A2-CAP: A CAUSAL ATTRIBUTE–AFFORDANCE GRAPH HARNESS FOR FROZEN VLA MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon robotic manipulation requires reasoning about which object properties enable skills and whether their intended effects occur. We introduce **A2-CAP**, an inference-time harness around a frozen vision-language-action (VLA) policy. Its registry of 12 mechanism-level strategies, plus an LLM fallback, defines a causal attribute–affordance graph whose directed edges encode preconditions and effects. **PiCausal** grounds video evidence in this graph; **CAP-Plan** selects skills, verifies their outcomes, and handles failures without updating the policy. We introduce **A2-CausalHOI**, 1,305 hand–object interaction videos spanning 85 object categories and 256 tasks, with registry-derived target graphs. On a 100-clip graph-grounding sample, PiCausal raises edge F1 from $\mathbf{0.24}$ to $\mathbf{0.61}$ over a matched VLM-only baseline. On 50 scored EPIC-KITCHENS-100 segments, gains are $+\mathbf{0.43}$ to $+\mathbf{0.52}$ across three backbones; on 14 segments annotated before viewing model drafts, they are $+\mathbf{0.19}$ to $+\mathbf{0.36}$. With pre-generated graphs on two dual-arm tasks, the frozen $\pi0.5$ policy under the harness scores $\mathbf{50.0}\%$ task SR over eight T1 rounds versus $\mathbf{0.0}\%$ over ten rounds for each of two baselines, under a joint execution-and-planning rubric with partial credit. On T2, which reverses the arm roles, it scores $\mathbf{59.1}\%$ task SR over 11 rounds without a baseline comparison.

1 INTRODUCTION

Vision-language-action (VLA) models have advanced robotic manipulation, but long-horizon tasks still expose two bottlenecks: demonstrations scale combinatorially with object–skill compositions, and failures are opaque because policies do not explicitly represent which precondition or effect broke Zhang et al. (2026b); Song et al. (2026). Both reflect the absence of a structured interface between physical object properties and executable skills. We argue that the missing component is not another policy: it is a *harness*—an inference-time layer around the policy that defines a skill vocabulary, schedules execution, verifies effects, and recovers from failure without modifying policy weights.

Classical causal discovery assumes predefined tabular variables, video methods recover event-level graphs rather than object-centric physical mechanisms, and planning approaches based on skill graphs or VLM decomposition rarely ground actions in verifiable attributes Spirtes et al. (2001); Zheng et al. (2018); Chen et al. (2024; 2026); Li et al. (2026); Tang et al. (2024). Current systems can therefore describe what happened but not why an action is feasible, which state should change, or where to resume after failure. Harnesses are beginning to reach robot manipulation Zhang et al. (2026b); Gao et al. (2026); Song et al. (2026), but they add learned memory, verifiers, or test-time steering, whereas A2-CAP exposes a mechanism-level symbolic graph interface between physical attributes and executable skills.

We propose **A2-CAP**, a registry-constrained attribute–affordance mechanism graph instantiated as a neuro-symbolic harness around a frozen VLA policy (Fig. 1). Its nodes are probabilistic attributes (e.g., `sharp`, `portable`) and affordances (e.g., `knife.affd.cut`, `fruit.affd.pick_up`), and its directed edges encode why a skill is feasible and what state change it produces. **PiCausal** grounds video evidence into this graph and **CAP-Plan** (Causal Attribute–Affordance Planner) reads it as an ordered set of executable skills with preconditions, postconditions, and recovery traces.

Our contributions are:

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.

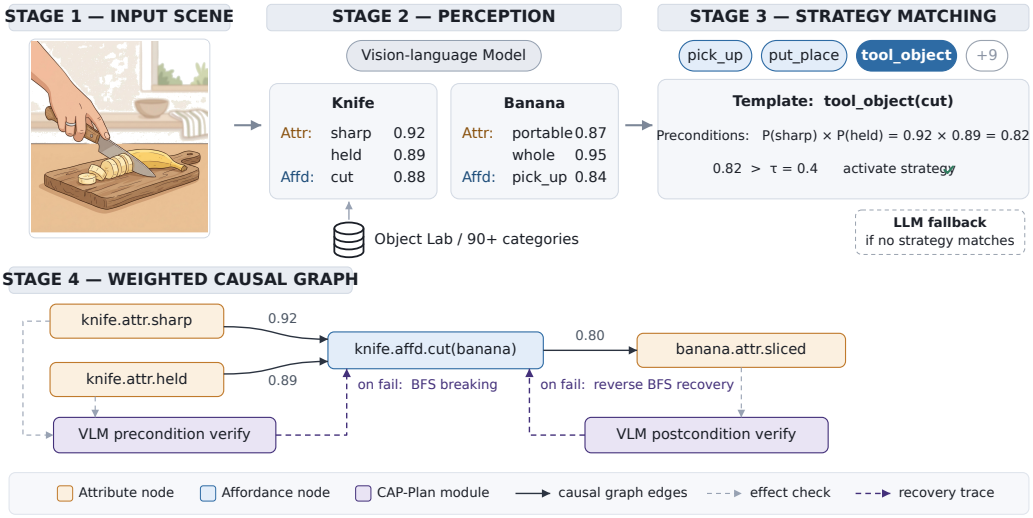


Figure 1: **A2-CAP overview.** A VLM extracts probabilistic attributes (e.g., *sharp*: 0.92) and affordances (e.g., *cut*: 0.88) per object, from which the strategy registry instantiates typed mechanism edges. The resulting weighted graph (orange = attributes, blue = affordances) is the interface between PiCausal and CAP-Plan, which together form an inference-time harness around a frozen VLA.

