WHAT'S PAST IS PROLOGUE: BENCHMARKING PROSPECTIVE MEMORY GROUNDED IN RETROSPECTIVE ACTION UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory enables an agent to use experience that is no longer directly observable when making a current decision. Existing multimodal memory evaluations have largely examined whether models retain historical information, answer explicit questions about past events, or execute deferred intentions under specified conditions. A complementary challenge arises in cooperation: an agent must determine which elements of a shared interaction history remain applicable in the current situation and ground them in immediate perception to choose a concrete next action. We introduce **CoopPortal**, a multimodal benchmark for studying this problem in long-horizon two-player cooperative puzzle solving. CoopPortal contains two independently annotated tracks over 118 Portal 2 levels. **Track A** evaluates retrospective action understanding by requiring models to reconstruct the 1,585 key atomic actions that directly contributed to the observed successful solutions from complete dual-player episodes. **Track B** evaluates prospective action prediction under strict partial observability: given only a designated player's first-person video and the shared dialogue before a decision boundary, models must predict the next atomic action in the recorded trajectory, its purpose, and the historical cues reported in support of the prediction. Track B contains 369 decision points and prevents access to the partner's current view and to all future observations. To provide a reproducible reference approach, we propose **APEX**, a training-free inference pipeline that constructs dialogue-anchored multimodal micro-events, distills them into structured experience memory over trials, mechanisms, commitments, and uncertainties, and combines this memory with dense recent visual grounding for action prediction. Evaluations across frontier, open-source, and task-specific video models reveal substantial variation across retrospective reconstruction and prospective action prediction, particularly on complex and long-tail actions. CoopPortal provides a diagnostic setting for studying how multimodal agents transform accumulated cooperative experience into decision-relevant action understanding.

1 INTRODUCTION

Memory allows an agent to use information that is no longer directly observable when making a current decision. During a long interaction, an agent accumulates visual observations, dialogue, action outcomes, and environmental changes, only a small portion of which may matter at a given moment. Recent benchmarks have made substantial progress in evaluating whether multimodal models can retain and retrieve such information through long-video question answering, streaming interaction, and deferred-intention tasks (Einstein & McDaniel, 1990; Park et al., 2023; Shi et al., 2026; Long et al., 2025). These evaluations establish whether historical information remains available when queried or triggered, providing an important foundation for studying long-horizon memory.

However, availability alone does not determine whether memory can support action. In cooperative tasks, past experience is not merely a collection of facts to recall: a failed attempt can rule out an approach, an observed mechanism can determine which operation is feasible, and a verbal agreement can make one participant's action contingent on another's progress. Moreover, these constraints may change as the environment evolves or the team revises its plan. The missing question is therefore whether a model can determine which available historical evidence remains applicable at the current

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.

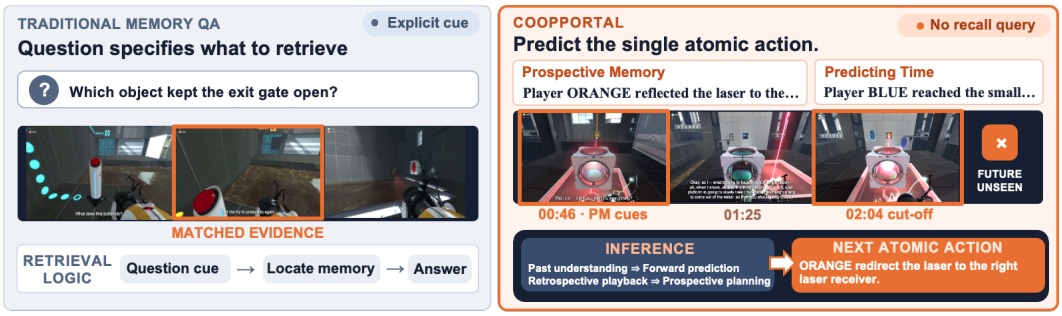


Figure 1: Comparison between the Traditional Memory QA and CoopPortal.

decision point and combine it with immediate perception to predict a concrete cooperative action. As illustrated in Figure 1, existing tasks often specify a relevant query, trigger condition, or deferred intention, whereas our setting requires the model to identify the evidence relevant to a designated action prediction from the available history.

We introduce **CoopPortal**, a multimodal benchmark built from Portal2 two-player cooperative puzzle-solving videos. CoopPortal contains two independently annotated tracks over the same source levels: **TrackA** evaluates retrospective action understanding by asking a model to reconstruct the key actions that directly contributed to an observed successful solution from a complete dual-player episode; **TrackB** evaluates prospective action prediction by asking a model to predict the designated player's next atomic action, its purpose, and supporting historical cues from a temporally truncated first-person view and dialogue history. These tasks require models to compress long and redundant history, align temporally offset dialogue with physical execution, and select evidence by its functional role in the current decision. To address these challenges in TrackB, we propose **APEX**, a training-free inference pipeline with three components: dialogue-anchored multimodal evidence construction extracts informative visual-dialogue micro-events; structured experience memory organizes them into trials, mechanisms, commitments, and uncertainties; and experience-grounded action reasoning combines this memory with dense recent visual observations to predict the next action.

Our contributions can be summarized as:

- We present CoopPortal, a multimodal benchmark comprising two complementary tracks that separately evaluate retrospective action understanding and prospective action prediction , together with an evaluation protocol addressing class imbalance, semantic ambiguity, and the decoupling of evidence fidelity from action correctness.
- We propose APEX, an architecture-agnostic, training-free pipeline built on adaptive expansive windowing that first constructs a structured understanding of history and then grounds it in the current state for next-action prediction, without relying on explicit retrieval instructions.
- We conduct a comprehensive evaluation of frontier multimodal models on CoopPortal, surfacing diagnostic findings including the saturation gap on class-balanced metrics, a capability watershed on complex manipulation actions, and the separability of prospective memory contribution through controlled ablation.

2 Related Work

Spatio-Temporal and Action-Decision Benchmarks. Long-video comprehension benchmarks such as Ego4D (Grauman et al., 2022), EgoSchema (Mangalam et al., 2023), and Video-MME (Fu et al., 2024) mainly evaluate retrospective retrieval and temporal alignment through multiple-choice question answering. RIVER (Shi et al., 2026) extends evaluation to streaming perception and proactive response, but still relies on explicit natural-language queries rather than action selection. In contrast, cooperative-game benchmarks such as Overcooked (Carroll et al., 2019) and TeamCraft (Long et al., 2024) require spatial actions but largely emphasize short-horizon or fully observable settings, while GPTNT (Parekh et al., 2025) focuses on real-time communication rather than grounding dormant historical cues. Consequently, existing benchmarks do not test whether agents can implicitly retrieve and ground unprompted multimodal history to predict atomic next actions under restricted first-person

observability. CoopPortal fills this gap with Critical Action Reconstruction (Track A) and Next-Action Prediction (Track B), targeting retrospective and prospective action reasoning, respectively.

Long-Term Memory and Action-Oriented Reasoning. Memory-augmented architectures extend the temporal reasoning and planning capabilities of multimodal agents. Generative Agents (Park et al., 2023), MemGPT (Packer et al., 2023), and Voyager (Wang et al., 2023) use external memory operations, including retrieval and reflection, to maintain behavioral coherence beyond limited context windows. M3-Agent (Long et al., 2025) further organizes multimodal episodic and semantic memories into entity-centric repositories for multi-hop reasoning over long videos, while recent surveys (Hu et al., 2025) advocate structured, operation-aware architectures beyond flat episodic caches. However, existing approaches remain limited for prospective action reasoning: (1) similarity, recency, or entity-based organization cannot reliably distinguish failed attempts, verified mechanisms, and coordination commitments, causing plan–fact confusion; (2) insufficient separation between historical memory and immediate affordance verification allows stale evidence to override current observations; and (3) retrieve-then-generate pipelines evaluate historical relevance separately from action feasibility. APEX addresses these limitations through dialogue-anchored adaptive evidence construction, structured experience distillation, and single-pass evidence selection and action prediction.

3 THE COOPPORTAL BENCHMARK

We introduce **CoopPortal**, a dual-track multimodal benchmark evaluating whether agents can transform interaction history into decision-relevant action understanding in cooperative environments. Built upon 11.5 hours of synchronized *Portal 2* gameplay from the Portal Dialogue Corpus (Tomlin et al., 2025), CoopPortal evaluates both retrospective action understanding and prospective action prediction under partial observability.

3.1 DATA CONSTRUCTION AND ANNOTATION

Data collection and standardization. Starting from 153 dual-player level recordings in the Portal Dialogue Corpus, we build a unified multimodal trajectory index from player-perspective videos, speaker-attributed timestamped transcripts, and game-state data. Each recording is segmented into independent level episodes organized by player-pair session and puzzle course. The two tracks are constructed independently to prevent cross-track information leakage. To reduce lexical variation, we apply a standardized domain lexicon to unify action verbs, game objects, spatial landmarks, mechanism names, and portal colors.

Dual-track decoupled annotation. For Track A, we filter the 153 recordings to retain 111 episodes with confirmed successful completions whose valid action paths can be unambiguously determined. We then annotate the critical atomic action sequence $\mathcal{C}$ for each episode following a *causal minimality principle*: only actions that directly cause a puzzle state transition and constitute a necessary condition for the successful solution are retained; routine movements, ineffective exploration, failed attempts, and redundant operations are discarded. Each critical action identifies its executor as Player BLUE, Player ORANGE, or both, with object, location, and color specified when needed. For Track B, we select decision points t_{cut} that represent clear planning bottlenecks, retaining only the designated player's first-person video and joint dialogue up to that moment. For each sample, we annotate the target player's first immediately executable atomic action a after t_{cut}, its purpose u, and the historical cue set $\mathcal{M}$ that directly and causally supports the prediction. We annotated 390 Track B candidate samples; after quality review, 21 samples failing the admission criteria were removed, yielding 369 decision points.

3.2 DUAL-TRACK TASK FORMULATION

We represent a complete two-player cooperative trajectory as the multimodal observation space $\mathcal{O}_{\text{full}} = (V^{\text{blue}}, V^{\text{orange}}, D)$, where $V^p = \{v_t^p\}_{t=0}^T$ denotes player p's first-person video stream and $D = \{(s_j, \tau_j^s, \tau_j^e, \rho_j)\}_{j=1}^J$ denotes the shared dialogue stream with start time τ_j^s, end time τ_j^e, and speaker identity ρ_j for each utterance s_j. Built on the same raw trajectories, CoopPortal defines two tasks with complementary observation boundaries and temporal objectives: Track A tests retrospective action understanding over complete successful trajectories, while Track B tests prospective action prediction under restricted observations using historical experience.

Track A: Critical Action Reconstruction. Given the full observation $\mathcal{O}_{\text{full}}$, the model must output the chronologically ordered critical atomic action sequence $\hat{\mathcal{C}} = (\hat{c}_1, \ldots, \hat{c}_{\hat{N}})$ along the successful completion path. Each action must identify its executing player and annotate, where necessary, the action object, spatial location, or portal color. **Track B: Next-Action Prediction.** For a designated target player p and decision boundary $t_{\text{cut}} \in (0, T)$, the model may only access that player's first-person video up to t_{cut} and the joint dialogue up to that moment, denoted $\mathcal{O}^B = (V^p_{0:t_{\text{cut}}}, D_{\leq t_{\text{cut}}})$. All video frames and utterances after t_{cut} are strictly excluded to prevent future-information leakage. The model must jointly predict the next atomic action, its purpose, and the historical cues that directly and causally support the prediction: $(\hat{a},\ \hat{u},\ \hat{M}) = \arg\max_Y \mathcal{P}_\theta\big(Y \mid \mathcal{O}^B\big)$, where $\hat{a} \in \mathcal{A}_{\text{slots}}$ is the structured next atomic action with slots for action type, object, location, color, and necessary spatial landmarks; $\hat{u}$ is the action purpose; and $\hat{M}$ is the set of supporting historical cues.

3.3 Evaluation Metrics

The two evaluation tracks are scored independently to assess distinct cognitive dimensions. Below, we outline our protocols, with full details provided in Appendix B.5.

For **Track A**, let $\mathcal{C} = (c_1, \ldots, c_N)$ and $\hat{\mathcal{C}} = (\hat{c}_1, \ldots, \hat{c}_{\hat{N}})$ denote reference and predicted sequences of lengths N and $\hat{N}$, respectively. We score candidate pairs via combined semantic, lexical, and domain-specific signals: $q_{ij} = 0.65E(c_i, \hat{c}_j) + 0.20L(c_i, \hat{c}_j) + 0.15K(c_i, \hat{c}_j)$, where E, L, and K denote sentence-embedding cosine similarity, token Jaccard overlap, and *Portal 2* domain-lexicon overlap. Pairs are eligible if $q_{ij} \geq \tau_{\text{CA}} = 0.65$. Evaluated chronologically, each reference c_i matches the earliest unassigned eligible prediction. Unmatched references or predictions count as omissions or hallucinations. With matched set $\mathcal{M}_{\text{CA}}$ and total quality $Q = \sum_{(i,j)\in\mathcal{M}_{\text{CA}}} q_{ij}$, the primary metrics are: $\text{P}_{\text{CA}} = Q/\hat{N},\ \text{R}_{\text{CA}} = Q/N,\ \text{F1}_{\text{CA}} = 2\text{P}_{\text{CA}}\text{R}_{\text{CA}}/(\text{P}_{\text{CA}} + \text{R}_{\text{CA}})$.

