EGODUAL: MULTI-LEVEL TASK PLANNING FROM EGOCENTRIC VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

A system that guides a procedural task from video, whether a proactive assistant coaching a person or a planner instructing a robot, must decide not only what comes next, but when to intervene. The right level of guidance depends on who carries it out: a person can act on the next subtask, while a robot policy needs the next concrete action. These levels follow different timelines. Within a subtask, the next action is imminent while no subtask boundary is near. At a boundary, the opposite holds. A granularity-agnostic policy cannot be correct at both. We introduce granularity-conditioned procedural planning, in which the requested level (subtask or action) is an explicit input that determines both what the agent predicts and when it intervenes. We build hierarchical supervision over subtasks and actions, and introduce EgoDual, a dual benchmark of 7,017 queries from 93 egocentric procedural videos in which coarse and fine requests share the same visual input. Across four VLMs, zero-shot models are largely insensitive to granularity. Our dual supervision learns distinct coarse and fine intervention policies, raising the granularity separation score from 0.45 to 0.68, and improves action-level prediction. Our analysis further shows that subtask-level anticipation is bottlenecked by predicting the next procedural state, not by generating text at the right level of abstraction.

1 INTRODUCTION

An agent that guides an ongoing procedural task, whether a proactive assistant coaching a person through a recipe (Li et al., 2025; Kundu et al., 2026) or a high-level planner instructing a robot policy (Ahn et al., 2022; Huang et al., 2022), must continuously answer two questions from the video stream: has the current step finished, and what comes next? Both answers depend on the level at which a step is defined. The step after cutting the peppers can be *fill the peppers* or *set down the knife*, and which is appropriate depends on the executor: a person, or a capable policy on an easy task, needs only the goal (Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024), whereas a robot policy facing a difficult manipulation needs the detailed action (Belkhale et al., 2024; Shi et al., 2025). We call this problem *granularity-conditioned procedural planning*: given the observed video, the procedural context, and a requested granularity, answer both questions at that level.

Answering both questions is already demanding at a single level. On the assistance side, streaming vision–language models (VLMs) process continuous video (Chen et al., 2024; Qian et al., 2025), and proactive assistants built on them decide when to speak and what to say, which requires detecting step completion and responding before the moment has passed (Li et al., 2025; Wang et al., 2025; Zhang et al., 2025; Kundu et al., 2026). In robotics, language-model planners decompose a goal into steps and rely on success detection to decide when to advance (Ahn et al., 2022; Huang et al., 2022; Song et al., 2023), and vision–language–action (VLA) policies execute the instruction they are given (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024), so the planner's timing and wording directly determine what the robot does (Huang et al., 2022; Shi et al., 2025).

Planning at a requested granularity is harder still, and no existing system supports it. Proactive assistants inherit the guidance level of their annotations and are evaluated against a single ground-truth timeline (Li et al., 2025; Wang et al., 2025; Zhang et al., 2025; Kundu et al., 2026); VLA policies, even hierarchical ones with an intermediate language level (Belkhale et al., 2024; Shi et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2025), fix that level at training time. Datasets impose the same restriction: robot datasets attach one task-level instruction per episode (Walke et al., 2023; Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024), and egocentric corpora narrate at one level throughout (Damen et al., 2022; Grauman et al., 2022; Sener et al., 2022). Granularity moreover changes *when* an instruction is due: fine actions last only a few seconds (Damen et al., 2022; Sener et al., 2022), whereas a subtask spans many of them and a new one begins only when the previous one completes (Sener et al., 2022), so at most moments exactly one of the two levels calls for an instruction. No existing benchmark tests whether the requested level changes a model's decision (Zhang et al., 2025; Kundu et al., 2026; Tran et al., 2026), and in Section 4.2 we show that current models do not change theirs.

We close this gap by making granularity an explicit input. We annotate egocentric procedural videos (Tran et al., 2026) with coarse *subtasks*, each achieving one intermediate outcome, and the fine-grained *actions* that realize them, following the two-level structure of Assembly101 (Sener et al., 2022), and build EgoDual, a dual benchmark in which every query presents the same visual prefix under both granularities, with prompts differing in a single token (Section 3.2). Its key cases are the *divergent states*, where one granularity requires an instruction and the other silence; with the observation fixed, they isolate the effect of granularity from that of visual evidence.

We then train a VLM with a timing adapter for when to speak and a content adapter for what to say, supervised with dual coarse and fine labels, so that one model adapts both decisions to the requested level. We fine-tune Qwen3-VL-8B (Qwen Team, 2025) and Gemma3-4B (Gemma Team, 2025) under this scheme and compare them with four zero-shot VLMs, which time interventions poorly at either granularity and barely respond to the requested level. Granularity-conditioned training improves timing at both levels, yields distinct coarse and fine intervention policies (Section 4.2), and improves prediction of the next action on both backbones (Section 4.3).

2 Related Work

Progress Monitoring and Next-Step Prediction. Guiding a procedure from streaming video requires two decisions: whether the current step has finished, and what should come next. Existing work has largely addressed the two in isolation. Egocentric action anticipation predicts the next step at fixed or long horizons and from untrimmed streams (Furnari & Farinella, 2019; Girdhar & Grauman, 2021; Mascaró et al., 2023; Furnari & Farinella, 2023; Rodin et al., 2025), but it is evaluated on what comes next rather than on whether an instruction is due. Progress monitoring has instead been taken up by streaming video–language models, which align generation with the video timeline and decide at each moment whether to respond (Chen et al., 2024; Qian et al., 2025). Proactive assistants combine the two decisions, tracking task state to time their guidance and choosing what to say when they intervene (Li et al., 2025; Wang et al., 2025; Zhang et al., 2025; Kundu et al., 2026), and language-model planners in robotics do the same on behalf of a low-level policy by detecting success before advancing and re-planning from feedback (Huang et al., 2022; Song et al., 2023). In every case, however, the level at which a step is defined is fixed by the training annotations (Li et al., 2025; Wang et al., 2025; Zhang et al., 2025; Kundu et al., 2026): the system monitors and predicts at that one level, and cannot be asked at inference time to operate at another.