1. **Harness layer.** A2-CAP, a registry-constrained attribute–affordance mechanism graph operationalized as a neuro-symbolic manipulation harness around frozen VLA policies, with 12 neuro-symbolic strategies plus an LLM fallback.
2. **Benchmark. A2-CausalHOI**, 1,305 real hand–object videos organized as 256 interaction tasks over 85 object categories, each with a registry-derived attribute–affordance target graph.
3. **Graph grounding. PiCausal**, a video-to-graph module that grounds raw video into the A2-CAP formalism via VLM perception and strategy matching; under an identical evidence budget it more than doubles graph-recovery F1 over the VLM-only baseline sharing the same backbones and output grammar (Table 2).
4. **Scheduling and recovery. CAP-Plan**, a graph-conditioned planner that turns recovered graphs into interpretable long-horizon manipulation with effect verification and graph-guided recovery.

2 Related Work

2.1 Video-to-Graph Datasets

Visual causal benchmarks such as Causal3D, Causal3DIdent and CLEVRER are synthetic (Liu et al., 2025; von Kügelgen et al., 2021; Lippe et al., 2022; Yi et al., 2020), and their causal variables, though known, are not grounded in real hand–object manipulation. Real-video datasets are closer, but their nodes are event-level abstractions rather than object-centric physical states and executable affordances (Table 1). A2-CausalHOI instead targets everyday hand–object interactions with registry-derived target graphs whose nodes encode physical attributes and action affordances.

2.2 Structured Graph Recovery from Video

Video-native methods recover coarse temporal or event-level dependencies (Li et al., 2020; Kong et al., 2025; Zhang et al., 2025; Li et al., 2025), whereas classical causal discovery assumes predefined tabular variables and infers structure from scratch (Spirtes et al., 2001; Zheng et al., 2018). PiCausal does neither: the registry fixes the space of expressible mechanisms, and PiCausal estimates which are active and with what confidence, yielding graphs that are comparable across clips and machine-checkable as programs (Sec. 5).

Dataset	Videos	#Tasks	#Obj.	Obj. Int.	GT Graph	Nodes	Node Semantics
V-CDN (Li et al., 2020)	25,000	N/A	N/A	×	×	N/A	N/A
CausalWorld (Ahmed et al., 2021)	15,000	8	3	×	×	N/A	N/A
CLEVRER (Yi et al., 2020)	20,000	4	3	✓	✓	5	Event
MECD/MECD+ (Chen et al., 2024)	1,438	1	N/A	×	✓	11	Event
A2-CausalHOI	1,305	**256**	85	✓	✓	$\leq$12	Attr./Affd.

Table 1: Comparison of video causal datasets. "Obj. Int." marks benchmarks whose videos contain object interactions rather than rendered transformations or passive scenes. To our knowledge A2-CausalHOI is the only benchmark combining real hand–object interaction, ground-truth causal graphs, and attribute–affordance node semantics; it also carries by far the largest number of distinct annotated tasks. MECD+ (Chen et al., 2026) extends MECD with additional annotations.

2.3 Harnesses around Frozen Policies and Structured Planning

LLM agents wrap a frozen model in an external runtime—tools, memory, validators, execution loops—to plan, verify, and recover (Shinn et al., 2023); this harness view reached manipulation through affordance APIs (Ahn et al., 2022), executable code (Liang et al., 2023), language feedback (Huang et al., 2022) and value maps (Huang et al., 2023). A recent wave applies the same recipe to VLA policies kept frozen at inference time, with external scaffolds falling into three groups: memory-based harnesses that track task progress outside the policy (Zhang et al., 2026b; Gao et al., 2026), verification-based harnesses that detect execution errors and trigger correction (Song et al., 2026; Zhang et al., 2026a), and intervention-based harnesses that modify the control interface or action stream at test time (Naouali et al., 2026; Tan et al., 2026). Graph-structured planners instead encode task dependencies with scene graphs, skill graphs, or hierarchical state machines (Li et al., 2026; Ju et al., 2026; Duan et al., 2026), and progress-aware VLA frameworks reason over affordances without an explicit graph (Liu et al., 2026). A2-CAP shares the frozen-policy discipline but exposes the mechanism linking an object's attributes to an executable skill and then to a verifiable postcondition; unlike intervention-driven systems that revise hypotheses online (Ramirez Sanchez et al., 2025; Zhao et al., 2025), it validates declared mechanisms by execution and reads skill ordering, verification targets, and recovery decisions off graph topology.

3 A2-CAP: Registry-Constrained Attribute–Affordance Graph

A2-CAP formalizes object-interaction causality with probabilistic *attributes*, *affordances*, and neuro-symbolic *causal strategies*: symbolic rules provide mechanism-level structure, while VLM/object-prior confidence scores soften each rule for noisy videos. The goal is not to enumerate all visual events but to expose reusable mechanisms connecting physical object properties to executable actions and their effects.