For **Track B**, we evaluate action accuracy, purpose correctness, and prospective-memory evidence separately. ***Action Accuracy*** is measured via SlotScore (attribute precision) and ActionSimt (overall semantic similarity). SlotScore compares extracted slots against non-empty gold slots $\mathcal{S}_{\text{gold}} \subseteq \{\text{action}, \text{object}, \text{location}, \text{color}\}$. For text slots, all-mpnet-base-v2(Song et al., 2020) cosine similarity c_s yields partial credit: $m_s = \mathbb{I}(c_s \geq 0.75) + 0.5 \cdot \mathbb{I}(0.55 \leq c_s < 0.75)$, where $\mathbb{I}(\cdot)$ is the indicator function. The color slot requires exact set matching ($m_s = 1$). Overall scores are computed as:

$$\text{SlotScore} = \frac{1}{|\mathcal{S}_{\text{gold}}|} \sum\nolimits_{s\in\mathcal{S}_{\text{gold}}} m_s, \quad \text{ActionSim} = \max(0, \cos(e_{\text{pred}}, e_{\text{gold}}))\,, \tag{1}$$

where e_{pred} and e_{gold} are text embeddings of the predicted and gold action strings. ***Purpose Correctness*** is evaluated as a binary semantic judgment checking if the prediction shares the same functional objective and direct means-end chain without simply repeating the action. We use Claude Opus 4.8 as an LLM judge across 3 isolated evaluations under a strict rubric, determining final Purpose via majority voting. ***Prospective-Memory Evidence*** evaluates predicted cues $\hat{M}$ (u items) and reference cues $\mathcal{M}$ (v items) by forming a similarity matrix $\mathbf{R} \in \mathbb{R}^{u\times v}$ based on embedding similarity c_{ij}: $R_{ij} = \mathbb{I}(c_{ij} \geq 0.70) + 0.5 \cdot \mathbb{I}(0.42 \leq c_{ij} < 0.70)$. Solving maximum-weight bipartite matching for assignment matrix $\mathbf{X}^*$ yields matched mass $T_{\text{PM}} = \langle \mathbf{R}, \mathbf{X}^* \rangle_F$, where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. The final metrics are: $\text{P}_{\text{PM}} = T_{\text{PM}}/u,\ \text{R}_{\text{PM}} = T_{\text{PM}}/v,\ \text{F1}_{\text{PM}} = 2\text{P}_{\text{PM}}\text{R}_{\text{PM}}/(\text{P}_{\text{PM}} + \text{R}_{\text{PM}})$. Dataset-level results report macro-averages across all decision points.

3.4 Dataset Statistics and Analysis

CoopPortal spans 14 sessions, providing complementary annotations across both tracks. Track A comprises 111 successful full-episode trajectories containing 1,628 critical actions and 18,420 total utterances; episodes average 270.54s in duration, 14.67 critical actions, and 165.95 utterances, requiring models to distill multi-minute trajectories into causally essential actions rather than describing local events.

Track B covers 369 decision points across 108 clips (204 BLUE and 165 ORANGE player perspectives; 3.42 points/clip), with an average observable history of 151.78s. Additionally, Track B integrates 928 *invoked_pm* cues (2.51 cues/sample) to evaluate whether models can simultaneously ground immediate actions, infer functional purpose, and retrieve relevant historical evidence. More detailed dataset statistics are provided in the Appendix C.

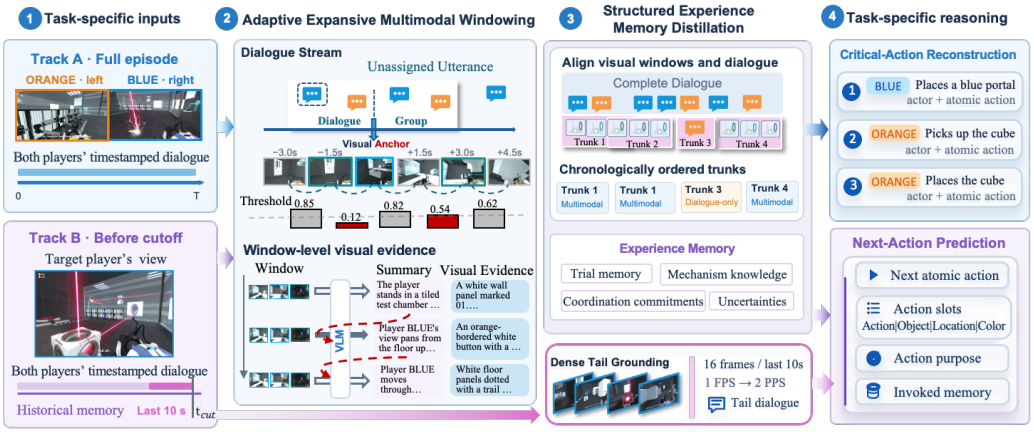


Figure 2: Overview of the APEX framework.

4 METHODOLOGY: APEX

We propose **APEX** (**A**daptive ex**P**ansive **EX**perience memory), a training-free, architecture-agnostic inference pipeline that converts redundant and interleaved multimodal streams into structured temporal experience representations for action planning. As illustrated in Figure 2, the framework operates via three sequential modules: Adaptive Expansive Multimodal Windowing combines dialogue-anchored visual sampling with state-diversity maximization to compress raw streaming inputs into time-aligned micro-events; Structured Experience Belief Distillation maps these micro-events into a four-way causal belief state covering tried strategies, environmental mechanisms, coordination commitments, and unresolved uncertainties to decouple verbal intent from visually grounded execution; finally, Experience-Grounded Action Reasoning fuses this distilled context with dense grounding-zone tail frames in a single forward pass, joint-predicting the next atomic action alongside its supporting causal historical cues.

4.1 PROBLEM SETUP

Let the full multimodal observation of a cooperative episode be $\mathcal{O}_{\text{full}} = (\{V^p\}_{p\in\mathcal{P}}, D)$, where $\mathcal{P} = \{\text{BLUE}, \text{ORANGE}\}$, $V^p = \{v_t^p\}_{t=0}^T$ is player p's first-person video, and $D = \{(s_j, \tau_j^s, \tau_j^e, \rho_j)\}_{j=1}^J$ is the speaker-attributed dialogue stream. CoopPortal derives a track-specific observation–target pair $(\mathcal{O}^k, Y^k)$ from this shared trajectory. Track A observes the complete dual-player trajectory, $\mathcal{O}^A = \mathcal{O}_{\text{full}}$, and reconstructs the critical action sequence $Y^A = \mathcal{C}$. Track B observes only the designated player's video and joint dialogue up to a decision boundary t_{cut}, i.e., $\mathcal{O}^B = (V^p_{0:t_{\text{cut}}}, D_{\le t_{\text{cut}}})$, and predicts $Y^B = (a, u, \mathcal{M})$, comprising the next atomic action, its purpose, and its supporting historical cues. APEX operates on the corresponding track-specific observation. For Track B, it applies a temporal partitioning operator Γ_Δ with $t_{\text{hist}} = t_{\text{cut}} - \Delta$, yielding a historical zone $\mathcal{H} = \mathcal{O}^B_{[0,t_{\text{hist}}]}$ and a grounding zone $\mathcal{G} = \mathcal{O}^B_{(t_{\text{hist}},t_{\text{cut}}]}$. The same boundary applies to video and dialogue: $\mathcal{H}$ is distilled into compact causal state estimates, whereas $\mathcal{G}$ retains recent visual evidence and verbal intentions for reasoning over immediately executable actions.

4.2 ADAPTIVE EXPANSIVE MULTIMODAL WINDOWING

Long-horizon videos often contain redundant static frames that exceed context budgets and dilute attention on critical state transitions. To address this, we propose an adaptive sampling mechanism that anchors visual frames at dialogue landmarks and maximizes local state diversity under a bounded context budget. Specifically, the dialogue stream $D_{\le t_{\text{hist}}}$ is partitioned via a greedy grouping algorithm. Starting at unassigned utterance times g_i^s, all utterances beginning within $[g_i^s, g_i^s + \delta)$ are clustered into a group ending at g_i^e. The group midpoint defines the visual anchor $t_i^a = (g_i^s + g_i^e)/2$. Given the available history duration T^r, which equals T for Track A and t_{hist} for Track B, the window budget K scales adaptively: $K = \text{clamp}(\lceil T^r/\tau \rceil,\ K_{\min},\ K_{\max})$, where τ parameterizes

temporal granularity that controls the average history span covered by each window. $K_{\min}$ and $K_{\max}$ explicitly bound window counts per task branch. Anchor count mismatches relative to K are resolved via uniform interpolation or temporal bucketing.

From each anchor t_i^a, we search bidirectionally within maximum radius r at step size Δ_f. Using L_2-normalized DINOv2-Small CLS embeddings ϕ, we select candidate frames f_c satisfying the diversity gate $\phi(f_i^a)^\top \phi(f_c) < \theta$. The resulting triplet $\mathbf{f}_i = (f_i^-, f_i^a, f_i^+)$ at timestamps $t_i^- < t_i^a < t_i^+$ maximizes local visual variance over candidate set $\mathcal{F}$:

$$\max_{\mathbf{f}_i \in \mathcal{F}^3} \sum_{u,v \in \{-,a,+\}} \left(1 - \frac{\phi(f_i^u)^\top \phi(f_i^v)}{\|\phi(f_i^u)\|_2 \|\phi(f_i^v)\|_2}\right) \quad \text{s.t.} \quad t_i^- < t_i^a < t_i^+. \tag{2}$$

If no candidate clears threshold θ, the frame with minimum similarity acts as a fallback. Finally, all dialogue utterances D_i with start times $\tau_j^s < t_i^+$ and end times $\tau_j^e > t_i^-$ are bound to $\mathbf{f}_i$. A vision-language model independently processes each window to yield a chronologically ordered trunk sequence $\mathcal{T} = \{T_1, \ldots, T_K\}$, where each unit $T_i = \langle \mathbf{f}_i, D_i \rangle$ aligns local visual state changes with contextualized dialogue.

4.3 Structured Experience Belief Distillation

Directly reasoning over time-aligned trunks can conflate verbal plans with executed facts and dilute attention near decision boundaries. We therefore distill the trunk sequence $\mathcal{T}$ into a four-way causal belief state:

$$\mathcal{Z} = \mathcal{D}_\theta(\mathcal{T}, D_{(t_{\text{hist}}, t_{\text{cut}}]}) = (\mathcal{Z}^{\text{trial}}, \mathcal{Z}^{\text{mech}}, \mathcal{Z}^{\text{commit}}, \mathcal{Z}^{\text{unc}}).$$

$\mathcal{Z}^{\text{trial}}$ records past actions a_j, observed outcomes $\mathbf{o}_j$, and their status $g_j \in \{\text{Success}, \text{Failure}, \text{Partial}\}$. $\mathcal{Z}^{\text{mech}}$ stores visually verified mechanisms γ_m with their activation conditions $\mathbf{c}_m$ and spatial scopes $\mathbf{s}_m$. $\mathcal{Z}^{\text{commit}}$ captures agreed role assignments d_c, participating players ρ_c, and agreement status $\mu_c \in \{\text{Stated}, \text{Acknowledged}, \text{Unresolved}\}$. Finally, $\mathcal{Z}^{\text{unc}}$ retains unresolved propositions u_n that may affect the next decision. Each entry $z \in \mathcal{Z}$ is linked to a bounded support set $\text{supp}(z) \subseteq \{T_1, \ldots, T_K\}$, with $1 \le |\text{supp}(z)| \le S_{\max}$. Including $D_{(t_{\text{hist}}, t_{\text{cut}}]}$ preserves recent commitments across the history–active boundary, separating verbal intentions from visually grounded execution.

4.4 Dual-Track Inference

The modular design of APEX supports two task-specific inference paths with distinct observation boundaries and temporal objectives.

Track A: Critical Action Reconstruction. Track A evaluates retrospective sequence understanding from the complete dual-player episode. Given the full observation $\mathcal{O}_{\text{full}}$, adaptive windowing constructs the time-aligned trunk sequence $\mathcal{T}_{1:K}^A$. Since the complete episode is observable and no online decision boundary is imposed, Track A bypasses experience-memory distillation and tail grounding. A reconstruction function $\mathcal{S}_\theta$ directly maps the trunks to a chronological sequence of critical atomic actions: $\hat{\mathcal{C}} = (\hat{c}_1, \ldots, \hat{c}_{\hat{N}}) = \mathcal{S}_\theta\big(\mathcal{T}_{1:K}^A\big)$. This inference path requires the model to suppress exploratory behavior and recover the causally necessary action skeleton of the successful solution.

