a few-shot-specific artifact.

5.4 Robustness against visual corruptions

We evaluate robustness to pixel-level corruptions of query frames using ImageNet-C (Hendrycks & Dietterich, 2019) operators applied per frame.

Protocol. We evaluate all methods on *1000 episodes of Diving48 5-way 1-shot* (8 frames; seed 18), corrupting only query clips while keeping support clips clean. We use the 15 ImageNet-C corruptions plus Gaussian blur, grouped into *noise* (Gaussian, shot, impulse), *blur* (defocus, glass, motion, zoom, Gaussian), *weather* (snow, frost, fog, brightness), and *digital* (contrast, elastic transform, pixelate, JPEG). We report results averaged over severities $\{1, 3, 5\}$. Blur and weather corruptions are shared across frames within a clip to avoid artificial flicker, whereas noise and JPEG are sampled per frame.

Let $\bar{A}_{m,c}$ denote the top-1 accuracy of method m under corruption c, averaged over severities $\{1, 3, 5\}$, and A_m^{clean} its clean accuracy. We report *mean Corruption Accuracy* (mCA), $\text{mCA}_m = \frac{1}{16}\sum_c \bar{A}_{m,c}$, and *relative Performance under Corruption* (rPC), $\text{rPC}_{m,c} = \bar{A}_{m,c}/A_m^{\text{clean}}$. Following ImageNet-C, we additionally report *mean Corruption Error* (mCE), using Task-Adapter++ as the reference:

$$\text{mCE}_m = \frac{100}{16}\sum_{c=1}^{16} \frac{100 - \bar{A}_{m,c}}{100 - \bar{A}_{\text{ref},c}}. \tag{9}$$

Task-Adapter++ therefore has $\text{mCE} = 100$, with lower values indicating lower corruption error.

Table 5: **Backbone scalability of GEET.** Top-1 accuracy (%) under 1-shot and 5-shot settings with frozen CLIP backbones. Gains carry over to the larger backbone across all three benchmarks.

GEET	Params	Diving48		FineGym99		FineGym288	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ViT-B/16	3.80	83.52	90.24	83.88	91.62	81.18	90.16
ViT-L/14	12.79	**84.58**	**90.89**	**84.67**	**93.58**	**81.67**	**91.72**

Table 6: **Robustness against visual corruptions** on Diving48 (5-way 1-shot). Values in parentheses denote rPC; mCE is normalized by Task-Adapter++ (mCE = 100). *Performance reported here on 1000 test episodes for all methods*.

Method	Clean Top-1 ↑	Corrupted queries (mean Top-1 over severities 1/3/5)↑				mCA (%) ↑	mCE ↓
		Noise	Blur	Weather	Digital		
D^{2}ST	72.4	49.9 (0.69)	48.0 (0.66)	61.1 (0.84)	59.2 (0.82)	54.4	68.4
Task-Adapter++	46.1	31.2 (0.68)	32.6 (0.71)	35.1 (0.76)	36.4 (0.79)	33.9	100.0
GEET	**80.6**	**62.8** (0.78)	**58.8** (0.73)	**73.0** (0.91)	**70.8** (0.88)	**66.1**	**50.9**

As shown in Table 6, GEET achieves the highest accuracy across all corruption groups and the best overall mCA (66.1) and mCE (50.9), with particularly large gains under noise and blur. Since mCA is an absolute measure, part of this advantage follows from GEET's higher clean accuracy; however, GEET also retains the largest fraction of its clean score in every corruption group (rPC in Table 13 in Sec. E in Appendix), so the gain does not stem from the clean model alone. These results indicate that GEET's lightweight temporal-first adaptation remains effective when spatial appearance is degraded. Per-type and per-severity results are reported in Sec. E in Appendix.

5.5 Robustness against temporal disruptions

We evaluate robustness to changes in query-frame order and timing following (Schiappa et al., 2023).

Protocol. We use the same Diving48 5-way 1-shot protocol as Section 5.4, with clean support clips and corrupted queries. Five disruptions are applied to the sampled 8-frame clips: *sampling rate* (faster/slower playback), *reversal* (local segments played backward), *freezing* (repeated frames), *jumbling* (frames shuffled within local segments), and *box jumbling* (segments shuffled while preserving local motion). Jumbling segment sizes are scaled to the 8-frame setting. Results are averaged over severities $\{1, 3, 5\}$. We report mCA, rPC, and mCE as above, with mCE averaged over the five temporal disruptions and Task-Adapter++ as reference.

Table 7 shows that GEET achieves the highest absolute accuracy under all five temporal disruptions, with an mCA of 73.4 and mCE of 48.8. Although D^{2}ST and Task-Adapter++ retain a larger fraction of their clean accuracy under some disruptions, their absolute corrupted accuracy remains lower. Overall, GEET's gains on clean fine-grained recognition persist under diverse perturbations of temporal order and timing. Per-severity results are reported in Sec. F in Appendix (Table 17).

5.6 Why GEET captures fine-grained details?

To examine whether GEET's gains arise from more temporally focused adaptation, we analyze its temporal saliency across fine-grained action sequences. Additional analysis on nearest-neighbor, single-frame blackout sensitivity analysis are provided in the Sec. B in Appendix.

5.6.1 Qualitative diagnostic on temporal saliency.

To examine how GEET allocates adaptation capacity over time, we measure the per-frame magnitude of the adapter's residual update. Let $\Delta X_{\text{patch}} \in \mathbb{R}^{HW\times(BT)\times C}$ denote the adapter's residual update inside a transformer block; the update magnitude at frame t is

$$s(t) \;=\; \frac{1}{HW}\sum_{n=1}^{HW} \|\Delta X_{n,t}\|_2\,, \tag{10}$$

Table 7: **Robustness against temporal disruptions** on Diving48 (5-way 1-shot). Values in parentheses denote rPC; mCE is normalized by Task-Adapter++ (mCE = 100). *Performance reported here on 1000 test episodes for all methods.*

Method	Clean Top-1 ↑	Corrupted queries (mean Top-1 over severities 1/3/5)↑					mCA (%) ↑	mCE ↓
		Sampling	Reversal	Freezing	Jumbling	Box jumb.		
D^{2}ST	72.4	64.6 (0.89)	69.8 (0.96)	66.4 (0.92)	69.3 (0.96)	68.0 (0.94)	67.6	59.3
Task-Adapter++	46.1	44.1 (0.96)	46.0 (1.00)	44.6 (0.97)	46.6 (0.99)†	45.8 (0.97)†	45.4	100.0
GEET	**80.5**	**72.7** (0.90)	**74.5** (0.93)	**71.9** (0.89)	**73.5** (0.91)	**74.3** (0.92)	**73.4**	**48.8**

† Follow-up run using the same protocol; rPC is computed using that run's clean accuracy (47.3).

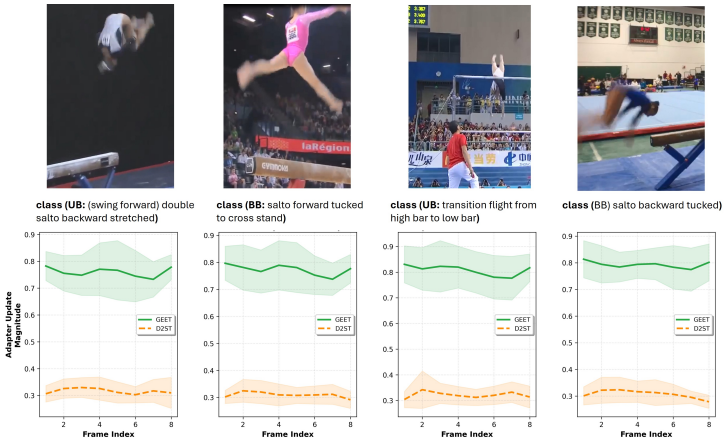

Figure 2: **Temporal saliency** measured by per-frame adapter update magnitude $s(t)$ (Eq. 10) on representative FineGym clips. For each example, we plot mean $\pm$ variability across episodes. GEET consistently produces substantially larger and more structured temporal responses than D^{2}ST.

where n indexes spatial tokens. A discriminative few-shot adapter should emphasize motion-critical frames (e.g., take-off, rotation, landing) rather than update all frames uniformly.

