

Streaming Atomic Action Segmentation in Egocentric Videos via Iterative LVLM-to-Streaming Distillation

Hyolim Kang¹ Jungsoo Lee¹ Janghoon Cho¹

Juntae Lee¹ Seon Joo Kim² Sungha Choi^{3*†}

¹Qualcomm AI Research[‡] ²Yonsei University ³Kyung Hee University

{hyolkang, jungsool, janghoon, juntlee}@qti.qualcomm.com

seonjookim@yonsei.ac.kr sunghac@khu.ac.kr

Abstract

Large Vision-Language Models (LVLMs) are promising scalable annotators for video understanding, but they are not directly suited for dense temporal annotation. We study this challenge in egocentric streaming understanding, where the goal is to segment continuous first-person video into atomic actions: short, concrete physical interactions with well-localized temporal boundaries. We propose **CoTT** (Chains of Temporal Transitions), an annotation-free framework that structures LVLM reasoning around temporal transitions and combines it with iterative streaming distillation. CoTT uses structured LVLM reasoning to generate dense atomic action pseudo-labels from unlabeled egocentric videos, distills these labels into a lightweight online model, and bootstraps the model’s predictions to improve temporal consistency. On large-scale egocentric benchmarks, EPIC-Kitchens-100 and Ego-Exo4D, CoTT achieves competitive performance against supervised online localization methods without using ground truth temporal annotations for training. It also transfers across egocentric datasets and, on THUMOS14, outperforms prior best training-free zero-shot Temporal Action Localization methods, even without any dataset-specific training or adaptation.

1 Introduction

Wearable cameras and augmented-reality devices are shifting video understanding toward egocentric streaming. Egocentric video (Grauman et al., 2022) centers understanding on the camera wearer’s physical interactions with the environment. Streaming video requires these interactions to be interpreted causally, without access to future frames. A key challenge is therefore to continuously identify when the wearer’s actions start and end.

*Work done while at Qualcomm AI Research.

†Corresponding author.

[‡]Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

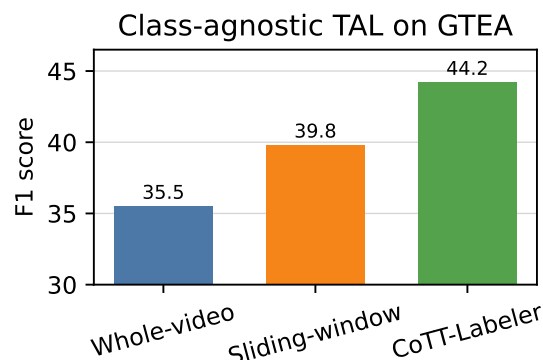


Figure 1: Compared to direct whole-video and sliding-window prompting, our structured LVLM inference pipeline (CoTT-Labeler) substantially improves atomic action localization.

This immediately raises the question of what temporal unit should be treated as an action. High-level activities such as *cooking* or *repairing a device* are often too coarse and temporally ambiguous for streaming perception. We therefore adopt the atomic-action granularity commonly used in established egocentric and procedural video benchmarks (Damen et al., 2020; Grauman et al., 2024; Fathi et al., 2011). At this level, segments are short, object-directed physical interactions with recognizable intent, such as *open fridge* or *pour coffee*. They are coarser than sub-movements such as *grasp the handle* or *pull the door*, yet finer and temporally better localized than high-level activities. We use *atomic actions* to refer to this established annotation level throughout the paper.

Despite its importance, scalable atomic action segmentation remains challenging due to its reliance on dense temporal supervision. Annotating long egocentric videos requires exhaustively marking precise start and end times for many brief, densely occurring interactions. Thus, large-scale temporally aligned supervision for generic atomic actions remains expensive to obtain through human

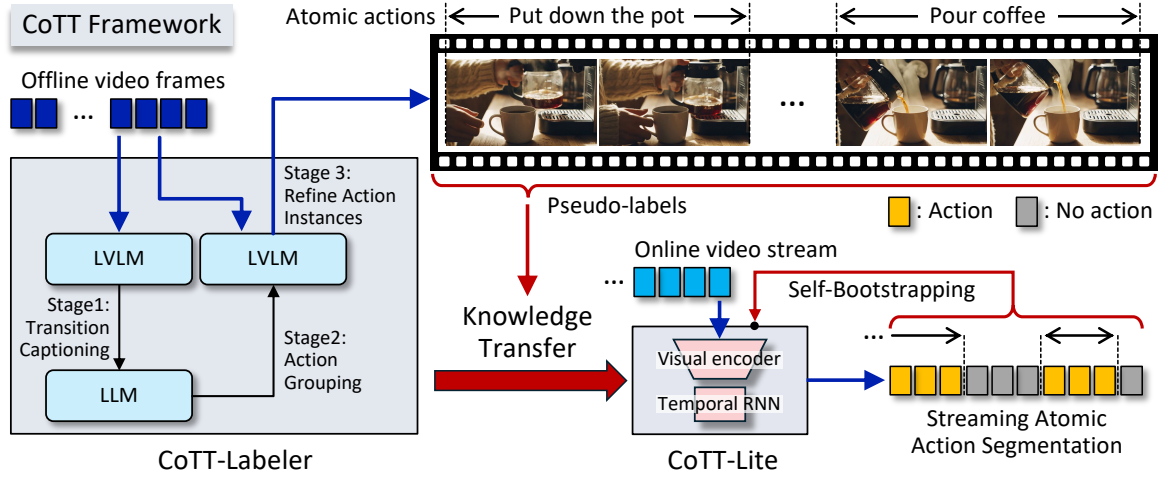


Figure 2: Overall framework of CoTT. CoTT-Labeler first processes offline egocentric videos with an LVLM/LLM pipeline. This produces dense atomic action pseudo-labels from unlabeled videos. CoTT then distills these pseudo-labels into CoTT-Lite, a lightweight causal segmenter that operates on online video streams without LVLM calls. Finally, CoTT bootstraps CoTT-Lite predictions as improved pseudo-labels for subsequent training rounds, transferring LVLM temporal reasoning into an efficient streaming model.

annotation.

Large Vision-Language Models (LVLMs) offer a promising alternative to manual supervision, as they can generate temporal annotations directly from unlabeled video. Yet, despite their success in long-form VQA (Cho et al., 2026; Maaz et al., 2023; Arnab et al., 2025) and video grounding (Wang et al., 2025; Ren et al., 2024; Zeng et al., 2024), straightforward LVLM inference struggles with exhaustive atomic action segmentation. A whole-video prompt asks the model to enumerate all actions in an untrimmed video at once, but often collapses into a coarse summary of salient events. Sliding-window prompting instead queries short clips obtained by splitting the video; it provides finer local evidence, but often fragments, duplicates, or misaligns actions that cross clip boundaries. (Figure 3)

To this end, we propose **CoTT** (Figure 2), an annotation-free framework for streaming egocentric atomic action segmentation based on **Chain of Temporal Transitions**. CoTT first uses a structured LVLM inference pipeline, **CoTT-Labeler**, to convert unlabeled videos into dense atomic action pseudo-labels through transition captioning, language-based grouping, and boundary refinement. This decomposition avoids asking the LVLM to enumerate all actions from a long video at once, while also avoiding the boundary fragmentation of direct sliding-window localization (Figure 1)

Although CoTT-Labeler provides dense pseudo-

labels, it remains an offline LVLM-based annotation pipeline, making it unsuitable for real-time streaming deployment. We therefore distill its pseudo-labels into **CoTT-Lite**, a lightweight recurrent model that performs streaming atomic action segmentation without LVLM calls. Beyond efficiency, this distillation also improves temporal coherence: CoTT-Labeler supplies broad semantic coverage, but its boundaries can be noisy or still fragmented due to the LVLM uncertainty. CoTT-Lite, by design, favors predictions that match the temporal structure of atomic actions, using actionness to detect action presence and progression drop to identify coherent action endpoints. This temporal inductive bias suppresses noisy boundary fluctuations and turns LVLM pseudo-labels into more consistent streaming predictions, which we then reuse as improved pseudo-labels in an iterative bootstrapping procedure.

Our experiments focus on large-scale egocentric benchmarks, EPIC-Kitchens-100 and Ego-Exo4D, where exhaustive atomic action annotation is costly and scalability is essential. Without human temporal annotations for training, CoTT-Lite achieves competitive performance against fully supervised online temporal action localization methods, and iterative bootstrapping further improves performance. We also show cross-dataset transfer from Ego-Exo4D to GTEA, and demonstrate that CoTT-Labeler can serve as a strong class-conditioned zero-shot temporal action localizer on

THUMOS14.