Track B: Next-Action Prediction. Track B evaluates prospective action reasoning under truncated, first-person observation: $\mathcal{O}^B = \{V_{0:t_{\text{cut}}}^p, D_{\le t_{\text{cut}}}\}$. Historical trunks are first distilled into the causal belief state $\mathcal{Z}$. However, $\mathcal{Z}$ alone may reflect stale spatial states and cannot verify immediate physical executability. We therefore sample the grounding zone over $\mathcal{I}_{\text{tail}}$ to construct dense tail frames $\mathcal{F}_{\text{tail}} = \{v_t^p \mid t \in \mathcal{I}_{\text{tail}}\}$, which provide current visual evidence of player state and available affordances. The model then jointly predicts the next atomic action, its purpose, and the supporting historical evidence:

$$(\hat{a}, \hat{u}, \hat{M}) = \arg\max_Y \mathcal{P}_\theta\left(Y \mid \mathcal{Z}, \mathcal{F}_{\text{tail}}, D_{(t_{\text{hist}}, t_{\text{cut}}]}\right). \tag{3}$$

Here, $\hat{a} \in \mathcal{A}_{\text{slots}}$ is the structured atomic action, $\hat{u}$ denotes its purpose, and $\hat{M} \subseteq \mathcal{Z}$ contains the historical entries that justify the prediction. When the current visual state fully determines the action, the model may output $\hat{M} = \varnothing$. This single-pass inference combines long-range causal experience with immediate affordance grounding while avoiding a separate retrieval stage.

Table 1: Main results on CoopPortal. Track A evaluates critical action reconstruction from dual-player episodes. Track B evaluates next-action prediction under truncated single-player observation

Model	Track A			Track B			
	CA-P	CA-R	CA-F1	SlotScore	ActSim	PM-F1	Purpose
Closed-source models							
GPT-5.4	39.66	35.93	35.52	**57.45**	**64.24**	60.10	**58.76**
Gemini-3.7-Flash	**48.84**	**38.75**	**40.55**	55.08	62.82	**62.94**	50.76
Open-source backbone models							
Qwen3-VL-4B	15.25	9.87	11.25	31.21	47.44	48.02	22.37
Qwen3-VL-8B	16.90	11.59	13.00	43.44	54.97	47.91	27.22
Kimi-k2.6	**42.47**	**32.52**	**34.77**	**52.17**	**61.58**	**58.70**	**48.25**
DeepSeek-V4.1-Flash	34.42	24.80	27.26	50.91	60.7	58.01	43.15
MiniMax-M3	31.81	21.89	24.50	48.69	60.57	57.51	53.37
Task-specific trained models							
VideoChat3	40.38	22.23	26.99	27.50	41.34	37.76	21.14
InternVL2.5	22.03	16.16	16.77	29.97	46.36	40.09	18.43
Open-source backbone + APEX (ours)							
APEX (Qwen3-VL-4B)	22.53	**26.02**	21.97	43.02	48.13	49.18	32.14
APEX (Qwen3-VL-8B)	36.91	20.54	24.01	50.90	58.68	54.60	36.04
APEX (DeepSeek-V4.1-Flash)	**39.37**	25.82	**29.70**	**52.63**	**63.38**	**58.44**	**49.05**

5 EXPERIMENT

5.1 EXPERIMENTAL SETTING

Evaluated models. We evaluate nine representative multimodal models, together with our proposed APEX, across three categories. *Closed-source models* include GPT-5.4 and Gemini-3.5-Flash. *Open-source models* include Kimi-K2.6, DeepSeek-V4.1-Flash, MiniMax-M3, Qwen3-VL-4B, and Qwen3-VL-8B. *Video-understanding models* include InternVL2.5-8B (Chen et al., 2024), VideoChat3 (Li et al., 2026). This selection covers models with different accessibility and video-understanding capabilities. We use the term *open-source* to denote models with publicly available weights, without conflating differences in their licenses or deployment interfaces.

Evaluation protocol. Each evaluated model receives the same track-specific input package, including the sampled visual evidence and dialogue context available at the prediction point. For Track A, the input contains the complete dual-player episode and the model reconstructs the critical action sequence. For Track B, the input is restricted to the designated single-player observation and dialogue history before t_{cut}, from which the model predicts the next atomic action. We use the same task instruction and structured output schema across models. The evaluated models directly map this shared evidence to the required prediction. All results are obtained in a training-free, zero-shot setting without benchmark-specific fine-tuning. The full prompts and frame-sampling protocol are provided in the appendix D.2 and appendix E.

Implementation details. The temporal partitioning buffer width is $\Delta = 10$s. For windowing, the maximum dialogue-group width is $\delta = 3$s, the temporal granularity is $\tau = 12$s; the bidirectional search uses step size $\Delta_f = 1.5$s and maximum radius $r = 4.5$s (at most three steps per side), with diversity gate threshold $\theta = 0.80$. The grounding zone is sampled at 1 FPS over $[t_{\text{cut}} - 10\text{s}, t_{\text{cut}} - 5\text{s}]$ (6 frames) and 2 FPS over $[t_{\text{cut}} - 4.5\text{s}, t_{\text{cut}}]$ (10 frames), yielding the index set $\mathcal{I}_{\text{tail}} = \{t_{\text{hist}} + 1.0 \times k\}_{k=0}^{5} \cup \{t_{\text{cut}} - 0.5 \times j\}_{j=0}^{9}$.

5.2 MAIN RESULTS

Table 1 presents the overall results on CoopPortal. The best CA-F1 on Track A is only 40.55, while the highest SlotScore on Track B reaches 57.45, revealing substantial limitations in both retrospective reconstruction and prospective action prediction. Moreover, no model dominates across both tracks or

Table 2: Ablation results on the two CoopPortal tracks.

(a) Track A: Critical Action Reconstruction.

Method	CA-P	CA-R	CA-F1
APEX (ours)	**36.91**	**20.54**	**24.01**
w/o adaptive	30.06	17.93	19.49
w/o merged views	31.27	18.69	21.44

(b) Track B: Next-Action Prediction.

Method	SlotScore	ActSim	PM-F1	Purpose
APEX (ours)	**50.90**	58.68	**54.60**	**36.04**
w/o experience	47.30	56.40	45.55	29.27
w/o adaptive	48.64	**58.81**	50.92	34.42

all Track B metrics, indicating that action understanding, purpose inference, and prospective-memory activation remain distinct challenges.

Gemini-3.7-Flash achieves the best overall performance, with 48.84 CA-P, 38.75 CA-R, and 40.55 CA-F1. Several models nevertheless exhibit substantial precision–recall imbalance; for example, VideoChat3 obtains 40.38 CA-P but only 22.23 CA-R. This suggests that recognizing a few salient actions is considerably easier than reconstructing the complete causal chain of a cooperative solution.

GPT-5.4 performs best in action grounding, action semantics, and purpose inference, whereas Gemini-3.7-Flash achieves the highest PM-F1. Scaling Qwen3-VL from 4B to 8B substantially improves SlotScore and ActionSim but does not improve PM-F1, while video-oriented models remain behind the strongest general-purpose models. These results show that stronger visual understanding or increased model scale alone does not ensure effective use of decision-relevant history.

APEX consistently improves its Qwen3-VL-8B backbone, raising CA-F1 from 13.00 to 24.01 on Track A and improving SlotScore, PM-F1, and Purpose from 43.44, 47.91, and 27.22 to 50.90, 54.60, and 36.04 on Track B. These gains demonstrate the value of structured experience memory and current-state grounding. The remaining gap to frontier models positions APEX as a competitive training-free reference rather than a complete solution to CoopPortal.

5.3 Ablation Study

Table 2 isolates the contributions of APEX's evidence construction and memory design. On Track A, removing adaptive windowing reduces CA-F1 by 4.52 points (24.01 to 19.49), mainly through lower CA-P (-6.85), while removing merged dual-player views lowers CA-F1 by 2.57 points. Both components are therefore important for recovering the causal skeleton of a cooperative solution. On Track B, removing experience memory yields the largest degradation, reducing PM-F1 by 9.05% and Purpose by 6.77%, which confirms the value of distilled historical evidence for prospective reasoning. Adaptive windowing selects historical frames before the fixed tail grounding zone and affects the evidence distilled into this memory. Without it, the retained tail preserves coarse action semantics and immediate purpose (ActSim: 58.81 to 58.68; Purpose: 36.04 to 34.42), but SlotScore and PM-F1 still drop by 2.26 and 3.68 points. Thus, tail observations support local recognition, whereas atomic-action selection and prospective-memory grounding require adaptive access to earlier history.

5.4 Fine-Grained Analysis of Cooperative Reasoning

Cross-Track Decoupling. For each model and player dyad, we identify *A-high* sessions as the top quartile of session-level Track A CA-F1 and measure the fraction of their Track B cuts with bottom-quartile SlotScore. Appendix D.4, Figure 7(a), shows that low-scoring next-action decisions remain in every included model's A-high sessions (16.13–38.24%). This persistent failure indicates that recovering the critical-action skeleton of a completed episode does not guarantee online decisions under truncated first-person observation, which also require phase localization, player-role identification, and immediate affordance grounding.

Evidence-Action Decoupling. SlotScore and PM-F1 are positively but moderately associated across models (Spearman's $\rho = 0.193$–0.352; Appendix D.4). Figure 7(b–c) further reveals an asymmetric pattern: higher PM-F1 consistently coincides with higher mean SlotScore, whereas high SlotScore does not reliably imply high PM-F1. This suggests that relevant historical evidence generally facilitates action selection, while some locally plausible actions can still be inferred from recent observations without correctly grounding the underlying history. Together, these results expose two non-interchangeable conversion gaps in prospective cooperative reasoning: from global solution-path

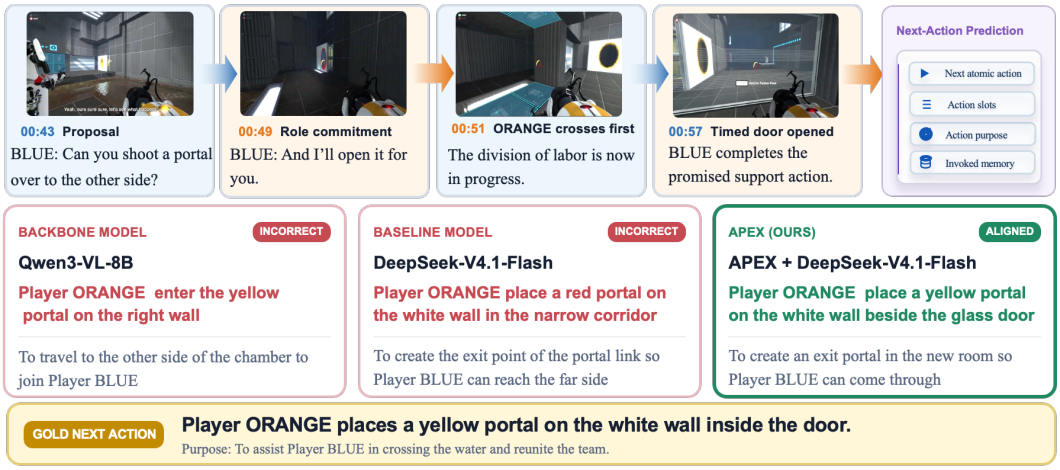


Figure 3: Case Study.

understanding to a state- and player-specific decision, and from relevant historical experience to the correct executable atomic action. Consequently, complete-episode reconstruction, local action plausibility, or evidence selection alone may each overestimate a model's capacity for grounded next-action prediction.

5.5 Case Study

Figure 3 presents a representative Track B example of prospective action prediction. Earlier in the trajectory, Player BLUE proposes that Player ORANGE cross the water by placing a portal on the far side and promises to open the timed door. After Player BLUE fulfills this commitment, Player ORANGE reaches the exit side while Player BLUE remains at the starting side. The next action cannot be recovered from a single past utterance or local cue; it requires reconstructing the accumulated joint state. Since Player ORANGE has already placed a red portal at the starting side, Player ORANGE must place the complementary yellow portal on the reachable white wall inside the door to create a route for Player BLUE. This case shows that retrospective understanding alone is insufficient. Qwen3-VL-8B incorrectly predicts that Player ORANGE should enter the existing yellow portal. DeepSeek-V4.1-Flash correctly predicts portal placement but assigns the wrong color, showing its failure to retain the earlier portal-color state. In contrast, APEX integrates the role commitment, evolving spatial configuration, and current visual affordance to correctly predict the yellow portal placement, enabling Player BLUE to reunite with Player ORANGE.