3.1 Causal Node Construction

A2-CAP uses two node types. *Attributes* are verifiable object states or properties, denoted `obj.attr.name` (e.g., `apple.attr.portable`, `knife.attr.sharp`, or relational states such as `cup.attr.full_of(water)`). Each has confidence $P \in [0, 1]$, so the graph retains uncertain but useful evidence instead of prematurely binarizing perception. *Affordances* are executable or observable interaction capabilities, denoted `obj.affd.action` (e.g., `knife.affd.cut`, `apple.affd.pick_up`). We use "affordance" in the action-capability sense common in manipulation planning: an action an object admits under its current attributes rather than a standalone property. Affordances are role-resolved by actor and recipient, so the same object can participate in different mechanisms; a strategy's edges (Sec. 3.3) are what make an affordance node meaningful, by carrying its precondition and effect attributes.

3.2 CAUSAL RELATION CONSTRUCTION

Object interactions induce three weighted causal relations: state changes that alter attributes (Attr $\rightarrow$ Attr), state changes that enable affordances (Attr $\rightarrow$ Affd), and actions that alter attributes (affd$\cdot \rightarrow$ attr; each is listed with an example in App. Table 7). The action-to-attribute relation compresses the observed chain of intention, action, and affordance into a single edge, since the three are simultaneous in manipulation video. Edge weights encode joint prerequisite confidence, not intervention effect size; edge direction declares the strategy's precondition–effect mechanism. For example, `knife.attr.sharp` and `knife.attr.held` jointly enable `knife.affd.cut(banana)`, whose effect is `banana.attr.sliced`.

3.3 NEURO-SYMBOLIC CAUSAL STRATEGIES

A causal strategy is $S = (\mathcal{C}_S, \mathcal{E}_S, \tau_S)$, where $\mathcal{C}_S$ are prerequisite nodes, $\mathcal{E}_S$ are instantiated weighted edges, and τ_S is an activation threshold. It fires when $P(S_{\text{active}}) = \prod_{c \in \mathcal{C}_S} P(c) > \tau_S$, adding subgraph $\mathcal{G}_S$ to the global graph. The registry implements 12 such strategies over an 89-category object database (Table 6), and both the prerequisite probabilities and the affordance verb that names each node are read from that database, so one interaction is instantiated the same way wherever it occurs; the VLM's per-object observations are fused with the category priors of Eq. (1) and carried alongside, which lets dynamic state overrides (e.g., whole $\rightarrow$ sliced) coexist with hard-to-see properties such as material or sharpness. Interactions the registry cannot express fall to an LLM fallback whose edges enter the same weighted graph rather than a separate planner.

4 DATASET DESIGN AND CONSTRUCTION

A2-CausalHOI is built by a three-stage pipeline in which a strategy-guided BFS proposes candidate object sets, an LLM planner drafts the structured interaction plan, and the deterministic Strategy Registry verifies preconditions and instantiates the registry-derived target graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$; the registry, never the LLM, writes the final mechanism edges (App. A).

The released version contains 1,305 first-person videos organized as 256 interaction tasks over 85 object categories and 12 mechanism strategies, and clips run 1.1–23.3 s (median 7.5 s). Each task carries one registry-derived target graph, with 2–12 nodes and 1–10 directed edges; collection continues, and the release is versioned so that reported numbers stay tied to a fixed snapshot. Table 1 compares A2-CausalHOI with existing video causal benchmarks.

5 METHOD: REGISTRY-CONSTRAINED GRAPH GROUNDING AND PLANNING

A2-CAP is operationalized as a neuro-symbolic manipulation harness with two inference-time modules: **PiCausal**, which grounds video evidence into the A2-CAP graph formalism, and **CAP-Plan**, which consumes the same graph for object manipulation planning.

5.1 PICAUSAL: VIDEO-TO-GRAPH GROUNDING

PiCausal converts unstructured interaction videos into weighted A2-CAP graphs in three coupled steps (Fig. 1). Edges are *mechanistic*: domain-knowledge precondition–effect relations declared in the strategy registry, not statistical dependencies estimated from data. Learning is confined to perception, which estimates where and with what confidence declared mechanisms fire. The implemented real-robot online-graph mode captures a scene frame, generates a graph from that visual input, and passes its plan to CAP-Plan for action execution and effect checking; the scored robot results below use pre-generated graphs. **Perceptual evidence.** A VLM assigns each graph node v (a physical attribute or an affordance of one of the scene's objects) an observation probability $P_{\text{obs}}(v)$, while the object database supplies a category prior $P_{\text{prior}}(v)$. We fuse the two by

$$P(v) = \begin{cases} \min\big(1, w_p P_{\text{prior}}(v) + w_o P_{\text{obs}}(v)\big) & P_{\text{prior}}(v) > 0,\ P_{\text{obs}}(v) > 0, \\ P_{\text{obs}}(v) & P_{\text{prior}}(v) = 0,\ P_{\text{obs}}(v) > 0, \\ w_h P_{\text{prior}}(v) & P_{\text{obs}}(v) = 0, \end{cases} \tag{1}$$