Levels of Abstraction in Procedures. A physical procedure can be described at many levels, from the overall goal down to motor-relevant actions (Sener et al., 2022; Belkhale et al., 2024), yet existing systems bind each component to a single level. Language-model planners emit high-level steps (Ahn et al., 2022), vision–language–action policies execute whatever instruction they receive (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024), and hierarchical systems insert an intermediate language level between the two (Belkhale et al., 2024; Shi et al., 2025). Even in the hierarchical systems the level is a property of the module, so changing the level of guidance means changing the module (Belkhale et al., 2024; Shi et al., 2025). Procedural understanding likewise models the structure among the steps of one level, for online mistake detection and progress tracking (Flaborea et al., 2024; Seminara et al., 2024), and although human-centered embodied agents have been argued to require assistance adapted to the actor's state and needs (Ding et al., 2026), how the level of guidance should adapt is left open. The consequence of the level for timing has not been examined at all. A fine action ends every few seconds (Damen et al., 2022; Sener et al., 2022), whereas a coarse step ends only when its outcome is reached (Sener et al., 2022), so the same observation can

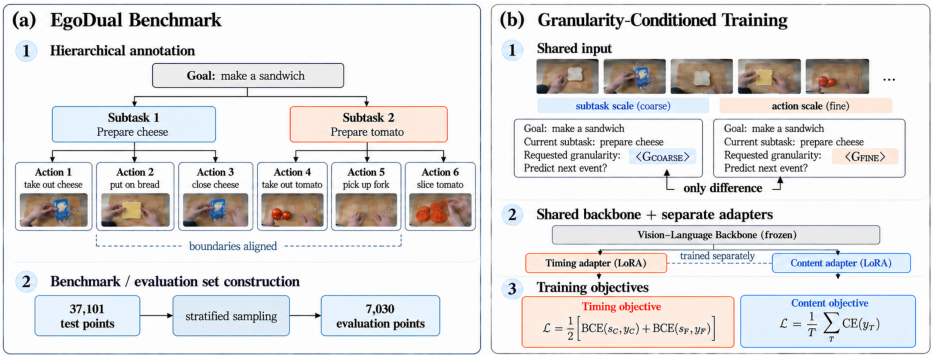

Figure 1: Overview. (a) The EgoDual benchmark annotates each video with coarse subtasks and the fine actions that realize them, together with transition attributes. Every query is evaluated under both granularities on the same observation. (b) The granularity-conditioned model receives inputs that differ only in the granularity token. A frozen vision–language backbone feeds a timing adapter, which reads an intervention score from the <DECIDE> token, and a separately trained content adapter, which generates the next subtask or action.

require an instruction at one level and silence at the other. We instead treat granularity as a single input token to one model and measure its effect on both when the model intervenes and what it says.

Benchmarks for Procedural Assistance. Existing benchmarks evaluate procedural assistance at a single level of abstraction. Benchmarks for intervention timing compare a model's decisions along a dense temporal grid against one reference timeline annotated at one level (Zhang et al., 2025; Kundu et al., 2026; Tran et al., 2026). Datasets for next-step prediction impose the same restriction: egocentric corpora narrate at one level throughout (Damen et al., 2022; Grauman et al., 2022), and robot learning datasets attach a single task-level instruction to each episode (Walke et al., 2023; Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024). The one corpus with both coarse and fine labels, Assembly101, defines its anticipation task at a single level and includes no timing task (Sener et al., 2022). No benchmark therefore tests whether a model's timing and content change appropriately when the requested level changes (Zhang et al., 2025; Kundu et al., 2026; Tran et al., 2026). Our dual benchmark adopts the two-level structure of Assembly101 (Sener et al., 2022), adds a timing decision at each level, and queries both levels on identical visual prefixes, so that sensitivity to granularity becomes a measurable quantity.

3 Method

We formalize when to intervene and what to say at a requested level (Section 3.1), build the EgoDual benchmark that poses both questions at both levels on identical observations (Section 3.2), and train timing and content adapters that answer them at that level (Section 3.3). As illustrated in Figure 1, the benchmark (a) annotates each video at two nested granularities and scores both on identical observations, while the training scheme (b) conditions a single model on which granularity is requested.

3.1 Problem Formulation

A procedural task is described at two levels (Figure 2). A *subtask* is a contiguous sequence of actions that achieves one intermediate outcome of the task, and an *action* is a single step that realizes it. Which level an executor needs is not inferred; the requested level is an input to the model.

At query time t, the model observes the video context x_t, the procedural context c_t, which contains the task goal and the subtask in progress, and a requested level $g \in \{\mathrm{C}, \mathrm{F}\}$, where C denotes the subtask level and F the action level. It predicts whether to intervene and, if so, what to say:

$$P_\theta(d_t, y_t \mid x_t, c_t, g), \tag{1}$$

where $d_t \in \{\mathrm{SPEAK}, \mathrm{SILENT}\}$ is the intervention decision and y_t is the next subtask under $g = \mathrm{C}$ or the next action within the current subtask under $g = \mathrm{F}$. Both outputs depend on g: a subtask

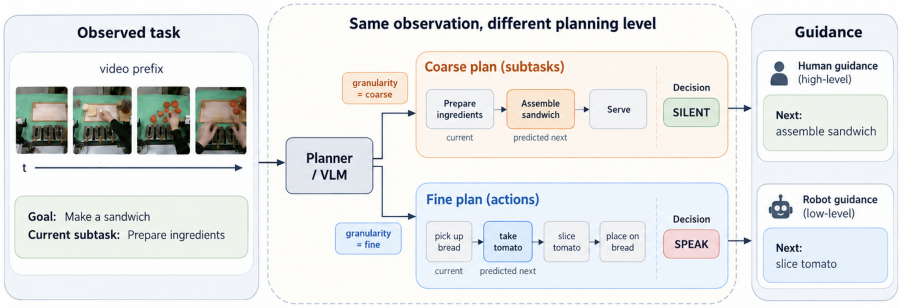

Figure 2: Granularity-conditioned procedural planning. The same observation is queried at two planning levels, which yield different predictions and different intervention decisions.

begins only at a subtask boundary, whereas an action begins at every step within one, so the same observation calls for an intervention at one level and silence at the other.

3.2 EGODUAL BENCHMARK

Source videos. The benchmark comprises 93 egocentric procedural videos covering cooking, repair, crafts, and tutorial activities, drawn from the validation split of an egocentric assistance corpus (Tran et al., 2026) and held out from training and model selection.

Hierarchical annotation. We manually segment every video into fine-grained action intervals. Qwen3-VL-235B then produces a description for each action, the task goal, a partition of the actions into coarse subtasks, and transition attributes; all outputs are manually reviewed and corrected. Each transition, at either level, is labeled STRONG when the next unit continues with the object currently in hand, WEAK when it introduces a new object or tool, and NONE when it enters an idle or task-unrelated state. These labels select the forecasting targets and support the analysis in Section 4.5.

Coarse and fine annotation. Following the protocol of the source corpus, each video is queried every $0.5\,\mathrm{s}$. At query time t, the model observes the video up to t and decides whether an eligible event begins before the next query. Every query is instantiated under both granularities with identical frames and context, the prompts differing only in the granularity token. The coarse target is SPEAK when a new subtask begins, and the fine target is SPEAK when a new action begins within the current subtask; the first action of a subtask is attributed to the coarse level. Only STRONG transitions define positive targets. Fine events that are too short, adjacent to an idle span, or textually ambiguous are excluded, and WEAK and NONE transitions are masked for the affected granularity.

