

One Identity, Many Roles: Multimodal Entity Coreference for Enhanced Video Situation Recognition

Supplementary Material

In this section, we first provide a detailed discussion of the metrics (A), followed by qualitative (B) and quantitative analyses (C). We then outline the limitations of our method (D), describe the FINCH clustering algorithm used in the EVC module (E), and present our annotation pipeline and dataset statistics (F).

A. Metrics

LEA [39] is a link-based coreference evaluation metric that weights entities by their importance and measures how well predicted mention clusters align with ground-truth clusters. The final score is computed as an F1 over precision and recall of entity links. In our setup, links are obtained by exact string matching: if two event roles share the same caption, they are treated as belonging to the same cluster.

LEA-Soft [51] extends LEA by incorporating semantic similarity of role captions, measured using CIDEr. This ensures that even when coreference links are correct, poor or incorrect captions reduce the score.

IoU [25] adapts Intersection-over-Union to VidSitu grounding by computing it at the role level and averaging across all semantic roles in the video. For each role, a single predicted box is compared with the ground-truth box in the corresponding frame, and IoU@0.5 counts cases with at least 50% overlap.

HOTA [33], Higher Order Tracking Accuracy, evaluates tracking by jointly considering detection, association, and localization. It balances precision and recall across trajectories, offering a unified measure of overall tracking quality.

B. Qualitative Analysis

MEC in VidSitu is a challenging problem that requires identifying actions, disambiguating roles, clustering entity tracks across shot boundaries with significant visual changes, grouping event roles by their mentions, and finally generating descriptive captions for each entity. The videos, drawn from complex movie scenes with fast motion, shot changes, and diverse contexts, make this even harder. Each video contains five events. Since we obtain YOLO tracks within each shot, our Entity Visual Clustering module associates and clusters these tracks across shots, extending them to full-video entity tracks. These final tracks are paired with event-role captions and entity mentions to provide coherent role assignments throughout the video. It is important to note that the model operates only on sub-sampled frames, but the

clusters can be extended throughout the video with the help of shot level tracks. Each event is represented as a table updated every two seconds. We present four representative samples in [qualitative_results.mp4](#) for full-video visualizations with tracks obtained without any explicit training.

Open-o3 Video results. Open-o3 Video [37] is a recently released open-source framework that performs grounding and object localization in videos using a large vision-language model. It associates textual queries with visual regions and outputs grounded predictions on selected frames. For qualitative comparison, we run the official Open-o3 Video inference pipeline on the same VidSitu samples used in our own qualitative analysis, following the authors’ default configuration.³ For clarity of visualization, we present only the frames that Open-o3 chooses to ground, rather than every frame in the video, allowing its grounding behavior to be viewed more clearly. The resulting visualizations are provided in [openo3_results.mp4](#).

Despite being one of the most recent models for open-world video grounding, Open-o3 Video exhibits several limitations in the context of fine-grained video entity tracking. First, it performs grounding on only a single frame per video, preventing it from capturing entities that appear across multiple moments or scenes. Second, even within that selected frame, the model often detects only a subset of the relevant objects, leading to incomplete grounding. Third, it lacks fine-grained temporal dynamics—capabilities that are core to the VidSitu task that we target. These constraints highlight the gap that remains in current open-source grounded conversation models. Our method directly addresses these challenges by producing dense, fine-grained, and temporally continuous entity tracks across the entire video.

C. Additional Quantitative Results

Multiple metrics. In Tab. 1, we present the results of Cine-MEC across multiple metrics. For brevity, only the primary metrics are reported in the main paper; however, Tab. 6 provides additional results on other commonly used metrics for the SRL task in the VidSitu benchmark. We show clear improvements over prior methods in these metrics as well.

Videos with many entities. Video models struggle with tracking and association of actions and entities as the number of entities increases. To systematically analyze this effect, we construct a subset of validation videos with ≥ 4 entities and report results in Tab. 7. As expected, performance drops

³<https://github.com/marinero4972/Open-o3-Video>

Table 6. CineMEC: Results for SRL on VidSitu Validation Set. R@5: Recall at 5, C: CIDEr, R-L: ROUGE-L, C-Vb: CIDEr scores averaged across verbs, C-Arg: CIDEr scores averaged over arguments.