Figure 2 shows $s(t)$ on representative FineGym clips. GEET produces substantially larger updates than D^{2}ST throughout, and, unlike D^{2}ST's flat, low-amplitude response, its magnitude peaks and decays across specific frames, consistent across the action classes tested. This indicates that GEET's temporal feature decomposition and gated temporal fusion concentrate adaptation on frames where short temporal changes occur, rather than spreading it uniformly across the clip, supporting its improved fine-grained discrimination.

6 Conclusion and Limitations

GEET demonstrates that prioritizing temporal specialization over broad spatio-temporal adaptation is a promising direction for few-shot fine-grained action recognition. The results further show that such focused adaptation can jointly deliver strong recognition, parameter and compute efficiency, and robustness, providing a practical approach for adapting foundation models to fine-grained video understanding.

Limitations: Our work focuses on few-shot recognition of short length fine-grained actions, where the action of interest is temporally localized. Extending temporal-first adaptation to long, untrimmed videos, where relevant motion must also be localized within extended temporal context, remains an important direction for future work. Our current robustness evaluation is not covering naturally occurring variations such as viewpoint changes and perspective distortions.

Reproducibility and LLM usage. Source code is provided in the supplementary material; LLMs were used only to improve writing clarity.

REFERENCES

Congqi Cao, Yueran Zhang, Yating Yu, Qinyi Lv, Lingtong Min, and Yanning Zhang. Task-adapter: task-specific adaptation of image models for few-shot action recognition. In *Proceedings of the ACM International Conference on Multimedia, (ACM MM)*, pp. 9038–9047, 2024.

Congqi Cao, Peiheng Han, Yueran Zhang, Yating Yu, Qinyi Lv, Lingtong Min, and Yanning Zhang. Task-adapter++: Task-specific adaptation with order-aware alignment for few-shot action recognition. *Pattern Recognition*, pp. 113237, 2026.

Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 10618–10627, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 21–26, 2017.

Xiaoyu Chen, Yigang Cen, Wanru Xu, Yue Zhang, Yi Jin, Yidong Li, and Linna Zhang. Hierarchical meta-prototypes network for few-shot action recognition. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 2643–2652, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754703. URL https://doi.org/10.1145/3746027.3754703.

Jihoon Chung, Cheng-hsin Wuu, Hsuan-ru Yang, Yu-Wing Tai, and Chi-Keung Tang. Haa500: Human-centric atomic action dataset with curated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, (CVPR)*, pp. 13465–13474, 2021.

Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *Proceedings of the IEEE/CVF international conference on Computer Vision, (CVPR)*, pp. 833–842, 2019.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, (ICCV)*, pp. 6202–6211, 2019.

Yuqian Fu, Li Zhang, Junke Wang, Yanwei Fu, and Yu-Gang Jiang. Depth guided adaptive meta-fusion network for few-shot video recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1142–1151, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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*, pp. 3148–3159, 2022.

James Hong, Matthew Fisher, Michaël Gharbi, and Kayvon Fatahalian. Video pose distillation for few-shot, fine-grained sports action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, (ICCV)*, pp. 9254–9263, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning, (ICML)*, pp. 2790–2799. PMLR, 2019.

Wenbo Huang, Jinghui Zhang, Xuwei Qian, Zhen Wu, Meng Wang, and Lei Zhang. Soap: Enhancing spatio-temporal relation and motion information capturing for few-shot action recognition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 4572–4580, 2024.

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, (ECCV)*, pp. 474–493. Springer, 2024.

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

Dongho Lee, Jongseo Lee, and Jinwoo Choi. Cast: cross-attention in space and time for video action recognition. *Advances in Neural Information Processing Systems*, 36:79399–79425, 2023.

SuBeen Lee, WonJun Moon, Hyun Seok Seong, and Jae-Pil Heo. Temporal alignment-free video matching for few-shot action recognition. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 5412–5421. IEEE, 2025.

Bozheng Li, Mushui Liu, Gaoang Wang, and Yunlong Yu. Frame order matters: A temporal sequence-aware model for few-shot action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI)*, volume 39, pp. 18218–18226, 2025.

Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European conference on Computer Vision (ECCV)*, pp. 513–528, 2018.

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, (ICCV)*, pp. 7083–7093, 2019.

Huabin Liu, Weixian Lv, John See, and Weiyao Lin. Task-adaptive spatial-temporal video sampler for few-shot action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 6230–6240, 2022a.

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 (CVPR)*, pp. 3202–3211, 2022b.

Fei Long, Xiaoou Li, Jiaming Lv, Haoyuan Yang, Xianjun Cheng, and Peihua Li. BDC-CLIP: Brownian distance covariance for adapting CLIP to action recognition. In *Proceedings of the International Conference on Machine Learning, (ICML)*, 2025.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European conference on Computer Vision, (ECCV)*, pp. 1–18. Springer, 2022.

Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022.

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, (CVPR)*, pp. 2203–2213, 2023.

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, (CVPR)*, pp. 11317–11326, 2025.

Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 475–484, 2021.

Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Yingya Zhang, Changxin Gao, Deli Zhao, and Nong Sang. Disentangling spatial and temporal learning for efficient image-to-video transfer learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, (ICCV)*, pp. 13934–13944, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning, (ICML)*, pp. 8748–8763. PmLR, 2021.

Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 6545–6554, 2023.

Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016.

Madeline C. Schiappa, Naman Biyani, Prudvi Kamtam, Shruti Vyas, Hamid Palangi, Vibhav Vineet, and Yogesh S. Rawat. A large-scale robustness analysis of video action recognition models. In *CVPR*, 2023.

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Hao Tang, Jun Liu, Shuanglin Yan, Rui Yan, Zechao Li, and Jinhui Tang. M3net: multi-view encoding, matching, and fusion for few-shot fine-grained action recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1719–1728, 2023.

Anirudh Thatipelli, Sanath Narayan, Salman Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Bernard Ghanem. Spatio-temporal relation modeling for few-shot action recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 19958–19967, 2022.

Bin Wang, Wentong Li, Wenqian Wang, Mingliang Gao, Runmin Cong, and Wei Zhang. Tds-clip: Temporal difference side network for efficient video action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

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 *European conference on Computer Vision, (ECCV)*, pp. 20–36. Springer, 2016.

Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. BEVT: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14733–14743, 2022a.

Xiang Wang, Shiwei Zhang, Zhiwu Qing, Mingqian Tang, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. Hybrid relation guided set matching for few-shot action recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 19948–19957, 2022b.

Xiang Wang, Shiwei Zhang, Zhiwu Qing, Changxin Gao, Yingya Zhang, Deli Zhao, and Nong Sang. Molo: Motion-augmented long-short contrastive learning for few-shot action recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 18011–18021. IEEE, 2023.

Xiang Wang, Shiwei Zhang, Jun Cen, Changxin Gao, Yingya Zhang, Deli Zhao, and Nong Sang. Clip-guided prototype modulating for few-shot action recognition. *International Journal of Computer Vision*, 132(6):1899–1912, 2024.