Evaluation set. The dense grid contains $37{,}101$ dual queries, most of which lie far from any transition. We keep every positive and transition-associated query and subsample the remaining negatives at a rate that decreases with distance to the nearest event: 25% within $2\,\mathrm{s}$, 10% between 2 and $4\,\mathrm{s}$, and 3% beyond $4\,\mathrm{s}$. This yields $7{,}030$ dual queries, of which $7{,}017$ are scored after masking, including 175 coarse and 562 fine positives. Every scored query is BOTH-SILENT, FINE-ONLY, or COARSE-ONLY. The 737 queries in the latter two states are the *divergent* cases at the center of the benchmark: the observation is the same but the correct decision differs, so any change in model behavior is attributable to granularity alone.

3.3 GRANULARITY-CONDITIONED TRAINING

We fine-tune a vision–language model to represent Eq. 1, with a prompt whose only granularity-dependent element is the token encoding g. Two components share this backbone and input: a *timing adapter* determines when to speak, and a *content adapter* generates what to say. They are trained separately, because under a joint objective the content term dominates the gradient and starves the decision (Section 4.6). Both are supervised at the coarse and the fine level.

3.3.1 WHEN TO SPEAK

The timing adapter appends a <DECIDE> token to the prompt. Let $\mathbf{h}(x_t, c_t, g)$ be the backbone's final-layer hidden state at this token, which summarizes the observation and the requested granularity. A linear readout with parameters $\mathbf{w}$ and b maps it to the intervention score $s_g = \mathbf{w}^\top \mathbf{h}(x_t, c_t, g) + b$. Let $d_g \in \{0, 1\}$ be the intervention label at granularity g, with 1 for SPEAK and 0 for SILENT. Each query contributes both of its decisions to the loss, $\mathcal{L}_{\text{decision}} = \frac{1}{2} \sum_{g \in \{\text{C,F}\}} \text{BCE}(s_g, d_g)$. On a divergent query the two terms demand opposite decisions from the same observation, and the granularity token is the only input that distinguishes them. A policy that ignores g therefore cannot minimize this objective.

Sampling. Coarse transitions are far rarer than fine ones, and both are rare relative to silent queries. We sample BOTH-SILENT, FINE-ONLY, and COARSE-ONLY examples in a 2:1:1 ratio, so each granularity is positive on a quarter of the training examples. Within BOTH-SILENT, queries closer to an upcoming event are sampled more often, concentrating the negatives on the temporally difficult decisions.

3.3.2 WHAT TO SPEAK

The content adapter is trained only on reference-SPEAK examples. The target $y = (y_1, \ldots, y_T)$ describes the next subtask under $g = \text{C}$ and the next action under $g = \text{F}$, and ends with a supervised <eos> token so that the model learns to stop. The loss is token-level cross-entropy, averaged within each example before averaging across the batch $\mathcal{B}$:

$$\mathcal{L}_{\text{content}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{T_i} \sum_{\tau=1}^{T_i} -\log P_\theta\big(y_\tau^{(i)} \mid y_{<\tau}^{(i)},\, x_{t_i},\, c_{t_i},\, g_i\big). \tag{2}$$

The per-example normalization by T_i matters because subtask descriptions are systematically longer than action descriptions; a loss summed over tokens would weight the coarse condition by its verbosity rather than by its frequency. The prompt, the frames, and the forced SPEAK marker are excluded from the sum, so gradients flow only through the predicted description. Because fine positives outnumber coarse positives by roughly an order of magnitude, we sample the two granularities in equal proportion; at the natural frequency the model would become fluent in action-level phrasing while rarely practicing the subtask level.

3.3.3 RELATIVE GRANULARITY OBJECTIVE

The decision loss $\mathcal{L}_{\text{decision}}$ supervises each granularity in absolute terms. We additionally study whether relating the two scores of the same query improves sensitivity to granularity. Let $\Delta = s_{\text{F}} - s_{\text{C}}$ denote the model's preference for fine over coarse intervention. For a FINE-ONLY query i and a COARSE-ONLY query j, the ranking objective

$$\mathcal{L}_{\text{gran}} = \text{softplus}\big(-(\Delta^i - \Delta^j)\big) \tag{3}$$

requires the preference for fine intervention to be larger on i than on j, which is exactly the pairwise ordering that the granularity separation score measures (Section 4.1). We evaluate it as an optional variant of the timing objective,

$$\mathcal{L}_{\text{timing}} = \mathcal{L}_{\text{decision}} + \lambda_g \mathcal{L}_{\text{gran}}, \qquad \lambda_g = 0.5, \tag{4}$$

while the default timing model uses $\mathcal{L}_{\text{decision}}$ alone.

4 EXPERIMENTS

The experiments address four questions: whether the trained model's intervention timing changes with the requested granularity, and whether the timing of zero-shot models already does (Section 4.2); how accurately it predicts the next step at each granularity (Section 4.3); at what temporal resolution it places an intervention (Section 4.4); and how much of its action accuracy is attributable to the next action reusing the object in hand (Section 4.5). Section 4.6 gives the reason for training timing and content on separate adapters.

System	coarse (C)		fine (F)		
	acc.	ROC-AUC	acc.	ROC-AUC	GranSep
Random	50.0	0.500	50.0	0.500	0.500
Qwen3-VL-8B	51.1	0.471	51.2	0.510	0.448
Qwen3.6-27B	51.1	0.468	50.4	0.510	0.420
Gemma3-4B	52.0	0.505	52.9	0.535	0.583
Llama 4 Maverick	48.3	0.494	50.4	0.490	0.490
Qwen3-VL-8B, ours	**63.7**	<u>0.692</u>	**62.1**	**0.612**	**0.675**
$+\mathcal{L}_{\text{gran}}$	<u>62.5</u>	**0.709**	<u>58.5</u>	<u>0.595</u>	<u>0.661</u>
Gemma3-4B, ours	56.4	0.639	53.6	0.574	0.627
$+\mathcal{L}_{\text{gran}}$	57.2	0.646	54.7	0.588	0.653

Table 1: **Timing.** Balanced accuracy (acc., chance 50%) and ROC-AUC (chance 0.5) at the coarse (C) and fine (F) granularity. GranSep is the granularity separation score (chance 0.5): the probability that a FINE-ONLY query receives a higher preference for a fine intervention than a COARSE-ONLY query. The upper block is zero-shot; the lower block is trained with supervision at both granularities, and $+\mathcal{L}_{\text{gran}}$ adds the optional ranking term of Eq. 3. Bold: best per column; underline: second.

