LogFA: Feature-Space Data Augmentation in Long Egocentric Videos Using Foundation Models

Zijia Lu and Ehsan Elhamifar

Northeastern University
lu.zij@northeastern.edu, e.elhamifar@northeastern.edu

Abstract. Egocentric AI assistants seek to infer user actions and intentions from long first-person videos to facilitate real-world tasks ranging from daily activities to industrial operations. However, collecting large-scale annotated egocentric videos that capture variations in objects, tools, environments and procedures is costly and impractical, motivating the need for data-efficient learning. Existing data augmentation approaches either offer limited visual diversity through simple transformations or rely on computationally expensive generative models that may introduce artifacts. We propose **LogFA** (**Lo**cal-**g**lobal **F**eature-space **A**ugmentation), a novel framework that performs **data augmentation directly in the feature space** for Temporal Action Segmentation (TAS). LogFA efficiently generates semantically consistent variations without pixel-level synthesis by modifying pre-extracted video features at both local and global levels. Locally, LogFA leverages vision-language models with a **Prompt-based Feature Enhancement** strategy to create diverse action-level variations through text-guided feature modifications. Globally, LogFA constructs a **Generalized Directed Acyclic Graph** to model procedural dependencies and sample alternative action sequences. Experiments on egocentric video benchmarks show LogFA significantly improves model generalization to unseen environments while maintaining low computational and data collection costs. Code will be released at https://github.com/ZijiaLewisLu/ECCV2026-LogFA.

Keywords: Temporal Action Segmentation · Data Augmentation · Long Video Understanding

1 Introduction

The visual perception capabilities of artificial intelligence models have advanced rapidly in recent years. Consequently, there is growing interest in *egocentric AI assistants* that perceive the world from the user's perspective, infer their actions and intentions, and provide assistance. Such systems have broad applications, ranging from supporting users in daily activities to helping factory workers adapt to new production procedures. They must also enable efficient inference on edge devices such as headsets and mobile phones. To support such assistants, **Temporal Action Segmentation** (TAS) aims to partition long videos into

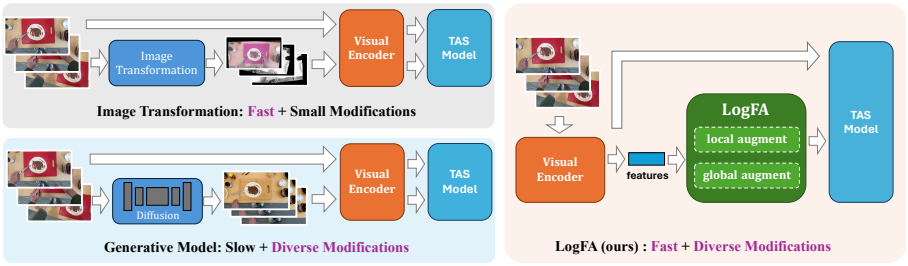

Fig. 1: Different Data Augmentation Methods. Our LogFA generates diverse data augmentations directly in feature space with low computational cost, whereas conventional image transformations provide limited diversity and generative methods incur high computational cost.

non-overlapping segments corresponding to distinct actions, enabling automatic video analysis and a deeper understanding of human behavior and intent.

Numerous studies have introduced datasets of egocentric videos for various procedural tasks [3, 6, 20]. However, because the same task can be performed using diverse objects, tools, procedures, and environments, most datasets include limited variations of each task, and TAS models often experience performance degradation when deployed in settings that are unseen during training [7, 31]. This highlights the importance of **data diversity**. As we show in our experiments, increasing data diversity is more effective than increasing data quantity in improving model performance. However, collecting diverse videos greatly increases data collection cost. This motivates the development of data augmentation methods that can synthesize diverse training examples from limited data, enabling efficient model training in new environments with minimal data collection effort.

Conventional data augmentation methods [56, 58, 61] rely on image transformations (e.g., color jittering, affine transformations, resizing) that exploit structural priors of images, such as semantic invariance to shifts or rotations, to generate modified training samples at low cost, as shown in Figure 1. However, such augmentations offer only *limited diversity* and fail to introduce meaningful semantic spatial and/or temporal variations such as modifying object appearances or scene attributes, while stronger transformations risk destroying essential semantic information. Another line of work [18, 52, 53, 57] leverages advances in generative models to produce richer and more meaningful data variations using pretrained image or video editing models. Although these approaches yield greater diversity without disrupting key semantics, they incur *heavy computational costs* due to reliance on large diffusion-based generative models and may introduce unintended artifacts depending on model performance.

Paper Contributions. To overcome these limitations, we propose a novel ***Lo**cal-**g**lobal **F**eature-space **A**ugmentation* (**LogFA**) framework for long egocentric video understanding. Our method generates diverse augmentations *directly in the feature space*, eliminating the need for pixel-level generative models while preserving the ability to induce large, semantically consistent variations. Cru-

cially, augmenting in feature space offers substantial computational efficiency. TAS models are typically trained on features produced by pretrained visual encoders rather than taking raw pixels as inputs. In contrast to generative approaches—which must first synthesize data pixels and then encode pixels with visual encoders—LogFA bypasses both steps, producing augmented features directly with minimal computational overhead.

To achieve effective and diverse augmentations, LogFA performs modifications at both the *local clip level* and the *global video level*. For ***Local Augmentation***, LogFA generates varied executions of individual actions—e.g., by altering the objects, tools, or their visual attributes. We demonstrate that these feature modifications can be realized by leveraging contrastive *vision-language models* (VLMs) [38, 54]. VLMs provide jointly aligned visual–textual semantic spaces, where the features of a video and its corresponding caption are close. LogFA leverages this property by modifying video captions to reflect desired augmentations, and then uses *enhanced* textual features as efficient approximations of the visual features of the modified videos. Specifically, because the visual and textual features of VLMs are not perfectly aligned, we introduce a ***Prompt-based Feature Enhancement*** mechanism to reduce the modality gap.

For ***Global Augmentation***, LogFA generates diverse procedures to execute a task by altering the sequence of actions. We propose a ***Generalized Directed Acyclic Graph*** that models action dependencies, co-occurrences, and repetitions of a task. During training, we sample varied action sequences from this graph and generate video features. In both local and global augmentation, LogFA leverages the common-sense knowledge embedded in large language models (LLMs) to automatically design augmentation strategies.

