Multi-Modal Few-Shot Temporal Action Segmentation

Zijia Lu
Northeastern University
lu.zij@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

Abstract

Procedural videos are critical for learning new tasks. Temporal action segmentation (TAS), which classifies the action in every video frame, has become essential for understanding procedural videos. Existing TAS models, however, learn a fixed-set of tasks at training and unable to adapt to novel tasks at test time. Thus, we introduce the new problem of Multi-Modal Few-shot Temporal Action Segmentation (MMF-TAS) to learn open-set models that can generalize to novel procedural tasks with minimal visual/textual examples. We propose the first MMF-TAS framework, by designing a Prototype Graph Network (PGNet). In PGNet, a Prototype Building Block summarizes action information from support videos of the novel tasks via an Action Relation Graph, and encodes this information into action prototypes via a Dynamic Graph Transformer. Next, a Matching Block compares action prototypes with query videos to infer framewise action labels. To exploit the advantages of both visual and textual modalities, we compute separate action prototypes for each modality and combine the two modalities through prediction fusion to avoid overfitting on one modality. By extensive experiments on procedural datasets, we show our method successfully adapts to novel tasks during inference and significantly outperforms baselines. Our code is available at https://github.com/ZijiaLewisLu/ICCV2025-MMF-TAS.

1. Introduction

Procedural videos have become increasingly important for learning new skills (e.g., parsing instructional videos) and understanding users' goal-oriented activities (e.g., inferring executed steps of a task). As a result, temporal action segmentation (TAS), which segments long procedural videos into non-overlapping action/step segments, has gained increasing attention [3, 36, 40, 53, 54, 58], with various applications, such as content retrieval and AI task assistants [4, 26, 27, 53, 73, 76].

Current TAS methods for procedural videos face a critical limitation: models are learned for a closed set of procedural tasks with hundreds of training videos, and cannot adapt to novel tasks without retraining. Given the vast number of procedural tasks and the high cost of annotating long videos, we need an open-set approach that can adapt to novel tasks using a few annotated videos.

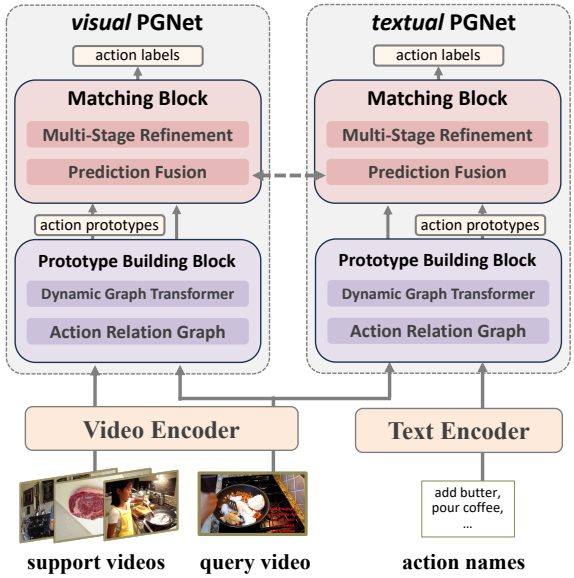

Figure 1. **Our MMF-TAS framework.** We propose a Prototype Graph Network (PGNet) that segments query videos of novel tasks by summarizing action information from support videos and matching it with query videos. We learn separate visual and textual PGNets. In each PGNet, *Prototype Building Block* computes action prototypes from the corresponding modality to summarize action information. *Matching Block* compares action prototypes with query videos to predict action labels and combine modalities via prediction fusion.

Multi-Modal Few-shot Temporal Action Segmentation (MMF-TAS). Inspired by humans' ability to learn new tasks from a few examples, we study the new problem of MMF-TAS. In this setting, models receive a few support videos of novel tasks at test time and leverage the information of both visual (action demonstration) and textual (action names) modalities to segment query videos. Standard few-shot setting assumes having both support videos and their framewise labels, which can be limiting given the long durations of procedural videos. On the other hand, the Multi-Modal Few-shot setting, which we study in the paper, allows handling more general and realistic settings, such as when only action classes are known but support videos are

unavailable (zero-shot setting) and when support videos are available but with partial or no labels (unlabeled or weakly-labeled few-shot setting).

Addressing MMF-TAS faces unique challenges. First, adapting to novel tasks requires understanding the *multiple actions* and their *dependencies* within tasks. However, prior few-shot video methods [19, 25, 43, 46, 67] learn to generalize to novel actions using single-action videos/clips and do not consider action dependencies. Recent Video-Language Models (VLMs) [18, 34, 77, 78] demonstrate open-world video understanding with the potential for identifying actions of novel tasks. Yet, constrained by their computation burden, they primarily excel in short-video recognition/captioning and are unable to capture long temporal action dependencies, which is critical for successful TAS.

Second, MMF-TAS requires combining the information in both visual and textual features. However, we found direct fusion of the features, commonly employed by prior MMF image or video methods [19, 44, 68], makes models over-rely on textual features and underutilize visual features, causing inferior performance. This can be observed in Figure 2: when fusing features, test accuracy on novel tasks is similar to that of using only textual features, yet *worse* than using only visual features (more details in Remark 1).

Paper Contributions. We introduce the MMF-TAS problem and address the aforementioned challenges with our proposed **Prototype Graph Network** (PGNet). PGNet can adapt to novel tasks using a *summarize-and-match* strategy: It first summarizes the information of actions in support videos with a *Prototype Building Block*, then matches this information with query videos to effectively infer framewise action labels with a *Matching Block* (see Figure 1).

The Prototype Building Block summarizes action information by computing representative features for each action (action prototypes) using either the visual demonstrations in support videos or the textual semantic of action names. We design an *Action Relation Graph* that can capture different aspects of action information (e.g., action pattern, distinctions, and dependencies). Our *Dynamic Graph Transformer* then computes action prototypes from the graph while learning specialized parameters tailored for different action information.

The Matching Block compares the similarity between action prototypes and query videos to infer and iteratively refine framewise action labels. To address the issue that fusing two modalities via features causes over-reliance on textual modality, Matching Block combines modalities through prediction fusion. This leverages the idea that predictions contain rich information about inputs and can transfer knowledge across models [15, 16]. Specifically, we build separate visual and textual prototypes in Prototype Building Block, and use predictions from one modality to guide the other in Matching Block.