Xiao Wang, Weirong Ye, Zhongang Qi, Xun Zhao, Guangge Wang, Ying Shan, and Hanzi Wang. Semantic-guided relation propagation network for few-shot action recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 816–825, 2021.

Cong Wu, Xiao-Jun Wu, Linze Li, Tianyang Xu, Zhenhua Feng, and Josef Kittler. Efficient few-shot action recognition via multi-level post-reasoning. In *European Conference on Computer Vision, (ECCV)*, pp. 38–56. Springer, 2024.

Jiazheng Xing, Jian Zhao, Chao Xu, Mengmeng Wang, Guang Dai, Yong Liu, Jingdong Wang, and Xuelong Li. Ma-fsar: Multimodal adaptation of clip for few-shot action recognition. *Pattern Recognition*, 169:111902, 2026.

Jinglin Xu, Yongming Rao, Xumin Yu, Guangyi Chen, Jie Zhou, and Jiwen Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 2949–2958, 2022.

Mengmeng Xu, Han Zhang, and Jian et al. Xu. Actionclip: A new paradigm for video action recognition. In *ICCV*, 2021.

Feng Yang, Jie Zhao, Fulin Luo, Anyong Qin, Tiecheng Song, Yue Zhao, Chenqiang Gao, and Junwei Han. Mpl: Match-guided prototype learning for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 34439–34448, 2026.

Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. AIM: Adapting image models for efficient video action recognition. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CIoSZ_HKHS7.

Tianwei Yu, Peng Chen, Yuanjie Dang, Ruohong Huan, and Ronghua Liang. Multi-speed global contextual subspace matching for few-shot action recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 2344–2352, 2023.

Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pp. 4486–4496, 2021.

Haosong Zhang, Mei Chee Leong, Liyuan Li, and Weisi Lin. Pevl: Pose-enhanced vision-language model for fine-grained human action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18857–18867, 2024.

Yilun Zhang, Yuqian Fu, Xingjun Ma, Lizhe Qi, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. On the importance of spatial relations for few-shot action recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 2243–2251, 2023.

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision, (ECCV)*, pp. 803–818, 2018.

Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *European Conference on Computer Vision, (ECCV)*, pp. 782–797. Springer, 2018.

Xiatian Zhu, Antoine Toisoul, Juan-Manuel Perez-Rua, Li Zhang, Brais Martinez, and Tao Xiang. Few-shot action recognition with prototype-centered attentive learning. *arXiv preprint arXiv:2101.08085*, 2021.

Appendix

A Ablation on the number of frames, blocks and different shot settings

We study how GEET behaves as we vary (i) the amount of adaptation-depth applied within the backbone, (ii) the temporal span of the input clip, and (iii) the number of labeled examples per class. We compare against D^2ST, our strongest adapter-based baseline. Across these axes, GEET shows robust scaling trends, suggesting that its temporal-first, parameter-efficient design extracts more benefit from additional capacity, longer context, and stronger supervision.

Adaptation-depth. Table 8a shows that GEET already outperforms D^2ST when adapters are inserted only in the final transformer blocks. The gap remains as adapters are progressively added to more blocks, indicating that GEET can capture fine-grained temporal structure with localized, lightweight modifications rather than requiring broad changes throughout the network.

Temporal context. Table 8b reports results under different numbers of input frames. While both methods improve as the clip length increases, GEET benefits more consistently from additional frames. This supports our claim that GEET is well-matched to fine-grained actions, where discriminative cues are often phase-dependent and unfold over time.

Amount of supervision. As shown in Table 9, GEET preserves its advantage over D^2ST from 1-shot up to 10-shot settings, indicating stable improvements as more labeled examples become available.

Table 8: Ablation on **adapter depth** and **temporal context** for GEET and D^2ST.

(a) Comparison of GEET and D^2ST when attaching adapters to the last 6, 8, and 10 CLIP blocks. We report 1-shot and 5-shot accuracy on Diving48 and FineGym99.

Method	Blocks	Params	Diving48		FineGym99	
			1-shot	5-shot	1-shot	5-shot
D^2ST	6	3.69M	70.01	79.84	73.84	84.80
GEET	6	1.83M	**80.62**	**88.60**	**78.51**	**88.62**
D^2ST	8	4.86M	71.02	81.43	75.12	86.10
GEET	8	2.40M	**81.05**	**89.02**	**79.10**	**89.30**
D^2ST	10	6.12M	72.70	82.64	76.32	87.13
GEET	10	3.03M	**81.89**	**89.54**	**79.78**	**90.21**

(b) Effect of input frame count on GEET and D^2ST. We report 1-shot and 5-shot accuracy using 8, 12, and 16 frames on Diving48 and FineGym99.

Method	Frames	Diving48		FineGym99	
		1-shot	5-shot	1-shot	5-shot
D^2ST	8	73.29	83.85	76.81	87.55
GEET	8	**83.52**	**90.24**	**83.88**	**91.62**
D^2ST	12	79.88	87.84	80.00	89.98
GEET	12	**87.22**	**92.03**	**83.34**	**91.81**
D^2ST	16	82.75	89.82	80.41	89.38
GEET	16	**87.18**	**91.53**	**83.68**	**92.53**

B More Qualitative analysis

B.1 Nearest-neighbor retrieval on confusing fine-grained categories.

To examine how GEET reshapes the embedding space for fine-grained discrimination, we perform nearest-neighbor (NN) retrieval on a curated 80 videos subset of highly confusable FineGym99 balance-beam actions (e.g., round-off, salta variants, free aerial walkover), where subtle GEET dif-

Table 9: **Amount of supervision** Evaluation of GEET and D^{2}ST across different shot settings.

Dataset	Method	1-shot	3-shot	5-shot	7-shot	9-shot	10-shot
Diving48	D^{2}ST	73.29	80.64	83.85	84.59	84.54	84.96
	GEET	**83.52**	**88.11**	**90.24**	**91.15**	**91.15**	**91.97**
FineGym99	D^{2}ST	76.81	85.52	87.55	89.30	90.03	90.70
	GEET	**83.88**	**88.84**	**91.62**	**92.05**	**92.56**	**92.92**

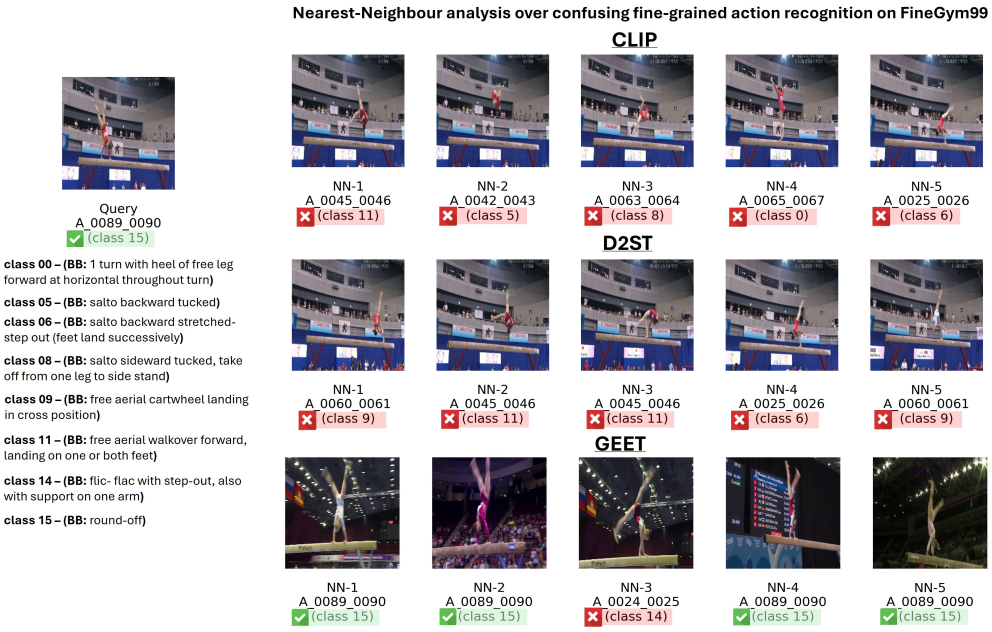

Figure 3: **Nearest-neighbor retrieval** on visually confusable FineGym99 classes. For the same query and gallery, frozen CLIP retrieves appearance-similar but label-mismatched neighbors, D^{2}ST partially improves alignment, while GEET produces substantially cleaner top-K neighborhoods with higher label consistency.