We provide extensive experiments showing that LogFA outperforms both image transformation and generative data augmentation approaches on egocentric video benchmarks, while maintaining a low computational cost. TAS models trained with LogFA achieve strong performance especially on test videos with large changes. We also provide ablation studies to validate our key design choices.

2 Related Works

Temporal Action Segmentation. Temporal Action Segmentation (TAS) has been studied in unsupervised [2,9,10,13,17,46,66], weakly-supervised [5,8,12,21–23,26,28,29,39–41,44,51] and fully supervised [1,4,11,14,16,19,20,24,25,27,35,42,45,48–51,59,60,62] settings. MSTCN [11] and subsequent works [25,50] use a multi-stage refinement mechanism for accurate predictions. DiffAct [27] extends this mechanism to a diffusion process. FACT [30] establishes new SOTA via parallel action and frame-level temporal modeling. Beyond TAS, related long-video understanding and tracking tasks include temporal grounding and multi-object tracking in videos [32]. Meanwhile, many work [3,6,20] have collected egocentric procedural video datasets. EgoPER [20] features cooking recipes collected in a controlled lab environment. EgoProceL [3] combines 5 datasets, thus has videos of diverse tasks. While videos in each dataset contain variance among them,

they are not sufficient to cover the diverse possible ways to complete a task. While most existing TAS methods [27,30] focus on achieving good performance on collected videos, it is overlooked that collecting new videos is often required to adapt models to new environments, incurring high computational costs. Motivated by this, we propose LogFA as a new approach to reduce data collection cost and improve the applicability of TAS methods to real-world scenarios.

Data Augmentation. Classical data augmentation in vision applies label-preserving transformations to inputs, such as geometric and photometric edits (random crop/resize, flip, rotation, color jitter, image composition, etc.) [56,58]. While effective, such transformations primarily capture low-level invariances and often struggle to induce semantic diversity (e.g., tool substitutions, object attribute changes, or contextual shifts) without risking label drift. An alternative line of work leverages generative models to synthesize richer variations. Image- and video-diffusion models, as well as text-conditioned editors, enable attribute- and object-centric edits (appearance changes, background replacement, style transfer) that better reflect distributional shifts [43, 52]. Despite improved semantic coverage, diffusion-based augmentation is computationally intensive, can introduce temporal incoherence, and may yield artifacts that confound downstream learning, particularly when training-time generation must be repeated at scale. Meanwhile, one thread of work [**?**,18,53] aims to learn domain-specific generations. However, these methods often struggle on model performance as they still face the challenge of collecting high-quality video data in the desired environment. In contrast, our LogFA method proposes to directly generate new data in feature space using existing contrastive vision-language models. It is capable of generating large semantic changes compared to conventional data augmentation methods and also enjoys high efficiency compared to generative methods. Domain adaptation methods [34,47,63] migrate models from a source domain to a target domain. However, they assumes access to source domain data and cannot handle unique actions/tasks in the target domain. In contrast, our method does not require data from a source domain and can directly generate data in the target domain.

3 Methodology

3.1 LogFA Framework

Building a personalized and computationally efficient AI assistant often requires a specialized TAS model tailored to a user's unique setting, such as a specific home, factory environment, or task procedure. To this end, we propose the ***Lo**cal-**g**lobal **F**eature-space **A**ugmentation* (**LogFA**) framework, which enables training reliable TAS models from only a few videos. Furthermore, we leverage the common-world knowledge encoded in LLMs to automate the design of our augmentation strategy.

LogFA performs augmentation at complementary *local (intra-action)* and *global (inter-action)* levels. For **local augmentation**, we modify video clips to

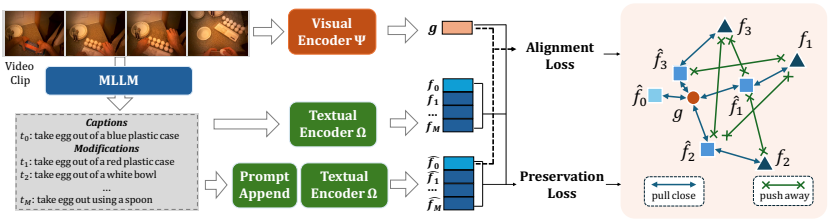

Fig. 2: Local Augmentation. We leverage a multimodal LLM to generate clip captions t_0 and diverse modifications $(t_1, \ldots, t_M)$ that describe different variations of the same action. We then extract the visual feature g and the textual features $\{f_m\}$ using a contrastive VLM. Finally, we enhance the textual features $\{\widehat{f}_m\}$ with our Prompt-based Feature Enhancement module and use them as approximations of the visual features of the modified video clips.

capture different variations of executing an *action*, ranging from subtle appearance or texture changes to larger alterations such as tool or object substitutions and background modifications. For **global augmentation**, we introduce diverse procedures for executing a *task*. Specifically, we construct a task graph that encodes dependencies, repetitions, and co-occurrences among actions, and leverage this graph to sample varied action sequences before rearranging the corresponding action segments in the video.

Overall, our training pipeline consists of two stages: (1) a *feature augmentation stage*, which uses LogFA to generate diverse augmented features from a few annotated videos, and (2) a *TAS training stage*, which trains a TAS model using both the original and augmented features. We describe both augmentation components in detail below.

3.2 Local Augmentation

Overview. Local augmentation requires making valid modifications to clips *without* changing the action performed in the clips. As shown in Figure 2, given a video clip v with an action label *"taking out egg"*, we leverage a multimodal LLM to obtain an action-centric caption t_0 of v, e.g., *"taking an egg out of a blue plastic case"*, which provide more detailed information about the action. Next, we leverage a text LLM to craft M plausible variants of the action, $(t_1, \ldots, t_M)$, such as changing the color of the plastic case or substituting the containers of eggs.[1] Given that the clip is encoded into a visual feature using a video encoder, an ideal augmentation approach is to edit its features. However, how features should change by changing e.g., the color/texture of some objects or tools is unknown. As a result, prior work often resort to changing pixels with a generative model and then re-encoding the frames to obtain augmented features. This incurs heavy computation cost. In contrast, we avoid pixel-level generation

[1] We use a text LLM rather than a multimodal LLM for modification because we find the former better follows prompts and generates diverse modifications. We defer details of caption generation to the supplementary material.

and instead propose to modify feature representations directly via contrastive VLMs [38, 54, 65].

