VidParse: Online Parsing of Egocentric Procedures Like a Pro

Anubhav Gupta, Archit Kambhamettu, Vatsal Agarwal, Pulkit Kumar, and Abhinav Shrivastava

University of Maryland, College Park, USA
{anubhav, architk, vatsalag, pulkit, abhinav2}@umd.edu

Abstract. Translating continuous, noisy egocentric video streams into discrete, temporally ordered action steps is fraught with visual challenges. Heavy ego-motion, transient occlusions, and the high intra-class variability of unscripted human-object interactions cause standard frame-level online temporal models to struggle, often resulting in severe over-segmentation and structural collapse. To bridge the gap between unstable low-level perception and high-level procedural logic, we present *VidParse*, a online, training-free framework that treats activity understanding as a graph-constrained inference problem. Rather than relying on learned temporal filters, we dynamically identify semantic transitions using a temporal similarity matrix over manipulation-anchored features, which are extracted from frozen foundation models to prioritize foreground hand-object interactions. A beam search decoder then leverages an induced procedural task graph to explicitly enforce valid action transitions and prune impossible trajectories. By anchoring robust visual segments to hard procedural constraints, our approach preserves long-range state transitions and achieves up to a 10x improvement in complex multi-step parsing accuracy over strong online baselines, all without requiring a single gradient update.

Keywords: Training-Free · Online Action Segmentation · Video Parsing
Project Page & Code: https://learn2phoenix.github.io/VidParse

1 Introduction

Egocentric procedure parsing aims to convert a streaming first-person video into an ordered sequence of action steps as a person performs a task. In many real-world applications, this reasoning must occur online as the video unfolds. Unlike offline temporal segmentation, which can rely on full video context to resolve ambiguities, the online setting must infer the current step using only past observations. Maintaining a coherent estimate of procedural state over long sequences is therefore the central challenge.

Most existing approaches address this problem by learning temporal models that capture action transitions from annotated training data. Progress-aware

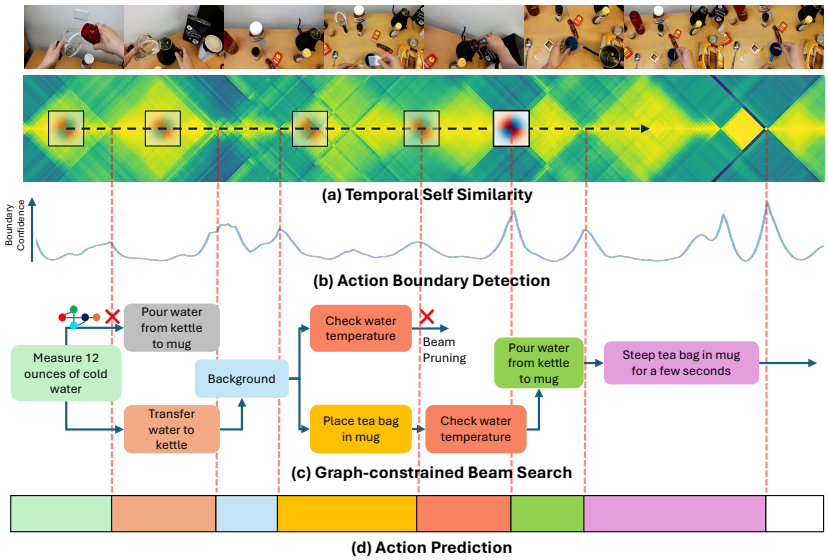


Fig. 1: VidParse: (a) *Object-Centric Self-Similarity*: We compute a Temporal Similarity Matrix (TSM) strip from frozen DINOv2 features. A Gaussian-tapered checkerboard kernel (red/blue overlay) slides along the diagonal to detect semantic transitions (b) *Training-Free Boundary Detection*: peaks indicate likely action boundaries. (c) *Graph-Constrained Beam Search*: These boundaries trigger a beam search that assigns action labels (d) *Final Prediction*: The result is a coherent, causally valid segmentation.

methods explicitly model task completion through structured state representations, while sequence models learn transition dynamics between actions using large amounts of labeled data. Although effective in controlled settings, these approaches depend heavily on training data and learned temporal dynamics, making them sensitive to domain shifts and difficult to deploy in settings where annotated data or retraining is impractical.

Rather than learning temporal models, we formulate egocentric activity understanding as a structured inference problem governed by procedural constraints. Many everyday tasks follow consistent structural dependencies: certain actions must precede others, while others cannot occur simultaneously. Exploiting these dependencies allows action sequences to be inferred even when local visual evidence is ambiguous.

Inspired by this, we introduce **VidParse**, a *training-free structured parsing framework* for online egocentric procedure understanding. We define "training-free" to indicate that our method requires no task-specific training or fine-tuning on the downstream procedure parsing dataset; instead, it relies entirely on frozen pretrained components (e.g., DINOv2 and hand-object detectors). Our approach combines three complementary components as shown in Fig. 1. First, we construct interaction-focused visual representations using Manipulation-Anchored Features (MAFs), which combine frozen DINO descriptors with hand-object de-

tections to emphasize regions where interactions occur. Second, we detect action boundaries online by computing a temporal similarity matrix over these features and applying a classical checkerboard kernel, enabling segmentation without learned temporal filters or boundary predictors. Finally, we perform structured decoding over an induced procedural task graph using graph-constrained beam search, selecting the action sequence that best satisfies both visual evidence and procedural constraints while maintaining causal inference.

We evaluate our approach on two challenging egocentric procedure parsing benchmarks, GTEA and EgoPER, which exhibit substantially different task structures and execution variability. Despite requiring no training or fine-tuning, our framework performs competitively with strong online baselines while achieving improved temporal consistency.