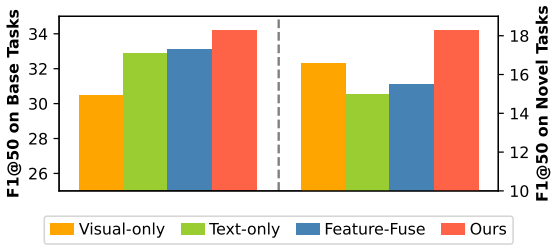

Figure 2. **Effectiveness of Visual and Textual Modality** on base and novel tasks. Visual features (orange) generalize better on novel tasks while textual features (green) are better on base tasks seen at training. When models learn to fuse features (blue), it over-relies on features suitable for training data (textual features) and underutilizes the other features (visual features). Hence, its result on novel tasks is *worse* than using only visual features. Our framework (red) harvests the advantages of both modalities.

Finally, we conduct extensive experiments on procedural datasets. We show that PGNet not only significantly outperforms baselines in the standard few-shot setting, but also allows flexible test scenarios to further reduce annotation costs, such as zero-shot and unlabeled/weakly-labeled few-shot settings.

2. Related Works

Multi-Modal Few-Shot Learning. Few-Shot (FS) learning aims to generalize to novel classes using a few labeled support examples. It has been studied in the image domain [20, 59, 62, 65, 66] and single-action video understanding [19, 25, 43, 46, 67]. Early FS video methods [46, 74] address Action Recognition that classifies the actions of trimmed video clips and finds frame alignments among support and query videos. Recently, [19, 25, 67, 71] study FS Action Localization that locates single-action in untrimmed videos by separating action relevant and irrelevant frames. However, the assumption of single-action videos leads those works to treat actions as isolated instances, ignoring the crucial action relations. Moreover, standard few-shot setting uses only visual modality, overlooking the readily available textual semantics in action names. Recent approaches [19, 35, 44] explore Multi-Modal FS to incorporate both visual and textual modalities, yet similarly focus on image and single-action video understanding. In real-world cases, however, videos are long and contain multiple actions with inherent causal relationships. Applying single-action FS methods to such videos cannot exploit the temporal dependencies among actions and obtain poor results. Thus, we propose MMF-TAS that leverages both visual/textual modalities to classify actions while also taking into account the relations of actions.

Temporal Action Segmentation. Temporal Action Segmentation (TAS) has been studied in unsupervised [2, 7, 8, 12, 23, 55, 79], weakly-supervised [11, 28–30, 33, 37, 38, 48, 50–52] and full-supervised [1, 3, 4, 13, 14, 17, 22, 24, 26, 27, 31, 36, 39–42, 45, 53, 54, 56–58, 60, 61, 69, 73, 75] settings. MSTCN [9] and subsequent works [32, 58] uses a

multi-stage refinement mechanism for accurate predictions. DiffAct [36] extends this mechanism to a diffusion process. FACT [40] establishes new SOTA via parallel action and frame-level temporal modeling. However, these methods focus on classifying a fixed set of known tasks/actions with hundreds of training videos. To handle new tasks, they require collecting new datasets of tasks. To reduce annotation cost, we propose the first MMF-TAS framework that adapts to novel tasks with a few support videos. We introduce prototype graph network to learn action prototypes that encode actions and their dependencies of novel tasks, and compare them with query videos to predict action labels.

3. Proposed MMF-TAS Framework

3.1. Problem Definition

Multi-Modal Few-shot Temporal Action Segmentation (MMF-TAS) aims to segment query videos from novel tasks by comparing them to the support videos of the tasks. We follow the standard K-*way* V-*shot setting*, which assumes we are extending models to K tasks, where each task has V labeled support videos, and query videos belong to one of the K tasks. Hence, the model inputs are a query video $\mathcal{V}$ and support videos $\{(\hat{\mathcal{V}}_{k,v}, \hat{Y}_{k,v})\}_{k=1,v=1}^{K,V}$, where $\hat{\mathcal{V}}_{k,v}$ and $\hat{Y}_{k,v}$ are the v-th support video of task k and its framewise action labels, respectively. Models predict the task and framewise action labels of the query video.

3.2. Framework Overview

Conventional TAS models segment videos by learning and memorizing the fixed set of actions given at training, thus cannot adapt to novel tasks or actions. To address MMF-TAS, we propose a Prototype Graph Network (PGNet) that follows a *summarize-and-match* strategy. Instead of learning any specific action, it aims to learn the ability to summarize the information of actions in support videos and match it with query videos to infer their action labels. Hence, we do not require models to memorize particular actions and can segment videos of novel tasks.

Our framework is shown in Figure 1. To leverage multi-modalities, we employ a video and a textual encoder to obtain frame features $\mathbf{F}$ for query video, and $\hat{\mathbf{F}}_{k,v}$ for the v-th support video in task k. For action a in task k, we obtain the textual feature $\hat{\mathbf{E}}_{k,a}$. Our *Prototype Building Block* (PBB) in PGNet computes action prototypes from support videos (using either visual or textual modality). Each prototype summarizes the information for one action class in the tasks,

$$\mathbf{R}^v = \mathrm{PBB}^v(\{\hat{\mathbf{F}}_{k,v}, \hat{Y}_{k,v}\}, \mathbf{F}); \quad (1)$$

$$\mathbf{R}^t = \mathrm{PBB}^t(\{\hat{\mathbf{E}}_{k,a}\}, \mathbf{F}); \quad (2)$$

where $\mathbf{R}^v$, $\mathbf{R}^t$ denote the action prototypes obtained from visual or textual features. Query video feature $\mathbf{F}$ is included to produce tailored prototypes for query video and enhance subsequent matching [25].

With the prototypes, our *Matching Block* (MB) can effectively compare the similarity between the query video and action prototypes and infer action labels accordingly,

$$Y^v = \mathrm{MB}^v(\mathbf{R}^v, \mathbf{F}), \quad Y^t = \mathrm{MB}^t(\mathbf{R}^t, \mathbf{F}), \quad (3)$$

where Y^v, Y^t are the framewise action labels of the query video from visual or textual modalities.

To enable learning of the summarization and matching ability, we mimic the test setting at training time – the model input at training time is also a group of query and support videos, sampled from a set of *base* tasks available at training. At test time, we evaluate on *novel* as well as base tasks. There is no overlap between base and novel tasks.