Contrastive VLMs jointly learn a visual encoder Ψ and a textual encoder Ω with *aligned output spaces*. Let the visual feature of the video clip v and the textual feature of its corresponding caption t_0 be denoted by

$$g = \Psi(v), \quad f_0 = \Omega(E_0), \tag{1}$$

where E_0 represents the text embedding tokens of t_0. These two features are expected to lie close to each other. Similarly, for each modified caption t_m (e.g., *"taking an egg out of a bowl"*), its textual feature, $f_m = \Omega(E_m)$, should lie close to the visual feature of a video clip that corresponds to the modified caption (note that we do not have this video clip). As a zero-shot retrieval check on EgoPER [20], we use the visual features of video clips to retrieve their corresponding captions from all clip captions in the video. This achieves 68% accuracy, compared with 0.13% for random guessing, indicating reliable alignment between visual and textual features. This gives us a starting point to use textual features of modified captions $t_1, \ldots, t_m$ as *approximations* of the visual features of clips demonstrating the captions, skipping pixel synthesis entirely. Compared to image transformations, this enables large and semantic-meaningful changes while preserving action labels.

Prompt-based Feature Enhancement (PFE). While the visual feature of a clip is more closely aligned with the textual feature of its corresponding caption than textual features of irrelevant captions, we observe that the visual and textual features still remain separated by a non-negligible distance. Empirically, the average cosine similarity between corresponding g and f_0 is 0.18, and training directly on textual features degrades performance, as we show in Section 4.2. One important reason is that the text encoder has a short input length and the short captions cannot capture all details of a clip (e.g., background, action-irrelevant objects), while the visual encoder processes the full clip.

To bridge this gap, we introduce a *Prompt-based Feature Enhancement* method. For each clip v, we aim to find a function τ that transforms text captions to improve the alignment between visual and textual features. Inspired by recent works [15, 55, 64] that use prompt learning to inject new information into prompts, we design a learning-based method. We append a sequence of learnable embedding tokens $\hat{E}_v$ to the original embedding tokens of a caption,

$$\tau(E_m) = \text{concat}(E_m, \widehat{E}_v), \quad \widehat{f}_m = \Omega\big(\tau(E_m)\big), \tag{2}$$

where $\widehat{f}_m$ denotes the new (enhanced) textual feature obtained after transformation τ. Notice that $\widehat{E}_v$ is specific to the clip v. We freeze Ω and only optimize $\widehat{E}_v$ to inject the missing context into the textual features. We provide more empirical analysis in Section 4.2.

Optimization Objectives. We optimize $\widehat{E}_v$ with the following two losses. First, we design an *Alignment Loss* to enhance the alignment between the visual feature

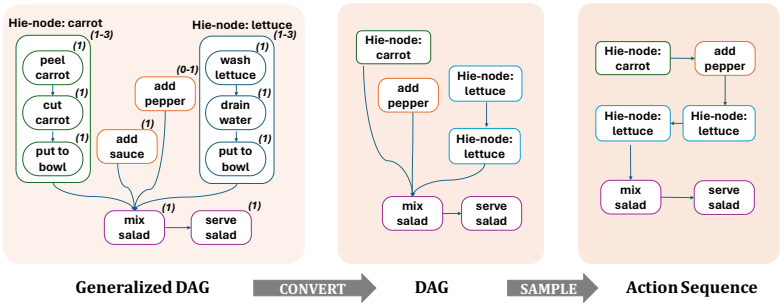

Fig. 3: Global Augmentation. We construct a Generalized Directed Acyclic Graph (GDAG) to model action dependencies, co-occurrences, and repetitions. To sample valid action sequences from GDAG, we propose a simple method of converting the graph into DAG and sample sequences from it via topological sort.

g of the clip and its unmodified caption feature $\widehat{f}_0$,

$$\mathcal{L}_{\text{align}} = 1 - \cos(\widehat{f}_0, g) = 1 - \frac{\widehat{f}_0 \cdot g}{\|\widehat{f}_0\| \, \|g\|}, \tag{3}$$

where $\cos(\cdot, \cdot)$ is the cosine similarity between two features.

Optimizing only $\mathcal{L}_{\text{align}}$ causes a potential issue of the model encoding all information into $\widehat{E}_v$, ignoring the original caption tokens E_m. In such a case, changing E_m would not change the textual feature, defeating the purpose of data augmentation. We therefore enforce that the semantics of E_m are correctly reflected in $\Omega(\tau(E_m))$. Thus we define the *Preservation Loss* $\mathcal{L}_{\text{preserve}}$ as

$$\frac{1}{M} \sum_{m=1}^{M} \left[-\log \frac{\exp(\cos(\widehat{f}_m, f_m))}{\sum_{n=1}^{M} \exp(\cos(\widehat{f}_m, f_n))} + w(1 - \cos(\widehat{f}_m, g)) \right], \tag{4}$$

where the first term ensures $\widehat{f}_m = \Omega(\tau(E_m))$ is closest to its original text features $f_m = \Omega(E_m)$ and farther from the textual features f_n of other captions, hence preserving its textual semantics. Meanwhile, as each $\widehat{f}_m$ represents a modification of the original clip and they share similarity, we introduce the second term to keep the features near to the visual feature g. We use the weight w to control the effect of this term. Although LogFA does not impose a hand-designed structural constraint on the augmented feature, the preservation loss anchors each edit to both its caption semantics and the original visual feature. This implicit constraint complements the GDAG constraint at the procedure level, which preserves valid action dependencies, optionality, repetitions, and hierarchy.

Overall, the optimization objective is the summation of the two losses. We optimize $\widehat{E}_v$ for a fixed number of iterations and use $\{\widehat{f}_m\}_{m=0}^{M}$ as features obtained by the local augmentation method.

3.3 Global Augmentation

While local augmentation captures intra-action diversity, a task also has inter-action diversity as one goal can often be achieved with flexible ordering of actions.