Beyond standard segmentation metrics, we introduce an N-step transition evaluation that measures long-range procedural consistency by analyzing multi-step action dependencies, enabling evaluation of structured procedural predictions. Our method maintains coherent multi-step trajectories more reliably than prior online approaches, demonstrating stronger preservation of global procedural structure. In summary, our contributions are as follows:

- **Training-free online procedure parsing.** We introduce **VidParse**, a framework for causal egocentric procedure parsing that operates without learning temporal models or fine-tuning foundation networks.
- **Structured procedural inference.** We formulate online procedure parsing as structured inference over an induced task graph and perform graph-constrained beam search to enforce valid action transitions.
- **Long-range procedural evaluation.** We introduce an N-step transition metric that measures the consistency of multi-step action dependencies beyond standard segmentation metrics.
- **Manipulation-anchored action segmentation.** We introduce Manipulation-Anchored Features (MAFs) and detect semantic action boundaries using a checkerboard kernel applied to an online temporal similarity matrix.

2 Related Work

Our work sits at the intersection of online temporal action segmentation, structure-aware (grammar/graph) procedural parsing, and similarity-based boundary detection. We briefly review each line of work and highlight how our method combines frozen foundation model features with non-parametric inference.

2.1 Temporal Action Segmentation

Temporal Action Segmentation (TAS) has evolved from offline models that utilize full video context (e.g., MS-TCN [9], ASFormer [41]) to causal Online Action Segmentation (OAS) [43] required for real-time egocentric applications. While the field initially focused on third-person videos, with datasets like 50 Salads [38], Breakfast [23], Kinetics [22], Something-Something [13], Cross-Task [44],

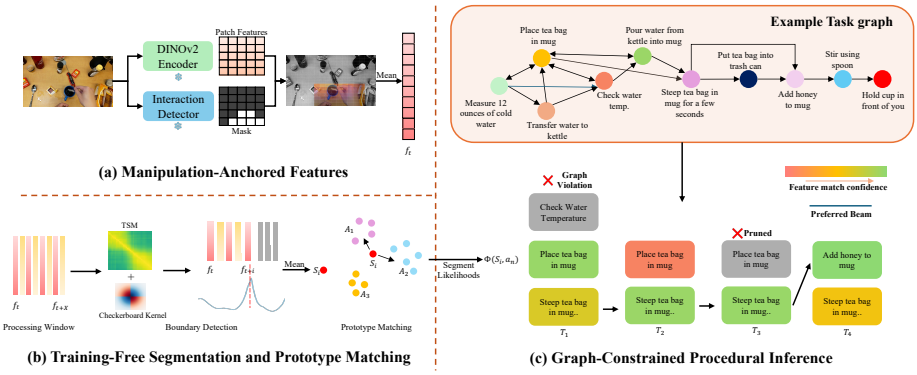


Fig. 2: VidParse Framework: (a) *Manipulation-Anchored Features*: We compute spatial features from a frozen DINOv2 backbone and apply a binary mask derived from a Hand-Object Detector (HOD)(b) *Training-Free Segmentation and Prototype Matching*: A sliding-window Temporal Similarity Matrix (TSM) is convolved with a Gaussian-tapered checkerboard kernel to detect semantic boundaries online. The resulting segments (S_i) are scored against pre-computed Action Centroids (c) *Graph-constrained Procedural Inference*: We infer the optimal sequence using a beam search that strictly enforces transitions defined in the induced Task Graph. Transitions that violate the procedural structure (e.g., pruned path at T_1) are assigned infinite cost and removed, ensuring causally valid segmentation.

COIN [39], recent focus has shifted to the egocentric perspective. With datasets like EPIC-Kitchens [6], Ego4D [14], CaptainCook4D [30], EgoPER [25], and HD-EPIC [32], the community is now addressing extreme ego-motion, occlusion, and long-tail distributions. Egocentric tasks like cooking and assembly often require real-time reasoning, necessitating Online Action Segmentation (OAS), where models must reason causally using only past frames. Without future context, online frame-level classifiers often suffer from severe over-segmentation. Recent mitigations include adaptive memory banks (OnlineTAS [43], MATR [37], sub-quadratic state-space models (Mamba-OTR [3]), and test-time vision-language adaptations (T3AL [27], FreeZAD [15]). While Large Multimodal Models offer broad generalization, their computational cost prohibits real-time dense segmentation. Our work bridges this gap by extracting semantic robustness directly from frozen foundation models within a lightweight, training-free framework, bypassing complex temporal optimization entirely.

2.2 Similarity-Based Boundary Detection

Generic Event Boundary Detection (GEBD) identifies semantic transitions without predefined taxonomies. Recent advancements replace classical techniques with predictive cognitive models (ESTimator [20]), dynamic multi-exit architectures (DyBDet [42]), and denoising diffusion processes (DiffGEBD [18]). Despite eliminating dense frame-level labels, unsupervised (UBoCo [21], OTAS [26])

and weakly-supervised (ATBA [40]) methods still require substantial feature or decoder optimization. We demonstrate that the inherent semantic stability of frozen foundation models like DINOv2 [29] revitalizes classical signal processing. By applying a Gaussian-tapered checkerboard kernel [5, 11] directly to object-centric patch embeddings, we achieve high-fidelity boundary emission. To our knowledge, this is the first to seamlessly integrate training-free GEBD into a graph-constrained action parsing pipeline.

2.3 Grammar-Based and Structural Parsing

To handle minutes-long procedural dependencies, recent works increasingly inject structural priors via neuro-symbolic reasoning. A foundational work in this domain is VideoGraph [17], which was among the first to model minutes-long human activities using a learned graph structure. Unlike purely sequential models, VideoGraph constructs a soft, undirected graph of latent concepts (unit-actions), demonstrating that modeling the structural relationships between atomic actions is essential for understanding long-form videos. Building on this structural intuition, subsequent works model activities using induced grammars [12], learn soft task graphs for progress estimation [36], or leverage large language models to extract logic for spatio-temporal alignment (LASER [16], PHGC [19], KML [28]).

Similarly, GTG2Vid [24] aligns videos with generalized task graphs for error detection. Other recent works explore temporal correspondence to segment procedural steps (e.g., [1, 2, 4, 31]); however, these are predominantly offline procedure learning and alignment methods. Our setting differs fundamentally as we focus on online, causal parsing from streaming video. Furthermore, while most prior works employ graphs as soft regularizers during training, we integrate the induced task graph as a strict, hard constraint during beam search. By assigning infinite costs to invalid procedural transitions, our decoder explicitly prevents the structural collapse endemic to local predictors.