ferences often cause misclassification. For each query clip, we retrieve top-K neighbors using cosine similarity over ℓ_2-normalized embeddings computed with the same 8-frame sampling protocol. We compare frozen CLIP, D^{2}ST, and GEET.

As shown in Fig. 3, frozen CLIP primarily retrieves visually similar clips (shared pose or background) but often from incorrect subclasses. D^{2}ST partially improves neighborhood consistency. In contrast, GEET retrieves predominantly same-class neighbors, indicating stronger alignment with GEET dynamics rather than static appearance. Quantitatively, we measure *Purity@5*, defined as the

Table 10: **Nearest-neighbor retrieval quality** on confusing FineGym99 classes. Purity@5 counts how many of the top-5 retrieved neighbors share the query label. Parentheses denote 95% bootstrap confidence intervals computed over the evaluation queries.

Method	Purity@5 ↑	Purity rate ↑	ΔPurity@5 vs. CLIP ↑
Frozen CLIP	0.85	0.17 (0.08–0.27)	0.00
D^{2}ST	2.80	0.56 (0.39–0.73)	+1.95 (1.20–2.70)
GEET	**3.40**	**0.68** (0.54–0.81)	+**2.55** (1.85–3.25)

number of same-label samples among the top-5 neighbors (Table 10). Frozen CLIP achieves 0.85 (0.17 purity rate), D^{2}ST improves to 2.80 (+1.95 over CLIP), and GEET further increases to 3.40 (+2.55 over CLIP).

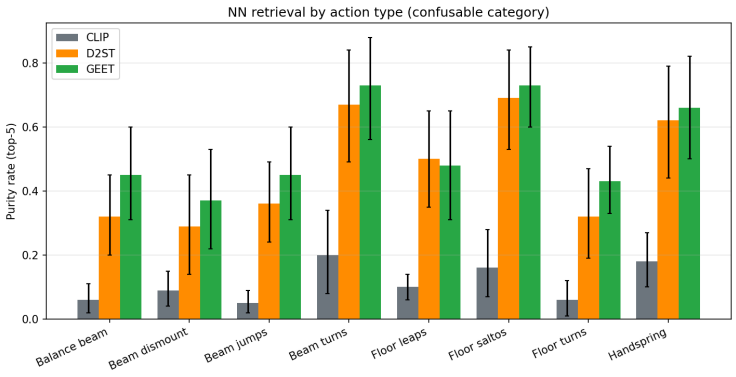

Figure 4: Top-5 purity by confusable action category. GEET matches or exceeds D^{2}ST in most categories, indicating consistent gains across fine-grained action types (95% bootstrap CIs).

A per-category breakdown across floor and beam actions (Fig. 4) shows that GEET matches or surpasses D^{2}ST in most categories, demonstrating that its temporal-phase-aware modeling consistently restructures local neighborhoods across diverse fine-grained action types. Overall, GEET improves retrieval purity by organizing embeddings according to discriminative motion phases rather than static visual similarity.

B.2 Single-frame blackout sensitivity analysis.

To analyze temporal importance allocation, we perform a single-frame blackout sensitivity study on 8-frame clips from FineGym99. For each frame index $t \in \{1, \ldots, 8\}$, we compute the margin variation

$$\Delta(t) = m(v) - m(v^{(t)}),$$

where $m(v) = s_{y^*}(v) - \max_{c \neq y^*} s_c(v)$ is the decision margin for the original clip v, and $m(v^{(t)})$ denotes the margin when frame t is replaced by a normalized black frame. A larger $|\Delta(t)|$ indicates stronger temporal influence, while the sign determines whether a frame contributes positively or negatively to the prediction.

We evaluate GEET and D^{2}ST on 20 randomly sampled 5-way 1-shot episodes and aggregate $\Delta(t)$ across episodes.

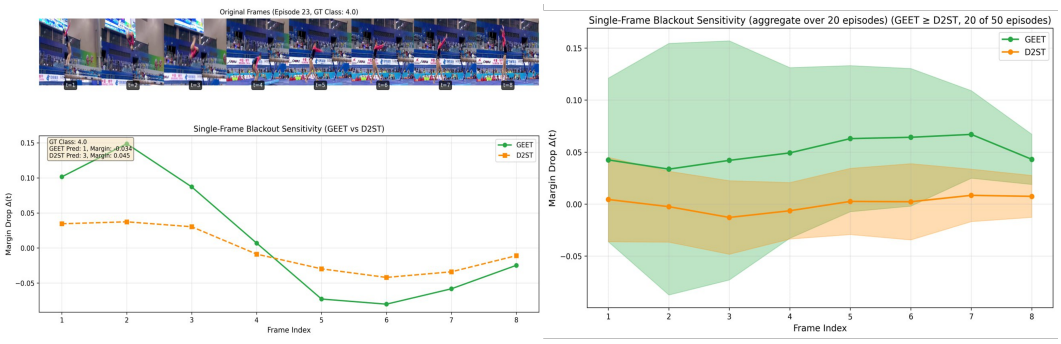

Figure 5: **Single-frame blackout sensitivity** for GEET vs. D^{2}ST. (Left) Example episode: GEET shows a sharper peak ($\Delta(2)$! $\approx$!0.15) than D^{2}ST ($\approx$!0.037). (Right) Mean $\Delta(t)$ over 20 episodes (shaded std): GEET exhibits consistently higher temporal sensitivity, especially in mid-to-late frames (5–7).

Episode-level behavior. Figure 5 (left) shows a representative example. GEET exhibits a pronounced sensitivity peak at frame 2 ($\Delta(2) \approx 0.15$), indicating strong reliance on this temporal position. In contrast, D^{2}ST's maximum response in the same episode remains substantially lower

(approximately 0.037), with a comparatively flat profile across frames. This illustrates GEET's sharper temporal selectivity at the episode level.

Aggregate analysis. The aggregate plot in Figure 5 (right) reports the mean $\Delta(t)$ over 20 episodes. GEET consistently produces larger margin drops across most temporal positions. Its highest mean sensitivity occurs in later frames (around frames 6–7), reaching approximately 0.065–0.07, whereas D^{2}ST remains near 0.01 at its peak. Even in early frames (1–3), GEET maintains mean sensitivity values around 0.035–0.045, compared to D^{2}ST's near-zero response.

The standard deviation band is also wider for GEET, indicating greater variability in which frames are deemed critical across episodes. GEET exhibits content-dependent temporal selectivity (via its temporal operators), producing sharper sensitivity peaks than D^{2}ST. D^{2}ST, by contrast, exhibits both lower peak magnitude and reduced dispersion, reflecting weaker differentiation between critical and non-critical frames.