Yet, with only a few training videos, a TAS model may overfit to the observed orders. Hence, we augment videos at global level by sampling diverse, valid action sequences of task and reassembling the action segments in a video accordingly.

Generalized Directed Acyclic Graph. A natural choice to encode the possible procedures of a task is via a directed acyclic graph (DAG), where an edge $u \rightarrow v$ denotes action u is a prerequisite of v. Valid orders are obtained via running topological sorting on the graph. However, in real-world scenarios, there often exist complex action patterns which DAG fails to encode, such as optional actions, repeated actions, or contiguous routines. For example, Figure 3 (left graph) shows the actions for an example salad task, where *add sauce* and *add pepper* are optional actions. Actions for preparing carrots form a routine that is generally performed together, although they can be interleaved with other actions. The routines of preparing carrot and lettuce can also be repeated multiple times based on a user's demand.

To accommodate those complex patterns, we therefore introduce a *Generalized DAG (GDAG)* that supports: ***(i)*** *flat* nodes (single actions) and *hierarchical* nodes (sub-GDAGs that contain multiple actions of a routine and should be executed contiguously), and ***(ii)*** a random variable $\mathcal{C}$ for each node, indicating how many times a node/action should be repeated. Standard actions have $\mathcal{C}{=}1$; optional actions allow $\mathcal{C}{\in}\{0,1\}$; repeated actions allow $\mathcal{C}{\geq}2$. For simplicity, we assume each $\mathcal{C}$ follows a uniform distribution parameterized by a lower bound c_1 and upper bound c_2. Figure 3 (left graph) shows the GDAG for the salad task, where the numbers on each node show the values of their repetition parameter (c_1, c_2). It includes two optional actions and also two hierarchical nodes denoting the two routines, while each routine can be performed repeatedly.

Augmentation with GDAG. To construct GDAG, we leverage an LLM by providing the GDAG definition, the action list, and example sequences from training videos as input. The LLM produces GDAG via JSON format. We provide the prompt details in the supplementary materials. Next, to sample action sequences from GDAG, we design a simple *convert-and-sample* approach: (1) We convert a GDAG to a DAG by sampling the value of $\mathcal{C}$ for each node, then remove nodes with $\mathcal{C}{=}0$ and duplicate nodes with $\mathcal{C}{>}1$. Figure 3 (middle graph) shows an example of the converted DAG. (2) We can easily sample an action sequence from the DAG via a topological-sort algorithm, which involves finding actions with no pending prerequisites and performing Breadth-First Search from those actions. Figure 3 (right graph) demonstrates an obtained action sequence. If the sequence contains hierarchical nodes, we sample action sequences from its sub-GDAG and replace the node with the sequence.

With the final sequence, we cut a video into action segments and reassemble the segments according to the sampled ordering. We also interpolate the features near the action boundaries to avoid sharp transitions between actions. GDAG generation can fail when the LLM proposes invalid dependencies or inaccurate repetition ranges. In practice, providing training action sequences in the prompt constrains the generated graph, while sampling repetitions at each iteration reduces sensitivity to a single inaccurate value of $\mathcal{C}$.

3.4 Training and Inference Pipeline

Training. As explained in Section 3.1, our framework includes a *feature augmentation stage* and a *TAS training stage*. In the feature augmentation stage, we obtain the augmented features for all video clips using local augmentation and obtain the GDAG for each task via the global augmentation method.

In the TAS training stage, we train a TAS model using the augmented data. In each training iteration, we use the augmented clip features with a probability ratio r_1, or use the original clip features otherwise. For augmented clips, we then randomly select a feature $\widehat{f}_m$ out of the M augmentations. We further interpolate it with the original visual feature, g, to expose the TAS model to the gradual changes from the original feature to the modified feature (e.g., the color of the egg plastic case shifting from blue to red),

$$\widehat{f}_m^{(\alpha)} = \alpha\, \widehat{f}_m + (1-\alpha)\, f^*, \quad \alpha \sim \mathcal{U}[0,1], \tag{5}$$

where α is a random variable uniformly sampled between 0 and 1. After this step, we re-order the video clips with a probability r_2. In this case, we randomly sample a valid action sequence from the task's GDAG and re-order the clips accordingly. In general, we set r_1=r_2 to avoid over-tuning. All augmentation models and sampled sequences are constructed solely from the training split; test videos and features are never accessed during augmentation or training and are used only for evaluation.

Inference. Following standard practice, we evaluate on visual clip features extracted from test videos, *without* any augmentation. The gains of LogFA come entirely from richer training signals created in feature space.

4 Experiments

In this section, we provide empirical analyses to verify the effectiveness of the LogFA framework. We first present ablations of key design choices in our method.

4.1 Experimental Setup

Dataset. We evaluate our method on popular egocentric TAS benchmarks, EgoPER [20] and EgoProceL [3]. **EgoPER** features videos from five cooking recipes. Each recipe includes videos collected from different subjects, and contains both correct videos and error videos with abnormal/incorrect action execution. Overall, it has 389 videos with 57 actions. **EgoProceL** contains diverse videos from five datasets. It has 16 tasks collected from different environments and includes only videos of correct task executions. It contains 1055 videos with 130 actions. To measure the effect of data augmentation, we do not use all available training videos. For each recipe in a dataset, we sample only three videos for training. If the dataset provides subject identities, we ensure the training videos

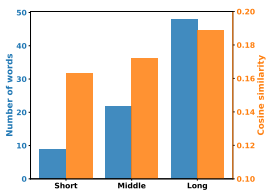

Fig. 4: Effect of caption length and granularity.

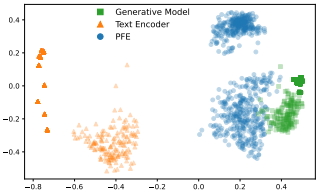

Fig. 5: Effect of Prompt-based Feature Enhancement.

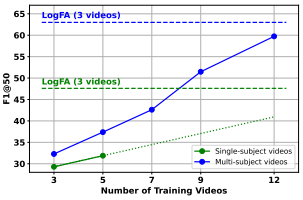

Fig. 6: Importance of Data Diversity.