Remark 1 *Notably, while existing multi-modal few-shot methods [19, 44, 68] typically fuse visual and textual features to create prototypes, we deliberately use separate* $\mathbf{R}^v$ *and* $\mathbf{R}^t$*. We show that the two modalities have inconsistent performance on base and novel tasks. In Figure 2, we compare the effect of using only* $\mathbf{R}^v$ *or* $\mathbf{R}^t$ *in Matching Block, or fusing them: i)* $\mathbf{R}^v$ *performs better on novel tasks than* $\mathbf{R}^t$*, as* $\mathbf{R}^v$ *and query video both belong to the visual modality and* $\mathbf{R}^t$ *is from a different modality. ii)* $\mathbf{R}^t$ *performs better on base tasks than* $\mathbf{R}^v$*, because it is computed from action names that are fixed across inputs, thus easy to learn at training.* $\mathbf{R}^v$ *is computed from support videos and varies across inputs. iii) When fusing their features, as model is trained on base tasks, it learns to over-rely on* $\mathbf{R}^t$ *and underutilize* $\mathbf{R}^v$*. Hence, its accuracy on novel task is close to that of using only* $\mathbf{R}^t$*, and worse than using only* $\mathbf{R}^v$*.*

Thus, we separate visual and textual prototypes, and fuse the predictions of two modalities in Matching Block, which combines their complementary information and allows penalizing over-reliance on one modality. The separation also enables flexible test scenarios that achieve lower annotation costs than the standard few-shot setting, including zero-shot and unlabeled/weakly-labeled few-shot (see Section 3.6).

3.3. Prototype Building Block

The goal of the Prototype Building Block is to compute action prototypes that summarize the information of action instances in support videos. For example, it should compare the action instances of one action class to capture its action pattern, or find the instances of the same video to capture action temporal dependencies. Therefore, it is crucial for models to understand these varied relations among action instances and aggregate different aspects of action information. Processing this information also requires different modeling capacity (e.g., measuring similarity among action instances vs comparing their temporal locations).

We introduce the *Action Relation Graph* that represents action instances and action prototypes as nodes and encodes

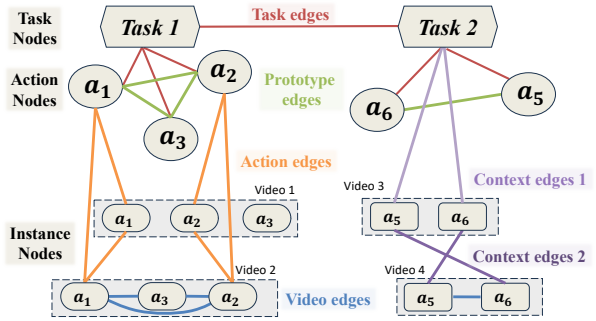

Figure 3. **Illustration of Action Relation Graph** *(only a subset of edges are drawn for readability).* Task, action and instance nodes learn task information, action prototypes and features of support action instances respectively, while relational edges encode their relations based on their node types and tasks, videos, action classes.

their relations as edges. With the graph, our *Dynamic Graph Transformer* learns specialized parameters for each relation to capture different action information. In the following, we first introduce the model details for visual modality, then the adjustment for textual modality.

3.3.1. Action Relation Graph

Nodes. As shown in Figure 3, to summarize the information of action instances into action prototypes, the graph contains Instance nodes that are created for each action instance in support videos to encode its frame features, and Action nodes, created for each unique action in a task to learn the associated prototype. To also summarize task information and infer the task label of query video, we create a Task node for each task.

To initialize the three types of nodes, we learn one embedding vector for each node type $(\boldsymbol{\rho}^T, \boldsymbol{\rho}^A, \boldsymbol{\rho}^I)$. Task and action nodes are directly initialized by their embeddings. Instance nodes are initialized by their frame features. Let $\hat{\mathbf{F}}_{k,v,i}$ denote the subset of $\hat{\mathbf{F}}_{k,v}$ that contains the features of the i-th action instance in $\hat{\mathcal{V}}_{k,v}$. The corresponding instance node is initialized as

$$\text{temporal-pooling}(\hat{\mathbf{F}}_{k,v,i}) + \boldsymbol{\rho}^I + \boldsymbol{\rho}^P_i, \tag{4}$$

where $\boldsymbol{\rho}^P_i$ is an absolute positional encoding to denote the action's ordering in the video.

Relational Edges. Edges capture the relations of the nodes they connect, such as if nodes belong to the same action, video, task or node type. Thus, they help identify action patterns, distinctions and dependencies and summarize information into task and action nodes. We propose the *relational edge type*: we build a densely-connected graph to preserve critical relations while assign edges with different types based on the relations they denote.

As illustrated in Figure 3, we define the Task edge type that connects a task node to action nodes in that task and to other task nodes, to learn task procedures and distinctions. Action edge connects action/instance nodes of the same action and same task to capture the motion pattern of each action. Prototype edge connects all action nodes of a task to learn the action distinctions. Video edge connects instance nodes of the same video to capture action ordering, hence the action dependencies critical for TAS. To have more informative connections, we also include two context edge types: Context edge 1 connects instance nodes to the task node; Context edge 2 connects instance nodes belonging to the same task but of different videos and action classes.

3.4. Dynamic Graph Transformer.

We aim to leverage the relational edges to capture different action information. Processing different edge types requires varied modal capacities, e.g., identifying action similarity with action edges and action distinctions with prototype edges. Thus, we propose a Dynamic Graph Transformer that learns specialized parameters for each edge type. As shown in Figure 4 (left), each transformer layer consists of dynamic graph attention (DGA), cross-attention, and fully connected layers,

$$\mathbf{N}' = \text{DGA}(\mathbf{N}), \tag{5}$$

$$\mathbf{N}'' = \text{fully-connected}(\text{cross-attention}(\mathbf{N}'; \mathbf{F})), \tag{6}$$

where $\mathbf{N}$ are the features of all nodes. DGA updates the nodes by computing self-attention among them. It represents the edge types as different adjacency matrices and learns specialized attention heads for them by constraining their attention maps with the matrices. Cross-attention refines nodes with query video features $\mathbf{F}$ to produce tailored prototypes.