Notably, GEET achieves higher peak sensitivity than D^{2}ST in 20 out of 50 sampled episodes, further supporting its stronger temporal discrimination capability.

B.3 GEET SHOWS MORE DECISIVE CORRECT PREDICTIONS ACROSS ACTION CATEGORIES.

Beyond accuracy, we evaluate *decision margin*, defined as the score of the correct class minus the highest competing score, where larger values indicate more decisive predictions.

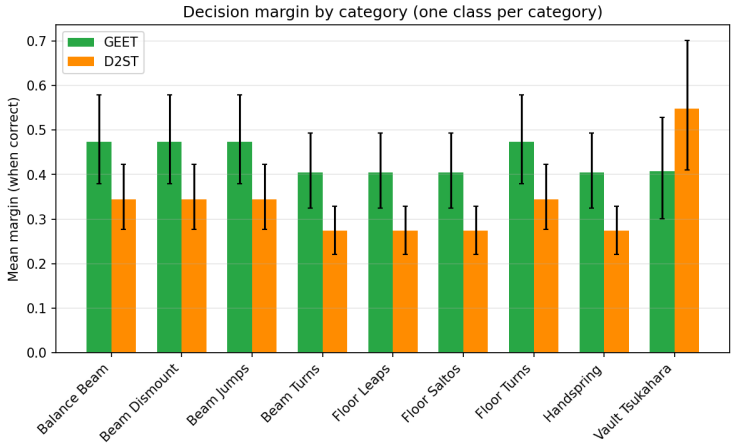

Figure 6: **Decision margin per action category**; error bars show 95% CIs. GEET (green) achieves higher margins than D^{2}ST (orange) in 8/9 categories.

We compute the mean margin (when predictions are correct) on the same few-shot episodes, grouped by FineGym99 action categories with representative classes and 95% bootstrap confidence intervals (Fig. 6). GEET achieves higher mean margins than D^{2}ST in eight of nine categories, indicating more confident and better-separated correct predictions. Overall, GEET improves not only accuracy but also discriminative confidence across diverse action types.

B.4 GEET IS MORE STABLE ACROSS DIFFERENT SUPPORT AND EPISODE DRAWS

Few-shot accuracy can depend on which support videos and episodes are sampled. We measure *stability* by running evaluation multiple times with different random seeds (and thus different episode and support sets) and measuring how much accuracy varies across runs. Lower variance means the method is less sensitive to the particular draw of support and query episodes.

We run 10 independent evaluation runs and define a stability score as $1/(1+\sigma)$, where σ is the standard deviation of accuracy (in %) across runs; higher score means less variance. Figure 7 reports this score for GEET and D^{2}ST. GEET attains a clearly higher stability score (with $\sigma \approx 0.29\%$) than D^{2}ST ($\sigma \approx 0.79\%$), i.e., roughly $2.7\times$ lower variance. Thus, GEET's few-shot performance is more

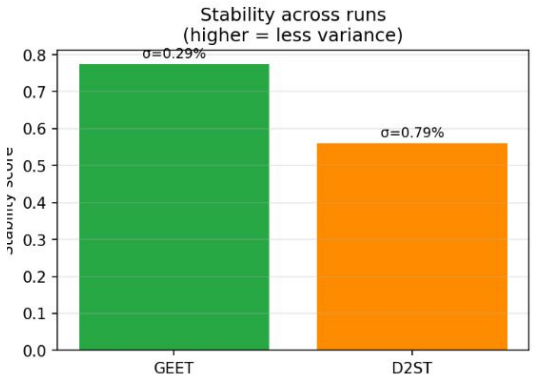

Figure 7: **Stability** across 10 runs with different random seeds (different episode and support draws). Stability score $= 1/(1+\sigma)$; higher means less variance. GEET (green) has substantially higher stability and lower σ than D^{2}ST (orange), indicating more consistent few-shot behavior.

consistent and less dependent on the random choice of support and episodes—a desirable property for deployment.

C Additional fully-supervised results on FineGym99/288

Table 11: **Fully-supervised results on FineGym99 and FineGym288.** PeVL (Zhang et al., 2024) results are reported using the video modality. Tunable parameters changes depending on the number of classes.

Method	Venue	Frames	Tunable Params (M)	Top-1	
				FineGym99	FineGym288
Non-CLIP-based methods					
TQN (Zhang et al., 2021)	CVPR'21	all	78.0	93.8	89.6
TSM (Lin et al., 2019)	ICCV'19	16+16	48.6	88.4	83.1
TRNms (Zhou et al., 2018)	ECCV'18	8+8	36.6	87.8	82.0
SlowOnly (Feichtenhofer et al., 2019)	ICCV'19	32	32.4	93.46	86.8
I3D (Carreira & Zisserman, 2017)	CVPR'17	16	28.0	75.6	77.3
CLIP-based methods ViT-B/16					
PeVL† (Zhang et al., 2024)	CVPR'24	16	28.0	91.2	85.4
AIM † (Yang et al., 2023)	ICLR'23	32	11.0	94.9	89.1
GEET (ours)	–	32	3.87	**95.4** $\pm$ **0.13**	**90.6** $\pm$ **0.10**

D Additional results on coarse-grained action datasets

Table 12 reports results on coarse-grained few-shot benchmarks. GEET is designed for fine-grained recognition, and on these benchmarks it does not reach the state of the art. It substantially improves over the frozen CLIP ViT-B baseline on all four benchmarks (e.g., $+22.0$/$+15.6$ points on HMDB51 and $+33.5$/$+39.4$ points on SSv2-Full in the 1-/5-shot settings), confirming that the adapter transfers beyond fine-grained data. Compared with general-purpose parameter-efficient adapters on the same backbone (ST-Adapter, AIM, DualPath), GEET is within 1.5 points on HMDB51 and UCF101, comparable on SSv2-Small 1-shot, and higher on SSv2-Small 5-shot, where its accuracy of 71.32% is the best among video-only ViT-B methods. However, GEET trails these adapters by 3.5–4.2 points on SSv2-Full, and it remains below D^{2}ST on most coarse-grained settings. The largest gaps are to multimodal methods that additionally use text, such as Task-Adapter++, which exceeds GEET by 13.1 points on HMDB51 1-shot and 12.6 points on SSv2-Full 1-shot.

We attribute these gaps to two properties of GEET's design, although we have not verified this experimentally. First, many HMDB51 and UCF101 classes can be recognized largely from scene

and object appearance, where methods that adapt spatial features more broadly have an advantage over GEET's deliberately limited spatial refinement. Second, although SSv2 is temporally driven, its categories (e.g., *"moving something up"*, *"pushing something so it falls off"*) depend on long-range temporal ordering and object-relational reasoning across the full clip, whereas GEET models short, local temporal transitions through a depthwise temporal convolution with a small receptive field over sparsely sampled frames. We therefore view long-range temporal and relational reasoning as a limitation of the current design and a direction for future work, consistent with the scope discussed in Sec. 6.