Our contributions are summarized as follows:

- We take a first step toward streaming atomic action segmentation in egocentric videos *without manual temporal annotations* by introducing **CoTT**, a framework that combines structured LVLM pseudo-labeling with iterative online distillation.
- We propose **CoTT-Labeler**, a structured LVLM annotation pipeline that generates reliable atomic action pseudo-labels by substantially improving over straightforward whole-video and sliding-window prompting.
- We introduce **CoTT-Lite**, a lightweight online segmenter trained from CoTT-Labeler pseudo-labels, and integrate it into an iterative LVLM-to-streaming distillation framework that bootstraps its predictions for improved temporal consistency.
- CoTT achieves competitive annotation-free online segmentation on EPIC-Kitchens-100 and Ego-Exo4D, transfers across egocentric datasets, and yields strong class-conditioned zero-shot TAL results on THUMOS14.

2 Related Work

2.1 Streaming Action Perception

Online Action Detection (OAD) (De Geest et al., 2016) is a well-established task in streaming video understanding that involves instantly identifying the action class of each incoming frame. While extensive research has been devoted to OAD (Xu et al., 2019; Eun et al., 2020; Yang et al., 2022; Wang et al., 2021; Xu et al., 2021; Chen et al., 2022; Zhao and Krähenbühl, 2022; An et al., 2023; Zhao et al., 2024), its frame-centric nature fails to capture the event-level semantics required for practical streaming analysis.

To address this limitation, Online Temporal Action Localization (On-TAL) (Tang et al., 2022; Reza et al., 2024; Song et al., 2024; Kim et al., 2022b,a; Han et al., 2026), recently introduced in (Kang et al., 2021), targets real-time event-level detection by identifying both the start and end timestamps of actions. Beyond the constraint of unobserved future frames, On-TAL demands immediate responsiveness, producing localization results as soon as an event terminates. Recent works (Kang

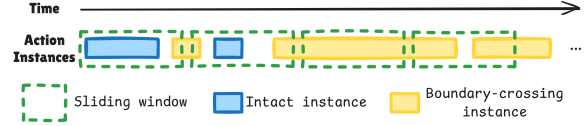


Figure 3: Challenges in sliding-window LVLM inference. Each window is processed independently. Actions fully contained within a window are easier to localize, whereas actions crossing window boundaries can be fragmented or misaligned.

et al., 2024; Jung et al., 2025) highlight the importance of a taxonomy-free understanding in streaming videos, which can be extended into a hierarchical, open-ended streaming understanding system when coupled with powerful LVLMs (Kang et al., 2025).

Our work. While recent On-TAL algorithms have reduced their reliance on predefined class labels, we argue that the key bottleneck toward a truly generalizable system lies in the scarcity of large-scale, taxonomy-free annotations. CoTT addresses this bottleneck by using LVLMs to generate dense atomic action pseudo-labels from untrimmed videos, distilling them into a lightweight online segmenter, and iteratively bootstrapping the resulting predictions to improve temporal consistency.

2.2 Large Vision-Language Models

In vision-language alignment learning, image captioning models have been employed to generate large-scale synthetic annotations that can rival or even surpass human-labeled data in quality (Santurkar et al., 2023; Nguyen et al., 2023). Extending this idea to videos, recent studies (Wang et al., 2025; Zeng et al., 2024; Ren et al., 2024) have demonstrated that LVLMs can describe and temporally ground complex events, highlighting their potential as scalable annotators for video understanding tasks (Han et al., 2025b). Importantly, TCoT (Arnab et al., 2025) showed that test-time scaling with a structured Chain-of-Thought inference strategy substantially improves long-form VQA performance.

Our work. We study the full pipeline of using LVLMs as supervision for egocentric atomic action segmentation, from pseudo-label generation to iterative bootstrapping of an online model.

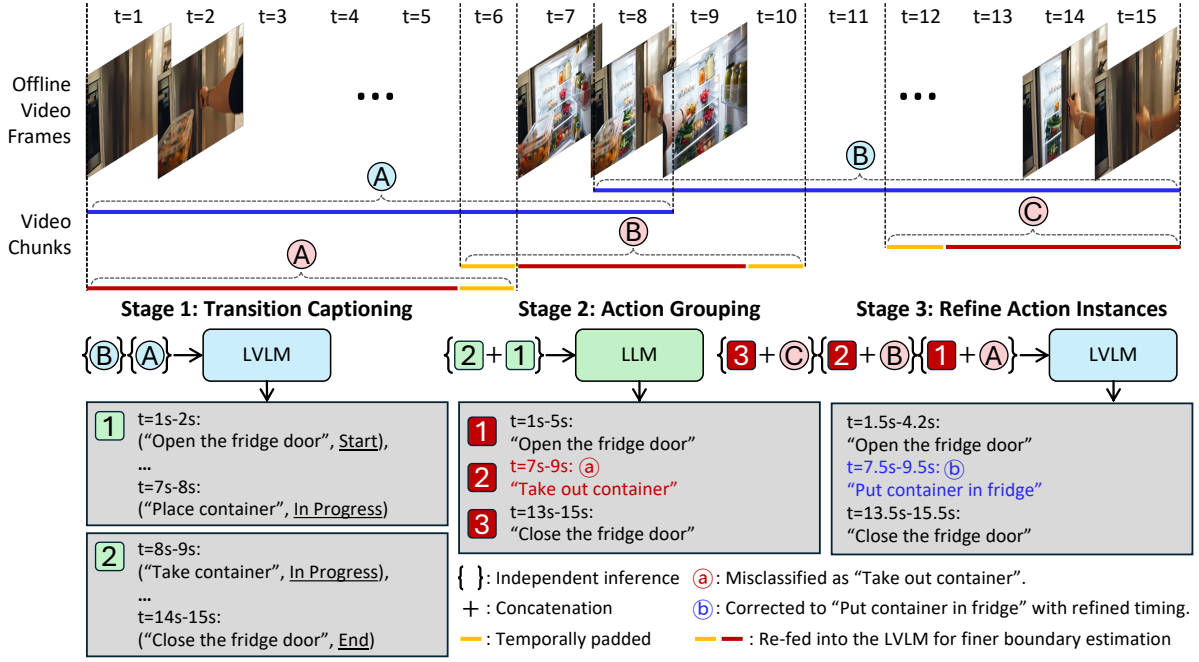


Figure 4: Overview of CoTT-Labeler. **Stage 1: Transition Captioning** extracts local frame-to-frame transition descriptions from short clips. **Stage 2: Action Grouping** merges transition captions into primitive atomic action instances using longer language context. **Stage 3: Boundary and Description Refinement** revisits each proposed instance with visual evidence to refine its temporal span and description.

3 Methodology

3.1 Problem Formulation

Streaming Atomic Action Segmentation in Egocentric Videos. Let an egocentric video stream be denoted as $V = \{x_\tau\}_{\tau=1}^T$, where x_τ is the τ -th incoming frame. Our goal is to segment the stream into the camera wearer’s atomic actions in an on-line manner. An action instance is represented as $\psi_m = (t_m^s, t_m^e, d_m)$, where (t_m^s, t_m^e) is its temporal span and d_m is a natural-language description. The target segmentation is the set of temporal spans

$$\Phi = \{\phi_m\}_{m=1}^M = \{(t_m^s, t_m^e)\}_{m=1}^M. \quad (1)$$

Following On-TAL convention (Kang et al., 2021; Kim et al., 2022b; Song et al., 2024; Reza et al., 2024), at time τ , an online segmenter can access only frames up to x_τ and must emit an action segment once its end is detected, without future frames or retrospective modification. We focus on “class-agnostic” segmentation, where action boundaries are detected without assuming a predefined action taxonomy. Following class-agnostic On-TAL (Kang et al., 2021, 2024, 2025) convention, the model processes streaming video and predicts only the start and end times of generic action instances.

Atomic Action Granularity. We do not introduce a new ontology of atomicity. Instead, we adopt the atomic-action granularity used in established egocentric and procedural video benchmarks, including EPIC-Kitchens-100 (Damen et al., 2020), GTEA (Fathi et al., 2011), and Ego-Exo4D (Grauman et al., 2024). In these datasets, annotated instances correspond to short, low-level physical interactions performed by the actor, such as *pick up knife* or *open fridge* in EPIC-Kitchens-100, *close water bottle* or *pour coffee* in GTEA, and *tighten bolt* or *remove wheel* in Ego-Exo4D. Although their vocabularies differ, these annotations share a common operational criterion: each segment captures a concrete interaction with relatively localized temporal boundaries, and longer activities can be composed from such units. We use *atomic action* to refer to this established annotation granularity throughout the paper.

Annotation-free Training. Given unlabeled training videos $\mathcal{V}_{\text{train}} = \{V_i\}_{i=1}^{N_{\text{train}}}$ and test videos $\mathcal{V}_{\text{test}} = \{V_j\}_{j=1}^{N_{\text{test}}}$, we assume no human temporal annotations are available for training. Ground-truth spans are used only for evaluation. The final model predicts $\hat{\Phi}^{(j)}$ for each test video V_j in a fully online manner.