4.1 SETUP

Models. We evaluate four instruction-tuned vision–language models zero-shot: Qwen3-VL-8B (Qwen Team, 2025), Qwen3.6-27B (Qwen Team, 2026), Gemma3-4B (Gemma Team, 2025), and Llama 4 Maverick (Meta AI, 2025). We fine-tune Qwen3-VL-8B and Gemma3-4B with the scheme of Section 3.3 on 8 H200 GPUs, using the training split of the source corpus annotated as in Section 3.2.

Metrics. ROC-AUC is the probability that a random SPEAK query receives a higher intervention score than a random SILENT query (chance 0.5); balanced accuracy is the accuracy of the binary decision on class-balanced subsets (chance 50%). A system that assigns identical scores under the coarse and the fine prompt can attain high ROC-AUC and high balanced accuracy at each granularity, so neither metric measures whether timing depends on the requested granularity.

The *granularity separation* score (GranSep) measures whether the intervention score depends on the granularity token. Let $\Delta = s_{\text{F}} - s_{\text{C}}$ denote the difference between the fine and the coarse intervention score of the same query (Section 3.3.3). GranSep is the probability that a random FINE-ONLY query has a larger Δ than a random COARSE-ONLY query; a system that ignores the granularity token scores exactly 0.5. Content is evaluated at reference SPEAK queries. A Qwen3-VL-235B judge determines whether the predicted verb and the predicted object each match the reference step; acc. is the fraction of predictions in which verb and object both match, and gran. is the fraction of predictions phrased at the requested granularity.

4.2 DUAL SUPERVISION PRODUCES GRANULARITY-DEPENDENT TIMING; ZERO-SHOT PROMPTING DOES NOT

Zero-shot models neither separate SPEAK from SILENT queries nor respond to the granularity token. In Table 1, their ROC-AUC lies between 0.468 and 0.535 and their GranSep between 0.420 and 0.583. Qwen3.6-27B does not outperform Qwen3-VL-8B, so a larger model of the same family supplies neither ability.

Supervision at both granularities yields timing that separates SPEAK from SILENT queries and changes with the granularity token. Trained Qwen3-VL-8B attains ROC-AUC 0.692 at the coarse and 0.612 at the fine level and GranSep 0.675; trained Gemma3-4B attains 0.639, 0.574, and 0.627. The same observation now receives different intervention scores under the coarse and the fine prompt, as BCE loss requires on divergent queries. ROC-AUC is lower at the fine level on both backbones: actions are shorter and more frequent than subtasks, so the SILENT queries nearest a fine positive lie within one to two seconds of it (Section 4.4).

	coarse (C), $n = 179$				fine (F), $n = 562$			
System	acc.	verb	object	gran.	acc.	verb	object	gran.
Qwen3-VL-8B	0.128	0.128	0.475	0.631	0.206	0.212	<u>0.546</u>	0.573
Qwen3.6-27B	0.073	0.073	0.346	0.564	0.158	0.164	0.406	0.546
Gemma3-4B	0.000	0.000	0.000	0.011	0.018	0.018	0.037	0.043
Llama 4 Maverick	0.006	0.006	0.006	0.006	0.000	0.000	0.000	0.000
Single-adapter	0.117	0.117	0.397	**0.866**	<u>0.219</u>	<u>0.222</u>	0.521	0.514
Qwen3-VL-8B, ours	<u>0.196</u>	<u>0.196</u>	**0.642**	<u>0.749</u>	**0.310**	**0.331**	**0.641**	**0.980**
Gemma3-4B, ours	**0.212**	**0.229**	<u>0.581</u>	0.726	0.185	0.215	0.479	<u>0.968</u>

Table 2: **Content at reference steps.** Agreement with the reference step as judged by Qwen3-VL-235B: acc. requires both verb and object to agree; verb and object score each field alone; gran. is the fraction of predictions phrased at the requested granularity. Single-adapter fine-tunes Qwen3-VL-8B with timing and content on one adapter under a shared objective; ours uses separate adapters. n is the number of reference steps per granularity. Bold: best per column; underline: second.

The ranking term improves Gemma3-4B and degrades Qwen3-VL-8B. Adding $\mathcal{L}_{\text{gran}}$ (Eq. 3, $\lambda_g = 0.5$), which directly optimizes the pairwise ordering that $\mathrm{GranSep}$ measures, improves every metric for Gemma3-4B but degrades four of the five metrics for Qwen3-VL-8B, including $\mathrm{GranSep}$ (0.675 to 0.661). The default model therefore omits the ranking term.

4.3 Training Improves Step Prediction; the Next Subtask Remains Difficult for All Systems

Table 2 scores the predicted step at reference SPEAK queries, so the content score is independent of the timing decision. The *single-adapter* baseline fine-tunes Qwen3-VL-8B with timing and content on one adapter, using the sum of the decision loss and the content loss; comparison with our model isolates the effect of training timing and content separately.

Training raises step accuracy at the coarse and the fine level on Qwen3-VL-8B and on Gemma3-4B. Qwen3-VL-8B improves from 0.128 to 0.196 at the coarse and from 0.206 to 0.310 at the fine level; Gemma3-4B improves from 0.000 to 0.212 and from 0.018 to 0.185. In the zero-shot setting, Gemma3-4B and Llama 4 Maverick rarely produce a step description that the judge can match to the reference. Trained models also respond at the requested granularity: at the fine level, 0.980 and 0.968 of their predictions are phrased as actions, compared with at most 0.573 for the zero-shot models.

At the fine level, errors of the trained model are predominantly verb errors. For trained Qwen3-VL-8B the object matches in 0.641 of predictions and the verb in 0.331. The object agreement is inflated by the evaluation set, which contains only STRONG transitions, at which the next action by definition uses the object in hand (Section 3.2); Section 4.5 quantifies the inflation.

Coarse accuracy is limited by anticipating the next subtask; phrasing at the subtask level is already learned. No system exceeds 0.212 at the coarse level. Verb agreement equals acc. in every coarse row but one, so the errors consist of incorrect verbs: the predictions typically name the object of the ongoing activity without stating the next intermediate outcome. Rewriting the coarse training targets as outcome statements raised the fraction of predictions phrased at the coarse level from 0.698 to 0.749 and left accuracy unchanged, so phrasing does not limit coarse accuracy.