Table 12: **GEET on coarse action recognition**. *"*" denotes multi-modal methods incorporating additional text input and text encoder.*

Method	Reference	Pretrain	HMDB51		UCF101		SSv2-Small		SSv2-Full	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CMN (Zhu & Yang, 2018)	ECCV'18	ResNet50	–	–	–	–	34.4	43.8	36.2	48.9
OTAM (Cao et al., 2020)	CVPR'20	ResNet50	–	–	–	–	–	–	42.8	52.3
AmeFuNet (Fu et al., 2020)	MM'20	ResNet50	60.2	75.5	85.1	95.5	–	–	–	–
TRX (Perrett et al., 2021)	CVPR'21	ResNet50	–	75.6	–	96.1	–	59.1	–	64.6
SPRN (Wang et al., 2021)	MM'21	ResNet50	61.6	76.2	86.5	95.8	–	–	–	–
HyRSM (Wang et al., 2022b)	CVPR'22	ResNet50	60.3	76.0	83.9	94.7	40.6	56.1	54.3	69.0
TA^{2}N+Sampler (Liu et al., 2022a)	MM'22	ResNet50	59.9	73.5	83.5	96.0	–	–	47.1	61.6
MoLo (Wang et al., 2023)	CVPR'23	ResNet50	60.8	77.4	86.0	95.5	41.9	56.2	55.0	69.6
MGCSM (Yu et al., 2023)	MM'23	ResNet50	61.3	79.3	86.5	97.1	–	–	–	–
SA-CT (Zhang et al., 2023)	MM'23	ResNet50	60.4	78.3	85.4	96.4	–	–	48.9	69.1
CLIP-FSAR (Wang et al., 2024)	IJCV'24	ResNet50	69.4	80.7	92.4	97.0	52.1	55.8	58.7	62.8
SOAP (Huang et al., 2024)	MM'24	ResNet50	76.5	88.4	94.1	99.3	–	–	61.9	79.8
HMP-Net (Chen et al., 2025)	MM'25	ResNet50	77.6	88.7	94.9	99.1	54.6	60.9	61.2	76.5
MPL (Yang et al., 2026)	CVPR'26	ResNet50	63.3	78.1	90.1	97.5	–	–	43.5	56.6
Trokens (Kumar et al., 2025)	ICCV'25	DinoV2	69.8	82.3	94.0	97.9	53.4	68.9	61.5	76.7
TEAM (Lee et al., 2025)	CVPR'25	ResNet50	62.8	78.4	87.2	96.2	–	–	–	–
CLIP-Freeze (Radford et al., 2021)	ICML'21	ViT-B	51.4	71.0	84.6	94.5	29.5	42.5	26.8	36.3
CLIP-FSAR (Wang et al., 2024)*	IJCV'24	ViT-B	77.1	87.7	97.0	99.1	54.6	–	–	–
MA-FSAR (Xing et al., 2026)*	PR'26	ViT-B	83.4	87.9	97.2	99.2	59.1	64.5	63.3	72.3
EMP-Net (Wu et al., 2024)*	ECCV'24	ViT-B	76.8	85.8	94.3	98.2	57.1	65.7	63.1	73.0
Task-Adapter (Cao et al., 2024)*	MM'24	ViT-B	83.6	88.8	98.0	99.4	60.2	70.2	71.3	74.2
Task-Adapter++ (Cao et al., 2026)*	PR'26	ViT-B	**86.5**	**92.0**	**98.9**	**99.7**	**63.6**	**73.2**	**72.8**	75.5
SOAP (Huang et al., 2024)	MM'24	ViT-B	79.3	89.8	96.8	99.5	–	–	66.7	81.2
HMP-Net (Chen et al., 2025)	MM'25	ViT-B	83.8	89.9	98.7	99.5	60.5	70.4	68.4	79.5
AIM (Yang et al., 2023)	ICLR'23	ViT-B	74.2	86.9	95.4	98.5	52.8	67.5	63.7	79.2
DUALPATH (Park et al., 2023)	CVPR'23	ViT-B	74.9	87.5	95.7	98.7	53.5	68.1	64.5	79.8
ST-Adapter (Pan et al., 2022)	NeurIPS'22	ViT-B	74.1	87.3	95.9	98.9	53.1	68.0	64.2	79.5
D^{2}ST (Pei et al., 2025)	ICCV'25	ViT-B	77.1	88.2	96.4	99.1	55.0	69.3	66.7	**81.9**
TSAM (Li et al., 2025)	AAAI'25	ViT-B	84.5	88.9	98.3	99.3	60.5	66.7	65.8	74.6
MPL (Yang et al., 2026)	CVPR'26	ViT-B	68.0	84.9	93.0	98.5	–	–	40.1	55.0
TEAM (Lee et al., 2025)	CVPR'25	ViT-B	70.9	85.5	94.5	98.8	47.2	63.1	–	–
GEET	–	ViT-B	73.37	86.60	94.56	98.60	53.23	71.32	60.25	75.66

E ADDITIONAL RESULTS FOR ROBUSTNESS AGAINST VISUAL CORRUPTIONS

This section expands the visual-corruption study from main paper. (Table 6) with group-level (Table 13), per-type (Table 14), and per-severity (Tables 15 and 16) breakdowns. We follow the same protocol: Diving48 novel classes, 5-way 1-shot, 8 frames per clip, and 1000 test episodes (seed 18), with corruptions applied only to query clips while support clips remain clean. We use the 15 ImageNet-C corruptions (Hendrycks & Dietterich, 2019) plus Gaussian blur, grouped into noise (Gaussian, shot, impulse), blur (defocus, glass, motion, zoom, Gaussian), weather (snow, frost, fog, brightness), and digital (contrast, elastic transform, pixelate, JPEG), each evaluated at severities $\{1, 3, 5\}$. Blur and weather corruptions are shared across the frames of a clip to avoid artificial flicker, whereas noise and JPEG are sampled independently per frame. Clean accuracies reported here are measured on these 1000 episodes and therefore differ from the 10,000-episode results in Table 1 and 2.

In addition to mCA, rPC, and mCE (Eq. equation 9), we report chance-corrected retention, $\mathrm{Ret}_{m,c} = (\bar{A}_{m,c} - 20)/(A_m^{\mathrm{clean}} - 20)$, since 5-way chance accuracy is 20%. Unlike rPC, retention measures only the accuracy above chance that survives the corruption, and therefore does not favour methods whose clean accuracy is already close to chance. Group-level mCE applies Eq. equation 9 restricted to the corruption types within each group, with Task-Adapter++ as the reference ($= 100$).

As shown in Table 13, GEET obtains the highest mCA, rPC, and retention, and the lowest mCE, in every corruption group. The retention gap over D^{2}ST is largest for noise (0.71 vs. 0.57) and blur (0.64 vs. 0.53), and blur is the most challenging group for both GEET and D^{2}ST. At the level of individual corruptions (Table 14), GEET has a lower corruption error than D^{2}ST on all 16 types. The most damaging corruptions for GEET are glass blur (48.9%), motion blur (56.3%), and elastic transform (59.1%), which distort local spatial structure across the frame, while photometric corruptions such as brightness (78.7%), JPEG compression (77.1%), and fog (76.6%) remain close to its clean accuracy (80.6%).

The per-severity results in Tables 15 and 16 show that GEET achieves the highest accuracy in all 48 corruption–severity pairs. From severity 1 to 5, GEET and D^{2}ST degrade by similar absolute amounts (noise: 33.5 vs. 33.8 points; blur: 32.6 vs. 33.9 points), so GEET's advantage is largely preserved rather than amplified at high severity. At severity 5, noise and blur reduce all methods substantially, with Task-Adapter++ approaching chance (25.7% and 28.1%), whereas weather

Table 13: **Group-level spatial scores**. mCA is mean top-1 in the group. Ret. is chance-corrected. Group mCE uses equation 9 restricted to the types in that group (Task-Adapter++ $= 100$ in every group).