	Edit	F1@10	F1@25	F1@50	Acc
Base w/o DP	50.4	52.3	46.7	30.2	44.8
Base	47.1	52.7	47.5	32.2	36.7
Base + Type1	51.9	55.8	50.3	34.1	43.7
Base + Type2	53.7	56.6	51.9	35.8	44.7
Base + Type3	53.1	55.4	51.5	36.8	45.6
Base + Type4	53.2	55.5	51.3	37.0	45.4
Base + All	**55.8**	**57.6**	**53.5**	**38.4**	45.0
Base + All (EgoVLP)	53.5	56.9	52.5	36.7	44.9

Table 1: Effect of different caption modifications and VLM backbones on EgoPER Coffee.

are from the same subject and use videos of the other subjects for testing, thus evaluating robustness on novel test videos.

Metrics. Following [4, 11, 25, 30, 50, 60], we compute segmental Edit distance score (*Edit*), segmental F1 score (*F1*) at three overlapping thresholds 10%, 25%, 50%, denoted by F1@{10, 25, 50} and framewise accuracy (*Acc*).

Implementation Details. We set $r_1 = r_2 = 0.3$ and use the state-of-the-art model [30] as our TAS backbone. For local augmentation, we use SigLIP2 [54] as our visual and textual backbone. While it is an image-based model, we found empirically that its feature quality is better than a recent video-based encoder [65]. SigLIP2 can be viewed as a video encoder with input clip length equal to 1. We use Qwen2.5-VL-8B [36] and Qwen3-8B [37] as our LLMs for the local augmentation method. We extract captions with a 16-frame sliding window and 8-frame overlap at 10 fps. For each video clip, we train PFE with the Adam optimizer, using a learning rate of 1e-4 for 800 iterations; the loss saturates within 700–900 iterations in 94% of cases. For generating GDAG in global augmentation, we use GPT-5o with thinking mode. We also evaluate Qwen3-30B for GDAG generation and find comparable results, indicating that LogFA is not tied to a closed-source LLM. For baselines, we combine multiple image transformations, including horizontal flip, rotation, grayscale, perspective change, affine transformation, and color jittering. We found compositional methods such as MixUp [61] often reduce performance, as changing clip labels may create false action orderings and confuse models. For the generative model, we use the latest method, Qwen-Edit [**?**], for its state-of-the-art image editing performance. *We will release our code and data.*

4.2 LogFA Design

We discuss the effectiveness of our local and global augmentations. Unless otherwise specified, we evaluate on the coffee recipe of the EgoPER dataset.

PFE in Local Augmentation. We first examine PFE by verifying the assumption that misalignment between visual and textual features is caused by limited information in captions. Hence, we prompt the LLM to generate captions with different levels of detail, producing short, medium, and long captions for each clip. We then compute the cosine similarity between the clip's visual feature and the textual features of the three captions. As shown in Figure 4, using longer, more detailed captions improves the alignment between visual and textual features. This shows that adding details bridges the feature gap.

Next, Figure 5 shows that PFE improves alignment between visual and textual features: we use an LLM to generate modifications for each clip, then compare (1) editing the clip with a generative model and extracting its visual feature (green square), (2) directly computing textual features of the modified captions (orange triangle), and (3) using PFE to obtain enhanced textual features (blue circle). We visualize the features using t-SNE [33]. We observe a clear separation between raw textual features and visual features, indicating feature misalignment. *PFE features align more closely with the visual features.* To quantitatively measure the alignment, we further train linear classifiers to distinguish (1) features of true video clips vs. features of generated clips and (2) features of true clips vs. features of PFE. The accuracies are 61.2% and 65.1%, respectively, indicating that PFE features align well with true features and are comparable in quality to generative methods. Indeed, while the cosine similarity between textual features and visual clip features is 0.18, the similarity between PFE features and visual features is 0.72, showing a significant improvement. Hence, training with local augmentation achieves performance similar to using a generative model, while avoiding heavy computational cost. This analysis does not claim that PFE features are superior to generated visual features; rather, it shows that PFE substantially closes the gap from raw text features. We provide more results in Section 4.3.

Caption Modification in Local Augmentation. In Table 1, we study the effect of different ways to generate modified captions. Specifically, *Base* denotes training with the three available videos and no augmentation, while *Base w/o DP* further removes dropout layers. We compare four modification types: **Type1** rephrases the caption without changing meaning; **Type2** adds extra, unimportant information (e.g., background scene, table layout); **Type3** changes the texture or color of objects; **Type4** changes how an action is executed (e.g., different objects). As shown in the second block of Table 1, all four types outperform *Base*, demonstrating the efficacy of local augmentation. Type1 has a smaller effect as it does not alter semantics but acts as textual jittering. Type2 is stronger as it improves robustness to action-irrelevant information. Type3 and Type4 yield the largest gains by enabling action-related semantic changes. In the last row of the table, we test switching the VLM backbone from SigLIP2 to EgoVLP. Using

	Edit	F1@10	F1@25	F1@50	Acc
Base w/o DP	46.4	43.4	38.7	27.5	44.0
Base	45.9	46.6	41.1	28.8	44.3
Base + DAG	46.2	46.1	41.1	28.8	46.1
Base + GDAG	**50.9**	**50.4**	**45.4**	**32.2**	**53.0**

Table 2: Effect of Generalized DAG.

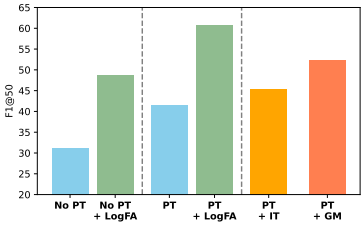

Fig. 7: Few-shot Adaptation.

EgoVLP as the local-augmentation backbone yields scores close to SigLIP2, with a small drop likely due to EgoVLP's more specific training prompt format.

GDAG in Global Augmentation. We next study the effectiveness of global augmentation. We compare capturing task procedures via DAG vs. GDAG on EgoProceL, which exhibits more complex procedural variation than EgoPER. In Table 2, *Base* shows training with only three videos and no augmentation. Using DAG fails to achieve consistent improvement across metrics because DAG only captures action dependencies. GDAG additionally models action repetition and co-occurrence, common in real-world videos, and improves F1@50 by 3.4%. Generating GDAGs with Qwen3-30B obtains comparable performance to GPT-5o (31.7 vs. 32.2 F1@50), supporting the reproducibility of the graph-construction step. This validates the importance of generating diverse procedures for a task, often ignored by image-based augmentation methods [56, 58].