6 Conclusion

We introduced CoopPortal, a dual-track multimodal benchmark that connects retrospective action understanding with prospective action prediction in long-horizon cooperative environments. By pairing critical action reconstruction from complete dual-player episodes with cue-free next-action prediction under truncated first-person observation, CoopPortal evaluates not only what agents retain from the past, but also whether they can activate and apply relevant experience at the moment of decision. Experiments across frontier and open-weight models reveal persistent capability gaps: reconstructing global solution paths does not reliably yield the correct local action, while identifying relevant historical evidence alone remains insufficient for grounded prediction. We further presented APEX, a training-free framework that integrates adaptive multimodal windowing, structured experience memory, and current-state grounding, consistently improving its open-weight backbone across both tracks. Together, CoopPortal and APEX provide a diagnostic foundation for studying how multimodal agents transform accumulated experience into contextually grounded future actions.

AI USE STATEMENT

Generative AI tools were used solely as assistive tools during manuscript preparation, including language polishing, clarity improvement, and limited code debugging. They were not used to generate experimental results, benchmark annotations, scientific claims, or citations. All research decisions, dataset construction and verification, experimental design, analyses, and final manuscript content were conducted and validated by the authors, who take full responsibility for the work.

REFERENCES

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for Human-AI coordination. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1910.05789.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Gilles O. Einstein and Mark A. McDaniel. Normal aging and prospective memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(4):717–726, 1990. doi: 10.1037/0278-7393.16.4.717. URL https://doi.org/10.1037/0278-7393.16.4.717.

Chaoyou Fu et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. URL https://arxiv.org/abs/2405.21075.

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, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Grauman_Ego4D_Around_the_World_in_3000_Hours_of_Egocentric_Video_CVPR_2022_paper.html.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, et al. Memory in the age of AI agents. *arXiv preprint arXiv:2506.02245*, 2025. URL https://arxiv.org/abs/2506.02245.

Kaini. File: Portal physics-3.svg. Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Portal_physics-3.svg, 2009. Based on derivative works by Pbroks13 and Dammit. Reproduced without modification under CC BY-SA 3.0, https://creativecommons.org/licenses/by-sa/3.0/deed.en.

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, Yuhao Dong, Haoning Wu, Zhiqiu Zhang, Yuandong Yang, Changlian Ma, Qingyu Zhang, Yansong Shi, et al. Videochat3: Fully open video mllm for efficient and generalist video understanding. *arXiv preprint arXiv:2607.14935*, 2026.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025. URL https://arxiv.org/abs/2508.09736.

Qian Long, Zhi Li, Ran Gong, Ying Nian Wu, Demetri Terzopoulos, and Xiaofeng Gao. TeamCraft: A benchmark for multi-modal multi-agent systems in Minecraft. *arXiv preprint arXiv:2412.05255*, 2024. URL https://arxiv.org/abs/2412.05255.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *arXiv preprint arXiv:2308.09126*, 2023. URL https://arxiv.org/abs/2308.09126.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Amit Parekh, Sabrina McCallum, Kareem Al-Hasan, Malvina Nikandrou, Alessandro Suglia, and Ioannis Konstas. GPTNT: Benchmarking real-time collaboration between multimodal agents on Keep Talking And Nobody Explodes. *arXiv preprint arXiv:2606.28514*, 2025. URL https://arxiv.org/abs/2606.28514.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. URL https://arxiv.org/abs/2304.03442.

Pbroks13. File: Portal physics-2.svg. Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Portal_physics-2.svg, 2008. Based on an earlier diagram by Dammit. Reproduced without modification under CC BY-SA 2.5 NL, https://creativecommons.org/licenses/by-sa/2.5/nl/deed.en.

Yansong Shi, Qingsong Zhao, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. RIVER: A real-time interaction benchmark for video LLMs. *arXiv preprint arXiv:2603.03985*, 2026. URL https://arxiv.org/abs/2603.03985.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867, 2020.

Nicholas Tomlin, Naitian Zhou, Eve Fleisig, Liangyuan Chen, Téa Wright, Lauren Vinh, Laura X Ma, Seun Eisape, Ellie French, Tingting Du, et al. Characterizing language use in a collaborative situated game. *arXiv preprint arXiv:2512.03381*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. URL https://arxiv.org/abs/2305.16291.

Wikipedia contributors. Portal (video game). English Wikipedia, https://en.wikipedia.org/wiki/Portal_(video_game). Gameplay section. Accessed: 2026-09-22.

A Game Setting and Terminology

Portal 2 is a first-person cooperative puzzle game in which two players solve a sequence of spatial test chambers by jointly manipulating the environment. This section introduces the game setting, the core portal rules that govern cooperative puzzle solving, and the processing of the original split-screen recordings.

Rule	Implication for play
Paired, bidirectional traversal	The two portals placed by the same player are mutually connected; neither portal is intrinsically an entrance or an exit. Portal placement and portal traversal are therefore distinct operations.
Momentum preservation and directional change	An object retains its speed magnitude while passing through a portal, whereas the orientation of the destination portal redirects its trajectory. Falling into one portal can thus become horizontal or upward motion after exiting the other.
Portal replacement	Creating a new portal of an existing color replaces the previous portal of that color. Players can consequently reconfigure a route while moving, which enables chained traversal and momentum-based maneuvers.

Table 3: Core portal rules relevant to cooperative puzzle solving.

A.1 Portal 2 Setup and Characters

The cooperative campaign takes place in the Aperture Science Enrichment Center, a controlled testing facility composed of self-contained puzzle chambers. Each chamber presents a spatially constrained environment with an exit that becomes accessible only after the players establish the required paths, activate the relevant devices, or transport necessary objects. Progress therefore depends on changing the chamber state rather than merely moving toward the exit.

The two controllable robotic test subjects are ATLAS and P-body. Throughout this paper, we refer to them by the stable color identities displayed in the gameplay footage: **Player BLUE** denotes ATLAS, and **Player ORANGE** denotes P-body. The two players have the same basic abilities, but their observations and immediate opportunities often differ because they occupy different locations and face different parts of the chamber. A mechanism, target surface, or route may be directly visible or reachable to one player while remaining outside the other player's first-person view. Spoken communication is consequently a natural part of play: players report local observations, request actions, assign temporary roles, and coordinate the timing of interdependent operations.

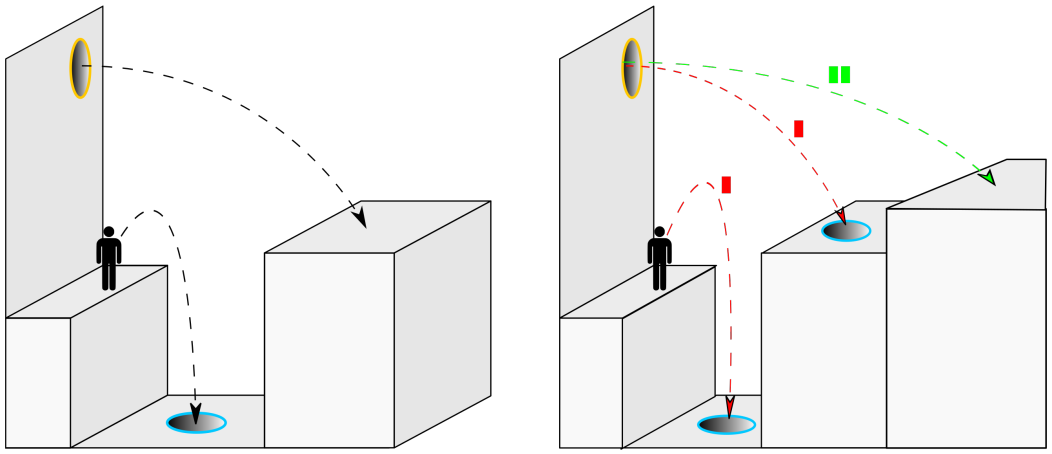
Each player carries a portal device associated with a fixed color pair. Player BLUE can place blue and purple portals, whereas Player ORANGE can place red and yellow portals. These color assignments are persistent within a chamber and identify which player created a particular portal pair.

A.2 Core Puzzle Mechanics

The defining mechanic of the *Portal* series is the placement of two linked portals on compatible white surfaces. A player or movable object entering either end of a portal pair emerges from the other end. This connection can create a shortcut, join otherwise disconnected areas, redirect access to elevated platforms, or transfer an object across a barrier. Table 3 summarizes the three portal rules that are most important for interpreting cooperative play (Wikipedia contributors).

Figure 4 visualizes the two dynamic rules in Table 3. Figure 4(a) illustrates how portal traversal preserves speed magnitude while the orientation of the destination portal redirects the trajectory. Figure 4(b) illustrates portal replacement: recreating a portal of an existing color removes its previous instance, enabling a player to reconfigure a route during motion and extend a momentum-based traversal. The blue/orange colors in these illustrations follow the original *Portal* mechanics and do not denote Player BLUE or Player ORANGE in our cooperative footage.

Chambers also contain devices and objects whose states determine which routes are available. Buttons, floor buttons, and switches can open doors, move platforms, or activate other mechanisms. Some triggers must be held continuously, which requires one player to preserve an environmental state while the other player uses the resulting opportunity. Movable objects, including cubes, balls, and discs, can be carried, placed on floor buttons, inserted into receptacles, or passed between players. Other common elements include lasers and receivers, light bridges, elevators, aerial faith plates, and crushers. These components create spatial and temporal dependencies: for example, one player may establish a portal route, hold a button, or redirect a laser so that the other player can reach a previously inaccessible region.



(a) Momentum preservation and directional change

(b) Portal replacement during traversal

Figure 4: Portal mechanics: (a) momentum preservation and directional change; (b) portal replacement during a momentum-based traversal. Images reproduced without modification from Wikimedia Commons (Pbroks13, 2008; Kaini, 2009).

A.3 Split-Screen Processing into Two Player-Centric Data Streams

The original gameplay recordings use a horizontal split-screen layout that simultaneously shows the two first-person views. The left half of each frame contains Player ORANGE's view, and the right half contains Player BLUE's view. In the source recordings, each frame has a resolution of 1920×600 pixels; we divide it at the vertical midpoint into two 960×600 player-specific frames.

This procedure produces two temporally synchronized first-person video streams, one for Player ORANGE and one for Player BLUE. The streams preserve the original frame order, timestamps, and visual content, so frames at the same timestamp correspond to the two players' local observations of the same underlying chamber state.

The decomposition retains both the shared and complementary properties of cooperative play. The two streams may simultaneously reveal a common event, such as a door opening or a portal being traversed, while each stream can also contain information unavailable from the other view, including the player's local pose, visible objects, nearby surfaces, and immediate route options. The dialogue transcript is retained on the original global timeline, allowing each utterance to be temporally aligned with both player-centric streams.

B Dataset Construction and Annotation

B.1 Annotation Setup and Shared Conventions

Each source recording contains temporally synchronized first-person videos for Player ORANGE and Player BLUE, together with a speaker-attributed dialogue transcript. The two tracks are annotated independently and have different evidence boundaries:

- **Track A: Critical Action Reconstruction** uses the complete two-player trajectory and its full dialogue history. Annotators identify the ordered sequence of critical atomic actions along the successful solution path.
- **Track B: Next-Action Prediction** uses only the target player's first-person observation and the joint dialogue available before a designated decision boundary t_{cut}. The partner's current view and all information after t_{cut} are unavailable.

Across both tracks, annotations use standardized terminology for player identities, game-level actions, objects and mechanisms, spatial landmarks, and portal colors. The complete controlled vocabulary

and its use rules are included directly in the task-specific guidelines below, so that each guideline is self-contained. Annotations must be grounded in observable gameplay and aligned dialogue. When an attribute is not required to distinguish the intended operation or is not supported by the available evidence, it is left unspecified rather than inferred.

B.2 Guidelines for Track A: Critical Action Reconstruction

The following guideline was provided to Track A annotators.

Guideline for Track A: Critical Action Reconstruction

Task. Given the full cooperative gameplay trajectory, annotate the minimal ordered sequence of atomic actions necessary for successful level completion.

Include. Retain an action only if it directly changes the puzzle state, establishes a route or mechanism required later, enables the teammate to progress, or resolves a necessary coordination dependency. Typical examples include placing a portal that creates a required route, activating a button that enables a traversal, carrying a required object to a mechanism, and traversing an established portal route to reach a previously inaccessible region.

Exclude. Do not annotate routine movement, camera adjustment, stationary observation, ineffective exploration, failed attempts, repeated corrections superseded by a later successful operation, or verbal plans without corresponding task-relevant execution.

Action unit, player identity, and format. Each label should record one atomic action with independent game-level significance. Use `Player BLUE` or `Player ORANGE` whenever the action has an identifiable individual executor, and preserve chronological order. For interdependent actions with a meaningful order, annotate the two players' actions separately (e.g., `Player ORANGE holds the floor button.` followed by `Player BLUE passes through the opened door.`). Use `Both players` only when both players perform the same necessary action simultaneously and no meaningful execution order exists. Do not split the low-level controls used to complete a single game-level action: aiming and firing to place a portal should be one portal-placement label.

Controlled terminology.