Method	Group	mCA	rPC	Ret.	mCE ↓
GEET	All 16	66.1	0.82	0.76	50.9
	Noise	62.8	0.78	0.71	54.1
	Blur	58.8	0.73	0.64	60.9
	Weather	73.0	0.91	0.88	41.1
	Digital	70.8	0.88	0.84	45.6
D^{2}ST	All 16	54.4	0.75	0.66	68.4
	Noise	49.9	0.69	0.57	72.8
	Blur	48.0	0.66	0.53	77.0
	Weather	61.1	0.84	0.79	59.3
	Digital	59.2	0.82	0.75	63.7
Task-Adapter++	All 16	33.9	0.74	0.53	100.0
	Noise	31.2	0.68	0.43	100.0
	Blur	32.6	0.71	0.48	100.0
	Weather	35.1	0.76	0.58	100.0
	Digital	36.4	0.79	0.63	100.0

F Additional results for robustness against temporal disruptions

This section provides per-severity results for the temporal-disruption study of Sec. 5.5. We use the same episodes as Appendix E (Diving48 novel classes, 5-way 1-shot, 8 frames, 1000 episodes, seed 18), with clean support clips and disrupted query clips. Following (Schiappa et al., 2023), we apply five temporal disruptions with identical operators for all methods: sampling rate (faster and slower playback, reported as their mean), reversal of local segments, frame freezing, jumbling of frames within local segments, and box jumbling, which shuffles segments while preserving local motion. Segment sizes are scaled to the 8-frame setting. We report mean top-1 over severities $\{1, 3, 5\}$, $\mathrm{rPC} = \mathrm{acc}_{\mathrm{corr}}/\mathrm{acc}_{\mathrm{clean}}$, and mCE averaged over the five disruptions with Task-Adapter++ as reference. For Task-Adapter++, jumbling and box jumbling are taken from a follow-up run under the same protocol, whose clean accuracy is 47.3% (vs. 46.1%); their rPC values in Table 7 use this run's clean accuracy.

Table 17 shows that GEET achieves the highest top-1 accuracy for every disruption at every severity. However, GEET degrades more than D^{2}ST from severity 1 to 5 (e.g., freezing: 15.3 vs. 9.6 points; reversal: 6.8 vs. 3.3 points), so the gap narrows at the strongest severity, most notably for freezing (63.7% vs. 61.7%). This is consistent with GEET relying more strongly on temporal structure: freezing repeats frames and thus suppresses the frame-to-frame differences used by the explicit-motion branch, while reversal and jumbling alter the temporal context captured by the appearance-

Table 14: **Per-type mean top-1** (%) over severities $1/3/5$, and the CE term $(100 - \bar{A}_{m,c})/(100 - \bar{A}_{\mathrm{ref},c})$ from equation 9. mCE in Table 6 is 100 times the mean of these 16 CE values.

	Top-1			CE		
Corruption	GEET	D^{2}ST	TA++	GEET	D^{2}ST	TA++
Gaussian noise	61.8	51.4	31.6	0.56	0.71	1.00
Shot noise	62.6	50.4	31.0	0.54	0.72	1.00
Impulse noise	63.9	47.9	30.9	0.52	0.75	1.00
Defocus blur	59.0	49.4	33.1	0.61	0.76	1.00
Glass blur	48.9	37.4	28.2	0.71	0.87	1.00
Motion blur	56.3	46.0	33.4	0.66	0.81	1.00
Zoom blur	70.1	55.4	34.5	0.46	0.68	1.00
Gaussian blur	59.7	51.8	34.0	0.61	0.73	1.00
Snow	62.6	48.1	27.0	0.51	0.71	1.00
Frost	74.0	60.1	35.7	0.40	0.62	1.00
Fog	76.6	67.1	36.2	0.37	0.52	1.00
Brightness	78.7	69.2	41.3	0.36	0.52	1.00
Contrast	75.5	68.7	36.0	0.38	0.49	1.00
Elastic transform	59.1	43.0	31.3	0.60	0.83	1.00
Pixelate	71.3	59.4	37.2	0.46	0.65	1.00
JPEG compression	77.1	65.8	41.0	0.39	0.58	1.00
Mean (mCA / mCE)	66.1	54.4	33.9	50.9*	68.4*	100.0*

* Last row, CE columns: mCE $= 100\times$ mean of the 16 CE terms.

evolution branch. Task-Adapter++ changes by at most 4.2 points across severities, so its rPC close to 1 in Table 17 reflects low sensitivity to temporal order at a low clean accuracy rather than stronger temporal modeling.

G Implementation Details

We set temporal kernel k=3, dilation d=2, and adapter bottleneck ratio 0.25. Training runs for 10 epochs with Adam, base LR 2×10^{-3} (2 warmup epochs from 2×10^{-4}), weight decay 5×10^{-4}, and a step-wise LR decay. All experiments use a CLIP ViT-B/16 backbone for consistency.

FineGym99 and FineGym288. We train GEET for 60k episodes using Adam with a base learning rate of 2×10^{-3}, weight decay 5×10^{-4}, and 2 warmup epochs linearly increasing the learning rate from 2×10^{-4} to the base value. We use a step-based learning-rate schedule with relative factors $\{1, 0.1, 0.05, 0.01\}$ applied at epochs 0, 4, 6, and 8. For GEET, the temporal depthwise convolution kernel size is set to k=3 on FineGym99 and k=5 on FineGym288. Video clips are sampled as 8-frame segments at 12 FPS with a 224×224 crop for both training and testing.

Additional datasets. For HAA500, MPII Cooking 2, and Toyota SmartHome, we use the same training setup as above and set the temporal kernel size to k=5 for GEET. Unless stated otherwise, all experiments use a frozen CLIP ViT-B/16 backbone and optimize only the GEET adapter parameters.

H GEET Algorithm

Table 15: **Per-corruption Per-severity top-1 accuracy** (%). Sev. 1 / 3 / 5 = mild / medium / strong. *GEET outperforms consistently across corruptions.*

	GEET			D^2ST			Task-Adapter++		
Corruption	1	3	5	1	3	5	1	3	5
Gaussian noise	77.0	68.3	40.0	66.6	57.7	29.8	39.0	30.8	24.9
Shot noise	77.8	68.2	41.9	67.1	54.8	29.3	38.7	29.7	24.7
Impulse noise	75.4	68.8	47.7	60.4	49.8	33.4	34.5	30.6	27.6
Defocus blur	75.2	62.4	39.5	66.9	51.6	29.7	39.5	30.9	28.8
Glass blur	67.8	45.3	33.7	55.8	31.0	25.4	32.1	25.6	26.8
Motion blur	72.4	56.3	40.1	63.7	46.6	27.8	39.8	32.7	27.9
Zoom blur	77.4	72.0	60.9	66.2	57.0	43.1	39.5	34.1	29.7
Gaussian blur	79.1	65.4	34.6	71.7	55.4	28.3	42.8	31.6	27.5
Snow	67.9	62.0	57.9	52.6	47.4	44.3	29.2	26.5	25.2
Frost	78.6	74.7	68.6	66.0	61.3	53.0	40.6	35.6	30.8
Fog	79.6	77.0	73.4	70.3	67.7	63.4	40.3	36.4	32.0
Brightness	80.6	80.2	75.4	72.4	71.4	63.8	46.3	42.5	35.2
Contrast	80.3	79.0	67.3	72.1	69.7	64.2	42.9	35.3	29.7
Elastic transform	78.2	59.4	39.8	63.6	39.4	26.1	40.7	28.4	24.9
Pixelate	77.9	73.6	62.4	67.7	63.4	47.1	42.1	38.0	31.7
JPEG compression	80.3	79.5	71.4	71.2	68.6	57.4	45.3	43.8	33.9

Table 16: **Group mean top-1** (%) at each severity.