Importance of Data Diversity. After validating the efficacy of both local and global augmentations, we verify the importance of data diversity. In Figure 6, we first compare the effect of training a base model with different numbers of videos. The blue curve shows the results when training videos are collected from multiple subjects, while the green curve uses videos from a single subject. As in EgoPER, each subject has at most five videos, we use the dotted green curve to show estimated performance when training on more than five videos of the same subject. We observe that (1) training with three multi-subject videos has higher performance than training with five single-subject videos; and (2) the slope of the blue curve is much larger than the green curve. This validates that *increasing data diversity is more important than increasing data quantity.*

However, collecting diverse videos is costly, as it requires recruiting more subjects, purchasing varied equipment, and designing different task procedures. In comparison, LogFA generates diverse training features with only a few videos. In Figure 6, the two dashed lines show the performance of LogFA with *only three single/multi-subject videos.* Both outperform the baseline trained with 12 single/multi-subject videos. Hence, LogFA effectively reduces annotation cost by *75%.* Furthermore, we compare using all available videos for training. LogFA achieves F1@50 of 74.9%, while the base model using no augmentation, image transformation, or generative methods achieves 63.1%, 63.5%, and 68.2%, respectively. This shows that *lack of data diversity cannot be addressed by simply*

	In-Domain Videos					Out-of-Domain Videos				
	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}			Acc
Base w/o DP	76.7	76.7	72.7	59.3	65.9	62.0	64.8	60.1	45.8	56.2
Base	77.5	80.9	77.3	64.1	65.3	60.2	67.8	62.9	47.7	53.8
Base + Image Transform	79.3	82.2	78.9	66.4	67.7	62.0	69.1	64.8	50.0	55.5
Base + Generative Model	77.3	80.3	76.7	63.8	65.3	59.4	66.8	61.9	47.4	52.6
Base + Local Aug	80.6	82.4	79.1	67.3	71.6	65.0	70.6	66.3	52.7	62.1
Base + Global Aug	80.8	84.5	81.1	67.4	71.3	65.2	73.2	68.6	54.3	62.9
Base + LogFA	**82.8**	**86.1**	**83.0**	**70.7**	**74.0**	**68.6**	**76.0**	**71.9**	**57.0**	**66.0**

Table 3: Performance on EgoPER dataset.

adding more data, and LogFA consistently improves performance even with full data.

Few-Shot Adaptation. Another common setting that often encounters data scarcity is few-shot adaptation, where a pretrained model needs to adapt to a new task with only a few videos. Thus, we pretrain the base model with all videos on EgoPER except those of *coffee recipe*, then finetune it using three videos of *coffee recipe*. In Figure 7, PT, IT, GM stands for pretraining, image transformation, and generative model, respectively. Even with pretraining, LogFA generalizes to few-shot adaptation settings, exceeds the base model by 19%, and outperforms other augmentation methods. This shows that *LogFA is a generic augmentation framework that can be applied to various data-scarce settings*. In the remaining experiments, for easy comparison and alleviating the impact of different pretraining strategies, we focus on training from scratch to better demonstrate the effect of data augmentation.

4.3 Performance on TAS Benchmark

We compare LogFA against prior augmentation methods on EgoPER and EgoProceL. In Tables 3 and 5, *Base* uses three training videos with dropout enabled and no augmentation; *Base w/o DP* removes the dropout layers.

EgoPER. Table 3 reports results on EgoPER. EgoPER includes normal videos and error videos where tasks are deliberately executed in unusual ways. We train on normal videos. In testing, we consider normal videos as in-domain and error videos as out-of-domain. Some error videos contain action classes absent in training (e.g., steps to correct mistakes); we exclude frames of those actions in evaluation.

Training with LogFA achieves the highest performance on *all* metrics. Compared to *Base*, LogFA boosts F1@50 by 6.6% on in-domain videos, reducing data needs. More importantly, LogFA yields a larger improvement on out-of-domain videos, with 9.3% higher F1@50 than *Base*, indicating greater diversity in augmented training data. Image transformations fail to generate sufficiently diverse data and yield limited gains, especially on out-of-domain videos (+2.3% F1@50). Generative augmentation does not consistently outperform the base model in our

	Edit	F1@10	F1@25	F1@50	Acc
Base	54.7	64.6	62.4	54.5	77.9
Base + Image Transform	61.0	70.1	67.4	58.1	77.8
Base + Generative Model	67.0	75.0	71.5	61.2	74.9
Base + LogFA	**68.6**	**76.4**	**73.3**	**63.3**	77.8

Table 4: Generalization to MSTCN++ on EgoPER all tasks.

	Edit	F1@10	F1@25	F1@50	Acc
Base w/o DP	46.4	43.4	38.7	27.5	44.0
Base	45.9	46.6	41.1	28.8	44.3
Base + Image Transform	46.7	48.2	44.6	32.4	46.3
Base + Local Aug	50.9	50.3	44.9	34.5	50.8
Base + Global Aug	51.9	50.4	45.4	32.2	53.0
Base + LogFA	**52.0**	**51.2**	**46.9**	**35.0**	**54.7**

Table 5: Performance on EgoProceL dataset.

reproduced runs, while LogFA remains substantially stronger by combining local and global feature-space augmentation.

Critically, running a generative model incurs large computational cost. A recent method [**?**] takes ∼1 minute to generate one image edit using two 40,GB GPUs. An average EgoPER video has ∼400 frames at 1 fps, demanding ∼13.3 GPU-hours to generate a single augmentation for one video. Captioning and text modification are shared by LogFA and generative baselines, taking 7.1 sec/clip and 1.4 sec/clip, respectively. After these shared steps, LogFA operates directly in feature space: for local augmentation, PFE takes only ∼6 seconds per augmentation on one GPU. Global augmentation's cost lies mainly in generating a GDAG with an LLM, which is computed once per task. Therefore, LogFA achieves on-par augmentation performance with ∼10× less computation time.