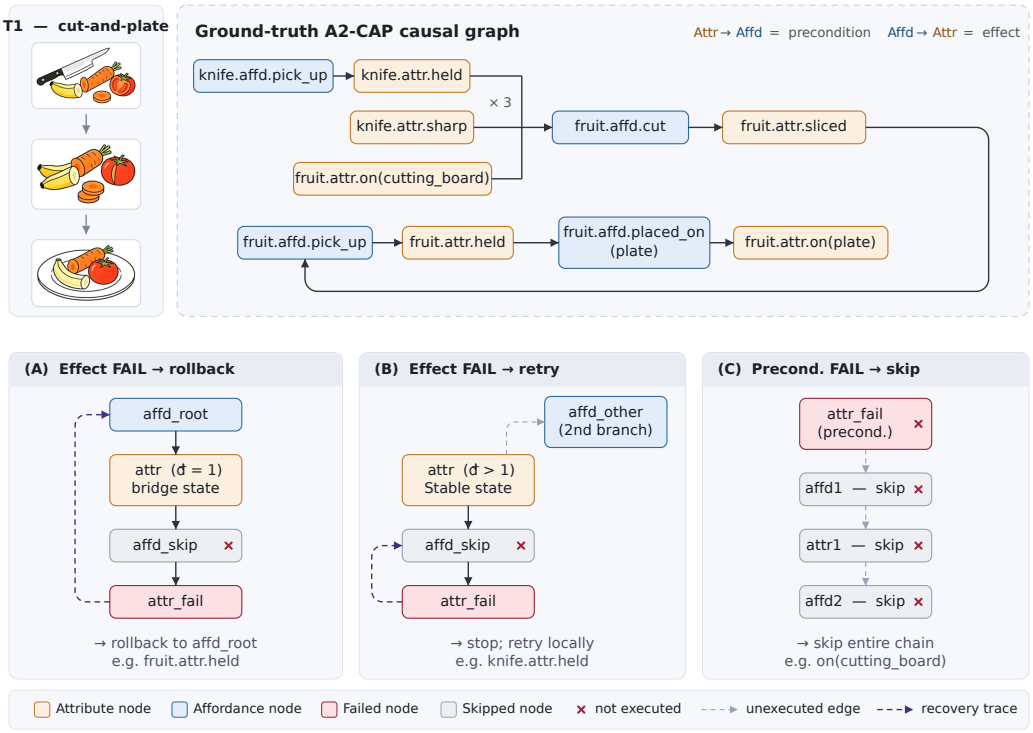


Figure 2: **CAP-Plan graph-conditioned execution and recovery.** *Top:* the A2-CAP causal graph for T1, where Attr↛Affd edges are preconditions and Affd ⇢ Attr edges are effects. *Bottom:* the three topology-decided recovery modes—rollback at a bridge state, local retry at a stable state, and truncation when a precondition is missing.

with consensus weights $w_p = 0.3$, $w_o = 0.7$ and hidden-state decay $w_h = 0.8$. **Strategy matching and instantiation.** A registry strategy fires when the joint probability of its prerequisites exceeds the threshold $\tau_S = 0.4$ (Sec. 3.3) and emits weighted edges (v_i, v_j, w_{ij}) with w_{ij} inherited from prerequisite confidence; interactions the registry cannot express are repaired by an LLM fallback whose self-reported confidence is written on the same $[0, 1]$ scale (App. C.3). All four constants are swept in App. E.6.

5.2 CAP-Plan: A2-CAP Graph-Conditioned Object Manipulation Planning

A recovered A2-CAP graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \mathcal{V}_{\text{attr}} \cup \mathcal{V}_{\text{affd}}$, is an executable program: attributes are verifiable states, affordances are robot skills, $\text{Attr} \to \text{Affd}$ edges are preconditions, and $\text{Affd} \dashrightarrow \text{Attr}$ edges are effects. This gives CAP-Plan a finite action set, a causal order, and the state variables to check after each action, in place of the unconstrained decomposition an open-ended VLM planner produces (Fig. 2).

Topological task extraction. CAP-Plan topologically sorts $\mathcal{G}$ and emits one subtask per affordance node,

$$z_i = \big(a_i, \text{Pre}(a_i), \text{Eff}(a_i), s_i, c_i\big), \tag{2}$$

where $\text{Pre}(a_i) = \{v \in \mathcal{V}_{\text{attr}} : (v, a_i) \in \mathcal{E}\}$, $\text{Eff}(a_i) = \{v \in \mathcal{V}_{\text{attr}} : (a_i, v) \in \mathcal{E}\}$, s_i is the matched strategy, and c_i is the weakest prerequisite confidence of a_i. Pure affordance–affordance activation edges update graph state but are not dispatched as robot actions. The resulting topological order is deterministic given the recovered graph. Each subtask is translated into a strategy-specific VLA instruction, dispatched to the policy, and verified by querying $\text{Eff}(a_i)$ from the current observation.

Causal recovery from effect failure. When a verified effect $v_{\text{fail}} \in \text{Eff}(a_i)$ is absent after executing subtask z_i, A2-CAP localizes the recovery point by running a reverse breadth-first search on the same

causal graph. Because the first predecessor of the failed effect is usually the action that just failed, the search treats the current affordance as transparent and applies a single graph-theoretic gate at attribute nodes,

$$d^+(v) = \big|\{u : (v, u) \in \mathcal{E}\}\big|. \tag{3}$$

An attribute with $d^+(v) = 1$ is a bridge state, so recovery continues upstream; an attribute with $d^+(v) > 1$ is a stable shared state, so the search stops and the current subtask is retried.

The gate thus makes rollback a consequence of topology rather than a per-strategy rule; Table 10 traces both cases on concrete failures.