Method	R@5	C	R-L	C-Vb	C-Arg
VAL SET					
VidSitu-SlowFast [51] CVPR'21	23.38	45.52	42.66	55.47	42.82
OME+OIE [73] AAAI'23	28.72	47.16	40.86	53.96	42.78
HostSG [82] ACMMM'23	29.38	55.09	43.13	64.24	47.68
TypesDev [68] ICMR'25	25.67	90.12	48.08	100.9	81.14
VideoWhisperer [25] NeurIPS'22	25.25	73.73	46.21	82.99	65.52
CineMEC (Ours)	27.14	76.34	46.83	86.01	69.91
Human	-	84.85	39.77	91.7	80.15

Table 7. Comparison of models performance on samples from the validation set with ≥ 4 entities (n=359 of 1324).

Method	Acc@1	CIDEr	LEA	L-Soft	IoU@0.5	HOTA
VideoWhisperer [25]	40.7	60.3	43.3	37.8	37.5	7.1
CineMEC (Ours)	44.9	61.5	46.1	40.1	51.0	31.7

for all methods due to the increased complexity and long-tail nature of these samples. However, CineMEC consistently outperforms GVSR [25] across all metrics. This suggests that CineMEC better enforces entity-centric understanding of videos. Thus, while ERG is still affected by the long-tail problem, it handles such challenging scenarios more effectively than the baseline.

Clustering purity of ERG. We analyze the clustering purity of entity role groups predicted by the ERG module. We first identify *correct* and *wrong* event-roles in each predicted entity cluster (purity 76%). When captions are generated independently (no grouping), CIDEr is (mostly) unaffected: 76.2 and 75.0. However, with predicted grouping, CIDEr for correct roles goes up to 82.3 *i.e.* close to human performance (83.7) and that for wrong roles reduces to 67.8. This causes the overall CIDEr to reduce by 2 points compared to no-grouping in Fig. 4.

Compute cost. Inference time for CineMEC is only 0.65 s per video on $1 \times A6000$ GPU. FINCH is a fast algorithm with $\mathcal{O}(N \log N)$ complexity [53] and running it on the fly is fast. The preprocessing time to extract Yolo + Siglip2, SlowFast visual features is 3.7 s per video. Our model has 245M parameters as compared to GVSR’s 220M.

D. Limitations

Semantic Role Labeling. Entity Role Grouping. The long-tail distribution also affects grouping: entities appearing early in a video or more frequently across events dominate ID assignments, while rare entities are clustered wrongly. This makes grouping harder and constrains downstream tasks. As

shown in our second analysis, improving grouping accuracy could substantially benefit both coreference and captioning.

Visual Clustering. The object proposal boxes occasionally miss relevant entities, leading to incomplete clusters and weaker grounding and captioning. This limitation arises not just from our framework but also from the underlying object proposal model.

E. FINCH-based Clustering

Input. We begin by describing how the box representations used for clustering are obtained. For each shot in the video, we apply a prompt-free tracker (YOLOE [65]) to generate short per-shot tracklets. From these tracklets, we collect the proposals corresponding to our sub-sampled frames. Each proposal is then passed through the VO encoder, yielding contextualised object features \mathbf{x}_{tl}^o for box l of frame t .

Overview of FINCH. FINCH [53] is a hierarchical clustering algorithm that forms partitions purely from first-order nearest-neighbor relations. At every level, each point is linked to its closest neighbor, and clusters naturally emerge without the need for specifying the number of clusters or setting a distance threshold. The algorithm operates on a pairwise distance matrix that captures how dissimilar the feature representations of any two points are, and uses this matrix to determine nearest-neighbor links. In our case, FINCH clusters the set of contextualised box features \mathbf{x}_{tl}^o into long, video-level entity tracks, which requires constructing such a distance matrix over these features.

Why the Standard FINCH Matrix Requires Adaptation. Directly applying FINCH assumes that all points are independent, which is violated in video-based tracking. Our boxes exhibit two domain-specific constraints that, if ignored, lead to invalid cluster assignments. We therefore modify the FINCH distance matrix to respect these constraints.