Category	Canonical terminology and usage
Action	`place`, `enter`, `pass through`, `step on`, `stand on`, `press`, `redirect`, `insert`, `pull`, `pick up`, `drop`, `catch`, `activate`, `launch`, `throw`, `jump`, `carry`. Use `place` for creating or relocating a portal, `enter` for moving into a room, and `pass through` for portal traversal.
Objects and mechanisms	`portal`, `laser`, `floor button`, `button`, `ball`, `disc`, `reflective cube`, `switch`, `laser receiver`, `moving platform`, `platform`, `crusher`. Prefer the most specific supported term.
Location and color	Form locations compositionally from `starting room`, `middle room`, `exit room`, `white wall`, `floor`, `platform`, `light bridge`, `entrance`, `exit`, `left`, `right`, `upper`, `lower`, `near`, and `opposite`. Use `blue`/`purple` for Player BLUE's portals and `red`/`yellow` for Player ORANGE's portals. Add the object, location, and portal color only when required to distinguish the operation.

B.3 Guidelines for Track B: Next-Action Prediction

The following guideline was provided to Track B annotators.

Guideline for Track B: Next-Action Prediction

Task. At the marked decision boundary, annotate the target player's first immediately executable atomic action.

Available evidence. Use only the target player's first-person video and the joint dialogue available before the boundary. Do not use later video frames, later dialogue, the partner's current first-person view, or the eventual puzzle outcome.

Next-action criterion. Write one concrete operation that the target player can immediately perform. Do not write a multi-step plan, a broad task objective, or a future consequence. For example, "place a portal on the reachable white wall" is valid, whereas "place a portal, go through it, and press the button" and "open the exit door" are not.

Output fields and controlled terminology.

Field	Annotation rule
`action`	Use `place`, `enter`, `pass through`, `step on`, `stand on`, `press`, `redirect`, `insert`, `pull`, `pick up`, `drop`, `catch`, `activate`, `launch`, `throw`, `jump`, or `carry`.
`object`	Use `portal`, `laser`, `floor button`, `button`, `ball`, `disc`, `reflective cube`, `switch`, `laser receiver`, `moving platform`, `platform`, or `crusher`; choose the most specific supported term.
`location`	Form locations compositionally with `starting room`, `middle room`, `exit room`, `white wall`, `floor`, `platform`, `light bridge`, `entrance`, `exit`, `left`, `right`, `upper`, `lower`, `near`, and `opposite`.
`color`	Use `blue`/`purple` for Player BLUE's portals and `red`/`yellow` for Player ORANGE's portals.

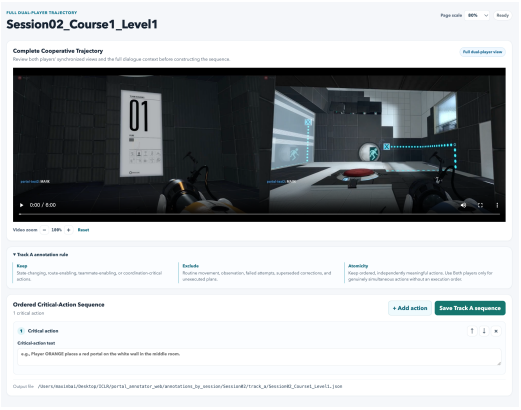
The four fields must match the next-action description; leave any field that is absent from that description empty. Then annotate its direct purpose and the earlier dialogue or gameplay events that directly and causally support the decision. All supporting cues must precede the decision boundary. Leave every unsupported attribute unspecified rather than guessing.

B.4 Annotation Workflow and Quality Control

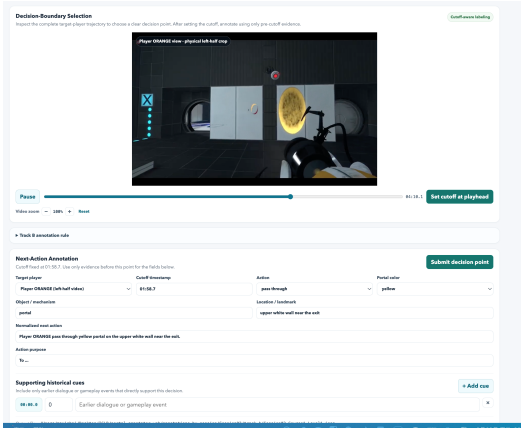
Annotators first receive the track-specific guideline and then complete the task through a dedicated web interface. Track A exposes the complete dual-player trajectory and supports the construction of a chronologically ordered critical action sequence. Track B presents the complete trajectory from the target player's first-person view, from which annotators first identify an appropriate decision boundary t_{cut}. Once the boundary is fixed, the next action, its purpose, and its supporting cues are annotated using only the target player's observation and the joint dialogue available before t_{cut}.

Track A quality control. Each Track A sample is independently annotated by two annotators. After initial annotation, they align, merge, and discuss their respective critical action sequences, checking whether each action satisfies causal minimality, whether atomic-action granularity is consistent, whether the executing player is correct, and whether the temporal order is appropriate. The two annotators discuss discrepant actions and produce a unified sequence. If ambiguity remains, a third annotator joins the discussion and the final label is determined by vote.

Track B quality control. Because different annotators may select different decision boundaries within the same level, the Track B candidate data are initially divided into two subsets and annotated by annotator A and annotator B, respectively. After initial annotation, the two annotators cross-review the samples completed by the other annotator. They check whether the selected cutoff defines a clear decision point; whether the next action is the target player's first immediately executable atomic action after that point; whether the action purpose and supporting cues lie, respectively, on the direct means–end chain and within the valid pre-cutoff evidence boundary; and whether the structured slots are consistent with the next-action description. Samples judged not to satisfy the admission criteria or to contain annotation defects are discussed and revised jointly to establish the final content. If ambiguity remains, a third annotator joins the discussion and the final label is determined by vote. Samples for which an unambiguous label cannot be established under these criteria are excluded from the final benchmark.



(a) Track A annotation interface for critical action reconstruction.



(b) Track B annotation interface for next-action prediction.

Figure 5: Annotation interfaces used to construct CoopPortal. Track A provides the complete two-player trajectory for reconstructing an ordered critical action sequence. Track B provides the target player's complete first-person trajectory for identifying a decision boundary; once the boundary is fixed, the next-action label is constructed only from the observation and dialogue available before that boundary.

B.5 Evaluation Metrics

The two tracks produce different output structures and are evaluated independently.

Track A: critical action sequence matching. Let $\mathcal{C} = (c_1, \ldots, c_N)$ denote the reference sequence and $\hat{\mathcal{C}} = (\hat{c}_1, \ldots, \hat{c}_{\hat{N}})$ the predicted sequence, consistent with the task definition above. Since semantically equivalent actions may differ substantially in wording, we score each candidate pair using complementary semantic, lexical, and game-specific signals:

$$q_{ij} = 0.65E(c_i, \hat{c}_j) + 0.20L(c_i, \hat{c}_j) + 0.15K(c_i, \hat{c}_j), \tag{4}$$

where E is sentence-embedding cosine similarity, L is token-level Jaccard overlap, and K is weighted overlap over a fixed *Portal 2* gameplay lexicon covering players, mechanisms, portal colors, and spatial terms. A pair is eligible only if $q_{ij} \geq \tau_{\mathrm{CA}}$, with $\tau_{\mathrm{CA}} = 0.65$.

We process reference actions in chronological order. For each c_i, the evaluator selects the earliest-indexed eligible prediction that has not already been assigned; each reference and predicted action can be matched at most once. Unmatched reference actions are omissions, while unmatched predictions are hallucinations. If $\mathcal{M}_{\mathrm{CA}}$ is the resulting match set and $Q = \sum_{(i,j)\in\mathcal{M}_{\mathrm{CA}}} q_{ij}$, the primary metrics are

$$\text{CA-P} = \frac{Q}{\hat{N}}, \qquad \text{CA-R} = \frac{Q}{N}, \qquad \text{CA-F1} = \frac{2\,\text{CA-P}\,\text{CA-R}}{\text{CA-P} + \text{CA-R}}. \tag{5}$$

This soft matching formulation gives more credit to high-quality alignments while penalizing both missing and unsupported actions.

Track B: next-action prediction. We evaluate action accuracy, purpose correctness, and prospective-memory evidence separately.

Action accuracy. We use two complementary measures. SlotScore tests whether the action is grounded in the required entities and spatial attributes, while ActionSim captures semantic agreement of the complete action description. The slot score compares the model-provided slots against the non-empty reference slots in $\mathcal{S}_{\mathrm{gold}} \subseteq \{\texttt{action}, \texttt{object}, \texttt{location}, \texttt{color}\}$. For action, object, and location, we encode the reference and predicted values with `all-mpnet-base-v2`

and convert cosine similarity c_s into partial credit:

$$m_s = \begin{cases} 1, & c_s \geq 0.75, \\ 0.5, & 0.55 \leq c_s < 0.75, \\ 0, & c_s < 0.55. \end{cases} \tag{6}$$

The color slot receives 1 only when the extracted portal-color sets match exactly. Missing predictions for non-empty reference slots receive 0. The sample-level score is

$$\text{SlotScore} = \frac{1}{|\mathcal{S}_{\text{gold}}|} \sum_{s \in \mathcal{S}_{\text{gold}}} m_s. \tag{7}$$

We additionally compute

$$\text{ActionSim} = \max(0, \cos(e_{\text{pred}}, e_{\text{gold}})), \tag{8}$$

where e_{pred} and e_{gold} are embeddings of the complete predicted and reference `next_action` strings. Thus, SlotScore exposes errors in critical action attributes that a broadly similar sentence may conceal, whereas ActionSim remains sensitive to the overall semantic description.

Purpose correctness. We treat purpose evaluation as a binary semantic judgment. A prediction is correct only if it identifies the same beneficiary and functional objective as the reference, lies on the same direct means-end chain, and does more than restate the action. We use Claude Opus 4.8 to score each example three times in isolated judging contexts under a fixed rubric. Each judgment returns 0 or 1, and the final PURPOSE score is determined by two-out-of-three majority vote. To assess the reliability of the LLM-based purpose judge, we randomly sampled 100 predictions generated by Gemini 3.7 Flash. Two human annotators independently labeled each prediction as purpose-correct (1) or purpose-incorrect (0), following the same rubric used by the automated judge. We then compared each annotator's labels separately with the final Claude Opus 4.8 decision, which was obtained by majority vote over three independent judging runs. Claude Opus 4.8 achieved 81.00 agreement with Annotator A ($\kappa = 0.6200$) and 89.00 agreement with Annotator B ($\kappa = 0.7764$), indicating substantial agreement with independent human judgments.

Prospective-memory evidence. The predicted cue set $\hat{M} = \{\hat{m}_i\}_{i=1}^{u}$ and reference set $\mathcal{M} = \{m_j\}_{j=1}^{v}$ are evaluated as unordered sets. We first construct a pairwise score matrix $\mathbf{R} \in \mathbb{R}^{u \times v}$, where each entry is based on the cosine similarity c_{ij} between cue embeddings:

$$R_{ij} = \begin{cases} 1, & c_{ij} \geq 0.70, \\ 0.5, & 0.42 \leq c_{ij} < 0.70, \\ 0, & c_{ij} < 0.42. \end{cases} \tag{9}$$

We then solve a maximum-weight one-to-one bipartite matching. Let $\mathbf{X}^* \in \{0,1\}^{u \times v}$ denote the optimal assignment matrix, subject to at most one match per predicted or reference cue. The soft matched-evidence mass is the Frobenius inner product

$$T_{\text{PM}} = \langle \mathbf{R}, \mathbf{X}^* \rangle_F. \tag{10}$$

The prospective-memory metrics are

$$\text{PM-P} = \frac{T_{\text{PM}}}{u}, \qquad \text{PM-R} = \frac{T_{\text{PM}}}{v}, \qquad \text{PM-F1} = \frac{2\,\text{PM-P}\,\text{PM-R}}{\text{PM-P} + \text{PM-R}}. \tag{11}$$

Dataset-level results report the mean SlotScore, ActionSim, Purpose, PM-P, PM-R, and PM-F1 over all decision points.

C Dataset Statistics and Visualizations

C.1 Trajectory Coverage and Temporal Scale

Table 4 characterizes the source coverage and temporal scale of the two tracks. CoopPortal spans 14 cooperative gameplay sessions in both tracks. Track A contains 111 complete level trajectories, with a median duration of 249.03 s (IQR: 172.78–345.83 s). Track B is derived from 108 underlying clips

Table 4: Trajectory coverage and temporal scale of CoopPortal. Track A uses complete dual-player level trajectories. Track B uses pre-cutoff single-player observations, and multiple decision points may originate from the same source clip. IQR denotes the interquartile range.