DGA. Specifically, Let $\mathbf{M}_j \in \{0,1\}^{H\times H}$ denote the adjacency matrix of edge type e. H is the number of nodes. In DGA, we compute the attention $\boldsymbol{\Delta}$ of an attention head as

$$\boldsymbol{\Delta} = (\mathbf{W}^Q\mathbf{N})(\mathbf{W}^K\mathbf{N}) + \mu(\textstyle\sum_j \alpha_j \mathbf{M}_j); \tag{7}$$

where the first term is the attention logit and $\mathbf{W}^{Q/K}$ is the query/key projection weight. The second term is the attention mask, where α_j is the learnable *dynamic edge weight* to direct the focus of this head to certain edge types, $\sum_j \alpha_j = 1$. As α_j and $\mathbf{W}^{Q/K}$ are learned together at training, it allows the head to *specialize on different edge types*. It also removes the need to stipulate the number of attention heads per each edge type. The model can use α to assign heads to different edge types based on its need. Lastly, μ is a learned scaling factor to match $\mathbf{M}$ with the magnitude of the attention logit.

We use multiple transformer layers, and take the features of action nodes from the last layer as the visual action prototypes, $\mathbf{R}^v$. To obtain textual prototype $\mathbf{R}^t$, we build a similar prototype learning block except 1) Action Relation Graph contains no instance node, 2) action nodes are initialized with the textual feature of the actions, $\hat{\mathbf{E}}_{k,a} + \boldsymbol{\rho}^A$.

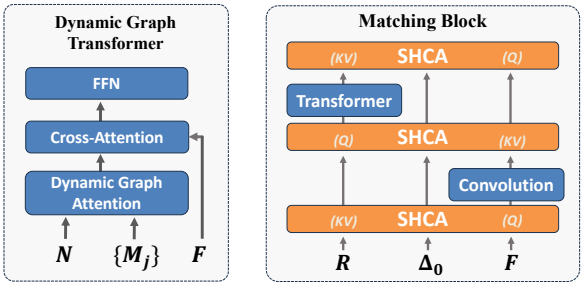

Figure 4. **Dynamic Graph Transformer and Matching Block**. SHCA standards for Single-Head Cross-Attention and *(Q)* and *(KV)* denote the input is used as attention query, or attention key and value.

3.5. Matching Block

Matching block infers action labels of the query video by comparing its frame features $\mathbf{F}$ with the action prototypes $\mathbf{R}$ (of either visual or textual modality). Since this block has the same structure for the two modalities, we drop the superscript (t, v) for simplicity.

As shown in Figure 4 (right), we use Single-Head Cross-Attention (SHCA) to compute the attention between $\mathbf{F}$ and $\mathbf{R}$. Therefore, the attention map reflects the similarity between $\mathbf{F}$ and $\mathbf{R}$, hence can serve as the action label prediction. We also extend SHCA to take *a reference attention map* as input that improves predictions and *enables the prediction fusion to combine modalities*, as we introduce next.

Following the multi-stage refinement technique in [40], we alternate between using $\mathbf{F}$ (or $\mathbf{R}$) as attention query in SHCA and $\mathbf{R}$ (or $\mathbf{F}$) as key and value. This process produces predictions and refines features based on the predictions. We also include temporal convolutions and transformers to further enhance features. We use the attention of the last layer as the final prediction.

SHCA. We define the l-th SHCA layer as

$$\mathbf{O}_l, \mathbf{\Delta}_l = \mathrm{SHCA}_l(\mathbf{Q}_l; \mathbf{K}_l; \mathbf{\Delta}_{l-1}), \tag{8}$$

where $\mathbf{Q}_l$ is the attention query and $\mathbf{K}_l$ is the key and value. $\mathbf{\Delta}_l$ is the attention map. Importantly, we also use the attention map from the previous layer $\mathbf{\Delta}_{l-1}$ as *reference* – we first compute the initial attention map $\hat{\mathbf{\Delta}}_l = (\mathbf{W}^Q\mathbf{Q}_l)(\mathbf{W}^K\mathbf{K}_l)$, then apply a function ψ to fuse attentions $\mathbf{\Delta}_l = \psi(\hat{\mathbf{\Delta}}_l, \mathbf{\Delta}_{l-1})$. Lastly, we use $\mathbf{\Delta}_l$ to obtain output feature $\mathbf{O}_l$. Here, we define ψ as weighted sum,

$$\mathbf{\Delta}_l = \hat{\mathbf{\Delta}}_l + \tau_l \mathbf{\Delta}_{l-1}, \tag{9}$$

where $\tau_l \in [0, 1]$ is a learnable *influence factor* to control the impact of $\mathbf{\Delta}_{l-1}$. This creates skip connections between the attentions and allows retaining useful information from earlier layers while integrating new matching results.

As the first SHCA layer has no previous layer, we set its reference attention $\mathbf{\Delta}_0$ as the initial attention $\hat{\mathbf{\Delta}}_1$ from *the other modality* (i.e., in visual modality, $\mathbf{\Delta}_0^v = \hat{\mathbf{\Delta}}_1^t$ and vice versa). This enables the *prediction fusion* where the prediction of one modality guides that of the other modality, as model predictions can effectively transfer knowledge across models [15, 16, 47]. The improved prediction also enhances the output feature of SHCA, consequently benefiting all subsequent layers. Meanwhile, we impose a penalty loss on τ_1 to prevent over-reliance on the other modality.

	Zero-Shot	Few-Shot (no-label)	Few-Shot (weak-label)	Few-Shot (full-label)
Textual Action Names	✓	✓	✓	✓
Support Videos	✗	✓	✓	✓
Support Video Labels	✗	✗	partial	✓

Table 1. Model Inputs of Different Test Settings.

3.6. Training and Inference

Training. We devise four losses to supervise the PGNet of both visual and textual modality: 1) To infer the task of the query video, we compute task prediction $\boldsymbol{P}_{\text{task}}$ by applying a linear classifier to the task nodes, as they are computed with access to both query and support videos features. We impose cross-entropy loss on $\boldsymbol{P}_{\text{task}}$. 2) To learn action labels, we treat the attention maps $\{\mathbf{\Delta}_l\}_{l=1}^3$ as prediction logits and apply cross-entropy loss on them. 3) We also apply on them the smoothing loss from prior TAS method [9] to ensure temporally smooth predictions. 4) We impose an L2 loss on τ_1 to prevent over-reliance on the other modality.