Precondition-driven causal truncation. CAP-Plan also handles failures that occur *before* execution: prior to dispatching z_i, the VLM checks whether $\mathrm{Pre}(a_i)$ holds. If a precondition v_{fail} is absent, the action is infeasible and CAP-Plan forward-propagates through $\mathcal{G}$ via BFS to identify and skip all downstream affordances that transitively depend on the missing state. Tool-state attributes whose detection is unreliable under arm occlusion are exempt from truncation and raise a warning instead (App. D.3).

6 Experiments and Analysis

We evaluate PiCausal's video-to-graph grounding and, separately, CAP-Plan's real-robot execution with pre-generated task graphs. The real-robot online mode implements a visual-input-to-graph-to-action path using a captured scene frame, but has no valid scored rounds; the reported robot metrics therefore do not evaluate that end-to-end path.

6.1 Experimental Setup

Datasets. We use two benchmarks. **A2-CausalHOI**, our real hand–object video benchmark with attribute–affordance graph labels, is evaluated on a duration-stratified sample of 100 clips drawn from 98 of its 256 interaction tasks; it covers 38% of tasks and matches the corpus on candidate-object count (3.77 vs. 3.61 per task), graph size (4.45 vs. 4.34 edges) and strategy mix (maximum deviation 1.8 percentage points). The paired effect is stable from n=20 upwards (App. E.2), although the longest-video tail is underrepresented (95th-percentile duration 11.4 vs. 13.3 s). **CLEVRER** Yi et al. (2020) tests transfer beyond hand–object manipulation: we map kinematic states to attributes and collisions to affordances (physics-collision strategy only) and evaluate on a 200-clip stratified subset of the validation split. **What the graph-recovery task measures.** Target graphs project annotated interaction steps into the same registry vocabulary used by PiCausal; they are not independently discovered structures. Edge F1 thus measures step-and-role recovery plus vocabulary mapping. Re-expanding annotated steps reproduces $174/211$ stored in-domain graphs exactly and all reviewed EPIC targets with steps. On 14 EPIC segments annotated before viewing model drafts, PiCausal reaches 0.30–0.44 F1, versus 0.50–0.64 on 32 draft-assisted segments (App. Table 13).

Metrics. Graph recovery is measured by SHD, F1 and precision. Robot planning uses the dual-arm Xtrainer on **T1** (cut and plate food) and **T2** (brew tea with reversed arm roles). Policy training episodes are skill-level segments with standardized start/end states and VLM-verified target effects; reset motions and recovery behavior are not counted as demonstrations.

6.2 Baselines and Implementation

We compare two methods, each instantiated with three recent VLM backbones (Qwen3.8-Flash, DeepSeek-V4-Flash-Vision, GPT-5.6-Luna; temperature 0): **PiCausal**, and a **VLM-only** baseline that removes the strategy registry entirely and asks the same backbone to propose nodes and causal edges zero-shot in the same output grammar. Both share one evidence policy—frames sampled on the video timeline at 2 FPS (at most 64 frames), longest side 1024 px, JPEG quality 90, with the exact frame indices logged; we re-run both methods at 2–4 FPS (App. E.3)—and identical object grounding (identity attributes from benchmark metadata only). The baseline loses the registry, its templates and LLM fallback, but is also spared its hand-annotated category priors and gating attributes; the ablation isolates the whole symbolic contract, not the priors alone. Manipulation uses one language-conditioned $\pi0.5$ policy fine-tuned on 600 in-house skill-level episodes across six registry strategies

Method	A2-CausalHOI			CLEVRER			EPIC-KITCHENS		
	SHD↓	F1↑	Prec.↑	SHD↓	F1↑	Prec.↑	SHD↓	F1↑	Prec.↑
PiCausal (Qwen3.8-Flash)	3.91	<u>0.610</u>	0.538	<u>4.13</u>	<u>0.537</u>	<u>0.588</u>	**3.76**	**0.610**	**0.575**
VLM-only (Qwen3.8-Flash)	14.77	0.241	0.168	5.52	0.485	0.470	13.48	0.093	0.066
PiCausal (DeepSeek-V4-FV)	<u>3.64</u>	0.593	<u>0.585</u>	4.49	0.405	0.533	4.28	0.523	0.502
VLM-only (DeepSeek-V4-FV)	11.68	0.207	0.164	7.50	0.227	0.227	13.02	0.041	0.029
PiCausal (GPT-5.6-Luna)	**2.51**	**0.709**	**0.705**	**4.02**	**0.560**	**0.606**	<u>4.14</u>	<u>0.534</u>	<u>0.552</u>
VLM-only (GPT-5.6-Luna)	11.41	0.241	0.182	6.09	0.470	0.425	10.34	0.105	0.080

Table 2: Graph recovery on A2-CausalHOI (100 clips), a 200-clip stratified CLEVRER subset, and 50 EPIC-KITCHENS-100 segments with a non-empty target (including blind-first segments; annotation arms are in App. Table 13). "VLM-only" is a zero-shot ablation that keeps the same backbones, frames and output grammar but removes the strategy registry. F1 and Precision are each means of the per-clip values; F1 is not recomputed from pooled Precision and Recall. SHD is a distance. Best scores are bold and second-best underlined, except in the EPIC column where we mark per method. All columns share one evidence policy, one symbolic contract and one ground-truth rule (a segment without a target is reported as coverage, never averaged in as a zero).

(four instantiated by T1/T2), then frozen at step 43,000 (App. G). CAP-Plan dispatches and verifies subtasks from a pre-generated graph in both scored robot tasks.

6.3 Graph Recovery Results

Table 2 reports graph recovery on both benchmarks.