3.2 CoTT Overview

CoTT is an annotation-free framework that combines structured LVLM annotation with iterative streaming distillation. It consists of three components. First, **CoTT-Labeler** uses an open-source LVLM to generate initial dense pseudo-labels from unlabeled videos. Second, **CoTT-Lite** distills these pseudo-labels into a lightweight online segmenter. Third, CoTT iteratively bootstraps CoTT-Lite predictions to obtain temporally regularized pseudo-labels for the next training round.

Formally, CoTT-Labeler produces initial pseudo action instances

$$\tilde{\Psi}_i^{(0)} = \{(\tilde{t}_m^s, \tilde{t}_m^e, \tilde{d}_m)\}_{m=1}^{\tilde{M}_i} \quad (2)$$

for each training video V_i . The corresponding spans

$$\tilde{\Phi}_i^{(0)} = \{(\tilde{t}_m^s, \tilde{t}_m^e)\}_{m=1}^{\tilde{M}_i} \quad (3)$$

are used to train the first CoTT-Lite model. After each training round, CoTT-Lite is run over $\mathcal{V}_{\text{train}}$, and its predictions become pseudo-labels for the next round.

3.3 CoTT-Labeler: Structured LVLM Annotation Pipeline

Directly using LVLMs for dense atomic action annotation is unreliable. Whole-video prompting often produces coarse summaries instead of exhaustive fine-grained actions. Sliding-window prompting instead splits the video into short clips and queries each clip independently; it provides finer local evidence, but often fragments, duplicates, or misaligns actions that cross clip boundaries. (Figure 3) CoTT-Labeler avoids this issue by separating local visual evidence extraction from global action formation.

Stage 1: Transition Captioning. We split a long video into overlapping clips of K sampled frames. For each clip, the LVLM does not predict complete action instances. Instead, it describes what changes between consecutive frames. Given sampled frames $\{x_{\tau_\ell}\}_{\ell=1}^K$, the LVLM outputs transition captions

$$\mathcal{D}^{\text{trans}} = \{(\tau_\ell, \tau_{\ell+1}, d_\ell^{\text{trans}}, s_\ell)\}_{\ell=1}^{K-1}, \quad (4)$$

where d_ℓ^{trans} describes the visible physical interaction and $s_\ell \in \{\text{start}, \text{continue}, \text{end}, \text{none}\}$ indicates its transition state. Since Stage 1 extracts only local transition evidence, it does not need to decide final action boundaries inside each window.

Stage 2: Action Grouping. We concatenate transition captions across the video and use an LLM to group consecutive transitions into primitive atomic action instances. Grouping is guided by both semantic consistency in d_ℓ^{trans} and transition states s_ℓ . Because this step is language-only, it can use a much longer temporal context than the LVLM clip window. The output is a set of primitive proposals

$$\hat{\Psi} = \{(\hat{t}_m^s, \hat{t}_m^e, \hat{d}_m)\}_{m=1}^{\hat{M}}. \quad (5)$$

Stage 3: Refine Action Instances. For each primitive proposal, we pad its temporal span by a short margin and re-feed the corresponding frames to the LVLM together with the proposal description. The LVLM verifies the visual evidence, refines the start and end times, and produces a concise action description. This yields the initial pseudo-label set $\tilde{\Psi}^{(0)}$. When a class vocabulary is provided, the same stage can be used for class-conditioned zero-shot temporal action localization.

3.4 CoTT-Lite: Streaming Distillation

CoTT-Labeler provides dense pseudo-labels, but it is offline and requires multiple LVLM calls. We therefore distill its pseudo-label spans $\tilde{\Phi}^{(r)}$ into CoTT-Lite, a lightweight causal online model that requires no LVLM calls at deployment. Inspired from OpenHOUSE (Kang et al., 2025), CoTT-Lite uses an actionness head to detect action starts and a progression head to detect action ends via progression drops. This provides a temporal inductive bias well suited to dense atomic actions, where consecutive interactions often have only short gaps.

At inference time, CoTT-Lite processes frames sequentially and emits an action segment once its end is detected, without accessing future frames. In our framework, CoTT-Labeler provides semantic coverage through LVLM-generated pseudo-labels, while CoTT-Lite transfers them into an efficient online model.

Iterative bootstrapping. We further connect this temporal inductive bias to bootstrapping. LVLM-generated pseudo-labels can be semantically meaningful but temporally noisy due to sparse visual sampling and LVLM uncertainty. After training on these labels, CoTT-Lite can produce temporally more coherent predictions by regularizing starts and ends through actionness and progression. At bootstrapping round r , we train CoTT-Lite $S_\theta^{(r+1)}$ using pseudo-labels $\{\tilde{\Phi}_i^{(r)}\}_{i=1}^{N_{\text{train}}}$, then run it on the

same unlabeled training videos to obtain

$$\tilde{\Phi}_i^{(r+1)} = S_{\theta}^{(r+1)}(V_i). \quad (6)$$

These predictions supervise the next round. After R rounds, the final model $S_{\theta}^{(R)}$ is evaluated online on test videos.

4 Experiments

We evaluate CoTT under four aspects: large-scale egocentric atomic action segmentation, iterative streaming distillation, cross-dataset transfer, and zero-shot temporal action localization.

4.1 Experimental Setup

Datasets. Our main experiments use two large-scale egocentric benchmarks with ground-truth atomic action annotations: EPIC-Kitchens-100 (Damen et al., 2020) (EK100) and the ego-view subset of Ego-Exo4D (Grauman et al., 2024) (EgEx). These datasets allow us to test both scalability and supervision quality: their scale evaluates whether CoTT can generate supervision for long and diverse egocentric videos, while their human annotated ground-truth boundaries provide a *supervised upper bound* for measuring how closely annotation-free training approaches GT-trained models. EK100 contains over 100 hours of footage from 700 cooking sessions, and EgEx contains approximately 68.4 hours of video with 37,932 atomic action annotations after de-duplication. We use GTEA (Fathi et al., 2011) as a smaller controlled dataset for CoTT-Labeler ablations and cross-dataset transfer. Finally, to enable direct comparison with prior zero-shot TAL methods (Liberatori et al., 2024; Han et al., 2025a; Ju et al., 2022), we evaluate CoTT-Labeler on THUMOS14 (Jiang et al., 2014) as a class-conditioned zero-shot temporal action localizer.

Training protocol. For EK100, EgEx, and GTEA, all training videos are treated as unlabeled for CoTT. CoTT-Labeler first generates pseudo-labels on the training split, and CoTT-Lite is trained only from these pseudo-labels. Ground-truth temporal annotations are used only for evaluation, except for GT-trained upper-bound models and supervised baselines. For cross-dataset transfer, we train CoTT-Lite on EgEx pseudo-labels and evaluate it directly on the GTEA test split.

Metrics. For egocentric atomic action segmentation, we report Hungarian F1 (Kuhn, 1955), precision, and recall at tIoU=0.5, following class-agnostic On-TAL evaluation (Kang et al., 2021,

2024, 2025). Predictions are matched to ground-truth segments by maximizing temporal IoU, and a match is counted as correct if it exceeds the tIoU threshold. For THUMOS14, we follow the zero-shot TAL protocol (Liberatori et al., 2024; Han et al., 2025a) and report average mAP over tIoU thresholds 0.3–0.7.

Direct LVLM baselines. We define two direct prompting baselines using the same output format as CoTT-Labeler. Baseline_{whole} prompts the LVLM on the entire video and asks it to enumerate all action instances; frames are sampled at 2 fps, so the model receives the full temporal extent of the video. Baseline_{slide} splits the video into short clips and asks the LVLM to localize action instances inside each clip independently; we use the same clip setting as CoTT-Labeler Stage 1, with $K=12$ frames sampled at 2 fps for the fair comparison.

Implementation. For CoTT-Labeler, we use Qwen2.5VL-32B (Team, 2025b) for LVLM/LLM inference unless stated otherwise. Stage 1 uses clips of $K = 12$ sampled frames at 2 fps. Stage 3 re-evaluates each primitive proposal using 12 sampled frames from its padded temporal region. For CoTT-Lite, we follow the dual-head recurrent architecture of (Kang et al., 2025), with an actionness head and a progression head. We also report a GT-trained upper bound, which uses the same architecture trained with ground-truth temporal labels. For class-conditioned zero-shot TAL on THUMOS14, target action classes and descriptions are provided in the prompt. For all class-agnostic experiments, we use *identical prompts* across datasets without dataset-specific tuning. All LVLM stages process at most 12 frames per call and run on a single NVIDIA H100 GPU; exact prompts, architecture details, and hyperparameters are provided in the appendix.

Efficiency. CoTT-Labeler is fully parallelizable across videos and segments. With 8 H100 GPUs, its wall-clock runtime is comparable to video duration. CoTT-Lite runs at over 500 FPS and requires no LVLM calls at deployment. For reference, exhaustive human annotation of atomic action boundaries requires roughly $3\times$ video duration in our empirical estimate.