Inference. At test time, we use $\mathbf{\Delta}_3$ of visual modality to predict for novel tasks and that of textual modality for base tasks[1]. Specifically, we first use $\boldsymbol{P}_{\text{task}}$ of the corresponding modality to estimate the task of query video, then mask out the entries in $\mathbf{\Delta}_3$ that correspond to similarity with action prototypes of the other tasks. Action labels are predicted by applying argmax on the masked attention map.

Generalized Inference. While PGNet is trained with standard few-shot setting, it can handle more *difficult, data-sparse scenarios* at inference time. As shown in Table 1, it includes: 1) **Zero-shot setting**: only knowing the actions in tasks but no support videos. Thanks to the separation of visual and textual PGNets, we can use textual PGNet to obtain prediction with only action names. (In its first SHCA, we set $\mathbf{\Delta}_0 = \mathbf{0}$ as there is no prediction from visual modality.) 2) **Few-shot (no-label)**: having support videos but no action labels. We predict action labels of the support videos with textual PGNet as in zero-shot setting. Then we use the support videos and predicted labels to run our model as in the standard few-shot setting. 3) **Few-shot (weak-label)**: support videos are labeled with only one frame per action instance in a video[2]. We predict action labels for support videos in zero-shot setting, use method from [5] to refine

[1] This can be easily inferred by comparing the actions and their ordering in support videos with training data.

[2] It corresponds to annotate just *2% and 3% of the frames* in CrossTask and COIN, respectively, and significantly reduces annotation cost.

predictions with the given sparse action labels and then run our model in the standard few-shot setting.

4. Experiments

Datasets. We evaluate on two procedural video datasets, CrossTask [79] and COIN [63]. Compared to other TAS datasets [10, 21, 27], both datasets contain a large number of procedural tasks, suitable for few-shot evaluation. CrossTask has 18 procedural tasks, 2750 videos and 106 actions. We choose 12 tasks as base tasks and 6 as novel tasks. COIN has more diverse tasks than CrossTask, with 180 tasks, 10177 videos and 750 actions. We choose 120 tasks as base and 60 as novel. For both datasets, we classify frames not relevant to any action into a *background* class.

Metrics. Following prior TAS works [3, 9, 32, 40, 58, 70], 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*). We follow conventional methods [9, 40] to exclude background frames in evaluation.

Implementation. We use 4 graph transformer layers in visual PGNet and 2 layers in textual PGNet. Each dynamic graph attention has 9 attention heads. For Matching Block, temporal transformer has 3 self-attention layers and temporal convolution has 8 layers [9]. We use video and text encoders from [77]. We provide more details in the supplementary materials.

4.1. Comparison with the State-of-the-Art

In Table 2, we extensively evaluate our models on CrossTask and COIN datasets with four different settings – Zero-Shot (3way-3shot), and Multi-Modal Few-Shot (3way-3shot and 5way-3shot). For each setting, we report results on novel tasks.

Competitors. We consider two sets of methods as our baselines – VLM and Few-Shot methods. **VLM** include recent contrastive VLMs, *ProcedureVRL* [77] and *LanguageBind* [78], and generative VLM, *Chat-UniVi* [18]. We finetune contrastive VLMs on base tasks of our datasets. As generative VLM is not designed for classification task, we do not finetune it but build a pipeline that first uses it to describe video clips, then uses a sentence embedding model [49] to compute the similarity between clip captions and action class names to obtain predictions. We use them as zero-shot baselines. Next, **Few-Shot** baselines include *LinearProbe* that first uses temporal convolution [32] to improve query/support video features, then linear layers to align the query video features with the video/text features of support videos, and *MUPPET* [44] that addresses MMF Action Localization and is adapted to MMF-TAS by us.

Zero-Shot. In the top-section of Table 2, we compare with recent VLMs for zero-shot setting. We outperform all VLMs, showing PGNet can generalize to novel tasks by only knowing the names of their actions. Action names can be obtained from knowledge bases (e.g., WikiHow) *without the need of video collection and annotation*. VLMs have high computation cost and need to split long procedural videos into clips and process each clip separately, thus miss the temporal action dependencies. It also causes the over-segmentation issue, where models predict many short erroneous segments. This results in an inflated Acc and low Edit and F1 scores [9], especially for LanguageBind on COIN. PGNet shows higher accuracy on CrossTask than on COIN, as novel tasks in COIN are more diverse and less similar to the base tasks. It represents a difficult scenario where textual common knowledge is insufficient to describe novel tasks and visual demonstrations are required.

Few-Shot. In the second to fourth sections of Table 2, we test models on three multi-modal few-shot settings, each having more novel tasks to adapt to. Our PGNet significantly outperforms all baselines, surpassing in F1@50 by ***8.8%*** to ***9.3%*** on CrossTask and ***6.9%*** to ***7.6%*** on COIN. As the number of novel tasks increases, our improvement over baselines enlarges, underscoring that PGNet can understand and distinguish complex task procedures. LinearProbe has low performance due to its inability to capture the relations among action instances in support videos. It also has the over-segmentation issue as VLMs. MUPPET employs advanced temporal modeling and localization techniques. However, it models each action individually and ignores action dependencies. Thus, it often predicts wrong action ordering, as indicated by its low Edit scores. It also fuses visual/textual modalities via features, therefore over-relies on textual features, as discussed in Remark 1. Our PGNet successfully leverages the advantages of both modalities.

Generalized Few-Shot. We also show extensive results for *unlabeled, weakly-labeled and standard (fully-labeled) few-shot settings*. The unlabeled setting only requires support videos, which can be collected online via automatic scripts. The weakly-labeled setting only needs one labeled frame per action instance and has a substantially lower annotation cost. On CrossTask, PGNet performs competitively in unlabeled and weakly-labeled settings, comparable to the fully-labeled setting. On COIN, simply providing support videos without label improves F1@50 by 1.6% from zero-shot setting to unlabeled few-shot settting. Providing weak labels further boosts F1@50 by 3.6%, largely bridging the gap with fully-labeled setting. This shows our PGNet is a versatile framework – *one model is applicable under various data collection budgets*. It also shows the necessity of visual modality in adapting to rare or unfamiliar tasks, where action names fall short of fully describing the task.