(1) The registry, not the backbone, drives recovery. On A2-CausalHOI the paired registry gain is $+0.370$ F1 (95% CI $[+0.312, +0.427]$, 84/100 clips won) with Qwen3.8-Flash, $+0.386$ (84/100) with DeepSeek-V4-Flash-Vision, and $+0.468$ (90/100) with GPT-5.6-Luna. A stronger backbone raises PiCausal's ceiling (0.593–0.709 F1) but barely moves the zero-shot baseline (0.207–0.241). Under the reported object-lab contract, graph recovery is invariant to the activation threshold over $\tau \in [0.2, 0.4]$; fusion weights affect F1 by under 0.02 only in the alternative fused-state variant (App. E.6).

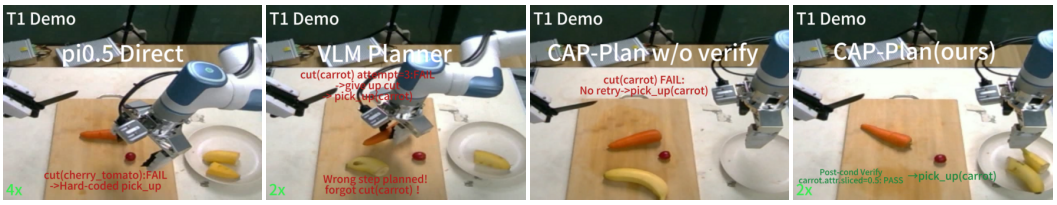
(2) Zero-shot VLMs find edges but cannot select a graph. The baseline reaches non-trivial recall (0.34–0.50) yet emits 10.2–14.8 edges per clip against 4.5 ground-truth edges, collapsing precision to 0.16–0.18 (FDR ≥ 0.82); PiCausal stays near the ground-truth budget (4.8–6.4) and cuts SHD from 11.4–14.8 to 2.5–3.9. The registry thus supplies both the mechanism vocabulary and the selection pressure that keeps a recovered graph compact.

(3) The advantage transfers but shrinks on CLEVRER. The registry still wins on the 200-clip subset ($+0.051$ for Qwen3.8-Flash, $+0.178$ for DeepSeek-V4-Flash-Vision, $+0.090$ for GPT-5.6-Luna, all with CIs excluding zero) but by much less: collision direction is a single mechanism a strong VLM can often infer zero-shot—GPT-5.6-Luna reaches 0.470 F1 alone, within 0.09 of its own harnessed score, and here the backbone spread (0.155) exceeds the registry gain for every backbone—whereas real hand–object clips require distinguishing many interaction types. The registry-not-backbone claim thus holds on A2-CausalHOI and EPIC-KITCHENS-100, not here; CLEVRER is a sanity check, not a like-for-like benchmark.

Ablation and robustness checks. The graph-recovery advantage is not explained by the evidence budget or one symbolic setting. On a paired 10-clip sensitivity subset, increasing sampling from 2 to 4 FPS changes PiCausal F1 by only $+0.030$ (DeepSeek) and $+0.005$ (Qwen), while changing VLM-only by -0.041 and -0.000; all four 95% CIs include zero, and the roughly 0.44 F1 registry gap persists (App. Table 12). Under the reported object-lab contract, sweeping τ from 0.2 to 0.4 is bit-identical, with at most 0.005 movement at 0.5 and a 0.14–0.15 F1 drop only at 0.6; the alternative fused-state sweep changes fusion weights by at most 0.018 F1 (App. E.6).

6.4 Cross-Dataset Transfer to EPIC-KITCHENS-100

To test transfer to independently collected real-world videos, we use the EPIC-KITCHENS-100 validation split Damen et al. (2022): 80 pre-registered segments sampled with a fixed seed under a



(a) $\pi 0.5$ Direct (b) VLM Planner (c) w/o verify (d) Ours

Figure 3: **Real-robot execution of the same frozen $\pi 0.5$ policy on T1** (dual-arm Xtrainer, representative frames; overlay text is each method's own runtime log). Without the harness the policy falls back on a hard-coded skill, the VLM Planner mis-orders the plan, and the harness without verification runs the causal plan with no checks, so the failed cut passes undetected; the full harness verifies the postcondition and continues.

closed object vocabulary and 1.5–10 s clips, annotated by three authors who wrote interaction steps from the clip alone—never from EPIC labels—using the same 12 strategies. Of these, 53 carry a reviewed graph and 50 a non-empty target, the pool scored in the EPIC column. In the initial EPIC object lab, 82% of objects have no strategy-gating attribute. Before scoring, fixed rules derive those attributes from its existing affordance lists and noun-class keywords, leaving 0% uncovered; the transfer result is conditional on this repaired vocabulary. EPIC targets expand annotated steps through the same registry templates used in-domain (App. E.4).

(1) The registry advantage transfers with all three backbones. PiCausal beats the zero-shot baseline by $+0.52$, $+0.48$ and $+0.43$ F1 with Qwen3.8-Flash, DeepSeek-V4-Flash-Vision and GPT-5.6-Luna, while the baseline emits 1.8–2.6$\times$ the target edge budget (10.7/9.4/7.4 vs. 4.1 per clip). **(2) The scoring vocabulary matters.** EPIC targets are instantiated by the same 12 templates; aligning predictions with the target's object lab adds $+0.12$ F1 here.