Statistic	Track A	Track B
Sessions	14	14
Source units	111 episodes	108 clips
Evaluation instances	111 trajectories	369 decision points
Units / session (median [IQR])	8 [6.25–9]	8 [6.25–8.75]
Instances / session (median [IQR])	8 [6.25–9]	21.5 [17.25–30]
Time span, mean (s)	275.45	151.78
Time span, median [IQR] (s)	249.03 [172.78–345.83]	113.00 [64.00–203.00]
Time span, range (s)	85.07–776.10	12.00–682.00
Decision points / clip (median [IQR])	–	3 [3–4]
Decision points / clip (range)	–	1–10

Table 5: Annotation structures of the two CoopPortal tracks. Track A actions are categorized by annotated executor. Track B slot counts include only valid non-empty values; color is present only for portal-related operations that require color to disambiguate the action.

(a) Track A

Statistic	Value
Complete trajectories	111
Critical actions	1,628
Avg. actions / trajectory	14.67
BLUE-only actions	716
ORANGE-only actions	748
Joint actions	164

(b) Track B

Statistic	Value
Decision points	369
BLUE targets	204
ORANGE targets	165
Next-action labels	369
Purpose labels	369
PM cues	928
Avg. PM cues / point	2.51
Median PM cues / point [IQR]	2 [2–3]
Non-empty action slots	369
Non-empty object slots	368
Non-empty location slots	295
Non-empty color slots	254

and contains 369 decision points. Its evaluation instances are therefore temporally correlated cuts rather than independent episodes: each source clip contributes a median of 3 decision points (IQR: 3–4; range: 1–10).

The observable history available at a Track B decision point is heterogeneous, with a median length of 113.00 s (IQR: 64.00–203.00 s) and a maximum of 682.00 s. The temporal summaries in Table 4 show that both tracks cover broad time ranges rather than a fixed short-horizon setting, while the clip-level accounting for Track B makes explicit that multiple decisions can arise within one cooperative trajectory.

C.2 Critical Action Structure in Track A

Track A represents each successful level as a chronologically ordered sequence of causally necessary atomic actions. Table **??** summarizes the resulting action structure. The 111 complete trajectories contain 1,628 critical actions, or 14.67 actions per trajectory on average. The annotations are distributed across both players, with 716 BLUE-only actions and 748 ORANGE-only actions; the remaining 164 actions are joint operations involving both players and are counted once. This composition reflects that successful reconstruction requires retaining ordered cross-player dependencies while excluding exploration, failed attempts, and other non-essential behavior. The example below shows the complete five-action solution skeleton for one level, in which the two players alternately open paths, configure portals, and complete the final traversal.

Example: complete critical action sequence (Session14, Course1, Level1)

1. Player BLUE steps on the red button on the right side of the floor in the main room to open the door of the middle room.
2. Player ORANGE places a red portal on the floor of the starting room.
3. Player BLUE steps on the red button on the left side of the floor to open the exit door.
4. Player ORANGE places a yellow portal on the white wall in the exit room.
5. Player BLUE passes through the red and yellow portals to reach the exit.

C.3 Prospective-Memory Structure in Track B

Each Track B instance pairs a pre-cutoff first-person observation with one immediately executable next action, its functional purpose, and a set of causally supporting historical cues. Table 5b summarizes this annotation structure. The 369 decision points contain 928 prospective-memory cues, with 2.51 cues per decision point on average (median: 2; IQR: 2–3). The target-player distribution is balanced across the two roles, with 204 BLUE and 165 ORANGE instances. Each decision point has one action and one purpose label; its action is further grounded by the action, object, location, and color slots whenever these attributes are required to disambiguate the immediate operation.

The example below illustrates a complete Track B annotation at a decision boundary. The target action is supported jointly by an established portal state and two preceding coordination utterances, while the purpose specifies the direct collaborative outcome rather than restating the action. This structure separates the immediate operation to be predicted from the historical information that causally justifies it.

Example: prospective-memory-grounded next-action prediction (Session10, Course1, Level2)

Target player: ORANGE **Decision boundary:** 70.0 s
Next action: Place a yellow portal on the white wall on the far side.
Purpose: Allow Player ORANGE to reach the room on the far side.
Action slots: action = place; object = portal; location = white wall on the far side; color = yellow.
Supporting historical cues:

1. Player ORANGE placed a red portal on the white wall on the starting side.
2. Player BLUE proposed that Player ORANGE place an orange portal and a yellow portal on the white wall on the starting platform and the white wall on the far-side platform.
3. Player BLUE asked Player ORANGE to project a yellow portal on Player BLUE's blue portal.

D Experimental Details

D.1 Model Details

We evaluate a diverse set of baseline models spanning proprietary general-purpose multimodal models, open-weight multimodal models, and dedicated video-understanding models. This diversity tests whether performance on CoopPortal is attributable to general multimodal reasoning, publicly reproducible model capacity, or video-specialized architectural and training choices. All baselines are evaluated in a training-free, zero-shot setting. They receive the same track-specific visual and dialogue evidence; no model is fine-tuned on CoopPortal or given benchmark-specific demonstrations.

GPT-5.4, Gemini-3.7-Flash, Kimi-K2.6, DeepSeek-V4.1-Flash, and MiniMax-M3 are accessed through API-based inference. Qwen3-VL-4B, Qwen3-VL-8B, InternVL2.5-8B and VideoChat3 are evaluated using locally deployed inference. To ensure a fair comparison, the benchmark constructs the visual and dialogue input package before model invocation; differences in results therefore do not arise from model-specific access to additional episode evidence.

The final reproducibility table will report the exact model or checkpoint identifier, API version or local serving framework, access date, decoding parameters, maximum visual input budget, and hardware

Table 6: Baseline models and their inference modes.

Category	Baseline	Inference mode
Closed-weight multimodal models	GPT-5.4	API-based inference
	Gemini-3.7-Flash	API-based inference
Open-weight multimodal models	Kimi-K2.6	API-based inference
	DeepSeek-V4.1-Flash	API-based inference
	MiniMax-M3	API-based inference
	Qwen3-VL-4B	Locally deployed inference
	Qwen3-VL-8B	Locally deployed inference
Video-understanding models	InternVL2.5-8B	Locally deployed inference
	VideoChat3	Locally deployed inference

configuration. These implementation-level fields are separated from the prompt specification, which is provided in a later appendix section.

D.2 Input Construction and Frame Sampling

All methods receive an identical, track-specific input package constructed from temporally aligned gameplay video and dialogue. The construction procedure enforces the observation boundary of each track before any model is invoked. Figure 6 visualizes the complete frame-selection process for one representative Track B sample, from eligible dialogue anchors to the final chronologically ordered visual input. In particular, visual frames and dialogue occurring after the designated prediction point are never provided for Track B.

Track-specific observation protocol. Track A: Critical Action Reconstruction. The input contains the complete episode, including synchronized visual observations from both players and the episode dialogue. The model must retrospectively reconstruct the critical action sequence that forms the causal backbone of the successful cooperative solution.

Track B: Next-Action Prediction. The input is limited to the designated player's first-person observation and the dialogue history strictly preceding the prediction point t_{cut}. The target action, subsequent visual evidence, and subsequent dialogue are excluded. As illustrated in Figure 6, all selected frames lie on or before the cutoff; the visualization marks the cutoff explicitly and separates the excluded future segment from the admissible observation history.

Dialogue-driven frame selection. Rather than uniformly sampling the video at a fixed interval, we use dialogue-driven sampling to compress each episode into temporally informative visual evidence. Dialogue onset is used as the primary anchor because utterances frequently coincide with cooperative instructions, state confirmation, or updates to the shared situation. For each non-empty subtitle segment, the sampler extracts its start time as a candidate timestamp. Track A pools dialogue anchors from both players to support reconstruction of cross-player dependencies, whereas Track B uses anchors from the acting player's subtitle stream to prioritize the evidence available to the player whose next action is predicted. Figure 6 shows these dialogue-derived candidate timestamps together with their corresponding sampled frames for a complete example.

For each sample, only candidate timestamps within its observable range are retained. In Track B, this range is $0 \leq t \leq t_{\text{cut}}$, which prevents any post-cutoff visual evidence from entering the input. The sorted candidate timestamps are then deduplicated with a minimum gap of 1.5 s: when two consecutive candidates are less than 1.5 s apart, only the earlier one is retained.

Decision-point anchor and frame-budget control. To ensure that the model observes the game state immediately before the prediction point, we explicitly add a decision-point anchor

$$t_{\text{end}} = \max(t_{\text{cut}} - 0.2, 0.0). \tag{12}$$

This anchor is merged with the dialogue timestamps and deduplicated again using the same 1.5 s minimum gap. It is therefore not merely an additional frame: it guarantees that the sampled sequence

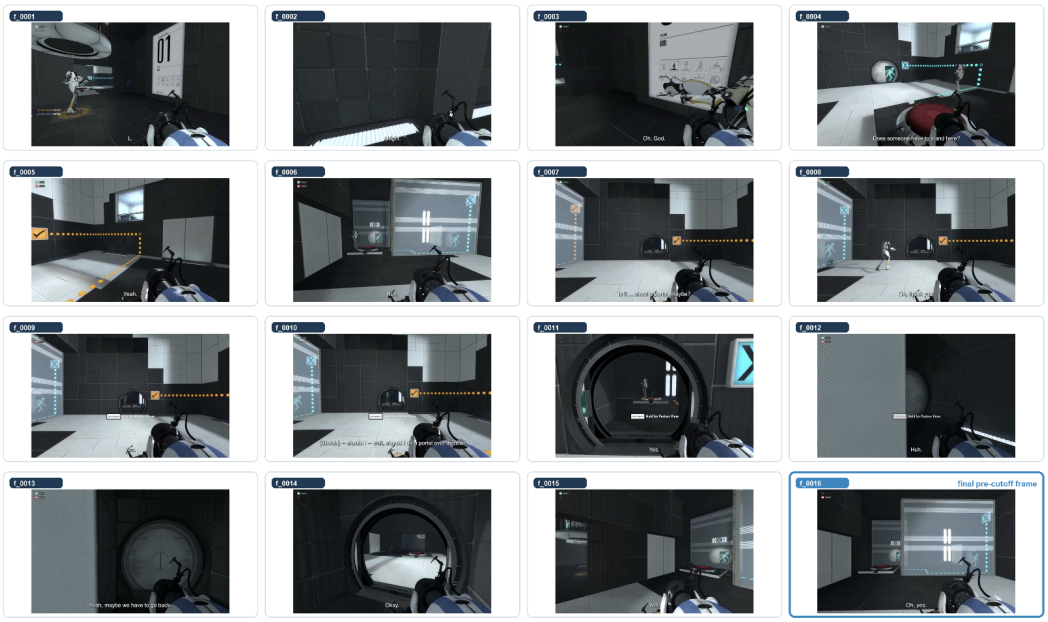


Figure 6: Complete dialogue-driven frame sampling for a Track B example. The figure shows every selected first-person frame in chronological order after dialogue-anchor selection, temporal deduplication, and frame-budget control. The final highlighted frame is the pre-cutoff decision anchor, which preserves the visual state immediately before prediction.

includes visual evidence close to the decision boundary. In Figure 6, this final pre-cutoff frame makes the immediate local state visible even when the most recent dialogue anchor occurs earlier.

We constrain the number of sampled frames to the interval $[6, 28]$. If the merged set contains fewer than six timestamps, we add uniformly spaced timestamps over $[0, t_{\text{end}}]$ and merge them with the existing anchors. During this completion step, we use the relaxed deduplication threshold $\min(1.5, 0.5) = 0.5$ s to avoid near-duplicate frames while preserving sufficient coverage. If more than 28 timestamps remain, we uniformly subsample the earlier timestamps while always preserving the final decision-point anchor. The complete example in Figure 6 demonstrates that the resulting sequence combines dialogue-relevant moments with broad temporal coverage under a bounded visual-input budget.

Visual-view formatting. For Track A, the sampled visual evidence preserves the two synchronized player views so that the model can recover cross-player action dependencies. For Track B, only the designated player's first-person view is retained. Within each input package, frames are presented in chronological order and are temporally aligned with the available dialogue context, as shown in Figure 6. Each output frame is resized so that its longest side does not exceed 768 pixels. The sampled timestamps are deterministically recorded in a manifest, enabling cache reuse and direct verification of the sampling policy.

D.3 APEX Implementation Details

APEX operates only over the observation interval permitted by the corresponding track. Its additional stages do not expose the model to extra or future observations; instead, they reorganize admissible visual and dialogue evidence into a compact, time-aligned representation before final reasoning.

Adaptive windowing. For Track B, APEX applies adaptive windowing only to the historical interval $[0, t_{\text{hist}}]$, where $t_{\text{hist}} = t_{\text{cut}} - 10$ s; the final 10 s are reserved for immediate grounding. For Track A, the same procedure is applied over the complete observable episode. We sort the eligible subtitle utterances by start time and group consecutive utterances whose start times fall within a 3 s span. Each group defines an anchor at the midpoint between the start of its first utterance and the end

of its last utterance. Given the available historical duration T^r, the target number of windows is

$$K = \mathrm{clamp}\left(\left\lceil \frac{T^r}{12\ \mathrm{s}} \right\rceil, K_{\min}, K_{\max}\right), \tag{13}$$

where $K_{\min} = 6$ and $K_{\max} = 40$ for Track A or 25 for Track B. When the dialogue-derived anchors are insufficient, we insert uniformly spaced anchors; when they exceed the budget, we retain temporally representative anchors by bucketed subsampling.