4.4 The Trained Model Anticipates a Step by a Few Seconds Without Localizing Its Onset Query

Table 3 partitions the SILENT queries of Table 1 by their distance to the nearest step and reports the ROC-AUC of the SPEAK queries against each partition, revealing the temporal resolution of its timing.

	coarse (C)			fine (F)		
System	≤ 2 s	2–4 s	> 4 s	≤ 2 s	2–4 s	> 4 s
Qwen3-VL-8B	0.503	0.473	0.467	0.504	0.513	0.561
Qwen3.6-27B	0.447	0.459	0.471	0.505	0.520	0.514
Gemma3-4B	0.490	0.488	0.509	0.536	0.545	0.522
Qwen3-VL-8B, ours	**0.567**	**0.646**	**0.698**	**0.564**	**0.739**	**0.726**

Table 3: **Timing precision.** ROC-AUC of positives against negatives grouped by their temporal distance to the nearest step, per granularity. A model that only detects approaching steps separates distant but not near negatives; a model that resolves onsets separates both. Bold: best per column.

	STRONG, $n = 562$			WEAK, $n = 517$		
System	acc.	verb	object	acc.	verb	object
Repetition	0.101	0.103	0.423	0.035	0.039	0.190
Qwen3-VL-8B	<u>0.206</u>	0.212	<u>0.546</u>	0.093	0.103	0.300
Qwen3-VL-8B, ours	**0.310**	**0.331**	**0.641**	**0.184**	**0.236**	**0.383**
Gemma3-4B, ours	0.185	<u>0.215</u>	0.479	<u>0.145</u>	<u>0.188</u>	<u>0.323</u>

Table 4: **Fine content by transition type.** Agreement with the reference as in Table 2. STRONG transitions (the evaluation set of Table 2) keep the object in hand; WEAK transitions introduce a new object or tool. The two sets are disjoint. Repetition predicts the previous action again and never observes the video. Bold: best per column; underline: second.

The trained model detects the approach of a step to within a few seconds but assigns the onset query no higher a score than its neighbors. For trained Qwen3-VL-8B, ROC-AUC against SILENT queries more than 4 s from any step is 0.698 at the coarse and 0.726 at the fine level; against SILENT queries within 2 s of a step it is 0.567 and 0.564, close to chance. Zero-shot models perform at chance level against every partition, between 0.447 and 0.561.

Errors within 2 s partly reflect the temporal resolution of the annotation; timing should therefore also be scored with a tolerance window. Following its source corpus, the benchmark labels exactly one query per step as SPEAK. The annotation precision of an action onset is approximately one query interval, so part of the error within 2 s consists of disagreements below the resolution of the annotation, which no model can resolve. We retain query-level scores for comparability with prior work and recommend additionally scoring timing under a tolerance window of a few queries around each reference onset.

4.5 Action Accuracy Depends on Whether the Next Object Is Already in Hand

Table 2 scored only STRONG transitions, at which the next action uses the same object as the current action, so a model identifies the correct object by naming the current object without recourse to the video. Table 4 measures the fraction of the object agreement obtained in this way, using the disjoint set of WEAK transitions, at which the next action introduces a new object or tool (masked in the timing benchmark, Section 3.2), and the repetition baseline.

Two thirds of the object agreement on STRONG transitions is attainable without visual input. The repetition baseline matches the object on 0.423 of STRONG and on 0.190 of WEAK transitions. Our model attains 0.641 and 0.383, exceeding the repetition baseline by 0.218 and 0.193. The contribution of visual input to the object agreement is therefore approximately 0.2 on both transition types, and the remainder of the drop from 0.641 to 0.383 reflects the loss of object continuity. All systems show the same asymmetry: from STRONG to WEAK transitions, object agreement decreases by 0.16 to 0.26, whereas verb agreement decreases by 0.03 to 0.11.

Verb agreement is conditional on object agreement. For our model, verb agreement given a matching object is 0.483 on STRONG and 0.480 on WEAK transitions, whereas given a non-matching

Form of decision loss	decision term	content term
Per token	3.6%	96.3%
Per answer string	42.2%	57.6%

Table 5: Share of the gradient norm on training batches contributed by the decision term and by the content term when both are trained on adapter. Per token applies the decision loss to each token; per answer string normalizes it over the two answer strings. The content term dominates in both settings.

object it is 0.06 and 0.09. The additional errors on WEAK transitions are therefore joint verb-and-object errors, and the fraction of predictions incorrect on both fields rises from 0.34 to 0.57. Object identification is thus the principal failure mode of action prediction, and it fails most at transitions that introduce a new tool, where a robot executor most requires an explicit instruction.

4.6 Under a Shared Adapter the Content Loss Dominates the Gradient

The single-adapter baseline in Table 2 trains timing and content on one adapter and is less accurate at each granularity (0.117 against 0.196 at the coarse and 0.219 against 0.310 at the fine level); at the fine level it also reaches the generation limit on most queries, so that only 0.514 of its outputs are phrased as actions.

Under a shared adapter with a per-token decision loss, the content loss contributes 96% of the gradient norm. In Table 5, with the decision loss applied per token, the content loss contributes 96.3% of the gradient norm and the decision loss 3.6%. The imbalance is intrinsic to the two objectives: next-token prediction of a free-form description has an attainable loss far above that of a binary decision, so its gradient is larger. Applying the decision loss per token amplifies the imbalance, because the two decision strings share their first token and differ only at the second, so most of the decision gradient concerns string format. Normalizing the decision loss over the two answer strings raises its share to 42.2%, and the content loss still dominates. Separate adapters remove the imbalance, and the linear readout removes the format component of the decision loss.

5 Conclusion and Future Work

We study procedural planning from egocentric video when the requested level of guidance, coarse subtask or fine action, is given as input. Because subtasks and actions begin at different moments, the level determines both the content and the timing of the next instruction. Existing benchmarks fix a single level and cannot measure whether a model responds to the request. The EgoDual benchmark evaluates every observation under both levels, isolating the effect of the request from that of the video. Under this protocol, zero-shot vision–language models time interventions nearly identically under both requests, whereas a model trained with dual supervision on a timing adapter and a content adapter times them differently and predicts the next step more accurately at each level. Granularity sensitivity therefore requires explicit supervision.

Two limitations point to future work. First, the model receives the granularity as an input rather than inferring it. In practice the right level depends on who is following the instructions. A proactive assistant can learn it from user feedback, since a user who asks for clarification wants finer actions and one who skips ahead wants coarser subtasks. A robot can derive it from the confidence of its executing policy, issuing coarse subtasks for routine tasks and finer actions for tasks it has not seen before. Second, the benchmark assumes a single correct next step, but many states admit several valid continuations. Capturing them requires a larger corpus and a multimodal judge that can reason about the consequence of each candidate and decide which ones the task actually requires.