Summary and Positioning: Prior work typically focuses on (i) online temporal segmentation with learned temporal models, (ii) structural priors via soft graph or grammar regularization, or (iii) boundary detection through trained or optimized objectives. Our framework combines these threads by using a classical, training-free boundary detector to emit action segments, and then enforcing procedure as a hard constraint during inference.

3 Method

We propose VidParse, a training-free framework for online egocentric procedure parsing. Given a first-person video, the goal is to convert it into a sequence of discrete action steps while respecting the procedural structure of the task.

VidParse proceeds in four stages. First, we extract interaction-focused visual representations called Manipulation-Anchored Features (MAFs) that isolate hand-object interactions using frozen foundation models. Second, we detect action boundaries online using a training-free temporal similarity detector, which

identifies transitions between manipulation phases. Third, we assign action labels to segments through prototype-based matching that maps visual segments to semantic actions without training. Finally, we decode the resulting sequence through structured procedural inference using graph-constrained beam search, which enforces valid task transitions.

3.1 Problem Formulation

Given a streaming egocentric video, the objective is to infer an ordered sequence of procedural actions online while respecting task constraints. We formulate online procedure parsing as a Maximum A Posteriori (MAP) inference problem over a sequence of temporal segments.

Formally, let $\mathcal{V}$ denote a streaming video. At each time step t, we extract a frame-level visual representation $\mathbf{v}_t \in \mathbb{R}^d$. An online boundary detector $\phi_{boundary}$ operates over these features and partitions the stream into N segments

$$\mathcal{S} = \{s_1, s_2, \ldots, s_N\}.$$

Each segment s_i corresponds to a contiguous block of frame-level features representing a single procedural step.

Our objective is to assign an action label to each segment while respecting the structure encoded by a procedural graph $\mathcal{G}$. The optimal sequence of actions is obtained by solving

$$\mathbf{A}^* = \arg\max_{\mathbf{A}} P(\mathbf{A} \mid \mathcal{S}, \mathcal{G}). \tag{1}$$

3.2 Manipulation-Anchored Features (MAFs)

Actions in egocentric tasks are primarily defined by how the hands interact with objects. Consequently, the most informative visual signal lies in regions where these interactions occur rather than in the entire scene. To capture this signal, we construct Manipulation-Anchored Features (MAFs), which focus the representation on these interaction regions while suppressing background clutter.

We begin by extracting patch-level descriptors using a frozen DINOv2 backbone [29]. Specifically, we use the patch tokens from the final layer of a ViT-L/14 encoder [7]. To localize the interaction region, we use a frozen Hand-Object Detector (HOD) [35] to detect hands and manipulated objects in each frame. The detected bounding boxes are merged using an enclosing strategy (ref. 4.2) that produces a single bounding region covering the interaction area.

This region is projected onto the patch grid to define a spatial slice

$$R_t = [p_{x1}, p_{y1}, p_{x2}, p_{y2}]$$

The frame-level representation is then computed as the spatial average of patch tokens within this region:

$$\mathbf{f}_t = \frac{1}{|R_t|} \sum_{i=p_{y1}}^{p_{y2}} \sum_{j=p_{x1}}^{p_{x2}} \mathbf{P}_{t,i,j}. \tag{2}$$

If no detections are present in a frame, we propagate the last valid feature forward. This simple temporal smoothing prevents head motion or short occlusions from introducing spurious changes in the feature stream.

The resulting interaction-focused descriptors form the **Manipulation Anchored Features (MAFs)** used throughout the remainder of the pipeline.

3.3 Training-Free Boundary Detection

Procedural parsing requires identifying transitions between consecutive actions. Rather than learning a boundary predictor, we detect transitions directly from the temporal similarity structure of the MAF feature stream using a training-free novelty detector.

Our approach follows three steps: constructing a local similarity matrix over a sliding temporal window, computing a novelty signal using a checkerboard kernel, and emitting boundaries through online peak detection. This ensures that the system remains "online," operating only on a short buffer of the most recent frames rather than the entire video sequence.

Gaussian-tapered checkerboard kernel. The use of a Gaussian-tapered checkerboard kernel to detect boundaries in a Self-Similarity Matrix (SSM) was pioneered in [11] and [5] where they demonstrated that transitions between homogeneous signal segments manifest as checkerboard patterns along the SSM diagonal. The core of our detector is a checkerboard kernel $\mathbf{K} \in \mathbb{R}^{2L \times 2L}$, where L defines the temporal horizon of the sliding window. To emphasize frames near the center of the window and suppress noise at the edges, we apply a radial Gaussian taper to the kernel. The kernel is defined as

$$\mathbf{K} = (\mathbf{C} \otimes \mathbf{J}_L) \odot \mathbf{G},$$

where

$$\mathbf{C} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

is the base checkerboard pattern, $\mathbf{J}_L$ is an $L \times L$ matrix of ones, and $\mathbf{G}$ is a Gaussian weighting matrix centered at the kernel midpoint.

Local novelty computation. At each time step we maintain a buffer containing the most recent $2L$ MAF features. Using this buffer we compute a local self-similarity matrix $\mathbf{S} \in \mathbb{R}^{2L \times 2L}$, whose entries measure cosine similarity between normalized features:

$$S_{i,j} = \frac{\mathbf{f}_i^\top \mathbf{f}_j}{\|\mathbf{f}_i\| \|\mathbf{f}_j\|}. \tag{3}$$

The novelty score at time t is obtained through the Frobenius inner product of this matrix with the kernel

$$\Delta_t = \sum_{i=1}^{2L} \sum_{j=1}^{2L} S_{i,j} K_{i,j}. \tag{4}$$

Large novelty values indicate abrupt changes in the structure and therefore suggest an action transition. Because DINOv2 is trained to be semantically stable, an action (e.g., "chopping") appears as a homogeneous block in the MAF, allowing linear filters to perform complex semantic segmentation tasks that previously required deep temporal networks.