EgoProceL. Table 5 reports results on EgoProceL. We evaluate on normal videos only, as error videos are unavailable. Because EgoProceL has more and longer videos than EgoPER, generative augmentation would require *3480 GPU-days*, so we report only image transformation results.

Local augmentation outperforms image transformations by 2% F1@50, again validating that PFE aligns textual and visual features, enabling us to leverage contrastive VLMs to generate diverse clip features with enhanced textual features. Meanwhile, global augmentation improves over *Base* by 3.4% F1@50 and raises the Edit score by 6%, indicating exposure to more diverse procedures. Combining both local and global augmentation achieves the best performance, with F1@50 of 35.05%.

To verify that LogFA is not specific to the FACT backbone, we also evaluate it with MSTCN++ [25]. As shown in Table 4, LogFA generalizes across TAS models and exceeds both image transformation and generative augmentation baselines.

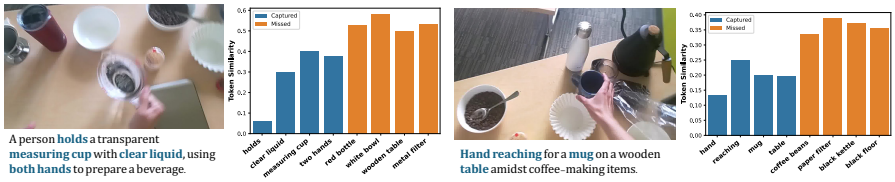

Fig. 8: Study of tokens learned by PFE.

4.4 Qualitative Results

We further study the semantics of the tokens learned by PFE. Specifically, given a clip and its unmodified caption t_0, we use an LLM to extract two sets of phrases: *(1)* phrases whose information is captured in the caption, and *(2)* phrases whose information is missed by the caption. We compare the similarity between the learned tokens and these phrases. Specifically, we convert each phrase into text embedding tokens and compute the average cosine similarity between each learned token and the phrase tokens. As shown in Figure 8, we visualize the similarity scores for two example clips. The similarity with the missed phrases (orange bars) are consistently higher than those of the captured phrases (blue bars). This also validates that PFE is able to capture the information missing in the original captions, hence improving the alignment between textual and visual features.

5 Conclusion

We introduced LogFA, a feature-space data augmentation framework for egocentric Temporal Action Segmentation that combines *local* text-guided edits via Prompt-based Feature Enhancement (PFE) with *global* procedural diversification via a Generalized DAG (GDAG). By operating directly in feature space, LogFA produces semantically rich variations without pixel-level synthesis, yielding substantial gains in generalization to unseen environments at a fraction of the computational cost of generative approaches. Extensive experiments on EgoPER and EgoProceL validate that both components contribute complementary improvements and that their combination achieves state-of-the-art performance under low-data settings. While our experiments focus on egocentric data, LogFA is not specific to viewpoint; evaluating third-person TAS benchmarks such as 50Salads, Breakfast, and GTEA is an important direction for future work.

Acknowledgements

This work was funded, in part, by ARPA-H (1AY2AX000062), NSF (IIS-2115110), ONR (N000142512287) and DARPA (HR00112220001). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Government.

References

1. Ahn, H., Lee, D.: Refining action segmentation with hierarchical video representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16302–16310 (October 2021)
2. Alayrac, J.B., Bojanowski, P., Agrawal, N., Sivic, J., Laptev, I., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. IEEE Conference on Computer Vision and Pattern Recognition (2016)
3. Bansal, S., Arora, C., Jawahar, C.: My view is the best view: Procedure learning from egocentric videos. In: European Conference on Computer Vision (ECCV) (2022)
4. Behrmann, N., Golestaneh, S.A., Kolter, Z., Gall, J., Noroozi, M.: Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In: ECCV (2022)
5. Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C.: D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2019)
6. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Ma, J., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision (IJCV) **130**, 33–55 (2022), `https://doi.org/10.1007/s11263-021-01531-2`
7. Ding, G., Golong, H., Yao, A.: Coherent temporal synthesis for incremental action segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2024)
8. Ding, L., Xu, C.: Weakly-supervised action segmentation with iterative soft boundary assignment. IEEE Conference on Computer Vision and Pattern Recognition (2018)
9. Elhamifar, E., Huynh, D.: Self-supervised multi-task procedure learning from instructional videos. European Conference on Computer Vision (2020)
10. Elhamifar, E., Naing, Z.: Unsupervised procedure learning via joint dynamic summarization. International Conference on Computer Vision (2019)
11. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3575–3584 (2019)
12. Fayyaz, M., Gall, J.: Sct: Set constrained temporal transformer for set supervised action segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2020)
13. Fried, D., Alayrac, J.B., Blunsom, P., Dyer, C., Clark, S., Nematzadeh, A.: Learning to segment actions from observation and narration. Annual Meeting of the Association for Computational Linguistics (2020)
14. Hussein, N., Gavves, E., Smeulders, A.W.: Videograph: Recognizing minutes-long human activities in videos. In: ICCV Workshop on Scene Graph Representation and Learning (2019)
15. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19113–19122 (2023)
16. Kuehne, H., Gall, J., Serre, T.: An end-to-end generative framework for video segmentation and recognition. IEEE Winter Conference on Applications of Computer Vision (2016)