AI use statement

In this work, we used generative AI tools to assist with several parts of the research workflow. Vision-language models were used to assist in constructing task annotations, including task goals, subtask descriptions, action descriptions, and transition attributes. All model-assisted annotations

used in the benchmark were manually reviewed and corrected by the authors. All AI-assisted annotations, code, figures and written content were independently reviewed by the authors. We take responsibility for the final content of this work, including text, claims, code, annotations, and artifacts produced with the aid of generative AI.

Reproducibility statement

We provide an anonymized code repository containing the implementation and supporting scripts used in this work. The main paper and appendix describe the benchmark construction, annotation procedure, model configuration, training objectives, hyperparameters, and evaluation protocol. Together with the released code, these details are intended to support reproduction of the main experimental results. Code: anonymous repository.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. RT-H: Action hierarchies using language. In *Robotics: Science and Systems (RSS)*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *CVPR*, pp. 18407–18418, 2024.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

Qianggang Ding, Xingyao Wang, Rui Feng, Zhibin Wang, Feixiang Wang, Kelong Mao, Hao Sun, Zhiyao Luo, Jiankai Tang, Lei Li, Jiadong Guo, Minheng Ni, Weicong Lin, Chenxi Yang, Hongxiang Gao, Zhenghua Chen, Yang Bai, Min Wu, Jun Cheng, Huazhu Fu, Dacheng Tao, and Bang Liu. ComBodied agents: A new paradigm of human-centric agentic AI. *arXiv preprint arXiv:2608.10915*, 2026.

Alessandro Flaborea, Guido Maria D'Amely di Melendugno, Leonardo Plini, Luca Scofano, Edoardo De Matteis, Antonino Furnari, Giovanni Maria Farinella, and Fabio Galasso. PREGO: Online mistake detection in procedural egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18483–18492, 2024.

Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling LSTMs and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6252–6261, 2019.

Antonino Furnari and Giovanni Maria Farinella. Streaming egocentric action anticipation: An evaluation scheme and approach. *Computer Vision and Image Understanding*, 234:103763, 2023.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13505–13515, 2021.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, 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 (CVPR)*, pp. 18995–19012, 2022.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Kaustav Kundu, Ritvik Shrivastava, Maxim Arap, Nanshu Wang, Xianhui Zhu, Quintin Fettes, Gautam Tiwari, Parth Suresh, Théo Moutakanni, Alejandro Castillejo Munoz, Allen Bolourchi, Pascale Fung, Pinar Donmez, Babak Damavandi, Anuj Kumar, and Seungwhan Moon. Plan, watch, recover: A benchmark and architectures for proactive procedural assistance. *arXiv preprint arXiv:2606.04970*, 2026.

Chenyi Li, Guande Wu, Gromit Yeuk-Yin Chan, Dishita G. Turakhia, Sonia Castelo Quispe, Dong Li, Leslie Welch, Claudio Silva, and Jing Qian. Satori: Towards proactive AR assistant with Belief-Desire-Intention user modeling. In *CHI Conference on Human Factors in Computing Systems*, 2025. doi: 10.1145/3706598.3714188.

Esteve Valls Mascaró, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action anticipation. In *WACV*, pp. 6048–6057, 2023.

Meta AI. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`, 2025. Accessed 2026-09-26.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video LLMs with active real-time interaction via disentangled perception, decision, and reaction. In *CVPR*, pp. 24045–24055, 2025.

Qwen Team. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model. `https://qwen.ai/blog?id=qwen3.6-27b`, April 2026.

Ivan Rodin, Antonino Furnari, and Giovanni Maria Farinella. Egocentric action anticipation from untrimmed videos. *IET Computer Vision*, 19(1):e12342, 2025. doi: 10.1049/cvi2.12342.

Luigi Seminara, Giovanni Maria Farinella, and Antonino Furnari. Differentiable task graph learning: Procedural activity representation and online mistake detection from egocentric videos. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21064–21074. IEEE, 2022.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi Robot: Open-ended instruction following with hierarchical vision-language-action models. In *International Conference on Machine Learning (ICML)*, 2025.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. LLM-Planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2998–3009, 2023.

Tuyen (Harry) Tran, Maxim Arap, Seungwhan Moon, Raffay Hamid, Alessandro Suglia, Zsolt Kira, Pascale Fung, and Mubarak Shah. Wearable AI Workshop at ECCV 2026. `https://wearable-ai-workshop.github.io/`, 2026. Workshop at the European Conference on Computer Vision (ECCV) 2026.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. StreamBridge: Turning your offline video large language model into a proactive streaming assistant. In *Advances in Neural Information Processing Systems*, 2025.

Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. Proactive assistant dialogue generation from streaming egocentric videos. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12044–12068. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.605.

A Appendix

Ethics statement

This work uses existing egocentric video datasets and does not involve new human-subject data collection. The introduced annotations focus on task goals, subtasks, actions, and temporal transitions rather than personal attributes. Future deployment in assistive or embodied AI systems should include appropriate validation and safeguards, particularly when model outputs are connected to real-world actions.

A.1 Evaluation points

Table 6: Construction of the evaluation set from the complete query grid. Sampling operates on the pair, so both granularities are always scored at the same instants; retained points within a video are at least 2 s apart, and the retention rate is equal across videos under a fixed seed.

Category	Grid	Retained	Evaluation set
COARSE_ONLY (STRONG positives)	179	all	179
FINE_ONLY (STRONG positives)	565	all	565
points carrying a WEAK or NONE event	718	all	718
ineligible action onsets	2,053	30%	613
BOTH_SILENT, ≤ 2 s from an event	15,862	25%	3,964
BOTH_SILENT, 2–4 s	6,543	10%	654
BOTH_SILENT, > 4 s	11,181	3%	337
total	37,101		7,030

Table 6 gives the construction of the evaluation set. Four sample counts recur in the main text and denote different things. There are 179 coarse positives, of which 175 are COARSE_ONLY and thus

enter GranSep; at the remaining four an ineligible fine event coincides, so the fine side is masked and the pair is not a clean divergent state, although the coarse side is still scored as a positive. The divergent set therefore has $175 + 562 = 737$ queries, while the content evaluation set, which is all reference SPEAK points at either granularity, has $179 + 562 = 741$. Of the 565 FINE_ONLY positives, 562 are scored. The frame-extraction path is not modified to recover them, since evaluation and training must decode identically.