Cross-Dataset Adaptation. We further challenge models by training on base tasks of COIN and testing on novel tasks of CrossTask in Table 3. Interestingly, we found PGNet trained on COIN benefits from the larger and more diverse

	CrossTask					COIN				
	Edit	F1@10	F1@25	F1@50	Acc	Edit	F1@10	F1@25	F1@50	Acc
Zero-Shot 3way-3shot										
ProceduralVRL[77]	13.7	13.0	9.5	5.8	30.8	12.6	14.0	10.8	5.3	34.7
LanguageBind[78]	11.4	10.4	7.8	4.3	20.1	12.1	11.3	8.9	4.9	35.6
Chat-UniVi[18]	9.5	7.9	6.2	3.8	9.2	9.4	6.9	5.4	2.3	6.3
PGNet *(zero-shot)*	**36.1**	**32.2**	**26.4**	**15.4**	**36.4**	**31.8**	**24.0**	**17.4**	**8.6**	**36.6**
	+22.4	*+19.2*	*+16.9*	*+9.6*	*+5.6*	*+19.2*	*+10.0*	*+6.6*	*+3.3*	*+1.0*
Few-Shot 3way-3shot										
Linear Probe	15.4	14.8	11.0	6.0	31.4	12.6	18.5	14.2	7.5	34.6
MUPPET[44]	18.1	19.5	15.8	9.5	13.1	13.1	19.9	15.4	8.1	15.9
PGNet *(no-label)*	35.8	33.4	27.5	16.9	35.9	37.2	27.8	19.9	10.2	29.6
PGNet *(weak-label)*	36.5	35.4	29.1	17.3	37.6	46.1	35.3	26.1	13.8	42.4
PGNet *(full-label)*	**37.7**	**36.6**	**30.1**	**18.3**	**39.3**	**46.6**	**36.7**	**27.6**	**15.0**	**43.8**
	+19.6	*+17.1*	*+14.3*	*+8.8*	*+7.9*	*+33.5*	*+16.8*	*+12.2*	*+6.9*	*+9.2*
Few-Shot 5way-3shot										
Linear Probe	13.4	12.7	9.8	5.4	27.2	13.0	15.6	11.5	6.3	33.1
MUPPET [44]	15.8	17.3	13.7	8.6	12.5	11.5	19.2	14.5	7.6	14.2
PGNet *(no-label)*	33.2	31.9	26.6	16.4	33.4	34.4	27.4	19.9	9.8	29.6
PGNet *(weak-label)*	34.2	33.3	28.2	17.4	35.8	44.6	34.9	26.1	13.1	41.9
PGNet *(full-label)*	**34.7**	**33.7**	**28.2**	**17.5**	**35.7**	**45.4**	**36.2**	**27.5**	**14.3**	**43.3**
	+18.9	*+16.4*	*+14.5*	*+8.9*	*+8.5*	*+32.4*	*+17.0*	*+13.0*	*+6.7*	*+10.2*

Table 2. **Performance on CrossTask and COIN datasets**. We test with three settings. In each setting, we show in blue the PGNet's improvement on novel tasks over the best baseline.

	Train	Test	Edit	F1@10	F1@25	F1@50	Acc
Linear Probe	COIN	CrossTask	3.2	2.5	1.9	1.0	9.7
MUPPET [44]			19.4	10.9	9.2	5.2	17.2
PGNet			**41.5**	**33.9**	25.1	12.4	33.9
PGNet	CrossTask	CrossTask	34.7	33.7	**28.2**	**17.5**	**35.7**

Table 3. **Cross-dataset Adaptation** with few-shot 5way-3shot setting.

training set. It better captures the procedures of novel tasks than PGNet trained on CrossTask, achieving *higher Edit*. Its lower F1@50 indicates less accurate action boundaries, which is mainly caused by annotation difference between the two datasets. Actions in CrossTask are typically labeled with tighter action boundaries than in COIN, which also creates more background frames (70% vs 50%) – a trait created by annotations rather than task procedures. The PGNet trained on COIN still yields better action localization, reflected by its higher F1@10, underscoring our method can scale to larger training data and learn better generalization.

In this section, we test our key model designs on COIN dataset with few-shot 3way-3shot setting. For better readability, we only show the results for novel tasks.

Effect of Action Relation Graph. In Table 4, we first test removing the types of relational edges (row 1), such that all edges are treated uniformly. It reduces F1@50 by 6.1% as the model cannot identify the relations among action instances to capture corresponding action information. Then, we test using a sparse Action Relation Graph instead of the proposed dense one (row 2), by building a hierarchical graph with only task, action and video edges. It drops F1@50 by 4.7%, as the sparse graph limits the message-passing among nodes and can miss edges of critial node relations. Manually designing the edges for sparse graphs is also restrictive and not scalable. Lastly, we remove task nodes (row 3) and only learn on action-level information. It reduces F1@50 by 6.4%, as it loses the task-level information and cannot infer task labels.

			Edit	F1@10	F1@25	F1@50	Acc
1	ARG	no-edge-type	31.0	23.3	17.3	8.9	23.6
2		sparse-graph	34.1	26.4	19.7	10.3	29.0
3		no-task-node	29.2	22.7	16.6	8.6	26.5
4	DGT	fixed-α	44.5	33.6	24.9	12.1	39.7
5		same-α	42.9	30.9	22.9	11.9	37.7
6	PGNet		**46.6**	**36.7**	**27.6**	**15.0**	**43.8**

Table 4. **Ablation for Prototype Building Block** to study the designs for Action Relation Graph (ARG) and Dynamic Graph Transformer (DGT).