4.2 Large-Scale Egocentric Atomic Action Segmentation

Tables 1 and 2 show the main results on EK100 and EgEx. CoTT-Labeler already provides useful pseudo-labels without human annotations. Distill-

EK100 (Damen et al., 2020) (tIoU=0.5)					
Method	Prog.	GT	F1	Prec.	Rec.
OAD-Grouping (Kang et al., 2021)	–	✓	21.4	25.5	18.4
CAG-QIL (Kang et al., 2021)	–	✓	23.1	21.3	25.2
SimOn (Tang et al., 2022)	–	✓	4.3	2.3	33.4
OAT (Kim et al., 2022a)	–	✓	27.5	17.5	63.7
ActionSwitch (Kang et al., 2024)	–	✓	32.4	29.8	35.5
CoTT-Labeler	–	×	31.1	31.8	30.5
CoTT-Lite (w/o Prog.)	×	×	30.4	39.8	24.6
CoTT-Lite (round 1)	✓	×	37.4	45.0	31.9
CoTT-Lite (round 2)	✓	×	39.2	36.9	41.9
GT-trained (Kang et al., 2025)	✓	✓	48.5	46.7	50.4

Table 1: Main results on EK100. “Prog.” denotes progression head; “GT” denotes human temporal labels.

EgEx (Grauman et al., 2024) (tIoU=0.5)					
Method	Prog.	GT	F1	Prec.	Rec.
ActionSwitch [†] (Kang et al., 2024)	–	✓	18.7	25.0	15.0
MAT [†] (Song et al., 2024)	–	✓	20.3	18.0	23.4
OAT [†] (Kim et al., 2022a)	–	✓	31.7	36.1	28.3
CoTT-Labeler	–	×	23.1	23.9	22.5
CoTT-Lite (w/o Prog.)	×	×	20.5	24.0	18.0
CoTT-Lite (round 1)	✓	×	25.5	31.3	21.5
CoTT-Lite (round 2)	✓	×	28.8	31.0	26.9
GT-trained (Kang et al., 2025)	✓	✓	34.3	34.1	34.5

Table 2: Main results on EgEx. [†] indicates our implementation.

ing these labels into CoTT-Lite yields a streaming segmenter, and iterative bootstrapping further improves performance. On EK100, CoTT-Lite improves from 37.4 F1 after the first bootstrapping round to 39.2 after the second round. On EgEx, the same trend holds, improving from 25.5 to 28.8 F1. The final bootstrapped model closes a substantial portion of the gap to GT-trained models and is competitive with fully supervised On-TAL baselines.

Why does bootstrapping help? LVLM-generated labels are useful but not final: CoTT-Labeler identifies meaningful atomic interactions, but its boundaries can be noisy due to sparse frame sampling and LVLM uncertainty. CoTT-Lite inherits a temporal inductive bias from its actionness and progression heads (Kang et al., 2025), where action starts are detected by actionness and action ends by progression drops. This bias regularizes noisy pseudo-labels into smoother temporal segments, as shown in Figure 5. The progression-head ablation isolates this effect: without progression modeling, CoTT-Lite

underperforms CoTT-Labeler, while progression-based boundary modeling allows CoTT-Lite to surpass the initial pseudo-labels and improve further through a second bootstrapping round.

4.3 Ablation of CoTT-Labeler

We next analyze the structured LVLM annotation pipeline on GTEA in Figure 1 and Table 3.

Direct LVLM baselines CoTT-Labeler substantially improves performance compared to direct prompting baselines (Figure 1).

Transition captioning. To evaluate the role of contextual transition modeling, we compare our default setup with a variant that captions each frame independently before grouping them into actions. Removing transition captioning causes a large drop, showing that frame-level descriptions are not a stable basis for grouping. Transition captions provide essential cues that help the LLM form coherent action instances.

Boundary refinement and LVLM backbone. Re-

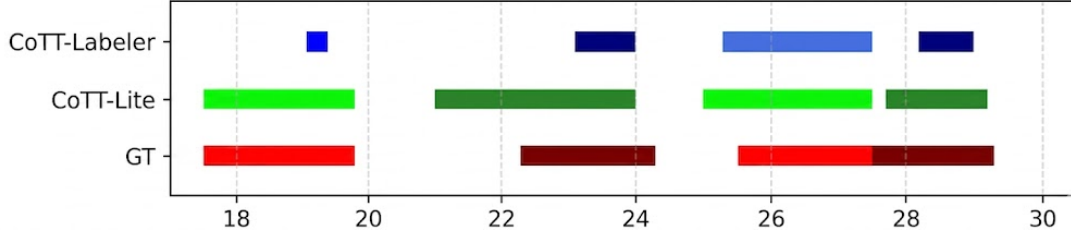


Figure 5: Effect of streaming distillation on EK100. CoTT-Labeler (blue) capture relevant interactions but can be temporally coarse or fragmented. CoTT-Lite predictions (green) benefit from progression-based temporal modeling, producing more coherent boundaries that better align with ground truth (red).

LVL	Trans.	Refine	F1	Prec.	Rec.
Qwen2.5	×	×	5.6	3.5	13.8
	×	✓	19.0	11.3	59.0
	✓	×	43.0	35.9	53.6
	✓	✓	44.2	36.9	55.1
Qwen3	✓	✓	45.6	34.6	66.7

Table 3: CoTT-Labeler ablation on GTEA at tIoU=0.5. “Trans.” denotes transition captioning across consecutive frames; without it, the model uses frame-level descriptions before grouping.

Training data	F1	Prec.	Rec.
GTEA (Fathi et al., 2011)	43.7	45.9	41.7
EgEx (Grauman et al., 2024)	41.8	60.9	31.9

Table 4: Cross-dataset validation on GTEA at tIoU=0.5. The EgEx row is trained only on EgEx pseudo-labels and evaluated directly on GTEA.

finement improves boundary quality by revisiting each proposal with visual evidence. A stronger LVL further improves performance, especially recall, indicating that CoTT-Labeler benefits from better LVL temporal reasoning.

4.4 Cross-Dataset Validation

To test whether the learned model transfers across egocentric domains, we train CoTT-Lite on EgEx pseudo-labels and evaluate it directly on GTEA. Table 4 shows that the EgEx-trained model performs comparably to the model trained on GTEA pseudo-labels. This suggests that CoTT pseudo-labels capture a transferable atomic-action granularity rather than overfitting to a single dataset.

4.5 Class-Conditioned Zero-Shot TAL

Although our main task is streaming egocentric atomic action segmentation, CoTT-Labeler also supports class-conditioned zero-shot Temporal Ac-

Method	Train	TTA	Avg. mAP
EffPrompt (Ju et al., 2022)	×	×	3.6
STALE (Nag et al., 2022)	✓	×	0.7
T3AL-0 (Liberatori et al., 2024)	×	×	4.8
T3AL (Liberatori et al., 2024)	×	✓	10.4
ZEAL (Aklilu et al., 2025)	×	×	10.7
FreeZAD (Han et al., 2025a)	×	×	9.9
AdaZAD (Han et al., 2025a)	×	✓	14.1
Baseline _{whole}	×	×	1.0
Baseline _{slide}	×	×	9.1
CoTT-Labeler	×	×	15.6

Table 5: Class-conditioned zero-shot TAL on THUMOS14. Avg. mAP is averaged over tIoU thresholds 0.3–0.7. TTA refers to Test-Time Adaptation.

tion Localization (TAL) (Han et al., 2025a; Liberatori et al., 2024): given a target action vocabulary, it localizes all action instances in an untrimmed video without benchmark-specific training. On THUMOS14, we provide class names and descriptions in the prompt and run CoTT-Labeler without training or Test-Time Adaptation (TTA). Table 5 reports average mAP over tIoU thresholds 0.3–0.7. CoTT-Labeler outperforms prior training-free methods and even surpasses TTA-based methods, while Baseline_{whole} and Baseline_{slide} remain limited by coarse summarization and boundary fragmentation, respectively.

5 Conclusion

In this work, we take a step toward building a streaming atomic action segmenter by extracting the action understanding capability of an LVL through CoTT and transferring it to a lightweight online model, CoTT-Lite. This demonstrates a practical pathway to bring LVL-level temporal reasoning outcome into efficient, real-time streaming video understanding systems.

6 Limitations

Dependence on LVLM. Our pipeline fundamentally relies on the reasoning ability of the LVLM used in CoTT-Labeler. While large models such as Qwen2.5VL-32B and Qwen3VL-32B reliably follow our transition-captioning and refinement instructions, we observed that smaller models (e.g., 7B-scale LVLMs) frequently fail even at basic instruction following, producing incomplete or inconsistent transitions. This directly affects the quality of the pseudo-labels and, consequently, the distilled performance of CoTT-Lite. In this sense, our framework inherits the limitations of the underlying LVLM. However, the opposite also holds: as LVLMs continue to improve in temporal reasoning, cross-frame consistency, and grounded understanding, the quality of CoTT-Labeler is expected to rise without any algorithmic change. Thus, the ceiling of our method is closely tied to progress in LVLM architectures, suggesting that future LVLM advancements will naturally translate into stronger pseudo-labels and better streaming models.