Around each anchor, we search in both temporal directions with step size 1.5 s and maximum radius 4.5 s. Candidate frames are represented by L_2-normalized DINOv2-Small CLS embeddings. On each side, the first candidate whose cosine similarity to the anchor frame is below $\theta = 0.80$ defines the corresponding window boundary. If no candidate crosses this threshold within the search radius, we use the candidate with the lowest similarity as a static-scene fallback. The resulting windows therefore expand around dialogue-relevant events only when the visual state changes, rather than imposing a fixed-duration visual context.

Time-aligned trunks. Each selected window is serialized as a time-aligned trunk. A trunk records its temporal interval, the visual observation distilled from that interval, and all subtitle utterances temporally overlapping the interval. In Track A, synchronized observations from both players are retained at the same timestamps; in Track B, only the designated actor's first-person observation is retained. The trunks are sorted chronologically and passed to the subsequent experience-distillation stage. This representation keeps dialogue grounded in the visual state in which it was spoken, while allowing dialogue-only intervals to be preserved when they convey a coordination-relevant plan or commitment.

Complete example of time-aligned trunks

Sample: `Session06_Course1_Level3_0053_ORANGE`; **current actor:** Player ORANGE; **history end:** 43.000 s; **cutoff:** 53.000 s.

[T_0001 | 1.436–5.936 s | multimodal]
Visual evidence [w_0001 | 1.436–5.936 s]. Technical schematics display a humanoid robot and then a close-up of a component labelled "P-body" with the annotation "FOR PROTECTION."

- A technical schematic of a humanoid robot is displayed, with annotations indicating dimensions and parts.
- The schematic transitions to a close-up of a component labelled "P-body" with the note "FOR PROTECTION."
- The P-body component is shown separately, with its label visible.

Aligned dialogue. *Player ORANGE:* "We have a P-body now."

[T_0002 | 5.936–11.723 s | dialogue_only]
Aligned dialogue. *Player BLUE:* "Interesting name." *Player ORANGE:* "Or an egg." *Player ORANGE:* "((laughter))" *Player BLUE:* "((laughter))"

[T_0003 | 11.723–14.723 s | multimodal]
Visual evidence [w_0002 | 11.723–14.723 s]. The actor observes a robotic entity near a portal surface, then looks upward at a red laser beam extending from a square platform.

- A robotic entity is visible in the distance near a white portal surface with a blue "X" marker and a vertical line of blue dots.
- The actor faces a dark corridor with a red laser beam extending upward from a square platform; a window is visible to the right.
- The view angles upward to show the red laser beam and the ceiling of the room.
- A yellow curved line, likely indicating a trajectory or path, is visible near the red laser beam.

Aligned dialogue. *Player ORANGE:* "Hello, Team Building 3." *Player BLUE:* "((laughter))" *Player ORANGE:* "Hello." *Player ORANGE:* "Ah, it's the cube."

[T_0004 | 14.723–18.870 s | dialogue_only]
Aligned dialogue. *Player ORANGE:* "I've heard about these." *Player BLUE:* "Oh, huh." *Player ORANGE:* "Wait, can I pick it up?"

[T_0005 | 18.870–24.870 s | multimodal]
Visual evidence [w_0003 | 18.870–24.870 s]. A red laser strikes the upper portal of a portal device and produces sparks while the actor holds a portal gun.

- The device contains a green portal in its lower half and a blue portal in its upper half; the red laser strikes the upper portal surface and produces sparks.
- A white portal surface is visible to the right of the device, with a glowing green energy trail extending from it.
- Part of the portal gun is visible in the lower-right of the actor's view.

Aligned dialogue. *Player ORANGE:* "((gasp))" *Player ORANGE:* "What?" *Player ORANGE:* "Okay." *Player BLUE:* "Yeah, you can." *Player ORANGE:* "This is beautiful, ()." *Player ORANGE:* "Can I, like –"

[T_0006 | 20.747–26.747 s | multimodal]
Visual evidence [w_0004 | 20.747–26.747 s]. The red laser continues to strike the upper portal, producing sparks, while the actor holds a portal gun in a dark tiled room.

- The red laser strikes the upper portal of the device and produces sparks.
- The device has a green portal on its lower surface and a red portal on its upper surface, with the red portal emitting sparks.
- The actor holds a portal gun with a glowing green energy trail.
- The actor faces the device; a white panel is visible to the right.

Aligned dialogue. *Player ORANGE:* "Okay." *Player BLUE:* "Yeah, you can." *Player ORANGE:* "This is beautiful, ()." *Player ORANGE:* "Can I, like –" *Player ORANGE:* "What do you do with him?" *Player ORANGE:* "Do you – do you –"

[T_0007 | 29.266–32.266 s | multimodal]
Visual evidence [w_0005 | 29.266–32.266 s]. A laser strikes a portal and the actor then moves forward through the environment.

- A red laser strikes the upper portal of a portal device, causing sparks to emit from the portal.
- The actor holds a portal gun and is positioned to the right of the portal device.
- The actor moves forward away from the device, revealing a distant white cube and a character in the room.

Aligned dialogue. *Player BLUE:* "Yeah yeah yeah yeah, it's supposed to reflect – reflect the laser." *Player ORANGE:* "Ah."

[T_0008 | 34.696–39.196 s | multimodal]
Visual evidence [w_0006 | 34.696–39.196 s]. The actor fires a laser at a portal cube, causing green light and sparks, after which a robotic entity appears in the room.

- A red laser is directed at a white portal cube, which emits a green glow and sparks upon impact.
- The actor holds a portal gun and is positioned to the right of the portal cube.
- After the laser strikes the portal, a robotic entity appears in the room facing the actor.

Aligned dialogue. *Player BLUE:* "So the laser is supposed to, like, hit this thing." *Player ORANGE:* "Can I go through a portal?" *Player BLUE:* "And I think, yeah, I think um what it should do is just put the portal –"

[T_0009 | 39.196–53.000 s | dialogue_only | recent]
Aligned dialogue. *Player ORANGE:* "Oh." *Player ORANGE:* "Shit I should –" *Player ORANGE:* "Yeah." *Player BLUE:* "Can you see it?" *Player ORANGE:* "I'm turning it." *Player BLUE:* "So like you turn it." *Player BLUE:* "And then –" *Player BLUE:* "Yep."

Structured experience memory. The chronologically ordered trunks are distilled into a compact historical experience memory that is explicitly separated from the current visual state. The memory retains four complementary forms of decision-relevant evidence: *trial memory* records attempted actions and their observed outcomes; *mechanism knowledge* records visually grounded environment regularities; *coordination commitments* records stated or unresolved division-of-labor plans; and *relevant uncertainties* records unresolved hypotheses that may still affect the next decision. Each retained item is grounded in a bounded set of supporting trunks, with at most $S_{\max} = 5$ support items per entry. The final prediction stage receives this compressed historical memory together with the recent dialogue and immediate visual grounding evidence; the memory itself does not assert the current physical state.

Example of structured historical experience memory

Trial memory

- Player BLUE aimed a red laser at a far-wall target, activating a portal on the wall and illuminating the lower portal of the device with pink light.
- Player ORANGE placed a portal while Player BLUE stood in front of the laser, indicating an unresolved attempt to redirect the beam.

Mechanism knowledge

- A red laser striking a wall target can activate a portal and illuminate the lower portal of the device with pink light. *[observed]*
- Portals can be placed on visible, portalable walls or floor surfaces with glowing indicators. *[observed]*

Coordination commitments

- Player ORANGE instructed Player BLUE to place a portal where the laser was pointing and then place another portal elsewhere. *[stated]*
- Player BLUE was instructed to stand in front of the laser while Player ORANGE placed a portal. *[stated; unresolved]*

Relevant uncertainties

- It remains unclear whether the laser redirection or portal placement achieved the goal of obtaining two boxes.
- The precise mechanism for activating the portal with the laser is not yet confirmed.

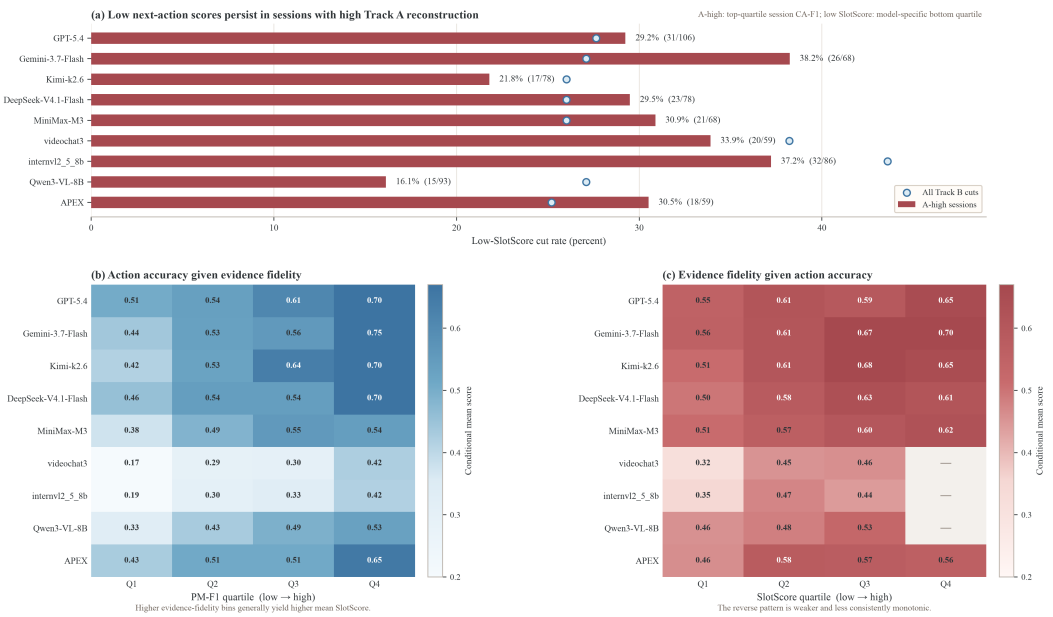


Figure 7: **Fine-grained cooperative reasoning diagnostics. (a) Cross-track decoupling:** bars show the fraction of Track B decision points whose SlotScore falls in the model-specific bottom quartile, restricted to sessions with top-quartile Track A CA-F1; open circles show the corresponding rate over all Track B decision points. **(b) Evidence-to-action conditioning:** mean SlotScore conditional on tie-aware PM-F1 quartiles. **(c) Action-to-evidence conditioning:** mean PM-F1 conditional on tie-aware SlotScore quartiles. Panels (b) and (c) jointly show that the two Track B signals are related but non-interchangeable.

D.4 Fine-Grained Cooperative Reasoning Diagnostics

This appendix provides the complete protocol and visual evidence for the fine-grained analyses in Section **??**. The diagnostics do not estimate causal effects between metrics. Instead, they test whether success at an upstream reasoning stage reliably eliminates failures at the subsequent decision stage.

Cross-track conditional failure. For each model and player dyad, we compute the mean Track A CA-F1 over all completed levels in a session and the mean Track B scores over all truncated decision points in the same session. We define an *A-high* session as one in the top quartile of session-level CA-F1 within that model. We then define a *low-Slot* decision point using the model-specific, tie-aware bottom quartile of SlotScore over its full Track B evaluation set. Figure 7(a) compares the low-Slot rate in all Track B cuts with that restricted to A-high sessions. The latter remains nonzero for every evaluated model: even where a model reconstructs the global critical action skeleton well, it can still fail to localize the current phase, identify the acting player, or ground the currently available affordance into the correct next atomic action.

Sample-level evidence–action association. We analyze the continuous relation between SlotScore and PM-F1 without introducing an arbitrary correct/incorrect threshold. Figure 7(b) reports $\mathbb{E}[\text{SlotScore} \mid \text{PM-F1 quartile}]$, whereas Figure 7(c) reports $\mathbb{E}[\text{PM-F1} \mid \text{SlotScore quartile}]$. The model-wise per-sample scatterplots and Spearman rank correlations are provided separately in Figure **??**. The two conditional directions are intentionally reported separately. Better prospective-memory evidence generally coincides with better next-action prediction, whereas a locally plausible action does not reliably imply that the model selected the appropriate historical evidence. The analysis therefore motivates scoring both action accuracy and evidence fidelity in Track B.

E PROMPT TEMPLATES

All baseline models receive the same task-specific prompt template together with the track-specific visual and dialogue inputs described in Section D.2. The templates below define only the shared inference instruction; the corresponding visual observations and instantiated metadata fields are provided at inference time without model-specific modification.

E.1 TRACK A: CRITICAL ACTION RECONSTRUCTION

For Track A, the prompt instructs a model to reconstruct the chronological sequence of atomic, necessary actions that constitute the successful cooperative solution from the complete two-player episode.