Effect of Dynamic Graph Transformer. In row 3 of Table 4, we use a preset dynamic edge weight α to assign a fixed number of attention heads per edge type. It is equivalent to methods [64, 72] that learn separate layers for each edge type and overlook that some edges may demand more complex reasoning ability than others. It decreases F1@50 by 2.9%. Next, we use shared edge weight α for all attention heads (row 4), equivalent to methods [6] that convert edge types into edge weights and learn one set of parameters. It reduces F1@50 by 3.6% as all attention heads focus

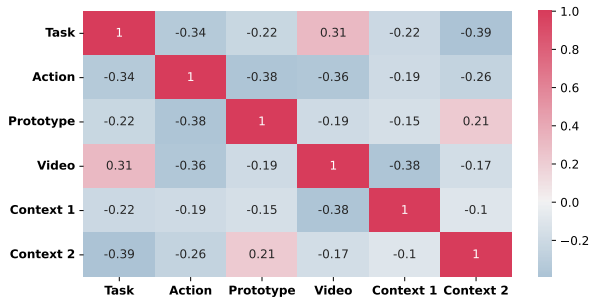

Figure 5. **Correlation between Dynamic Edge Weights α.**

			Edit	F1@10	F1@25	F1@50	Acc
1	PF	no-fusion	41.7	33.4	24.7	12.7	36.5
2		late-fusion	42.1	34.3	25.8	13.5	37.0
3		no-penalty	39.1	29.7	22.0	11.1	34.4
4	MSR	1^{st}-stage	35.9	27.8	20.9	11.4	42.9
5		2^{nd}-stage	42.5	33.8	25.4	13.9	43.7
6	PGNet		**46.6**	**36.7**	**27.6**	**15.0**	**43.8**

Table 5. **Ablation for Matching Block** to study the designs for Prediction Fusion (PF) and Multi-Stage Refinement (MSR).

on similar edges and cannot specialize to different ones.

4.2. Ablation Study

Effect of Matching Block. We study our Matching Block in Table 5. Recall that in prediction fusion, we use predictions of one modality to guide the other. We first test disabling prediction fusion (row 1). Without it, the model cannot leverage the information from both modalities, lowering F1@50 by 2.3%. Next, we perform prediction fusion in the last SHCA layer instead of the first layer (row 2). This prevents using the predictions of two modalities to refine features in the early layers, dropping F1@50 by 1.5%. Lastly, we test removing the L2 loss on the influence factor τ_1 (row 3). It cannot prevent models from over-relying on textual predictions, reducing F1@50 by 3.9%. Next, we also study the effect of multi-stage refinement. We measure the prediction accuracy at each stage, i.e., predicting with attention maps from the first, second and third SHCA (row 4-6). F1@50 increases steadily at each stage, rising from 11.4% to 15.0% from the first to the third stage. We found adding more stages yields diminishing improvement.

4.3. Qualitative Results

In Figure 5, we visualize the correlation of dynamic edge weights α among different edge types. α controls the focus of attention heads to edge types. Thus, the negative correlations among most edge types indicate each head specializes to one edge type instead of attending to multiple types. The positive correlations then show models share attention heads for edge types that encode similar information. Task edge and video edge has a positive correlation, as task edge

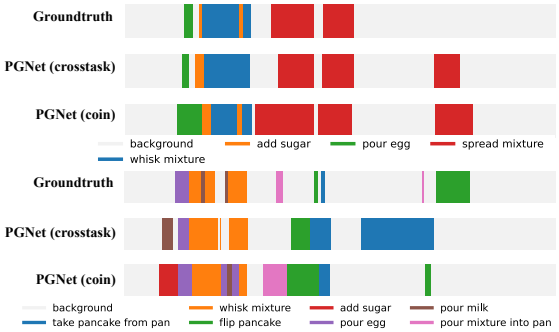

Figure 6. **Visualization of Model Predictions on CrossTask.**

captures task information and video edge captures temporal action dependencies, i.e., task procedure. Similarly for prototype edge and context edge, both edges connect action instances of different action classes, hence reflects action distinctions. It validates *the model understands the relation represented by each edge type, hence can leverage them to summarize useful action information.*

In Figure 6, we visualize the predictions of our PGNets on CrossTask videos. PGNet (crosstask) and PGNet (coin) denote models trained on CrossTask and COIN, respectively. The top video shows a case where PGNet (coin) predicts better action locations and orderings overall. Yet it is less accurate on action boundaries, since actions in COIN are typically annotated with looser action boundaries. The bottom video represents a challenging case that has many short and repeated actions. Both models still identify and locate most actions successfully, showing the reliability of our PGNet framework.

5. Conclusions

We introduced the new Multi-Modal Few-shot Temporal Action Segmentation (MMF-TAS) problem to adapt to novel tasks with minimal annotation cost and proposed the first MMF-TAS framework. Our Prototype Graph Network (PGNet) contains a Prototype Building Block to summarize critical action information using action prototypes, and a Matching Block to infer accurate action labels and fuse visual and textual modalities. By extensive experiments on COIN and CrossTask datasets, we showed our model generalizes well to novel tasks under various zero-shot and few-shot settings and substantially outperforms the prior works.

Acknowledgement

This work was funded, in part, by ARPA-H (1AY2AX000062), DARPA PTG (HR00112220001), NSF (IIS-2115110), ONR (N000142512287) and ARO (W911NF2110276). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either either expressed or implied, of the US Government.

References

[1] Hyemin Ahn and Dongheui Lee. Refining action segmentation with hierarchical video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16302–16310, 2021. 2

[2] J. B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[3] Nadine Behrmann, S. Alireza Golestaneh, Zico Kolter, Juergen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *ECCV*, 2022. 1, 2, 6

[4] G. Donahue and E. Elhamifar. Learning to predict activity progress by self-supervised video alignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[5] Dazhao Du, Enhan Li, Lingyu Si, Fanjiang Xu, and Fuchun Sun. Timestamp-supervised action segmentation from the perspective of clustering. In *IJCAI*, 2023. 5

[6] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020. 7

[7] E. Elhamifar and D. Huynh. Self-supervised multi-task procedure learning from instructional videos. *European Conference on Computer Vision*, 2020. 2

[8] E. Elhamifar and Z. Naing. Unsupervised procedure learning via joint dynamic summarization. *International Conference on Computer Vision*, 2019. 2

[9] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019. 2, 5, 6

[10] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 6

[11] Mohsen Fayyaz and Jurgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[12] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration. *Annual Meeting of the Association for Computational Linguistics*, 2020. 2

[13] Ziliang Gan, Lei Jin, Lei Nie, Zheng Wang, Li Zhou, Liang Li, Zhecan Wang, Jianshu Li, Junliang Xing, and Jian Zhao. Asquery: A query-based model for action segmentation. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 2024. 2

[14] Dayoung Gong, Suha Kwak, and Minsu Cho. Actfusion: a unified diffusion model for action segmentation and anticipation. *Advances in Neural Information Processing Systems*, 37:89913–89942, 2024. 2

[15] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *Int. J. Comput. Vision*, 2021. 2, 5

[16] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, 2015. 2, 5

[17] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. In *ICCV Workshop on Scene Graph Representation and Learning*, 2019. 2