Boundary emission. The novelty signal is converted into discrete boundaries using online peak detection. A boundary is emitted when the novelty score forms a local maximum and exceeds a saliency threshold h:

$$\tau = \{t \mid \Delta_t > \Delta_{t\pm k} \text{ and } \Delta_t > h\}. \tag{5}$$

To avoid over-segmentation, we enforce a minimum temporal distance d from the previous boundary. Once a peak is detected, the buffer is flushed up to index τ, producing a segment that is passed to the inference stage.

3.4 Prototype-Based Action Matching

To assign semantic labels to the detected segments, we employ a non-parametric, micro-prototype matching scheme. A naïve approach would average all instances of an action into a single global prototype. However, procedural actions are often lengthy and multi-phasic; averaging an entire execution sequence blurs distinct sub-states and dilutes the Manipulation-Anchored Feature (MAF) signal. To preserve local temporal semantics, we first group training examples into execution styles using agglomerative clustering and compute representative centroids. We then temporally slice each clustered action sequence representations into overlapping, short-duration windows (e.g., 2 seconds). This produces a dense library of *micro-prototypes*, denoted as $\mathcal{P}_a$, capturing the distinct temporal phases of each action a without requiring temporal warping. During online inference, when the boundary detector emits a segment s_k, we apply the Hand-Object Detector (HOD) mask to remove frames without valid interactions. We compute the mean descriptor $\mathbf{g}_k$ over this filtered segment and assign labels by finding the nearest neighbor in the micro-prototype space. The matching cost between the segment and action a is defined by the minimum cosine distance to its micro-prototypes:

$$\phi(s_k, a) = \min_{p \in \mathcal{P}_a} \left(1 - \frac{\mathbf{g}_k^\top p}{\|\mathbf{g}_k\| \|p\|}\right). \tag{6}$$

3.5 Structured Procedural Inference

Visual evidence alone can lead to ambiguous or noisy action assignments, particularly when multiple actions exhibit similar visual patterns. To enforce valid task progression, we perform structured procedural inference using a task graph that encodes allowable action transitions. We formulate decoding as the minimization of a structured energy function that integrates visual evidence, manipulation cues, and graph constraints. Finally, we incorporate a rectification mechanism that handles short visual gaps caused by occlusions or detector failures.

Task graph construction. We induce the task-graph from training sequences by aggregating step-ordered annotations and identifying prerequisite relationships between actions. First, we record all immediate sequential transitions, distinguishing between first-time visits and revisits. We then accumulate previously observed steps as potential prerequisites for each action's first occurrence. We also designate steps as omittable if valid instances exist without that step in the training set. Finally, actions with an empty minimal prerequisite set are classified as start nodes, and we fully connect all start nodes to each other

Energy formulation. We combine visual evidence, manipulation cues, and task-graph constraints to score candidate labelings.

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \sum_{i=1}^{N} \Big[D_i \cdot \phi(s_i, a_i) + \lambda D_i (1 - \bar{M}_i) \Big] + \Psi(\mathbf{A}, \mathcal{G}). \tag{7}$$

Here D_i denotes the duration of segment s_i, while $\bar{M}_i$ represents the mean manipulation confidence within the segment. While Eq. 7 is written over the entire segment sequence for clarity, inference is updated incrementally online. We enforce this using a fixed-lag commitment strategy: predictions older than few seconds are frozen and cannot be altered by future observations, ensuring the decoding relies only on a bounded temporal buffer. The first term encourages segments to match visually similar action prototypes, with longer segments contributing proportionally to the objective. The second term introduces a visibility prior that penalizes segments lacking reliable interaction evidence. Procedural structure is enforced through the following graph constraint:

$$\Psi(a_{i-1}, a_i) = \begin{cases} 0 & (a_{i-1}, a_i) \in E_{\mathcal{G}} \\ \infty & \text{otherwise.} \end{cases} \tag{8}$$

Low-Visibility Inertia and Background Assignment. Egocentric videos often suffer from transient occlusions or rapid head movements where hands momentarily leave the frame. To maintain stability under extremely low hand visibility, we employ a two-part strategy. If the system is actively tracking a procedural step, the decoder applies inertia, automatically extending the previously predicted action with a small constant cost to seamlessly bridge the visual gap. Conversely, if the sequence is just beginning or the system is already in a dynamically suppressed background state, this lack of visibility safely assigns the segment to the background (BG). This preserves task continuity during brief occlusions while preventing spurious predictions when no manipulation is visible.

Gap rectification. Egocentric videos often contain short occlusions or detector failures that interrupt otherwise continuous actions. To address these cases, we introduce a rectification mechanism during beam search. If a background segment is followed by a high-confidence return to the preceding action, the decoder retroactively assigns that action label to the gap. The rectified energy becomes

$$E_{\text{rectified}} = E_{\text{prev}} - \text{Cost}_{BG} + (D_{\text{gap}} \cdot \bar{\phi}_{a_{i-2}}). \tag{9}$$

This mechanism allows the model to bridge short visual interruptions while maintaining procedural consistency.

4 Experiments

4.1 Datasets and Metrics

We evaluate our framework on two benchmark datasets for egocentric procedural activity: GTEA [10] and EgoPER [25]. GTEA consists of 28 videos spanning 7 fine-grained kitchen activities, which we process at a temporal resolution of 15 fps. EgoPER contains 213 normal and 173 erroneous egocentric videos across 5 recipes; following [36], we evaluate only on the normal videos and use the same data splits. For EgoPER we operate at a temporal resolution of 10 fps. We report standard procedure parsing metrics including frame-wise Accuracy (Acc) without background frames, Edit distance, and F1 scores at overlap thresholds $\{0.1, 0.25, 0.5\}$ to capture both classification accuracy and temporal alignment quality. However, these standard segmentation metrics primarily measure local boundary accuracy and do not capture whether predicted action sequences preserve long-range procedural dependencies. To address this limitation, we additionally introduce *N-Step Transition Accuracy* to evaluate structured parsing.