Beyond Streaming. Although we develop CoTT primarily for streaming atomic action segmentation, the refined atomic segments paired with grounded descriptions naturally extend to broader video-language tasks. The outputs can serve as fine-grained units for grounded video QA, temporal video-text alignment, or open-world event retrieval. Exploring these applications is beyond the scope of this work as our goal is enabling streaming atomic action segmentation without human annotation, but represents a promising direction for leveraging CoTT as a general-purpose video-language annotation framework.

Proprietary LVLMs. Our experiments use open-source LVLMs rather than proprietary frontier models such as GPT-5 (Team, 2025a) or Gemini (Team, 2023). Evaluating CoTT with such models would be valuable, as stronger visual reasoning and temporal grounding may improve pseudo-label quality. However, large-scale egocentric atomic action segmentation requires exhaustive video-level or clip-level annotation over many hours of footage, resulting in a prohibitively large number of API calls. In our cost estimate, even the basic transition-captioning stage alone would cost roughly \$3K per 100 hours of video with proprietary APIs, before accounting for grouping, refinement, or iterative experimentation. This makes proprietary-model eval-

uation difficult to scale and hard to reproduce. We therefore focus on locally deployable open-source LVLMs, which better match our goal of scalable and reproducible annotation-free training.

References

- Josiah Aklilu, Xiaohan Wang, and Serena Yeung-Levy. 2025. Zero-shot action localization via the confidence of large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Jounghbin An, Hyolim Kang, Su Ho Han, Ming-Hsuan Yang, and Seon Joo Kim. 2023. Miniroad: Minimal rnn framework for online action detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Anurag Arnab, Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. 2025. Temporal chain of thought: Long-video understanding by thinking in frames. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Junwen Chen, Gaurav Mittal, Ye Yu, Yu Kong, and Mei Chen. 2022. Github: Gated history unit with background suppression for online action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Janghoon Cho, Jungsoo Lee, Munawar Hayat, Kyuwoong Hwang, Fatih Porikli, and Sungha Choi. 2026. Floc: Facility location-based efficient visual token compression for long video understanding. In *International Conference on Learning Representations (ICLR)*.
- Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2020. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*.
- Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. 2016. Online action detection. In *European Conference on Computer Vision (ECCV)*.
- Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. 2020. Learning to discriminate information for online action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Jesse Farebrother, Jordi Orbay, Quan Ho Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. 2024. Stop regressing: Training value functions via classification for scalable deep rl. *arXiv preprint arXiv:2403.03950*.
- Alireza Fathi, Xiaofeng Ren, and James M Rehg. 2011. Learning to recognize objects in egocentric activities. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, and 81 others. 2024. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chaolei Han, Hongsong Wang, Xin Gong, and Jie Gui. 2026. Oz-tal: Online zero-shot temporal action localization. *arXiv preprint arXiv:2605.09976*.
- Chaolei Han, Hongsong Wang, Jidong Kuang, Lei Zhang, and Jie Gui. 2025a. Training-free zero-shot temporal action detection with vision-language models. *arXiv preprint arXiv:2501.13795*.
- Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. 2025b. Videospresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ehsan Imani, Kai Luedemann, Sam Scholnick-Hughes, Esraa Elelimy, and Martha White. 2024. Investigating the histogram loss in regression. *arXiv preprint arXiv:2402.13425*.
- Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. THUMOS challenge: Action recognition with a large number of classes. <http://crcv.ucf.edu/THUMOS14/>.
- Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. 2022. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision (ECCV)*.
- Hyungrok Jung, Daneul Kim, Seunggyun Lim, Jeany Son, and Jonghyun Choi. 2025. Online generic event boundary detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Hyolim Kang, Jeongseok Hyun, Joungbin An, Youngjae Yu, and Seon Joo Kim. 2024. Actionswitch: Class-agnostic detection of simultaneous actions in streaming videos. In *European Conference on Computer Vision (ECCV)*.
- Hyolim Kang, Kyungmin Kim, Yumin Ko, and Seon Joo Kim. 2021. Cag-qil: Context-aware actionness grouping via q imitation learning for online temporal action localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Hyolim Kang, Yunsu Park, Youngbeom Yoo, Yeeun Choi, and Seon Joo Kim. 2025. Open-ended hierarchical streaming video understanding with vision language models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Young Hwi Kim, Hyolim Kang, and Seon Joo Kim. 2022a. A sliding window scheme for online temporal action localization. In *European Conference on Computer Vision (ECCV)*.
- Young Hwi Kim, Seonghyeon Nam, and Seon Joo Kim. 2022b. 2pesnet: Towards online processing of temporal action localization. *Pattern Recognition*.
- Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*.
- Benedetta Liberatori, Alessandro Conti, Paolo Rota, Yiming Wang, and Elisa Ricci. 2024. Test-time zero-shot temporal action localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.
- Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. 2022. Zero-shot temporal action detection via vision-language prompting. In *European Conference on Computer Vision (ECCV)*.
- Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. 2023. Improving multimodal datasets with image captioning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Baoqi Pei, Guo Chen, Jilan Xu, Yuping He, Yicheng Liu, Kanghua Pan, Yifei Huang, Yali Wang, Tong Lu, Limin Wang, and Yu Qiao. 2024. Egovideo: Exploring egocentric foundation model and downstream adaptation. *arXiv preprint arXiv:2406.18070*.

- Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multi-modal large language model for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sakib Reza, Yuexi Zhang, Mohsen Moghaddam, and Octavia Camps. 2024. Hat: History-augmented anchor transformer for online temporal action localization. In *European Conference on Computer Vision (ECCV)*.
- Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. 2023. Is a caption worth a thousand images? a study on representation learning. In *International Conference on Learning Representations (ICLR)*.
- Youngkil Song, Dongkeun Kim, Minsu Cho, and Suha Kwak. 2024. Online temporal action localization with memory-augmented transformer. In *European Conference on Computer Vision (ECCV)*.
- Tuan N Tang, Jungin Park, Kwonyoung Kim, and Kwanghoon Sohn. 2022. Simon: A simple framework for online temporal action localization. *arXiv preprint arXiv:2211.04905*.
- Gemini Team. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- OpenAI Team. 2025a. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.
- Qwen Team. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. 2021. Oadtr: Online action detection with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, Xiangnan Fang, Zewen He, Zhenbo Luo, Wenxuan Wang, Junqi Lin, Jian Luan, and Qin Jin. 2025. Time-r1: Post-training large vision language model for temporal video grounding. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. 2019. Temporal recurrent networks for online action detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. 2021. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Le Yang, Junwei Han, and Dingwen Zhang. 2022. Colar: Effective and efficient online action detection by consulting exemplars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. 2024. Timesuite: Improving mllms for long video understanding via grounded tuning. In *International Conference on Learning Representations (ICLR)*.
- Qingsong Zhao, Yi Wang, Jilan Xu, Yinan He, Zifan Song, Limin Wang, Yu Qiao, and Cairong Zhao. 2024. Does video-text pretraining help open-vocabulary online action detection? In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yue Zhao and Philipp Krähenbühl. 2022. Real-time online video detection with temporal smoothing transformers. In *European Conference on Computer Vision (ECCV)*.

A License

Artifact	Type	License
Epic Kitchens 100	Dataset	NCGL 2.0
EgoExo4D	Dataset	Ego-Exo4D License Agreement
GTEA	Dataset	Apache License 2.0
Qwen-VL Series	MLLM	Apache License 2.0

Table 6: **Summary of licenses for key artifacts.** We strictly adhere to the licensing terms and usage policies of all external datasets, models, and tools utilized in our framework.

We use all external datasets, models, and tools only for research and evaluation purposes, in accordance with their original licenses, access conditions, and intended uses. Derived artifacts produced by our framework are used solely within the research context of this work and are not intended for use outside the scope permitted by the corresponding source artifacts.

B Potential Risk

Potential risks are limited. CoTT is designed for offline pseudo-label generation and online action boundary segmentation, not for identifying individuals or making consequential decisions.

C Additional Implementation Details

Vision Encoder. For CoTT-Lite experiments, we follow the recent On-TAL work configuration (Kang et al., 2025), adopting the Egovideo (Pei et al., 2024) encoder. (2FPS) Note that for THU-MOS14, no vision encoder is used; raw frames are directly fed into the LVLM in CoTT experiments.

CoTT-Lite Architecture and Training Configuration. On top of the vision encoder, we employ a 3-layer GRU (Chung et al., 2014) with layer normalization (Ba et al., 2016) between layers for temporal modeling. The GRU hidden dimension is set to 768. Both the actionness head and the progression head consist of a single linear layer, trained with cross-entropy and histogram loss (Farebrother et al., 2024; Imani et al., 2024) respectively, following (Kang et al., 2024, 2025). We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 0.0003 and weight decay of 0.01. The history length is set to 64, and the model is trained for 15 epochs.