Against 43 preserved free-form graphs, PiCausal scores only 0.23–0.26 F1, versus 0.09–0.17 for VLM-only (App. Table 13). Edge agreement between two annotators is $\kappa = -0.47$ on the four double-annotated segments with complete bundles, an uncertain estimate but a warning that graph labels are not unique (App. E.5).

6.5 Object Manipulation Planning Evaluation

On T1 we compare four instantiations of the same $\pi 0.5$ policy: the policy alone, a VLM Planner on top of it, the harness without verification, and the full harness (Fig. 3). The full harness adds precondition gates, postcondition verification and graph-guided recovery; none updates the policy. The ablations isolate whether gains come from decomposition alone or from closed-loop graph-based monitoring. A human operator scores each subtask without being told which variant runs, and the rubric (App. B) caps a subtask at 0.5 whenever the method's planning decision departs from the graph's optimal action—so the baselines, which never consult a graph, cannot score fully there even when the action succeeds. Every variant stops by its own rule—Direct runs its fixed 16-step plan to the end, the VLM Planner terminates itself, and the causal variants stop when the topological plan is exhausted or a precondition truncation fires—while an operator keypress only resets the robot, so no method's stopping decision carries information about another's outcome.

Metrics. Task SR averages per-round scores in $\{0, 0.5, 1\}$; Sub-SR counts subtasks reaching at least partial success, using each method's own subtask list (8–16 on T1), so its percentages have different denominators. EE weights qualified subtasks by attempts; RR counts successful retries, and Att counts all attempts (App. C.6). The rubric also caps a subtask at 0.5 when its planning decision violates the graph's prescribed retry, rollback or skip action. SR is thus a joint execution-and-planning measure, not graph-neutral physical success.

On T1, w/o verification reaches 25% SR, compared with 0% for Direct and VLM Planner; full A2-CAP reaches 50% under the joint rubric. Its two retried rounds recover locally, and no valid round executes a graph-guided rollback (App. C.7); the table cannot measure the benefit of backward tracing. Full A2-CAP averages 13.12 attempts versus 12.80 without verification. Counting only complete rounds gives 2/8 for the harness versus 0/10 for each baseline, a small comparison without clear statistical separation (one-sided Fisher exact $p = 0.183$ against either baseline).

Method	SR↑	Sub-SR↑	EE↑	RR↑	Att↓
T1: cut-and-plate					
$\pi0.5$ Direct	0.0 ± 0.0	64.9 ± 10.9	40.4 ± 13.7	–	15.40 ± 1.90
VLM Planner + $\pi0.5$	0.0 ± 0.0	54.0 ± 19.5	21.4 ± 20.4	–	15.40 ± 4.62
A2-CAP w/o verification	25.0 ± 26.4	80.6 ± 22.7	64.3 ± 30.4	–	12.80 ± 1.40
A2-CAP + $\pi0.5$ **(Ours)**	**50.0** ± 37.8	**97.9** ± 4.0	**89.7** ± 9.2	**100.0** ± 0.0	13.12 ± 1.64
T2: tea brewing					
A2-CAP + $\pi0.5$ **(Ours)**	**59.1** ± 49.1	**89.1** ± 18.7	**81.1** ± 28.6	0.0 ± 0.0	**5.27** ± 0.90

Rounds: T1 Ours 8, T1 other variants 10, T2 11.
Evaluation coverage and excluded runs: App. C.7.

Table 3: Real-robot planning results ($\pm$ SD). Only rounds registered as valid in the original run log enter the headline rows (App. C.7). Task SR includes partial credit: T1 Ours has 2 complete, 4 partial and 2 failed rounds (50.0%); T2 has 6/1/4 (59.1%). The eight additional T1 harness rounds that ran the full plan but were not registered yield 3/16 complete and 31.2% mean SR when pooled with the valid rounds. VLM Planner has 11 further incomplete T1 runs; unregistered T2 runs also ended part-way and lack complete metric records. T1 compares all four variants; T2 has no baseline. Both use pre-generated graphs, and the online-graph variant has no valid evaluated rounds. RR is not applicable (–) without structured recovery; T2's 0% is from one retried round. All rows use the same frozen $\pi0.5$ policy.

6.6 Cross-Strategy Transfer on T2 (Tea Brewing)

T1 exercises one strategy combination under one arm assignment, so we also evaluate the full harness on **T2**, a 5-subtask tea-brewing task with a different combination and reversed arm roles (the chain is listed in App. B). The teabag is a graspable stand-in: a small, solid, 3D-printed cube (2–3 cm), since a real tea bag is too small and floppy for the gripper. T2 swaps the primary actor and activates `transfer_content`, which T1 never uses (including an anti-leak carrying pose for the teapot, App. B).

The T2 panel reports transfer to a second strategy combination: over 11 valid rounds the harness reaches 89.1% Sub-SR with 5.27 ± 0.90 attempts. These rounds span two sessions, with two repeated after a camera and venue change; the earlier nine score 55.6% task SR versus 59.1% pooled. Since only the harness was run and the SR interval is wide (± 49.1), we treat T2 as transfer evidence, not a ranking (App. C.7).

7 Conclusion

We presented **A2-CAP**, a registry-constrained attribute–affordance mechanism graph and inference-time harness around frozen VLA policies. **PiCausal** grounds video evidence over 12 mechanism-level strategies, while **CAP-Plan** schedules execution, verification, and recovery. The frozen $\pi0.5$ policy completes T1 in 2/8 rounds versus 0/20 across the two baselines, and T2 in 6/11. Because targets project annotated steps into the registry vocabulary, we interpret graph recovery as vocabulary-faithful grounding and the robot result as evidence under the joint execution-and-planning rubric.