This metric extracts all n-step action transitions from the predicted sequence and compares them with the ground-truth to measure consistency of multi-step dependencies. We compute precision–recall curves for these transitions and report the area under the curve (AUC). To evaluate performance under partial observations, we sweep the evaluation over progressive video completion levels. Specifically, each video is parsed using only the first $\{10\%, 20\%, \ldots, 100\%\}$ of its frames. This setting quantifies a model's ability to maintain a coherent procedural state as additional evidence becomes available.

4.2 Implementation Details

Our framework utilizes a frozen ViT-L/14 DINOv2 backbone [29] as the feature extractor. We apply a frozen hand-object detector (HOD) from [35] to obtain bounding boxes for hands and active objects in each frame. *Manipulation-Anchored Features.* We compute the minimum spanning box covering these detections to define the interaction region. We then identify the 16×16 DINOv2 patch tokens that intersect with this region. The final frame representation $\mathbf{f}_t$ is obtained by averaging the selected tokens. *Training-Free Boundary Detection.* The checkerboard kernel width L is set to 20 frames for EgoPER and 10 frames for GTEA. *Prototype Matching.* The number of clusters is set to $k = 3$ for GTEA and $k = 9$ for EgoPER. To construct the micro-prototypes, we temporally slice these cluster representations using a segment length of 4 seconds with a 2-second stride for EgoPER, and a 1.5-second segment length with a 0.5-second stride for GTEA. *Graph-Constrained Procedural Inference.* Beam search decoding uses beam width $B = 10$ for EgoPER and $B = 5$ for GTEA.

5 Results

Despite requiring no training or fine-tuning, VidParse achieves competitive performance with strong online baselines while maintaining improved procedural

Table 1: Action Segmentation on Procedural Datasets: Comparison to offline/online baselines on GTEA and EgoPer. We achieve SoTA result on all metrics.

			GTEA					EgoPer				
Method	**Inference**	**Train-Free?**	Acc	Edit	F1@{0.1,0.25,0.5}			Acc	Edit	F1@{0.1,0.25,0.5}		
MSTCN [34]	Offline	✗	79.35	84.46	86.54	83.79	71.86	87.52	92.34	92.60	91.81	86.12
MSTCN	Online	✗	47.28	60.26	66.75	60.12	40.29	25.40	44.81	44.09	31.78	15.72
ProTAS [36]	Online	✗	73.19	71.81	72.89	68.94	54.87	76.61	65.50	64.26	62.59	51.31
Ours	Online	✓	**89.1**	**87.4**	**91.9**	**89.9**	**80.5**	**80.7**	**88.7**	**91.1**	**88.5**	**77.6**

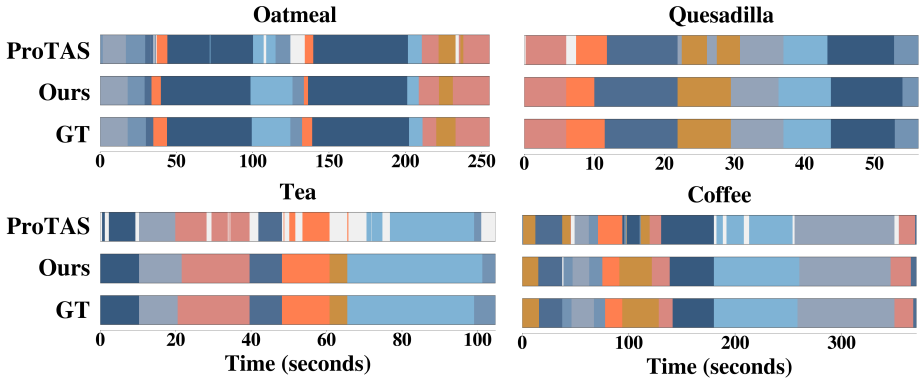


Fig. 3: Action Segmentation Performance: Frame-level action segmentation models, such as ProTAS are brittle and result in inconsistent parsing. Our structured procedural inference results in smooth and stable action transitions. (Background GT frames have been removed here for brevity)

consistency. We evaluate our method on several axes. First, We analyze action segmentation performance on standard benchmarks. We then study procedural parsing accuracy to access whether predicted sequences preserve long-range task structure. Next, we examine the temporal behavior of our online boundary detection mechanism. Finally, we present ablation studies to better understand the design choices of our framework.

5.1 Action Segmentation Performance

We evaluate action segmentation on the GTEA and EgoPER datasets. Tab. 1 reports results using Accuracy, Edit distance, and F1 scores at multiple overlap thresholds. Our method consistently outperforms the ProTAS baseline, improving F1 scores by more than 15% across all thresholds and achieving gains of up to 25% on F1@0.5 on GTEA. On EgoPER, we observe improvements of roughly 25% on F1 and Edit metrics while maintaining a 4% gain in frame-level accuracy.

Unlike frame-level models that predict labels independently for each frame, our approach produces temporally coherent action segments. As a result, small boundary offsets may slightly affect frame accuracy even when predicted segments align well with the ground-truth. Metrics such as Edit distance and F1@k

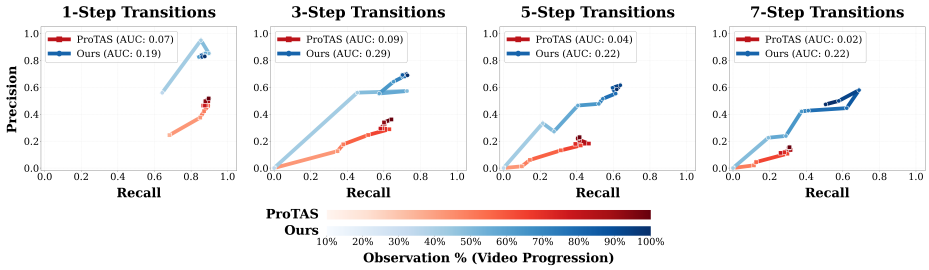


Fig. 4: N-Step transition performance on EgoPER: VidParse consistently outperforms the ProTAS baseline, with the gap widening for higher transition lengths.

therefore better reflect segmentation quality in this setting. As illustrated in Fig. 3, our predictions remain smooth and closely follow the ground-truth segmentation without the flickering commonly observed in frame-level models.