D Additional Quantitative and Qualitative Analysis

D.1 Bootstrapping Stability and Pseudo-Label Granularity

A natural concern for any self-training procedure is whether iterative bootstrapping remains stable over more rounds, and whether CoTT-Lite’s gain over its own teacher (CoTT-Labeler) reflects genuine recovery of missed actions rather than an artifact of implicit temporal smoothing. We extend self-bootstrapping to **round 5** on EK100 and additionally report a **GT-tuned smoothing baseline**: the strongest post-hoc combination of segment merging, short-segment dropping, and boundary dilation applied to the Labeler output, with thresholds swept to maximize F1 against GT (an oracle advantage no other row is given). Table 7 reports F1/Precision/Recall together with granularity statistics (segments per video, median/mean duration, coverage %) against human GT for every round. Here, *coverage %* denotes the fraction of total video time occupied by the union of a label source’s predicted spans, i.e., how much of the video is claimed by some predicted segment, irrespective of whether it aligns with a GT segment.

D.2 Cross-Domain Generalization

To probe whether CoTT-Lite learns a transferable notion of atomic-action granularity rather than overfitting to a single dataset’s annotation convention, we evaluate bidirectional cross-dataset transfer against two GT-trained On-TAL baselines, MAT (Song et al., 2024) and OAT (Kang et al., 2025), each trained with ground-truth temporal annotations on the source domain and evaluated zero-shot on the target domain’s test split (class-agnostic F1@0.5). This comparison is intentionally not apples-to-apples in our favor: only CoTT-Lite is trained without any human temporal annotation.

D.3 Vocabulary-Coverage Mismatch with Closed-Set GT

A residual gap remains between CoTT-Lite and GT-trained supervised upper bounds. Inspecting disagreements, we find this gap is partly explained by a mismatch between what GT annotates and what CoTT targets: GT is drawn from each dataset’s closed annotation vocabulary, whereas CoTT targets generic atomic actions defined by a single, dataset-agnostic system prompt. We frequently observe CoTT correctly localizing legitimate hand-

Label source	F1	Prec.	Rec.	#pred seg	seg/video	median dur	mean dur (s)	coverage %
GT (human)	–	–	–	–	75.6	42	3.71	78.9
Round 0 (CoTT-Labeler)	31.2	31.9	30.5	9,254	67.1	41	3.92	67.4
Labeler + GT-tuned smoothing	34.4	36.0	32.8	8,803	63.8	–	–	–
Round 1 (CoTT-Lite)	37.4	45.1	32.0	6,860	49.7	29	5.62	78.1
Round 2 (CoTT-Lite, peak)	39.2	36.9	41.9	10,981	79.6	49	3.41	75.7
Round 3	38.0	38.7	37.4	9,336	67.7	40	4.08	77.3
Round 4	35.7	31.3	41.5	12,841	93.1	56	2.90	75.7
Round 5	32.9	27.9	40.2	13,932	101.0	64	2.72	77.5

Table 7: **Round-wise evolution of pseudo-labels on EK100.** Performance peaks at round 2 and degrades *gradually* thereafter through over-segmentation (#pred seg and seg/video keep growing while mean duration shrinks), not catastrophic collapse. Crucially, even an oracle-tuned smoothing baseline over the Labeler reaches only 34.4 F1, well below CoTT-Lite’s 39.2: smoothing raises recall only to 32.8 (it can only reshape segments the Labeler already emitted), whereas CoTT-Lite reaches 41.9 by recovering actions the Labeler missed. Round 2 also best re-aligns pseudo-label granularity with human GT statistics (79.6 seg/video, 3.41s mean duration vs. GT’s 75.6/3.71s).

Direction	Model	F1	Prec.	Rec.
EgoExo4D→EK100	CoTT-Lite	35.2	40.5	31.1
	MAT (Song et al., 2024)	29.5	23.7	39.2
	OAT (Kim et al., 2022a)	27.4	31.9	24.1
EK100→GTEA	CoTT-Lite	42.7	51.6	36.4
	OAT (Kim et al., 2022a)	37.5	30.6	48.5
	MAT (Song et al., 2024)	24.5	15.9	53.8

Table 8: **Bidirectional cross-domain transfer.** CoTT-Lite achieves the best F1 in both directions, including the large-scale EgoExo4D→EK100 setting, despite being the only method trained without ground-truth temporal annotations.

object interactions that are simply absent from the closed-set vocabulary, rather than annotation noise. Consistent with this, the gap to the GT-trained score is smaller on the free-form, open-vocabulary EgoExo4D (84.0% at round 2) than on the stricter closed-vocabulary EK100 (80.8%): the more open the GT vocabulary, the smaller the gap.

As a related probe, we measure whether supplying the target closed vocabulary to CoTT-Labeler narrows this gap. On GTEA, providing GT action classes (class-conditioned) improves localization over the class-agnostic setting at all tIoU thresholds (F1: 48.5 vs. 44.3 at tIoU 0.5; 37.0 vs. 33.6 at 0.6; 25.6 vs. 21.0 at 0.7). However, with larger-vocabulary datasets such as EPIC-Kitchens, whose verb/noun combinations span hundreds of classes, the LVLM frequently invents its own vocabulary rather than adhering strictly to the provided list, so this class-conditioned variant does not currently scale and is not part of our main contribution; we view large-vocabulary class-conditioned localization as a direction for future work.

D.4 Robustness against Linguistic Variance and Status Noise

To validate how disentangling local evidence extraction (**Stage 1**) from atomic action grouping (**Stage 2**) in CoTT-Labeler enhances the robustness of our pipeline, we conducted a targeted quantitative analysis. We hypothesize that this architectural separation mitigates the boundary artifacts inherent in sliding-window approaches. Specifically, we posit that Stage 2 can effectively merge transition captions based on semantic consistency and temporal context, even when facing linguistic variations or noisy status cues. To test this, we performed a controlled experiment designed to isolate the grouping stage.

Stage 1 Output Generation. We constructed synthetic trials to simulate realistic Stage 1 outputs. Each trial consists of a 120-second sequence of 240 transition captions (0.5s per bin). Within each trial, we sampled 32 non-overlapping action segments with random durations ranging from 1.5 to 3.0s (3–6 bins), separated by random gaps. To mimic the textual interface of our pipeline, we utilized a template-filling mechanism to convert these intervals into records following the format: “Between t_{start} and t_{end} : Description: {text} Status: {flag}”. Half of the segments were designated as *Normal* (consistent inputs), while the other half were designated as *Variance* (perturbed inputs).

Controlled Paraphrase and Status Jittering To mimic real-world instability, Variance segments were subjected to two types of perturbations:

- **Linguistic Shift:** We enforced a mid-action description change, where the first half of the bins used one linguistic formulation (e.g.,

chopping the herbs), and the second half transitioned to a distinct alternative formulation (e.g., mince the herbs), while maintaining the same semantic.

- **Status Jittering:** We applied probabilistic jittering to the Status field within the logs. This simulates the noise often found in transition-level predictions, testing whether the framework can maintain continuity despite intermittent status inconsistencies. Normal segments maintained consistent descriptions and stable status flags across all consecutive bins.

Stage 2 Grouping Evaluation. We then applied Stage 2 to these synthetic textual records using the exact grouping prompt and decoding configuration from our main experiments. The goal was to recover the ground-truth structure solely from the noisy text logs. The output was a JSON-formatted list of grouped interactions.

Metrics and Results We evaluated performance using two recovery metrics with a 0.1s boundary tolerance: (i) **Total Action Recovery:** the fraction of all ground-truth actions correctly identified by a predicted group. (ii) **Variance Recovery:** the recovery rate specifically for Variance segments, quantifying robustness to both paraphrasing and status noise.