Track A baseline prompt template

```
# Task

You are a video understanding and task abstraction assistant for Portal 2 co-op
    gameplay.

You will be given:
- the complete level video or the video index for one Portal 2 co-op chamber
- full two-player dialogue / subtitle information for the level, with speaker labels
    written as `Player Blue` and `Player Orange` when available
- metadata including session, course, and level identifiers

Your task is to summarize the chronological sequence of critical actions that are
    necessary for clearing the chamber.

You need to output:
- `critical_actions`: a list of atomic, necessary, successful actions in chronological
    order

Do not change the output schema. Each item in `critical_actions` must be a plain
    English string, not a structured object.

# Core Decision Rule

- Summarize the successful solution path of the chamber, not every observed behavior.
- Keep only actions that are necessary for clearing the chamber. An action is critical
    if removing it would break an essential causal chain in the solution.
- Prefer actions that directly change the puzzle state, enable the teammate, unlock a
    path, move a required object, activate a mechanism, or complete an exit condition.
- Remove wandering, looking around, aiming without effect, communication-only steps,
    failed attempts, trial-and-error actions, and repeated actions that do not
    contribute to the final solution.
- If the same necessary action is attempted multiple times, keep only the successful or
     functionally decisive occurrence.
- If a button or floor button must be held continuously, write the decisive holding
    action once; do not repeat it for every moment it remains active.
- Preserve the chronological order of the successful critical actions.
- Prefer a shorter high-confidence list over a longer list with uncertain or non-
    essential steps.
- Do not include future plans, suggestions, purposes, explanations, or whole procedures
     as actions.

# Portal 2 Mechanics and Constraints

## Cooperation and Puzzle Logic

- The two players cooperate to clear the chamber. Common divisions of labor include
    placing portals, holding buttons, carrying cubes or balls, redirecting lasers,
    opening paths, and enabling the teammate to reach a new area.
- Some buttons must be held continuously, and some mechanisms have clear prerequisite
    dependencies.
- Critical actions often form causal chains such as: place portals -> let the teammate
    pass through -> trigger mechanisms -> transport objects -> activate receivers ->
    reach the exit.

## Portal and Environment Mechanics

- Portals can only be placed on portalable white surfaces, such as white walls, white
    floors, and white ceilings.
```

```
- After a player enters one portal, they exit from its paired portal.
- Momentum is preserved, so falling from height and then being launched through a
    portal to a distant or elevated area is a common mechanic.
- Buttons, floor switches, laser catchers, sphere sockets, cube droppers, lift
    platforms, faith plates, turrets, and light bridges are often directly relevant to
     puzzle progression.

## Roles and Portal Colors

- Use `Player BLUE` to refer to the blue player.
- Use `Player ORANGE` to refer to the orange player.
- Do not use bare `BLUE` or bare `ORANGE` as an action subject in `critical_actions`.
- `Player BLUE` can place only blue and purple portals.
- `Player ORANGE` can place only yellow and red portals.
- Every action string must explicitly state the acting player as `Player BLUE` or `
    Player ORANGE`.
- Do not use `Both players` as the action subject. If both players contribute, split
    their contributions into separate atomic actions. If one player's role is not
    identifiable, omit the uncertain action rather than inventing an actor.

# Canonical Wording and Naming Rules

> Whenever a standard verb, object, or color exists, use it verbatim so that the action
     list can be evaluated consistently.

## Standard Action Verbs

Use one of these standard verbs as the head verb whenever it applies:

`place`, `enter`, `pass`, `press`, `stand`, `redirect`, `aim`, `align`, `adjust`, `
    insert`, `pull`, `pick up`, `drop`, `throw`, `carry`, `take`, `catch`, `activate`,
     `launch`, `jump`, `walk`, `establish`

Disambiguate look-alike actions:
- `place` = shoot/place a portal, or place a cube/ball/disc in a position
- `enter` = enter a portal, room, or area
- `pass` = pass through a portal, barrier, or opened path
- `press` = press a hand button once
- `stand` = stand/step on a floor button to keep it triggered
- `redirect` = redirect a laser to a receiver or target
- `insert` = insert a ball, sphere, or disc into a socket/receptacle

## Standard Objects

Use these standard object names whenever possible:

`portal`, `button`, `floor button`, `laser`, `cube`, `reflective cube`, `ball`, `sphere
    `, `disc`, `room`, `switch`, `laser receiver`, `laser catcher`, `platform`, `lift
    platform`, `faith plate`, `light bridge`, `turret`, `exit door`, `white wall`, `
    white floor`, `white ceiling`

- Only describe a cube as a `reflective cube` when the current visual evidence,
    dialogue context, or confirmed chamber mechanics clearly show that it reflects,
    redirects, or guides a laser; otherwise write `cube`.
- If there are multiple objects of the same type, distinguish them by position, height,
     function, nearby landmarks, or relation to other devices.

## Colors

- Portal colors are limited to: `blue`, `purple`, `red`, `yellow`.
- Color applies only to portals unless the game object is explicitly color-coded in the
     evidence.
- Do not attach unsupported colors to buttons, lasers, cubes, platforms, or rooms.

## Location and Layout

- Each action should include key location information whenever possible.
- Avoid vague expressions such as `here`, `there`, `that side`, `this button`, `that
    door`, `the other room`, `somewhere`, or `over there` unless no more reliable
    information is available.

# Input Context

- Session: `{session_id}`
- Course: `{course_id}`
- Level: `{level_id}`
- Two-player dialogue / subtitle source: `{dialogue_source}`
- Optional video index: `{video_index_or_none}`
```

```
# Output Requirements

You must follow these rules when producing the output:

1. Output JSON only. Do not output explanations, reasoning, notes, or Markdown.
2. The output content must be in English.
3. The JSON object must contain exactly one top-level key: `critical_actions`.
4. `critical_actions` must be an array of strings.
5. Do not add fields such as `actor`, `purpose`, `evidence`, `timestamp`, `confidence`,
      or `rationale`.
6. Do not output timestamps.
7. Recommended length: 5 to 10 actions, but correctness is more important than length.
8. The JSON structure must be exactly as follows:

'''json
{
  "critical_actions": [
    "...",
    "...",
    "..."
  ]
}
'''
```

E.2 Track B: Next-Action Prediction

For Track B, the prompt asks a model to predict the immediately following atomic action of the designated player from the pre-cutoff first-person observation and dialogue history only.

Track B baseline prompt template

```
# Task

You are an action prediction and collaborative reasoning assistant for Portal 2 co-op
     video understanding.

You will be given:
- the current video clip, corresponding to a single-player first-person view in Portal
     2 co-op mode
- `actor`: the player in the current viewpoint, which can only be `BLUE` or `ORANGE`
- `dialogue_context`: the dialogue / subtitle context up to the end of the current clip

Your task is to predict: the single atomic action that the current-view player is most
     likely to execute immediately next.

You need to output:
- `next_action`: the complete natural-language description of the one atomic action
     that the current player will most likely execute immediately next
- `next_action_slots`: the direct slot decomposition of `next_action` into fixed
     structured fields
- `action_purpose`: the functional purpose of that action
- `invoked_pm`: the key past information or just-activated prospective memory cues that
      this action depends on

# Core Decision Rule

- Predict only the next action of the current `actor`. Do not replace the current
     players action with the teammates action.
- Prefer the action that is immediately executable under the current conditions, rather
      than a more important action that would happen several steps later.
- Prefer the action that has a direct causal role in advancing the puzzle, rather than
     loosely related behavior such as wandering, looking around, or tentative turning.
- Output only one action. Do not merge two or more actions into one step.
- Do not use the result, purpose, suggestion, plan, or an entire procedure as `
     next_action`.
- If multiple candidate actions exist, choose the one that is closest to the end of the
      clip and most likely to happen immediately next.

# Portal 2 Mechanics and Constraints

## Cooperation and Puzzle Logic
```

```
- The two players cooperate to clear the chamber. Common divisions of labor include
    placing portals, holding buttons, carrying cubes or spheres, adjusting lasers, and
     opening paths for the teammate.
- Some buttons must be held continuously, and some mechanisms have clear prerequisite
    dependencies.

## Portal and Environment Mechanics

- Portals can only be placed on portalable white surfaces, such as white walls, white
    floors, and white ceilings.
- After a player enters one portal, they exit from its paired portal.
- Momentum is preserved, so falling from height and then being launched through a
    portal to a distant or elevated area is a common mechanic.
- Buttons, floor switches, laser catchers, sphere sockets, lift platforms, and light
    bridges are usually directly relevant to puzzle progression.

## Roles and Portal Colors

- `BLUE` can place only blue and purple portals.
- `ORANGE` can place only yellow and red portals.
- When referring to the teammate in `invoked_pm`, you may use `Player BLUE`, `Player
    ORANGE`, or `Both players`.

# Canonical Wording and Naming Rules

> Whenever a standard verb, object, or color exists, use it verbatim so that `
    next_action` can be decomposed consistently.

## Standard Action Verbs

- Use one of these standard verbs as the head verb of `next_action` whenever it applies
    :
  `place`, `enter`, `pass`, `press`, `stand`, `redirect`, `aim`, `align`, `adjust`, `
    insert`, `pull`, `pick up`, `drop`, `throw`, `carry`, `take`, `catch`, `activate`,
     `launch`, `jump`, `walk`, `establish`
- Disambiguate look-alike actions:
  - `place` = shoot/place a portal, or place a cube/ball/disc
  - `enter` = enter a portal, or enter a room/area
  - `pass` = pass through a portal
  - `press` = press a button once
  - `stand` = stand/step on a floor button to keep it triggered
  - `redirect` = redirect a laser to a receiver

## Objects

- Use these standard object names verbatim:
  `portal`, `button`, `floor button`, `laser`, `cube`, `reflective cube`, `ball`, `disc
    `, `room`, `switch`, `laser receiver`, `platform`, `light bridge`
- Only describe a cube as a `reflective cube` when the current visual evidence,
    dialogue context, or confirmed chamber mechanics clearly show that it is used to
    reflect, redirect, or guide a laser; otherwise write `cube`.

## Colors (portal only)

- Portal colors are limited to: `blue`, `purple`, `red`, `yellow`.
- Color applies only to portals. Do not attach a color to buttons, lasers, cubes, or
    other objects.
- `Player BLUE` can place `blue` / `purple` portals; `Player ORANGE` can place `yellow`
     / `red` portals.

## Location and Layout

- For `location`, describe where the action happens or lands using a concrete spatial
    phrase, e.g. `the white wall in the exit room`, `on the upper right platform`, `in
     the middle room`.
- When possible, further specify `left`, `right`, `above`, `below`, `in front`, `behind
    `, `near`, `opposite`, `at the start of`, or `at the end of`.

# Field Requirements

## `next_action`

- It must be one atomic action that the current `actor` will execute immediately next.
- It should be written as much as possible in the form of `action + location / target /
     object`.
- It must use the most specific and safest wording supported by the current clip.
- If the action is placing a portal, specify as fully as possible what color portal is
    being placed and where it is being placed.
```

```
- If the color or location can be reliably inferred from the current visual evidence
    and context, include it.

## `next_action_slots`

- It must be a structured decomposition of `next_action`, not a separate prediction.
- It must contain exactly these four fields: `action`, `object`, `location`, `color`.
- `action` must be a single standard verb string from the standard verb list. It must
    not be null.
- `object`, `location` and `color` must each be either a list of strings or null.
- Use null for missing fields.
- `object` should contain the core object directly acted upon, using the standard
    object names whenever possible.
- `location` should contain the spatial phrase where the action happens, lands, or is
    applied.
- `color` may only describe portal color. For non-portal actions, `color` must be null.
- If `next_action` describes placing, entering, or passing through a portal, `object`
    should include `portal`, and `color` should include the portal color if it is
    known.
- The slot values must be consistent with the wording of `next_action`.

## `action_purpose`

- It must describe the functional purpose of the action, rather than merely restating
    the action in different words.
- It should answer what is this step for? rather than what is being done?
- Keep it concise and direct.

## `invoked_pm`

- Write 1 to 3 of the most important past cues. Do not pile up background information.
- Each cue must be something that has already been observed, heard, confirmed,
    completed, or just triggered in the past. Do not write future plans.
- Each `content` entry must have a clear subject.
- Recommended subjects are `Player BLUE`, `Player ORANGE`, or `Both players`.
- Each cue should be directly relevant to the predicted action, rather than vaguely
    saying it is `for solving the puzzle`.

# Input Context

- Current acting player: {actor}
- Dialogue / subtitle context:
{dialogue_context}

# Output Requirements

You must follow these rules when producing the output:

1. Output JSON only. Do not output explanations, reasoning, notes, or Markdown.
2. The output content must be in English.
3. The JSON structure must be exactly as follows:

'''json
{
  "next_action": "...",
  "next_action_slots": {
    "action": "...",
    "object": ["..."],
    "location": ["..."],
    "color": ["..."],
    "target": ["..."]
  },
  "action_purpose": "...",
  "invoked_pm": [
    "...",
    "..."
  ]
}
'''
```