5.2 Procedural Parsing Accuracy

Procedural tasks require maintaining a consistent sequence of actions rather than recognizing actions independently. To evaluate this property, we measure *N-step Transition Accuracy*, which captures whether predicted sequences preserve valid multi-step transitions. Let $\mathcal{H}_n(S)$ denote the *multiset* of all n-step transitions in sequence S:

$$\mathcal{H}_n(S) = \{(s_i, s_{i+1}, \ldots, s_{i+n}) \mid 1 \leq i \leq k - n\} \tag{10}$$

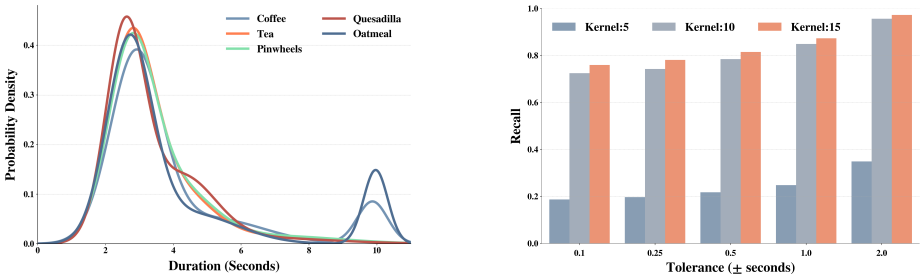
We evaluate performance by comparing the predicted multiset $\mathcal{H}_n(S_{\text{pred}})$ with the ground-truth multiset $\mathcal{H}_n(S_{\text{gt}})$. Let $\mathcal{U}$ be the set of unique transition tuples present in either sequence. For any transition tuple $h \in \mathcal{U}$, let $c(h, S)$ denote the count (multiplicity) of h in $\mathcal{H}_n(S)$. True Positive (TP) count is defined as:

$$TP_n = \sum_{h \in \mathcal{U}} \min\left(c(h, S_{\text{pred}}),\ c(h, S_{\text{gt}})\right) \tag{11}$$

As shown in Fig. 4, our method consistently outperforms the ProTAS baseline across all transition lengths. The gap increases with larger n: our AUC for 5-step and 7-step transitions is up to 5×–10× higher than the baseline. This indicates that our approach preserves the global task structure more effectively, while frame-level baselines accumulate local errors that disrupt long-range transitions and violate procedural constraints. Our method therefore reconstructs the underlying task graph rather than recognizing actions in isolation.

5.3 Online Boundary Detection Behavior

We analyze the temporal behavior of our online boundary detection mechanism. Fig. 5(a) shows the distribution of emitted segment durations. Approximately 75% of predicted segments fall within a 2–4 second window, and more than



(a) Action duration comparison. **(b)** Boundary recall analysis.

Fig. 5: Training-Free Boundary Analysis: (a) 75% of inference segments are under 4s showing our online nature (b) at *kernel*=10, we can recover a boundary within 2 s over 90% of the time showing that we do not mix actions

90% are shorter than 5 seconds. We compare the performance of our boundary detection method with other online unsupervised methods in Tab. 4(b) Bottom-Right. *Adjacent Frame Similarity* emits a boundary when the cosine distance between consecutive frame embeddings exceeds a fixed, predefined threshold. *z-score* triggers a boundary when the frame-to-frame distance deviates from the local mean and standard deviation of a rolling temporal window. Our method provides the best performance on the downstream action segmentation metrics.

Operating at the segment level also stabilizes predictions over time compared to frame-level methods. Frame-level models often produce rapid label fluctuations near action boundaries, whereas our method groups semantically consistent frames into discrete segments. As illustrated in Fig. 3, the resulting predictions closely match the ground-truth structure and remain temporally stable.

5.4 Ablation Studies

Feature Representation. We analyze the utility of using MAF representations in Tab. 2, and further compare them against egocentric video-language backbone, EgoVLPv2 [33] in Tab. 3 (Right). Starting from the ProTAS baseline, incorporating MAF features leads to consistent improvements across segmentation metrics, indicating that manipulation-anchored representations provide stronger cues for procedural actions. Applying our method on top of the same MAF features further improves performance across all metrics and achieves the best overall results. Full-frame features underperform MAFs, suggesting that explicit foreground information is necessary. Frame-level EgoVLPv2 remains competitive with DINOv2 MAFs, while clip-level features perform worse, indicating that coarse temporal representations can miss short procedural transitions.

Prototype Method. We evaluate prototype methods in Tab. 4(a). While computing a global mean collapses the action feature space, relying strictly on medoids limits the representation to previously observed instances. Our centroid-based approach better captures the action feature space while remaining representative of the underlying prototypes. This strategy yields the best performance

Table 2: MAFs vs other features: MAFs outperform other features across methods on EgoPER

			Coffee			Oatmeal			Pinwheels			Quesdilla			Tea			Overall		
Method	**Feat.**	**Train-Free?**	Acc	Edit	F1	Acc	Edit	F1	Acc	Edit	F1	Acc	Edit	F1	Acc	Edit	F1	Acc	Edit	F1
ProTAS [36]	I3D	✗	60.58	36.75	24.81	86.84	76.33	65.85	78.74	75.82	55.68	78.98	62.18	53.81	77.94	76.44	56.42	76.61	65.50	51.31
ProTAS	DINO CLS	✗	66.42	**50.40**	39.16	**88.85**	**77.22**	**74.55**	82.33	81.54	72.34	83.58	74.80	67.48	85.37	**79.18**	68.68	81.31	**72.63**	64.44
ProTAS	DINO MAFs	✗	**71.80**	49.17	**46.04**	87.47	70.67	70.16	**83.28**	**84.39**	**74.62**	**86.71**	**75.37**	**71.83**	**87.28**	74.75	**73.64**	**83.31**	70.87	**67.26**
Ours	DINO CLS	✓	65.06	74.10	58.66	74.74	86.11	60.54	69.38	**86.25**	56.06	69.68	86.92	65.41	65.77	78.04	64.52	69.09	82.68	61.42
Ours	DINO MAFs	✓	**75.82**	**76.45**	**73.37**	**80.40**	**91.57**	**69.70**	**74.58**	81.78	**57.94**	**83.58**	**91.90**	**89.71**	**86.42**	**96.74**	**91.97**	**80.69**	**88.69**	**77.61**

Table 3: Inference Speed and Feature Comparisons on EgoPER. (Left) Impact of beam width (B) on inference speed and parsing accuracy. The times shown are average seconds/video. Our CPU-based method is faster than the GPU-accelerated ProTAS baseline **(Right)** Feature comparisons highlighting the importance of frame-level MAFs over clip-level or full-frame features.