Constraint 1: Boxes from the same frame cannot belong to the same cluster. In frame t , multiple proposals \mathbf{x}_{tl}^o may appear simultaneously, but a valid temporal track may contain *only one* box per frame. Standard FINCH may incorrectly merge same-frame proposals into a single cluster. We enforce a hard exclusion by setting

$$\mathcal{D}(\mathbf{x}_{tl}^o, \mathbf{x}_{tl'}^o) = \infty \quad \text{for all } l \neq l'. \quad (1)$$

thereby preventing any clustering between proposals originating from the same frame.

Constraint 2: Boxes within the same shot-level tracklet must remain together. The tracker provides reliable identity continuity within each shot. Thus, all proposals belonging to the same shot-level tracklet correspond to the same underlying entity. Standard FINCH does not utilize this

Table 8. Statistics of our grounding annotations on the VidSitu dataset (together for validation and test sets).

Statistic	Value
Number of videos	2810
Number of unique captions	8157
Total boxes annotated	48026
Avg. boxes per unique caption	5.89
Avg. boxes per video	17.09

structural cue and may split such proposals across clusters. For any feature pair drawn from the same shot-tracklet, we strongly encourage early merging by scaling their distances:

$$\mathcal{D}(\mathbf{x}_{tl}^{'o}, \mathbf{x}_{t'l'}^{'o}) \leftarrow 10^{-5} \mathcal{D}(\mathbf{x}_{tl}^{'o}, \mathbf{x}_{t'l'}^{'o}). \quad (2)$$

where for all $(t,l),(t',l')$ in the same shot-level tracklet. This ensures that shot-consistent proposals cluster together before FINCH proceeds to merge across different shots.

With these modifications, FINCH produces clusters that extend shot-level tracklets into coherent, video-level entity trajectories. We employ a two-level hierarchy to achieve this: the first level groups boxes into stable intra-shot tracks while preventing same-frame overlaps, and the second level merges these shot-level groups across the full video to form long-range entity tracks.

Summary. Our modified FINCH pipeline (i) enforces frame-wise constraints and (ii) clusters tracks within shots. Together, these steps yield continuous visual clusters across shots, tightly integrating clustering with downstream video-language grounding.

F. Annotation Pipeline

Annotation Platform Setup. For a 10-second video containing 5 events, we sample frames at 1 fps, $\mathcal{F} = \{f_t\}_{t=1}^F$, resulting in 11 frames, following the protocol in GVSR [25]. From the SRL annotations, we collect captions for the primarily visual roles *agent* (*Arg0*), *patient* (*Arg1*), and *instrument* (*Arg2*) across all events. The unique captions from these roles are retained and treated as ground-truth labels for the video V . Each video is then used to create a dedicated annotation task in the CVAT tool [1], with video-specific labels as illustrated in Fig. 5. We annotated bounding boxes for videos in the validation and test splits of the VidSitu dataset, with overall statistics reported in Tab. 8.

F.1. Annotation Process

The annotation process begins with annotators first watching the complete video along with its associated role captions- Fig. 6. This initial pass provides a global understanding of the scene and helps identify entities and their continuity across shot changes. Annotators then return to the CVAT

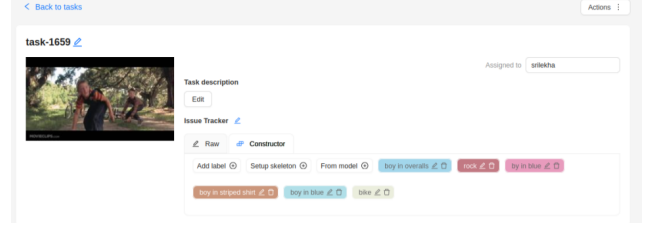


Figure 5. Example annotation task for a video. There are a total of 11 frames sub-sampled at T=1 second from a 10 second video. Text highlighted in colors represent different labels.