Across 96 trials, Stage 2 demonstrated strong resilience, achieving a Total Action Recovery of **94.40%** and a Variance Recovery of **92.97%**. These results empirically support our claim that the Stage 2 grouping mechanism is robust not only to realistic wording changes but also to noisy status estimates, successfully merging fragmented evidence into coherent atomic interactions

D.5 Qualitative Analysis

Figure 6 presents a representative example illustrating how Stage 2 grouping consolidates noisy, interval-wise evidence into coherent atomic interactions. In the inference pipeline, Stage 1 generates captions via a sliding window mechanism over the video stream, inducing paraphrastic variation across boundaries (e.g., “pour olive oil into the pan” vs. “drizzle olive oil into the pan”) even when the underlying atomic action remains continuous. Despite such surface-form inconsistencies, Stage 2 correctly merges semantically consistent transition captions into a single non-overlapping atomic instance, preventing over-segmentation. This qualitative behavior aligns with our quantitative robust-

Stage 1: Interval captions (example)

```
Between 2.50 and 3.00: **Description:**
    Begin chopping the herbs. **Status:**
    Starting within this interval
Between 3.00 and 3.50: **Description:**
    Begin chopping the herbs. **Status:**
    Continuing within this interval
Between 3.50 and 4.00: **Description:**
    Finely mince the herbs. **Status:**
    Continuing within this interval
Between 4.00 and 4.50: **Description:**
    Finely mince the herbs. **Status:**
    Continuing within this interval
Between 4.50 and 5.00: **Description:**
    Finely mince the herbs. **Status:**
    Ending within this interval
...
Between 8.50 and 9.00: **Description:** Pour
    olive oil into the pan. **Status:**
    Starting within this interval
Between 9.00 and 9.50: **Description:**
    Drizzle olive oil into the pan. **Status:**
    Continuing within this interval
Between 9.50 and 10.00: **Description:**
    Drizzle olive oil into the pan. **Status:**
    Ending within this interval
```

Stage 2: Grouped atomic actions

```
{
  "intervals": [
    {"start": 2.50, "end": 5.00, "
    description": "<start>mince the herbs<
    end>"},
    ...
    {"start": 8.50, "end": 10.00, "
    description": "<start>Pour olive oil<end
    >"},
    ...
  ]
}
```

Figure 6: **Qualitative example of linguistic-variance-robust grouping.** Stage 1 captions may paraphrase the same ongoing interaction (e.g., “pour” vs. “drizzle”). Stage 2 merges semantically consistent adjacent intervals into a single atomic interaction, avoiding over-segmentation.

ness results (Sec. D), confirming that the grouping stage effectively handles the linguistic variance inherent in streaming inference.

Sliding-window trade-off. Enlarging Stage 1’s sliding window reduces how often actions cross a window boundary, shrinking the boundary-fragmentation effect illustrated in Fig. 6. This relief comes at the cost of an opposing failure mode, however: as the window grows toward the full video, sliding-window prompting approaches whole-video prompting, which collapses into a coarse summary of only the most salient events rather than exhaustively enumerating densely packed atomic actions (Sec. 1, Table 5). CoTT-Labeler sidesteps this trade-off structurally instead of tuning a single window size: Stage 1 extracts only local transition evidence (no per-window action decisions), while Stage 2 forms complete action instances using whole-video language context, decoupling boundary formation from the window grid.

D.6 Deployment Efficiency Comparison

A single throughput number (FPS) is insufficient to characterize deployment cost, since it does not reflect latency, memory, or the number of LVLM calls. We therefore report a direct comparison at the deployment level between CoTT-Lite and the two LVLM prompting strategies it distills from (LVLM: Qwen2.5-VL-32B), measured on a single GPU for a 1-minute egocentric video (2 fps, 120 frames). We report *response latency* (time-to-decision), i.e., mandatory input buffering plus inference, the metric that matters for streaming: the delay from when an action occurs to when the system can emit a decision.

Method	Params	LVLM calls	Response latency	Streaming
Whole-video LVLM	32.5B	1	88.9 s (60 s buffer + 28.9 s)	✗
Sliding-window (K=12)	32.5B	10	9.2 s (6 s buffer + 3.2 s)	✗
CoTT-Lite	1.1B	0	59.9 ms (0 buffer + 59.9 ms)	✓

Table 9: **Deployment-time efficiency.** CoTT-Lite responds $\sim 1,500\times$ faster than whole-video and $\sim 150\times$ faster than sliding-window LVLM prompting. The dominant cost of the LVLM baselines is *mandatory buffering*, a hardware-independent wall-clock wait that no GPU upgrade removes; CoTT-Lite is causal and emits a decision the instant a frame arrives, with $30\times$ fewer parameters, no LVLM calls, and 2.1 GB vs. 20–24 GB peak GPU memory.

E Discussion: Semantic-Aware Streaming Understanding

A distinctive property of CoTT is that the final output of the pipeline is not just a set of temporal boundaries, but a collection of *atomic action segments paired with natural language descriptions*. After the grouping stage, each primitive instance is refined by the LVLM, which jointly adjusts its start/end timestamps and produces a concise segment-level description (Algorithm 1). As a result, each atomic action segment is associated with a human-readable summary such as “open the fridge door” or “put container in fridge,” rather than being represented only by its temporal span.

In the current work, however, these segment-level descriptions are used solely in the *offline* CoTT pipeline to obtain higher-quality pseudo-labels. Once the pseudo-labels are generated, CoTT-Lite is trained using only the start and end timestamps, without consuming any of the accompanying textual summaries. The online model effectively sees a supervision signal of the form

$$\{(t_m^{\text{start}}, t_m^{\text{end}})\}_{m=1}^M,$$

while the corresponding descriptions

$$\{d_m\}_{m=1}^M$$

are discarded during distillation. This is a deliberate design choice to keep the streaming model *class-agnostic*, lightweight, and free of any predefined vocabulary, but it also creates a strong information bottleneck: rich semantic signals present in the LVLM-refined labels are not propagated into the streaming regime.

We argue that these segment-level descriptions constitute a highly valuable resource for future work. They implicitly encode object identities, interaction types, temporal ordering, and even paraphrastic variation across instances. Leveraging this information would enable a shift from “purely class-agnostic boundary prediction” to *semantic-aware streaming atomic action segmentation*. For example, one could (i) distill embeddings of the descriptions into the online model, (ii) train an auxiliary head that predicts a textual representation or retrieves a matching description, or (iii) learn a joint space where streaming features are aligned with both temporal boundaries and segment-level language. Such extensions would preserve the open-vocabulary nature of our framework while enabling

the model to reason about “what” is happening, not only “when” actions start and end.

On possible LVLM training-data contamination. Since the LVLM used in CoTT-Labeler may have seen some of our evaluation videos during its own pretraining, one could argue that our results partly reflect memorization rather than generalization. Two observations partially mitigate this concern: (i) the LVLM is used only to produce *dense temporal boundaries*, a task it is not trained for and performs poorly at directly (whole-video/sliding-window prompting alone, Fig. 6, Table 5); and (ii) the strong cross-dataset transfer of CoTT-Lite (Sec. D.2) shows that the distilled model generalizes across distributions rather than memorizing a single benchmark. An “in-the-wild” evaluation on newly collected egocentric videos that the LVLM cannot have been trained on would further validate this generalization claim; we leave this as an important direction for future work.

F Additional Resources

Pseudocode To aid understanding, we include the pseudo-code of CoTT in Algorithm 1.

G Prompt Appendix

Below we provide the prompts used for CoTT. All prompts are included verbatim for reproducibility.

To ensure a fair comparison with CoTT-Labeler, we reuse the overall structure of the `REFINING_INTERVAL_USER_PROMPT` when implementing the **Baseline_{whole}** and **Baseline_{slide}**. Concretely, the instruction to “identify the most prominent physical interaction” is replaced with “provide all physical interactions in the given video,” and the example output is updated accordingly. This retains the same output format and reasoning constraints, allowing us to isolate the effect of CoTT-Labeler’s temporal transition decomposition while avoiding confounding differences in prompt design.

Regarding the experimental settings, we maintain a clear distinction in prompt strategy. In our class-agnostic experiments, we utilize a single, unified prompt across all evaluated datasets without modification. In contrast, the class-aware setting requires a modified system prompt by definition to enforce closed-set constraints. Crucially, this necessitates dataset-specific curation (e.g., explicitly enumerating valid action categories for each benchmark). We highlight that avoiding such dataset-dependent engineering is a primary motivation for

Algorithm 1: CoTT pipeline in a PyTorch style. (Class-agnostic Version)

```
# video: tensor of shape (T,C,H,W),
# untrimmed video frames
# return: refined pseudo action instances,
# e.g., a list of dicts:
# [{"t_start":float, "t_end":float, "desc":str},...]

# sliding_window(video, win_size, stride):
# - input: video (T,C,H,W)
# - output: iterable of short clips
# each clip: tensor (L,C,H,W),
# where L is about win_size
# (typically overlapping in time)

# transition_caption(clip):
# - input: single clip (L,C,H,W)
# - output: list of transition units within the clip,
# e.g., [{"t_start":float, "t_end":float,
# "text":str, "tag":str}, ...]
# where tag is in {"start", "middle", "end", "none"}

# group_actions(transitions):
# - input: list of transition units
# from the whole video
# - output: primitive action instances, e.g.,
# [{"t_start":float, "t_end":float, "summary":str},...]

# refine_action(video, act):
# - input: full video tensor + one primitive action
# instance
# - output: refined action (boundary + description),
# same dict schema
# {"t_start":float, "t_end":float, "summary":str,...}

# Stage 1: sliding-window transition captions (LVLM)
transitions = []
for clip in sliding_window(video, win_size, stride):
    # transition_caption returns a list of transition
    # units for this clip
    clip_trans = transition_caption(clip)
    transitions += clip_trans

# Stage 2: group transitions into primitive actions
# (LLM-based)
primitive_actions = group_actions(transitions)

# Stage 3: refine each primitive action instance
# (LVLM-guided)
refined = []
for act in primitive_actions:
    refined.append(refine_action(video, act))

return refined
```

prioritizing the class-agnostic approach as our main contribution.