B PROMPTS

B.1 TIMING

Both granularities receive identical text except for the granularity token on the penultimate line. The visual input precedes the text; there is no system prompt and no dialogue history.

```
Goal: {goal}
Current subtask: {label of the subtask in progress at t}
Event definitions:
<COARSE>: the next high-level subtask begins.
<FINE>: the next fine-grained action within the current
subtask
 begins. The first action of a new subtask does not count.
Requested granularity: <G_COARSE>
Should the assistant predict the next event now? <DECIDE>
```

The fine query is the same string with <G_COARSE> replaced by <G_FINE>. When no subtask is in progress the second line reads `No active subtask is available.`; in this evaluation set the subtask segmentation covers each video without gaps, so the substitution never occurs. Qwen's processor inserts an absolute timestamp before each temporal group of frames; we neither add nor remove these, and the blind control of §**??** removes them together with the frames, since a timestamp discloses elapsed video time and is therefore part of the visual input.

We refer to this template as *symmetric*. An earlier *asymmetric* template, used by several ablations below, supplied the `Current subtask` line only to the fine query and phrased the question differently at each granularity. It is reported where it was the configuration actually run, and it is not the template of any main-paper result.

B.2 CONTENT

The content prompt is the timing prompt without the <DECIDE> token, and generation is forced to begin with <SPEAK> so that the content score is independent of the timing decision. Decoding is greedy, is capped at `?` new tokens, and terminates at the first newline or full stop. Every system receives this prompt verbatim, with the same frames.

B.3 JUDGE

The judge is Qwen3-VL-235B. It receives the reference step and the prediction in shuffled order, without system identity, and returns structured fields whose conjunction is taken in code:

```
{"verb_match": true|false, "object_match": true|false,
 "level": "subtask"|"action"|"neither", "reason": "..."}
```

The judge is not asked for a verdict, for three reasons: the criterion is then auditable, verb and object failures can be counted separately, and a pilot in which the judge returned a single graded verdict was lenient at the "related action" grade. Synonymous verbs and hypernymic objects are delimited by positive and negative examples in the prompt; a hypernym accepted as a match is flagged in the output and counted.

C ZERO-SHOT OUTPUT FORMAT

Two zero-shot systems in Table 2 score at or near zero, and the cause is the output format rather than the content of their predictions. All systems receive the identical prompt of §B and are forced to begin with `<SPEAK>`; the failure is in what they emit after it.

Llama 4 Maverick treats the forced marker as an opening XML tag and closes it: 640 of its 741 outputs are the single word `Yes` and a further 72 are `yes`, so no step description is emitted and the judge has nothing to match. Gemma3-4B fails differently, emitting a paragraph of commentary in which the judge locates a step description at the requested level in 0.011 (coarse) and 0.043 (fine) of outputs.

Appending one sentence of format instruction to the prompt removes the failure: Llama 4 Maverick's object agreement rises from 0.006 to 0.447 at the fine granularity. We do not report that variant in Table 2, because the benchmark's comparability rests on every system receiving the same input, and adding an instruction for the systems that need one would break it. The zero-shot rows should accordingly be read as measuring whether a system follows the protocol *and* predicts the step, and not as an upper bound on what these backbones could do under a prompt tuned to each. The trained rows are unaffected: both trained models terminate on every query and emit a step description at the requested granularity on 0.968–0.980 of fine queries.

D GRANULARITY CONTROL OVER CONTENT

Table 7: Granularity control over content at the 175 COARSE_ONLY points. Each system generates twice from identical frames and identical text differing in one token. The pre-specified criterion asks whether the judge places the two outputs at different levels of description, and "direction" whether the coarse output is the broader of the two. The relative comparison was added after inspecting the results and is marked as such: it asks only which of the two is broader, without requiring the judge to assign either a level.

	pre-specified		relative (post-hoc)	
System	different level	direction	correct	reversed
Gemma3-4B	0.000 [0.000, 0.000]	0.000	0.000	0.034
Qwen3.6-27B	0.086 [0.039, 0.137]	0.097	0.194	0.149
Qwen3-VL-8B	0.143 [0.089, 0.198]	0.189	0.280	0.057
Qwen3-VL-8B, ours	**0.286** [0.209, 0.368]	**0.303**	**0.726**	0.080
w/o label rewriting	0.166 [0.109, 0.226]	0.171	0.623	0.091
Gemma3-4B, ours	0.223 [0.153, 0.299]	0.229	0.663	0.120

Because the fourth joint state does not occur (§A.1), no point asks for an utterance at both granularities, and content cannot be compared across granularities on reference points alone. Table 7 constructs the comparison instead: at the 175 COARSE_ONLY points the fine condition is forced to speak, and its output is judged against the coarse output from the same prefix. The probe is used for analysis only and never for training or selection.

Both trained models place the two outputs at different levels far more often than any zero-shot system, and zero-shot Gemma3-4B never does. The effect appears on both backbones, so control of granularity over content is separable from timing accuracy. Absolute rates remain low, and the relative comparison locates the reason: the trained model orders the pair correctly on 0.726 of points but the judge assigns the two outputs distinct levels on only 0.286. The token shifts the level of description reliably without carrying the coarse output all the way to the subtask level, which is the same limitation that caps coarse accuracy in Table 2.

E WHAT A SIGN-BASED STATISTIC CONCEALS

An apparently simpler statistic than $\mathrm{GranSep}$ is the accuracy of the sign of Δ, macro-averaged over the two divergent states. Table 8 shows why we do not use it. Every zero-shot system carries a

Table 8: Accuracy of the sign of $\Delta = s_{\mathrm{F}} - s_{\mathrm{C}}$ on the two divergent states, macro-averaged, against the mean of Δ in each joint state. Sign accuracy of 100% on one state and 0% on the other is the signature of a constant preference.

	sign accuracy			mean Δ			
System	F-only	C-only	macro	F-only	C-only	b.-silent	GranSep
Qwen3-VL-8B	0.553	0.326	0.440	$+0.02$	$+0.03$	$+0.03$	0.448
Qwen3.6-27B	0.034	0.926	0.480	-0.40	-0.33	-0.33	0.420
Gemma3-4B	0.012	0.994	0.503	-1.46	-1.65	-1.59	0.583
Qwen3-VL-8B, ours	0.858	0.309	0.583	$+0.90$	$+0.43$	$+0.98$	0.675

preference for one granularity that holds across all three joint states and whose direction is arbitrary: two of them prefer the coarse condition at 93% and 99% of COARSE_ONLY points while preferring it at 3% and 1% of FINE_ONLY points, which is the constant-preference degenerate solution in all but name. Removing the frames reverses the direction of these preferences, confirming that they do not originate in the input.

Macro-averaging does place such predictors near 0.5, as intended. It also, however, discards any signal that sits on top of an offset. When the offset between the two granularities exceeds the state-dependent difference, every pair falls on one side of zero and the sign carries no information, however well the two states are separated — and the mean-Δ columns show offsets of 1.5 against state-dependent differences of 0.2. Because GranSep ranks Δ rather than thresholding it, a shared offset leaves it unchanged, and it recovers exactly the state-dependent component. Our trained model illustrates the gap: its macro sign accuracy is 0.583 while its GranSep is 0.675, the difference being the offset of $+0.98$ that the sign statistic charges against it.