Video Annotations Dataset

Search Video Segment:
v_x2-MCPa_3rU_seg_10_20



Event	Arg0	Arg1	Arg2
Ev1 0-2 secs	Arg0 (thrower): boy in overalls	Arg1 (thing thrown): rock	Arg2 (thrown at, to, over, etc.): boy in blue
Ev2 2-4 secs	Arg0 (thrower): boy in striped shirt	Arg1 (thing thrown): rock	Arg2 (thrown at, to, over, etc.): boy in blue
Ev3 4-6 secs	Arg0 (walker): boy in blue	-	-
Ev4 6-8 secs	Arg0 (thrower): boy in overalls	Arg1 (thing thrown): rock	Arg2 (thrown at, to, over, etc.): boy in blue
Ev5 8-10 secs	Arg0 (elevator, Agent): boy in overalls	Arg1 (Logical subject, patient, thing rising): bike	-

Figure 6. Example visualization of ground truth semantic role labels for each event of a video. Annotators first go through the video and identify associations between captions and visual entities accurately.

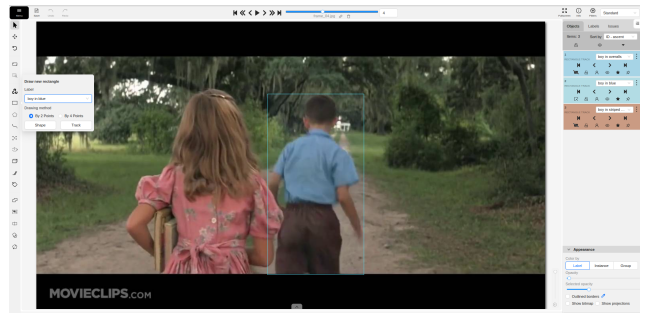


Figure 7. Choose a label that can be visually identified, then draw a bounding box around it or extend the existing track of the recognized entity. Label *boy in blue* is visible in frame 05.

tool [1] to assign each caption to its corresponding entity by drawing bounding boxes in the sampled 11 frames (see Fig. 7, 8).

While most captions can be localized visually, certain

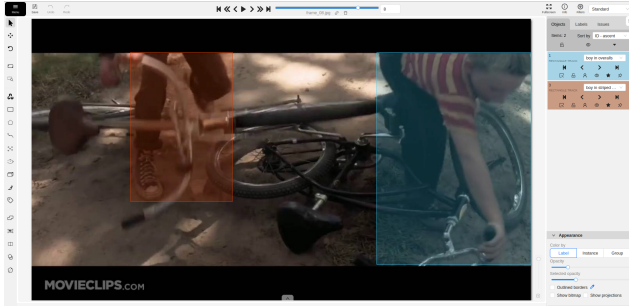


Figure 8. Identifying and linking entities to their captions becomes particularly challenging during shot changes like the one above. In such cases, association requires more than a simple object match; it demands a compositional understanding of semantic cues and spatial positioning within the scene to correctly distinguish between entities and assign the appropriate caption.

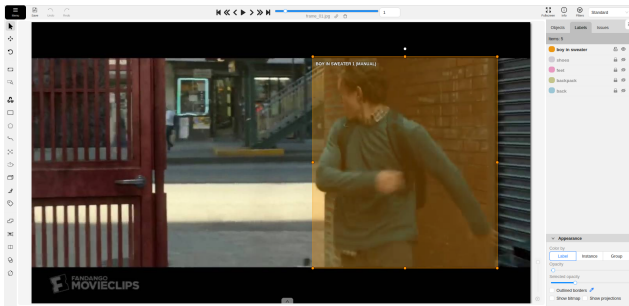


Figure 9. Label *back* is a non-visual role, hence it is not grounded.

cases are inherently non-annotatable, such as abstract expressions (e.g., *back* in Fig. 9). In other cases, annotation is non-trivial due to complex shot transitions. For example, in the *Forest Gump* sequence, multiple shot changes require viewing the full video to ensure consistent tracking of people. Similarly, Fig. 8 highlights a particularly challenging case, where multiple boys with bicycles must be distinguished and grounded. Here, visual cues such as dress color and spatial positioning, rather than faces, enable correct identification. These examples illustrate how annotating VidSitu requires not only bounding-box labeling but also a deeper semantic understanding of context and continuity. We assess potential annotator bias and observe a 92% inter-annotator agreement, measured using pairwise F1 (IoU@0.5) on 10 sampled videos, indicating strong consistency and minimal bias across annotations.

Compensation. We fairly compensated the annotators for their efforts.