Method	Group	Sev. 1	Sev. 3	Sev. 5
GEET	Noise	76.7	68.4	43.2
	Blur	74.4	60.3	41.8
	Weather	76.7	73.5	68.8
	Digital	79.2	72.9	60.2
D^2ST	Noise	64.7	54.1	30.9
	Blur	64.8	48.3	30.9
	Weather	65.3	62.0	56.2
	Digital	68.6	60.3	48.7
Task-Adapter++	Noise	37.4	30.4	25.7
	Blur	38.7	31.0	28.1
	Weather	39.1	35.3	30.8
	Digital	42.7	36.3	30.1

Table 17: **Per-severity top-1** (%) for the five Schiappa et al. (2023) temporal noises. Sampling rate is the mean of faster and slower playback at that severity.

Method	Noise	Sev. 1	Sev. 3	Sev. 5
Motion-phase	Sampling rate	77.2	74.3	66.6
	Reversal	78.6	73.1	71.8
	Freezing	79.0	73.1	63.7
	Jumbling	78.1	72.6	69.8
	Box jumbling	78.6	75.2	69.0
D2ST	Sampling rate	68.1	65.5	60.1
	Reversal	71.8	69.2	68.5
	Freezing	71.3	66.2	61.7
	Jumbling	71.4	69.5	66.8
	Box jumbling	69.7	68.8	65.6
Task-Adapter++	Sampling rate	45.4	44.4	42.5
	Reversal	46.7	45.9	45.4
	Freezing	46.3	45.5	42.1
	Jumbling	47.0	47.1	45.7
	Box jumbling	46.0	45.8	45.5

Algorithm 1 GEET: Gated Explicit-motion Extraction Tempo-spatial Adapter (training + inference). The adapter $\mathcal{G}(\cdot)$ is defined in sec. 3 in main paper (Eqs. 1–8).

Require: Few-shot episodes $\mathcal{E} = \{(\mathcal{S}, \mathcal{Q})\}$, frozen CLIP ViT backbone f_θ, GEET parameters ψ (FC-down/up projections $\phi_\downarrow, \phi_\uparrow$, depthwise temporal convolutions DWConv_t, depthwise spatial convolution DWConv_s, scalar gate g), optimizer $\mathcal{O}$, number of frames T

Ensure: Trained GEET parameters ψ

for all episode $(\mathcal{S}, \mathcal{Q}) \in \mathcal{E}$ **do**
  ▷ Build class prototypes from the support set
  $\mathcal{P} \leftarrow \emptyset$
  for all class k in $\mathcal{S}$ **do**
    $\mathbf{p}_k \leftarrow \mathbf{0}; \quad n_k \leftarrow 0$
    for all video x in support class k **do**
      $\mathbf{u} \leftarrow \text{EncodeWithGEET}(x; f_\theta, \psi, T)$ ▷ Video embedding
      $\mathbf{p}_k \leftarrow \mathbf{p}_k + \mathbf{u}; \quad n_k \leftarrow n_k + 1$
    end for
    $\mathbf{p}_k \leftarrow \mathbf{p}_k / n_k$ ▷ Prototype of class k
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathbf{p}_k\}$
  end for
  ▷ Query classification and episodic loss
  $\mathcal{L} \leftarrow 0$
  for all (video x, label y) in $\mathcal{Q}$ **do**
    $\mathbf{u} \leftarrow \text{EncodeWithGEET}(x; f_\theta, \psi, T)$
    for all prototype $\mathbf{p}_k \in \mathcal{P}$ **do**
      $\ell_k \leftarrow \cos(\mathbf{u}, \mathbf{p}_k)$
    end for
    $\mathcal{L} \leftarrow \mathcal{L} + \text{CE}(\text{Softmax}(\{\ell_k\}), y)$
  end for
  $\mathcal{L} \leftarrow \mathcal{L}/|\mathcal{Q}|$
  $\psi \leftarrow \mathcal{O}.\text{Step}(\nabla_\psi \mathcal{L})$ ▷ Update only GEET; f_θ is frozen
end for

Algorithm 2 ENCODEWITHGEET: CLIP ViT forward with GEET inserted per block

Require: Video frames x, frozen CLIP ViT f_θ with L transformer blocks, GEET parameters $\psi = \{\psi_\ell\}_{\ell=1}^{L}$, number of frames T
Ensure: Video embedding $\mathbf{u}$

$\mathbf{X} \leftarrow \text{PATCHEMBED}(x)$ ▷ $\mathbf{X} \in \mathbb{R}^{(HW+1)\times(BT)\times C}$ includes CLS
for $\ell = 1$ **to** L **do**
  $\mathbf{X} \leftarrow \text{VITBLOCK}_\ell(\mathbf{X})$ ▷ Frozen backbone block
  $\mathbf{X} \leftarrow \mathbf{X} + \text{GEET}(\mathbf{X}; \psi_\ell, T)$ ▷ Adapter residual update
end for
$\mathbf{u} \leftarrow \text{TEMPORALPOOL}(\text{CLS}(\mathbf{X}))$ ▷ e.g., mean over T
return $\mathbf{u}$

Algorithm 3 GEET: Gated explicit-motion temporal-first adapter (per transformer block); see Sec. 3, Eqs. 1–8.

Require: Tokens $\mathbf{X} \in \mathbb{R}^{(HW+1)\times(BT)\times C}$, GEET parameters $\psi_\ell = \{\phi_\downarrow, \phi_\uparrow, \text{DWConv}_t, \text{DWConv}_s, g\}$, number of frames T
Ensure: Adapter update $\Delta\mathbf{X} \in \mathbb{R}^{(HW+1)\times(BT)\times C}$

▷ Separate CLS and patch tokens; reshape patch tokens to video tensor
$\mathbf{X}_{cls} \leftarrow \mathbf{X}[0]; \quad \mathbf{X}_p \leftarrow \mathbf{X}[1:]$
$\mathbf{V} \leftarrow \text{RESHAPE}(\mathbf{X}_p)$ ▷ $\mathbf{V} \in \mathbb{R}^{B\times C\times T\times H\times W}$
▷ Bottleneck residual adaptation
$\mathbf{U} \leftarrow \phi_\downarrow(\mathbf{V})$ ▷ $\mathbf{U} \in \mathbb{R}^{B\times C_\alpha\times T\times H\times W}$
▷ Appearance evolution
$\mathbf{A} \leftarrow \text{DWConv}_t(\mathbf{U})$
▷ Motion transition
$\Delta\mathbf{U}_1 \leftarrow \mathbf{0}$
for $t = 2$ **to** T **do**
  $\Delta\mathbf{U}_t \leftarrow \mathbf{U}_t - \mathbf{U}_{t-1}$
end for
$\mathbf{M} \leftarrow \text{DWConv}_t(\Delta\mathbf{U})$
▷ Gated temporal fusion (g initialized to 0)
$\mathbf{Z} \leftarrow \mathbf{A} + g\,\mathbf{M}$
▷ Local spatial refinement
$\mathbf{Z}' \leftarrow \mathbf{Z} + \text{DWConv}_s(\mathbf{Z})$
▷ Project back and format as token update (no CLS update)
$\Delta\mathbf{V} \leftarrow \phi_\uparrow(\mathbf{Z}')$ ▷ $\Delta\mathbf{V} \in \mathbb{R}^{B\times C\times T\times H\times W}$
$\Delta\mathbf{X}_p \leftarrow \text{UNRESHAPE}(\Delta\mathbf{V})$ ▷ $\Delta\mathbf{X}_p \in \mathbb{R}^{HW\times(BT)\times C}$
$\Delta\mathbf{X} \leftarrow \text{CONCAT}(\mathbf{0}_{cls}, \Delta\mathbf{X}_p)$
return $\Delta\mathbf{X}$