Set	**HOD**	**Backbone**	**Method**	**Total**	**FPS**	**Acc**	**Edit**	**F1@0.5**
ProTAS [36]	-	79.5	706	785.5	4.5	76.6	65.5	51.3
Ours ($B = 1$)	339	36.8	**24.8**	**400.6**	**8.7**	58.5	77.5	52.1
Ours ($B = 3$)	339	36.8	28.2	404.0	**8.7**	76.3	86.0	72.4
Ours ($B = 5$)	339	36.8	28.4	404.2	8.6	78.5	86.9	74.5
Ours ($B = 10$)	339	36.8	32.8	408.6	8.6	**80.7**	**88.7**	**77.6**

Method	**MAF**	**Clip**	**Acc**	**Edit**	**F1**
EgoVLP2	✗	✗	78.1	89.4	73.0
EgoVLP2	✓	✗	80.3	**91.5**	77.5
EgoVLP2	✗	✓	53.4	62.8	43.7
EgoVLP2	✓	✓	60.1	70.6	52.1
Ours	✗	✗	69.1	82.7	61.4
Ours	✓	✗	**80.7**	88.7	**77.6**

across all metrics, reaching an Acc. of 80.69 and an F1@0.5 of 77.61. Further, Tab. 4(d) demonstrates that our clustering approach for action matching is robust. Varying the number of prototypes p, extracted per action class, from 3 to 11 results in less than a 1.2% variance in Accuracy and Edit score.

Inference Speed. In Tab. 3 Left, we analyze the impact of beam width (B) on inference speed. Although MAF extraction adds HOD overhead, our approach is significantly faster, achieving 8.6 FPS, than the ProTAS baseline (4.5 FPS), allowing for a highly efficient and tunable speed-accuracy trade-off. Also, our parsing and decoding stages, operating on an AMD EPYC 7443 CPU, require only 32.8s vs 706s needed by ProTAS running on an NVIDIA RTX A4000.

Minium distance *d* As shown in Table 4(c), the minimum gap parameter d controls how close consecutive boundaries can be. Smaller values allow denser boundaries and can mildly over-segment actions, while larger values impose rigid spacing and may miss rapid transitions. Overall, performance remains stable across settings and consistently outperforms the baseline.

6 Limitations

While our framework is causal and does not require global context, it operates at the segment level rather than per-frame, which may limit suitability for ultra-low-latency applications. Second, the structural prior is induced from training sequences to form a directed task graph, which may restrict generalization to out-of-distribution procedural executions. Finally, our semantic representation relies on the Hand-Object Detector (HOD); in scenarios involving severe hand–object

Table 4: Ablation and Hyperparameter Sensitivity Analyses on EgoPER(a) prototype matching strategies and (b) boundary detection methods, alongside sensitivity to (c) the gap threshold d and (d) the number of prototypes p.

(a) Matching

Type	Acc	Edit	F1
Global	78.5	88.8	74.5
Medoids	80.1	87.0	75.0
Centroids	**80.7**	**88.7**	**77.6**

(b) Boundary

Method	Acc	Edit	F1
Adj. Frame	75.7	83.4	68.0
Z-score	78.5	85.1	69.9
ABD [8]	78.5	88.1	72.2
Ours	**80.7**	**88.7**	**77.6**

(c) d

d	Acc	Edit	F1
10	81.0	84.1	73.5
15	**81.7**	87.9	75.7
20	80.7	**88.7**	**77.6**
25	78.1	87.7	72.8
30	76.1	87.7	71.3

(d) p

p	Acc	Edit	F1
3	80.5	89.0	75.8
5	79.7	89.2	76.0
7	80.1	**89.9**	76.5
9	**80.7**	88.7	**77.6**
11	80.4	89.2	77.1

occlusions, extreme motion blur, or purely non-interactive actions (e.g., "Using a Microwave"), this may result in spurious predictions or delayed parsing. Please refer to our supplementary and project page for more details on this. While our framework could theoretically extend to exocentric videos where manipulation cues remain clearly visible, performance would likely degrade if these cues are weak or absent. Future work could also explore probabilistic graphs to better accommodate open-world scenarios or loosely structured tasks.

7 Conclusion

We presented VidParse, an online, training-free framework for egocentric procedure parsing that treats activity understanding as a structured inference problem. By leveraging Manipulation-Anchored Features and a Gaussian-tapered checkerboard kernel applied to frozen DINOv2 representations, our method detects action boundaries directly from visual evidence without gradient-based learning. We further integrate task-graph constrained beam search with gap rectification to enforce procedural consistency and correct transient visual failures. Experiments demonstrate that combining structural priors with foundation model features yields competitive performance with strong online baselines while better preserving long-range procedural structure.

8 Acknowledgments

This work was partially supported by NSF CAREER Award (#2238769) to Abhinav Shrivastava. The authors acknowledge UMD's supercomputing resources made available for conducting this research. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or the U.S. Government.