17. Kukleva, A., Kuehne, H., Sener, F., Gall, J.: Unsupervised learning of action classes with continuous temporal embedding. IEEE Conference on Computer Vision and Pattern Recognition (2019)
18. Lai, B., Dai, X., Chen, L., Pang, G., Rehg, J.M., Liu, M.: Lego: Learning egocentric action frame generation via visual instruction tuning. In: European Conference on Computer Vision. pp. 135–155. Springer (2024)
19. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. IEEE Conference on Computer Vision and Pattern Recognition (2017)
20. Lee, S., Lu, Z., Zhang, Z., Hoai, M., Elhamifar, E.: Error detection in egocentric procedural task videos. IEEE Conference on Computer Vision and Pattern Recognition (2024)
21. Li, J., Lei, P., Todorovic, S.: Weakly supervised energy-based learning for action segmentation. International Conference on Computer Vision (2019)
22. Li, J., Todorovic, S.: Anchor-constrained viterbi for set-supervised action segmentation. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
23. Li, J., Todorovic, S.: Set-constrained viterbi for set-supervised action segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2020)
24. Li, M., Chen, L., Duarr, Y., Hu, Z., Feng, J., Zhou, J., Lu, J.: Bridge-prompt: Towards ordinal action understanding in instructional videos. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (jun 2022)
25. Li, S.J., AbuFarha, Y., Liu, Y., Cheng, M.M., Gall, J.: Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.3021756
26. Li, Z., Abu Farha, Y., Gall, J.: Temporal action segmentation from timestamp supervision. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
27. Liu, D., Li, Q., Dinh, A.D., Jiang, T., Shah, M., Xu, C.: Diffusion action segmentation. International Conference on Computer Vision (2023)
28. Lu, Z., Elhamifar, E.: Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. International Conference on Computer Vision (2021)
29. Lu, Z., Elhamifar, E.: Set-supervised action learning in procedural task videos via pairwise order consistency. IEEE Conference on Computer Vision and Pattern Recognition (2022)
30. Lu, Z., Elhamifar, E.: Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2024)
31. Lu, Z., Elhamifar, E.: Multi-modal few-shot temporal action segmentation. International Conference on Computer Vision (2025)
32. Lu, Z., Iftekhar, A., Mittal, G., Meng, T., Wang, X., Zhao, C., Kukkala, R., Elhamifar, E., Chen, M.: Decafnet: Delegate and conquer for efficient temporal grounding in long videos. IEEE Conference on Computer Vision and Pattern Recognition (2025)
33. Maaten, L.V.D., Hinton, G.E.: Visualizing data using t-sne. Journal of Machine Learning Research (2008)
34. Mikołajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. In: 2018 International Interdisciplinary PhD Workshop (IIPhDW). pp. 117–122 (2018). https://doi.org/10.1109/IIPHDW.2018.8388338

35. Nawhal, M., Mori, G.: Activity graph transformer for temporal action localization (2021)
36. Qwen-Team: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
37. Qwen-Team: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
39. Rahaman, R., Singhania, D., Thiery, A., Yao, A.: A generalized and robust framework for timestamp supervision in temporal action segmentation. In: Computer Vision–ECCV 2022: 17th European Conference (2022)
40. Richard, A., Kuehne, H., Gall, J.: Action sets: Weakly supervised action segmentation without ordering constraints. IEEE Conference on Computer Vision and Pattern Recognition (2018)
41. Richard, A., Kuehne, H., Iqbal, A., Gall, J.: Neuralnetwork-viterbi: A framework for weakly supervised video learning. IEEE Conference on Computer Vision and Pattern Recognition (2018)
42. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. IEEE Conference on Computer Vision and Pattern Recognition (2012)
43. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
44. Shen, Y., Elhamifar, E.: Semi-weakly-supervised learning of complex actions from instructional task videos. IEEE Conference on Computer Vision and Pattern Recognition (2022)
45. Shen, Y., Elhamifar, E.: Progress-aware online action segmentation for egocentric procedural task videos. IEEE Conference on Computer Vision and Pattern Recognition (2024)
46. Shen, Y., Wang, L., Elhamifar, E.: Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. IEEE Conference on Computer Vision and Pattern Recognition (2021)
47. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of big data **6**(1), 1–48 (2019)
48. Sigurdsson, G.A., Divvala, S., Farhadi, A., Gupta, A.: Asynchronous temporal fields for action recognition. IEEE Conference on Computer Vision and Pattern Recognition (2017)
49. Singh, B., Marks, T.K., Jones, M., Tuzel, O., Shao, M.: A multi-stream bi-directional recurrent neural network for finegrained action detection. IEEE Conference on Computer Vision and Pattern Recognition (2016)
50. Singhania, D., Rahaman, R., Yao, A.: Coarse to fine multi-resolution temporal convolutional network. CoRR **abs/2105.10859** (2021), https://arxiv.org/abs/2105.10859
51. Souri, Y., Fayyaz, M., Minciullo, L., Francesca, G., Gall, J.: Fast Weakly Supervised Action Segmentation Using Mutual Consistency. PAMI (2021)
52. Souček, T., Gatti, P., Wray, M., Laptev, I., Damen, D., Sivic, J.: Showhowto: Generating scene-conditioned step-by-step visual instructions (June 2025)
53. Sun, Y., Zhou, H., Yuan, L., Sun, J.J., Li, Y., Jia, X., Adam, H., Hariharan, B., Zhao, L., Liu, T.: Video creation by demonstration (2024), https://arxiv.org/abs/2412.09551

54. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., Hénaff, O., Harmsen, J., Steiner, A., Zhai, X.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features (2025), https://arxiv.org/abs/2502.14786
55. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 139–149 (2022)
56. Xu, M., Yoon, S., Fuentes, A., Park, D.S.: A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognition **137**, 109347 (2023). https://doi.org/https://doi.org/10.1016/j.patcog.2023.109347, https://www.sciencedirect.com/science/article/pii/S0031320323000481
57. Xu, M., Gao, M., Li, S., Lu, J., Gan, Z., Lai, Z., Cao, M., Kang, K., Yang, Y., Dehghan, A.: Slowfast-llava-1.5: A family of token-efficient video large language models for long-form video understanding. arXiv preprint arXiv:2503.18943 (2025)
58. Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., Shen, F.: Image data augmentation for deep learning: A survey (2023), https://arxiv.org/abs/2204.08610
59. Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: Dense detailed labeling of actions in complex videos. International Journal of Computer Vision (2018)
60. Yi, F., Wen, H., Jiang, T.: Asformer: Transformer for action segmentation. In: The British Machine Vision Conference (BMVC) (2021)
61. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=r1Ddp1-Rb
62. Zhang, J., Tsai, P.H., Tsai, M.H.: Semantic2graph: Graph-based multi-modal feature fusion for action segmentation in videos (2022)
63. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 13001–13008 (2020)
64. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022)
65. Zhu, B., Lin, B., Ning, M., Yan, Y., Cui, J., HongFa, W., Pang, Y., Jiang, W., Zhang, J., Li, Z., Zhang, C.W., Li, Z., Liu, W., Yuan, L.: Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment (2023)
66. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. IEEE Conference on Computer Vision and Pattern Recognition (2019)