[18] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023. 2, 6, 7

[19] Chen Ju, Zeqian Li, Peisen Zhao, Ya Zhang, Xiaopeng Zhang, Qi Tian, Yanfeng Wang, and Weidi Xie. Multi-modal prompting for low-shot temporal action localization, 2023. 2, 3

[20] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. Few-shot object detection via feature reweighting. *IEEE International Conference on Computer Vision*, 2019. 2

[21] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 6

[22] H. Kuehne, J. Gall, and T. Serre. An end-to-end generative framework for video segmentation and recognition. *IEEE Winter Conference on Applications of Computer Vision*, 2016. 2

[23] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[24] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[25] Juntae Lee, Mihir Jain, and Sungrack Yun. Few-shot common action localization via cross-attentional fusion of context and temporal dynamics. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 3

[26] S. Lee and E. Elhamifar. Error recognition in procedural videos using generalized task graph. *International Conference on Computer Vision*, 2025. 1, 2

[27] S. Lee, Z. Lu, Z. Zhang, M. Hoai, and E. Elhamifar. Error detection in egocentric procedural task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 6

[28] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[29] J. Li and S. Todorovic. Anchor-constrained viterbi for set-supervised action segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[30] J. Li, P. Lei, and S. Todorovic. Weakly supervised energy-based learning for action segmentation. *International Conference on Computer Vision*, 2019. 2

[31] M. Li, L. Chen, Y. Duarr, Z. Hu, J. Feng, J. Zhou, and J. Lu. Bridge-prompt: Towards ordinal action understanding in instructional videos. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[32] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 2, 6

[33] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[34] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2

[35] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models, 2023. 2

[36] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. *International Conference on Computer Vision*, 2023. 1, 2, 3

[37] Z. Lu and E. Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. *International Conference on Computer Vision*, 2021. 2

[38] Z. Lu and E. Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[39] Zijia Lu and Ehsan Elhamifar. Bit: Bi-level temporal modeling for efficient supervised action segmentation. 2023. 2

[40] Z. Lu and E. Elhamifar. Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 3, 5, 6

[41] Z. Lu, A. Iftekhar, G. Mittal, T. Meng, X. Wang, C. Zhao, R. Kukkala, E. Elhamifar, and M. Chen. Decafnet: Delegate and conquer for efficient temporal grounding in long videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

[42] Himangi Mittal, Nakul Agarwal, Shao-Yuan Lo, and Kwonjoon Lee. Can't make an omelette without breaking some eggs: Plausible action anticipation using large video-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18580–18590, 2024. 2

[43] Sauradip Nag, Xiatian Zhu, and Tao Xiang. Few-shot temporal action localization with query adaptive transformer, 2021. 2

[44] Sauradip Nag, Mengmeng Xu, Xiatian Zhu, Juan-Manuel Perez-Rua, Bernard Ghanem, Yi-Zhe Song, and Tao Xiang. Multi-modal few-shot temporal action detection via vision-language meta-adaptation. *arXiv preprint arXiv:2211.14905*, 2022. 2, 3, 6, 7

[45] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization, 2021. 2

[46] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[47] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. *International Conference on Machine learning*, 2019. 5

[48] Rahul Rahaman, Dipika Singhania, Alexandre Thiery, and Angela Yao. A generalized and robust framework for timestamp supervision in temporal action segmentation. In *Computer Vision–ECCV 2022: 17th European Conference*, 2022. 2

[49] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 6

[50] A. Richard, H. Kuehne, and J. Gall. Action sets: Weakly supervised action segmentation without ordering constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[51] A. Richard, H. Kuehne, A. Iqbal, and J. Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[52] Y. Shen and E. Elhamifar. Semi-weakly-supervised learning of complex actions from instructional task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[53] Y. Shen and E. Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[54] Y. Shen and E. Elhamifar. Understanding multi-task activities from single-task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2

[55] Y. Shen, L. Wang, and E. Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[56] Yuhan Shen, Linjie Yang, Longyin Wen, Haichao Yu, Ehsan Elhamifar, and Heng Wang. Exploring the Role of Audio in Video Captioning . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024.

[57] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[58] Dipika Singhania, Rahul Rahaman, and Angela Yao. Coarse to fine multi-resolution temporal convolutional network. *CoRR*, abs/2105.10859, 2021. 1, 2, 6

[59] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *Neural Information Processing Systems*, 2017. 2

[60] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast Weakly Supervised Action Segmentation Using Mutual Consistency. *PAMI*, 2021. 2

[61] Yuhao Su and Ehsan Elhamifar. Two-stage active learning for efficient temporal action segmentation. pages 161–183, 2024. 2

[62] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[63] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[64] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. 7

[65] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Neural Information Processing Systems*, 2016. 2

[66] X. Wang, T. E. Huang, T. Darrell, J. Gonzalez, and F. Yu. Frustratingly simple few-shot object detection. *International Conference on Machine learning*, 2020. 2

[67] Tingting Xie, Christos Tzelepis, Fan Fu, and Ioannis Patras. Few-shot action localization without knowing boundaries. *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021. 2

[68] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13623–13633, 2023. 2, 3

[69] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 2018. 2

[70] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *The British Machine Vision Conference (BMVC)*, 2021. 6

[71] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28202–28211, 2024. 2

[72] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019. 7

[73] P. Zameni, Y. Shen, and E. Elhamifar. Moscato: Predicting multiple object state change through actions. *International Conference on Computer Vision*, 2025. 1, 2

[74] H Zhang, L Zhang, X Qi, H Li, PHS Torr, and P Koniusz. Few-shot action recognition with permutation-invariant attention. In *Proceedings of the European Conference on Computer Vision (ECCV 2020)*. Springer, 2020. 2

[75] Junbin Zhang, Pei-Hsuan Tsai, and Meng-Hsun Tsai. Semantic2graph: Graph-based multi-modal feature fusion for action segmentation in videos, 2022. 2

[76] Qing Zhong, Guodong Ding, and Angela Yao. Onlinetas: An online baseline for temporal action segmentation. In *Advances in Neural Information Processing Systems*, 2024. 1

[77] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835, 2023. 2, 6, 7

[78] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, Wang HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment, 2023. 2, 6, 7

[79] D. Zhukov, J. B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic. Cross-task weakly supervised learning from instructional videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6