H AI Usage

We utilized GPT5 (Team, 2025a) to fix typos and polish sentences, and Claude Code for minor code revisions.

System Prompt

```

SYSTEM_PROMPT = \
'''
You are an egocentric video analyzer,
specializing in detecting physical interactions
between the camera wearer's hands and
objects.

Your task is to output all temporal intervals in the
format:

[<start_time>, <end_time>, "<interaction_description
>"]

where <interaction_description> is a concise, single
-sentence description of the physical
interaction.

## Interaction Description Rules

- Must describe only physical interactions with
objects.
- Use short, concrete, deterministic phrases.
- Do not use forms like "camera wearer is ...".
Instead, write the action directly.
- Do not use conjunctions ("and", "/", etc.).
Each description must contain exactly one
action.

## Examples
1. "spread jam on bread"
2. "pour water into a cup"
3. "stir coffee with a spoon"
4. "take spoon"
5. "put the lunch box into the bag"
6. "place the cup on the table"
7. "take the book out of the shelf"
8. "hold the cup"
9. "open the drawer"

## Temporal Annotation Guidelines
- Start time = the first frame where the hand makes
physical contact with the object.
- End time = the last frame where the interaction
ends.
- Prioritize accuracy of start and end frames.
- Always focus on the hands of the camera wearer.
'''

```

Figure 7: System prompt used for all stages in class-agnostic CoTT.

System Prompt

```

SYSTEM_PROMPT = \
'''
Here are the target action classes and their
detailed descriptions.
Pay close attention to the start and end conditions
for each action.

Target Action Classes:
- Shotput involves an athlete actively
performing a shot put throw,
starting from when they enter the throwing circle
and begin their preparatory movements
(e.g., holding the shot, initial turns/glides),
through the forceful release of the shot,
and concluding with their immediate follow-through
and recovery within or just outside the
throwing circle.
- CleanAndJerk involves a weightlifter
actively performing the clean and jerk lift,
starting from when they approach the barbell and
assume a preparatory stance, through the "
clean" phase
(lifting to the shoulders) and the "jerk" phase (
lifting overhead),
and concluding when the barbell is either
successfully held overhead and then dropped,
or when the lift is failed and the barbell is
dropped,
and the lifter moves away from the immediate
lifting position.
- Billiards involves a person actively playing
billiards,
starting from when they lean over the table with a
cue stick to prepare for a shot,
through the striking of the cue ball, and
concluding when they straighten up after
completing the shot.
Exclude any segments where players are merely
conversing, standing around the table
without playing,
or where the focus is on introductory credits,
static images, or general room views that do
not show active play.

...
'''

```

Figure 8: For THUMOS14, the action class descriptions are explicitly enumerated and appended to the system prompt. We show the descriptions of the first three classes (out of 20) for clarity. This highlights a limitation of the class-aware setting, which necessitates such dataset-specific manual curation to define the search space, in contrast to the scalable, unified prompt of our class-agnostic approach.

Frame-Level Transition Caption Prompt

```
FRAME_ANALYSE_USER_PROMPT = \
'''
Given $number consecutive frames from the video,
which means there are $number_minus_one
adjacent transitions,
provide a succinct description of the physical
interaction
occurring in each of these $number_minus_one
transitions between adjacent frames.

The output format MUST be like:
## Output
```json
{
 "intervals": [
 "Between <t1> and <t2>: <description>",
 "Between <t2> and <t3>: <description>",
 <possibly multiple plain strings of
 description on this list>
]
}
'''
You MUST begin the code block with three backticks
(```) followed immediately by the word json,
and then end the code block with three backticks
(```).
"intervals" here MUST be a list containing
$number_minus_one plain strings
that start with "Between <t1> and <t2>:" and
describe the intervals accordingly in <
description> field.

The <description> must contain these.
- Is there expected physical interaction between
these two frames?
- If yes, specify whether the interaction is:
 - Starting within this interval: The
 initial definitive movements of the
 interaction begin here.
 - Continuing within this interval: The
 interaction was already in progress
 and is still unfolding.
 - Ending within this interval: The final
 definitive movements of the
 interaction conclude here.
- If no, clearly state that no physical
interaction is observed.
- Never use Double Quotation Marks (") or Single
Quotation Marks ('). It brokes the format
.
- Describe interactions using only ONE, LOW-
LEVEL verb.
This means a concrete, low-level physical
action.
Ensure absolute clarity.
Low-Level Verbs can include, but not limited
to actions like
put into, take out, grasp, open, close, etc.

To be specific, each string within the intervals
list must strictly adhere to the following
precise format:

"Between <start_timestamp_of_interval> and <
end_timestamp_of_interval>:
<Content generated for description of the
interaction> <Content generated for status of
the interaction>"

For example, a valid member of the intervals list
would be:

"Between 5.47 and 5.97: Description: Stir coffee
with a spoon. Status: Starting within
this interval."

From now on, timestamps for each frame will be
provided, alternating with the frames
themselves.
'''
```

Figure 9: Prompt used for the transition captioning.

### Interval Grouping Prompt

```
INTERVAL_GROUPING_USER_PROMPT = \
'''
Task: Group Physical Interactions from Interval
Descriptions

You'll receive a list of temporal interval
descriptions detailing the camera wearer's
actions.
Your task is to group these intervals into
complete, non-overlapping physical
interactions.
```

### Key instructions

```
* Single Action Per Interval:
Each grouped interval must represent only one
distinct physical interaction.
This means a new or significantly different action
from the previous one.
* No Splitting:
Do not split a continuous, identical, and adjacent
action into multiple intervals.
* No Conjunctions:
When describing actions, do NOT use "and," commas,
or similar conjunctions.
The description must be a single, concrete, low-
level action.

* Low-level verbs can include, but are not
limited to: put into, take out, grasp,
open, close, pour, etc
* Avoid high-level or abstract verbs such as
: 'interact'. Must be very specific.
```

### Output Specification

```
* No Interactions: If the provided intervals
contain no physical interactions,
return an empty list for "intervals".

```json
{"intervals": []}
'''

* Description Markers: All description strings
MUST begin with <start> and end with <end>

* Format: Your output must be in the
following JSON format
(you MUST begin the code block with three
backticks (```) followed immediately by the
word json,
and then end the code block with three backticks
```).):

```json
{
  "intervals": [
    {
      "start": <start time>,
      "end": <end time>,
      "description": "<start><plain string
that describes the physical
interaction><end>"
    },
    <possibly multiple elements on this list
    >
  ]
}
'''

Here is a list of interval descriptions:
$interval_descriptions
'''
```

Figure 10: Prompt used for grouping transition captions into primitive action instances.

Refinement Prompt

```
REFINING_INTERVAL_USER_PROMPT = \
'''
You are an expert video analyzer, specialized in
    detecting specific physical interactions.

Given $nframes consecutive frames from a $duration-
    second video segment, your task is to:

1. Precisely localize the start and end of
    the most prominent physical interaction.
2. Provide a refined and concrete description.
3. Assign a confidence score (0-1).

You will be given an initial hypothesized
    interaction: $description.

WARNING: This hypothesis may be inaccurate.
Always rely on the actual visual evidence to
    identify the most prominent physical
    interaction.
'''

---
### Key Instructions
#### 1. Exact Localization
- Do NOT label the entire segment as the action.
- The segment contains padding; find the exact
    frame where the action begins and ends.

#### 2. Precise Description
- Revise the initial hypothesis to match actual
    contents.
- Never use Double Quotation Marks or Single
    Quotation Marks. It breaks the format.
- Each description must contain exactly ONE LOW-
    LEVEL verb.
- That verb must describe a concrete, physical
    action such as grasp, put into, take out, open
    , close.
    Other elements like adjectives, nouns, and
    prepositional phrases should be used to
    enrich the description.
- Do not use vague or abstract verbs and nouns.
- Use precise nouns such as yellow purse, coffee mug
    , door handle.

#### 3. Confidence Score
- Provide one float value between 0.0-1.0
    indicating certainty.

---
### Output Format
You must provide both justification and structured
    output.
Start Frame
- Identify exact start frame and explain why it is
    chosen.
End Frame
- Identify exact end frame and explain why it is
    chosen.
Description
- Justify your refined description.
    Ensure a refined description that contains a
    single, concrete verb without any commas,
    slashes, or conjunctions.
    You must explicitly validate that this rule is
    followed.
Final Output
- The final output must follow the given format
    and be enclosed in a fenced code block that begins
    with ```json and ends with ```.
- Description Markers description string MUST
    begin with <start> and end with <end>

```json
{
 "start": <start time>,
 "end": <end time>,
 "description": "<start><refined action
 description><end>",
 "confidence_score": <float>
}

From now on, timestamps for each frame will be
 provided, alternating with the frames
 themselves.
'''
```

Figure 11: Prompt used for refining primitive action instances.