References

1. Ali, A.S., Mahmood, S.A., Saeed, M., Konin, A., Zia, M.Z., Tran, Q.H.: Joint self-supervised video alignment and action segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10807–10818 (2025)
2. Bansal, S., Arora, C., Jawahar, C.: My view is the best view: Procedure learning from egocentric videos. In: European Conference on Computer Vision. pp. 657–675. Springer (2022)
3. Catinello, A.S., Farinella, G.M., Furnari, A.: Mamba-otr: A mamba-based solution for online take and release detection from untrimmed egocentric video. In: International Conference on Image Analysis and Processing. pp. 456–468. Springer (2025)
4. Chowdhury, S.S., Chandra, S., Roy, K.: Opel: Optimal transport guided procedure learning. Advances in Neural Information Processing Systems **37**, 59984–60011 (2024)
5. Cooper, M., Foote, J.: Scene boundary detection via video self-similarity analysis. In: Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205). vol. 3, pp. 378–381. IEEE (2001)
6. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: The epic-kitchens dataset: Collection, challenges and baselines. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(11), 4125–4141 (2020)
7. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Du, Z., Wang, X., Zhou, G., Wang, Q.: Fast and unsupervised action boundary detection for action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3323–3332 (2022)
9. 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)
10. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR 2011. pp. 3281–3288. IEEE (2011)
11. Foote, J.: Automatic audio segmentation using a measure of audio novelty. In: 2000 ieee international conference on multimedia and expo. icme2000. proceedings. latest advances in the fast changing world of multimedia (cat. no. 00th8532). vol. 1, pp. 452–455. IEEE (2000)
12. Gong, D., Lee, J., Jung, D., Kwak, S., Cho, M.: Activity grammars for temporal action segmentation. Advances in Neural Information Processing Systems **36**, 75413–75433 (2023)
13. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE international conference on computer vision. pp. 5842–5850 (2017)
14. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18995–19012 (2022)
15. Han, C., Wang, H., Kuang, J., Zhang, L., Gui, J.: Training-free zero-shot temporal action detection with vision-language models. arXiv preprint arXiv:2501.13795 (2025)

16. Huang, J., Li, Z., Naik, M., Lim, S.N.: Laser: A neuro-symbolic framework for learning spatial-temporal scene graphs with weak supervision. arXiv preprint arXiv:2304.07647 (2023)
17. Hussein, N., Gavves, E., Smeulders, A.W.: Videograph: Recognizing minutes-long human activities in videos. arXiv preprint arXiv:1905.05143 (2019)
18. Hwang, J., et al.: DiffGEBD: Diversity-aware Diffusion model for Generic Event Boundary Detection. Ph.D. thesis (2025)
19. Jiang, X., Huang, Z., Xu, X., Song, J., Shen, F., Shen, H.T.: Phgc: Procedural heterogeneous graph completion for natural language task verification in egocentric videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8615–8624 (2025)
20. Jung, H., Kim, D., Lim, S., Son, J., Choi, J.: Online generic event boundary detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13741–13750 (2025)
21. Kang, H., Kim, J., Kim, T., Kim, S.J.: Uboco: Unsupervised boundary contrastive learning for generic event boundary detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20073–20082 (2022)
22. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
23. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 780–787 (2014)
24. Lee, S.P., Elhamifar, E.: Error recognition in procedural videos using generalized task graph. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10009–10021 (2025)
25. Lee, S.P., Lu, Z., Zhang, Z., Hoai, M., Elhamifar, E.: Error detection in egocentric procedural task videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18655–18666 (2024)
26. Li, Y., Xue, Z., Xu, H.: Otas: unsupervised boundary detection for object-centric temporal action segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6437–6446 (2024)
27. Liberatori, B., Conti, A., Rota, P., Wang, Y., Ricci, E.: Test-time zero-shot temporal action localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18720–18729 (2024)
28. Nguyen, T.S., Yang, H., Neoh, T.Y., Zhang, H., Keat, E.Y., Fernando, B.: Pkr-qa: A benchmark for procedural knowledge reasoning with knowledge module learning. AAAI (2026)
29. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
30. Peddi, R., Arya, S., Challa, B., Pallapothula, L., Vyas, A., Gouripeddi, B., Zhang, Q., Wang, J., Komaragiri, V., Ragan, E., et al.: Captaincook4d: A dataset for understanding errors in procedural activities. Advances in Neural Information Processing Systems **37**, 135626–135679 (2024)
31. Peirone, S.A., Pistilli, F., Averta, G.: Hiero: understanding the hierarchy of human behavior enhances reasoning on egocentric videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19862–19871 (2025)
32. Perrett, T., Darkhalil, A., Sinha, S., Emara, O., Pollard, S., Parida, K.K., Liu, K., Gatti, P., Bansal, S., Flanagan, K., et al.: Hd-epic: A highly-detailed egocentric

video dataset. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 23901–23913 (2025)
33. Pramanick, S., Song, Y., Nag, S., Lin, K.Q., Shah, H., Shou, M.Z., Chellappa, R., Zhang, P.: Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5285–5297 (2023)
34. Sekaran, S.R., Pang, Y.H., Ling, G.F., Yin, O.S.: Mstcn: A multiscale temporal convolutional network for user independent human activity recognition. F1000Research **10**, 1261 (2022)
35. Shan, D., Geng, J., Shu, M., Fouhey, D.F.: Understanding human hands in contact at internet scale. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9869–9878 (2020)
36. Shen, Y., Elhamifar, E.: Progress-aware online action segmentation for egocentric procedural task videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18186–18197 (2024)
37. Song, Y., Kim, D., Cho, M., Kwak, S.: Online temporal action localization with memory-augmented transformer. In: European Conference on Computer Vision. pp. 74–91. Springer (2024)
38. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 729–738 (2013)
39. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1207–1216 (2019)
40. Xu, A., Zheng, W.S.: Efficient and effective weakly-supervised action segmentation via action-transition-aware boundary alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18253–18262 (2024)
41. Yi, F., Wen, H., Jiang, T.: Asformer: Transformer for action segmentation. arXiv preprint arXiv:2110.08568 (2021)
42. Zheng, Z., He, L., Yang, L., Li, F.: Fine-grained dynamic network for generic event boundary detection. In: European Conference on Computer Vision. pp. 107–123. Springer (2024)
43. Zhong, Q., Ding, G., Yao, A.: Onlinetas: An online baseline for temporal action segmentation. Advances in Neural Information Processing Systems **37**, 58984–59005 (2024)
44. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3537–3545 (2019)