F Decision Readout

Table 9: Decision score crossed with prompt template, at 150 steps on a 500-pair subset with data and sampling held fixed.

Score	Template	GranSep	coarse ROC-AUC	fine ROC-AUC
likelihood	asymmetric	0.470 [0.407, 0.526]	0.464	0.580
likelihood	symmetric	0.492 [0.425, 0.555]	0.481	0.618
head	asymmetric	0.579 [0.510, 0.649]	0.558	0.548
head	symmetric	0.608 [0.541, 0.673]	0.595	0.431

Both likelihood-scored configurations lie at chance; only the linear readout produces GranSep intervals excluding it. A readout that inverts one granularity separates the two conditions without discriminating either, and only a criterion requiring both per-granularity statistics to exceed chance rejects it.

The readout is also what makes the coarse granularity learnable at all. Coarse per-decision cross-entropy never left 0.68–0.69 against a chance value of 0.693. At matched sampling the readout raises coarse AUROC from 0.464 to 0.558, and at full scale to 0.692. <SPEAK> and <SILENT> share their first token and are determined at their second, so a likelihood score rests on one token position and inherits a lexical prior from pretraining.

G Learning Rate of the Decision Readout

At the adapter's rate the zero-initialised readout barely moves (its norm is 0.066 after 400 steps) and the early signal decays to chance. At 10^{-3} and 3×10^{-3} it reaches 0.551 and 0.600, the latter still rising at the final step, which is why the main configuration trains the readout two orders of magnitude faster than the adapter. The second and fourth rows share a rate and differ only in template, giving the template's contribution at matched optimisation.

Table 10: Learning rate of the readout and the new token embeddings, at 400 steps on a 500-pair subset. The trajectory column gives GranSep at 100-step intervals.

Template	Readout rate	Trajectory	At step 400	$\|\mathbf{w}\|$
asymmetric	5×10^{-5}	0.571, 0.555, 0.473, 0.500	0.500 [0.442, 0.559]	0.066
asymmetric	1×10^{-3}	0.607, 0.514, 0.517, 0.551	0.551 [0.491, 0.609]	0.792
asymmetric	3×10^{-3}	0.494, 0.494, 0.553, 0.600	0.600 [0.536, 0.659]	1.555
symmetric	1×10^{-3}	0.527, 0.611, 0.638, 0.629	0.629 [0.569, 0.690]	0.786

H Input, Decision Window and Negative Sampling

Table 11: Per-decision cross-entropy under the input protocol, the decision window and the negative sampling, at 150 steps under balanced per-condition sampling, averaged over the last 50. Chance is 0.693.

Factor	Setting	combined	coarse	fine
input	$[t-8, t]$, 16 frames uniform	0.655	0.681	0.628
	$[t-24, t-4]$ 20 frames + 16 at 4 fps	0.655	0.685	0.624
	$[0, t-4]$ 16 frames + 16 at 4 fps	0.656	0.691	0.620
	$[t-24, t]$, 36 frames uniform	0.656	0.686	0.627
window	0.5 s	0.658	0.689	0.627
	1.0 s	0.666	0.689	0.643
	2.0 s	0.658	0.693	0.623
negatives	uniform	0.658	0.689	0.627
	stratified by distance to event	0.623	0.676	0.570

Two hypotheses motivated varying the visual input: that the coarse granularity, observing 8 s of subtasks that unfold over tens of seconds, lacks temporal coverage; and that an inter-frame interval of 0.533 s, exceeding the 0.5 s decision window, leaves the fine granularity without temporal resolution. Neither is supported. All four protocols fall within 0.001 of one another, every step-paired difference is below 0.008 against standard errors of 0.005–0.009, and the loss curves coincide from the first step. That coarse learning is unaffected by extending the window to the whole elapsed video is also what directs the remedy for the coarse granularity to the procedural state rather than to visual coverage.

Widening the decision window does not help either, and would cost timing precision while forcing the benchmark to be recomputed. Stratifying negatives moves the fine granularity by -0.056 ± 0.015 and the coarse one not at all, but the comparison is confounded: cross-entropy is computed under each configuration's own sampling distribution, and stratification removes exactly the negatives on which discrimination is weakest.

I Coarse Label Rewriting

Public procedural corpora label subtask segments with action descriptions rather than outcomes, so the coarse targets available for training express the wrong level of abstraction. We relabel the training corpora's subtask segments with the benchmark's own labelling procedure (§3.3.2), leaving segment boundaries unchanged.

The rewriting changes the style of the coarse output and not its accuracy. Mean output length rises from 3.06 to 4.27 words, the fraction of predictions phrased at the requested level from 0.698 to 0.749, and the fraction of coarse outputs the judge places at the subtask level in the content probe from 0.18 to 0.31 (Table 7), while accuracy moves by -0.034 [-0.092, $+0.024$] at the coarse and $+0.011$ [-0.021, $+0.048$] at the fine granularity. We retain the rewriting because it gives training and evaluation one definition of what a subtask description is, and we report that it does not improve accuracy. The conclusion of §4.3 follows: what limits coarse accuracy is anticipating the next outcome, not phrasing it.

J TRANSFER, CAPACITY AND VOCABULARY

On a held-out split of the training corpora the timing adapter attains $\mathrm{GranSep}$ of 0.90, against 0.675 on the benchmark, and the same ordering holds for both per-granularity AUROCs. The conditioning is therefore learned within the training domain, and the shortfall arises in transfer to activities the model has not seen. Two observations bear on the nature of that shortfall. Vocabulary is not the binding constraint: partitioning the benchmark by whether the reference object appears anywhere in the training labels gives differences in accuracy that are not significant on any system at either granularity. Adaptation capacity is not evidently the constraint either, since the model fits its training domain well; all results use low-rank adaptation, and whether full fine-tuning narrows the transfer gap is open.

K JUDGE VALIDITY

Every content figure in Table 2, Table 4 and Table 7 rests on the judge. It is applied identically to all systems, blind to system identity and with shuffled order, and returns verb and object agreement as separate fields whose conjunction is computed in code (§B). Three limitations remain. The judge shares a backbone with the model that produced the reference subtask labels, which may favour outputs phrased as it phrases them; spot checks show it accepting partial object matches; and the level field it returns is also the basis of the "gran." column, so a systematic bias in that field would affect both the accuracy and the granularity statistics. Agreement between the judge and human ratings on 100 sampled pairs is not yet available, and the absolute content figures should be read as unvalidated until it is; the comparisons between systems, which use one judge instance on one set of points, are less exposed to this.