8 Limitations

A2-CAP grounds a fixed mechanism vocabulary rather than discovering causal structure; graph quality depends on registry coverage, annotation conventions and VLM perception. The two scored robot tasks together instantiate four of the 12 predefined registry strategies, and both use pre-generated graphs. T1 has eight registered harness rounds versus ten per baseline, with a lower inclusive score across 16 harness rounds; T2 has no baseline or complete inclusive record. The implemented online visual-input-to-graph-to-action mode lacks valid evaluation rounds; evaluating it remains future work. No valid round tests graph-guided rollback (App. F). Thus the robot evidence supports closed-loop execution from pre-generated graphs, while online grounding and graph-guided rollback remain unmeasured.

AI USE STATEMENT

Generative AI tools were used in this work for the following tasks: implementing and debugging the video-to-graph pipeline, the dataset and annotation tooling, and the robot-experiment harness; validating and re-running those pipelines; editing the manuscript for language and clarity; and drafting diagrams. An early version of the dataset-construction schematic was produced with a generative image model; the figure published here (App. Fig. 4) was redrawn as a vector graphic, and its content was taken from the pipeline code and checked against it. They were not used to design the method, to generate the benchmark or its ground-truth graphs, or to analyse the results. The LLM planner of the dataset pipeline and the VLM components of A2-CAP are part of the method we study, not writing assistance.

For completeness, the remaining tasks that ICLR asks authors to disclose are not applicable to this work: no generative AI was used to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate or prove mathematical claims, to propose or refine hypotheses, to design or give feedback on the methodology or experiments, to clean or reformat the released benchmark data, to translate material, to interpret the results, or to conduct qualitative or thematic data analysis.

All AI-assisted artifacts were reviewed and verified by the authors. Code was executed end-to-end by the authors rather than accepted as written: the graph-recovery pipelines were re-run from their configurations and every reported number regenerated from the stored predictions, and the robot metrics were replayed from the archived per-round records through the scoring code; the aggregation scripts are included in the supplementary code release. The manuscript text was checked against these artifacts. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The A2-CausalHOI videos are recorded by the authors with a head-mounted camera and show the authors' own hands and household objects; they are not recordings of other people and contain no additional participants, so no external human-subjects review was required. No faces or identifiable personal information are annotated or released, and we do not attempt to identify any person or scene that a clip incidentally captures in the background. The released annotations describe physical object states and affordances only. The interaction steps of the EPIC-KITCHENS-100 subset used in Sec. 6.4 were written by the authors themselves from the released clips, and the anchoring effect of showing those annotators a model-written draft is reported there rather than assumed away.

We use the third-party EPIC-KITCHENS-100 annotations under their research-use licence (CC BY-NC 4.0, non-commercial) and report results derived from them under that licence; we neither redistribute the EPIC videos or annotations nor attempt to identify the participants appearing in them, and we ask readers to obtain the source data from the original benchmark. The demonstration video in the supplementary archive shows only the robot platform, the manipulated objects and the workspace; it has no audio track and contains no participants. We see no additional ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

We describe the full experimental protocol in Sec. 6 and the appendix, including the per-round validity curation of the robot runs (Sec. C.7). All graph-recovery methods share one published sampling and image policy (2 FPS on the video timeline, at most 64 frames, longest side 1024 px, JPEG quality 90), and every per-video prediction stores the exact frame indices used, so the evidence budget of each comparison can be audited. The robot experiments use a single frozen $\pi 0.5$ checkpoint, and the per-round records bundled with the release carry the subtask outcomes, attempt counts, and execution steps that the tables aggregate; the aggregation is a deterministic script rather than hand-transcription. The driver scripts, suite and profile configurations, and the evaluation code used to produce every table are part of the same prepared release. Every graph-recovery model query runs at temperature 0; the registry's LLM fallback, which no released prediction exercises (each stored prediction records `llm_fallback: false` in its provenance field), runs at 0.3, and the robot VLM client leaves the temperature at the provider default. The hyperparameter replay is offline for cached perception

outputs, but interactions explicitly labelled `llm` may re-query the provider; this API-dependent case is documented in App. E.4. A single supplementary archive is attached. It contains an anonymized copy of that code, the per-round robot ledger and metric records, the stored graph-recovery summaries, and the aggregation scripts that emit each table, so re-running the aggregation reproduces the reported numbers without any further model queries; it also contains a video showing all four T1 variants executing the task end to end. The two controls of App. E.5 that read the benchmark or the annotation bundles are reported from the stored summary, since neither is part of the archive. The benchmark videos and graphs, the LeRobot-format training demonstrations, and the frozen $\pi 0.5$ checkpoint are not attached and will be released upon publication.

REFERENCES

Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Yoshua Bengio, Bernhard Schölkopf, Manuel Wüthrich, and Stefan Bauer. CausalWorld: A robotic manipulation benchmark for causal structure and transfer learning. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.04296.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022. URL https://arxiv.org/abs/2204.01691.

Tieyuan Chen, Huabin Liu, Tianyao He, Yihang Chen, Chaofan Gan, Xiao Ma, Cheng Zhong, Yang Zhang, Yingxue Wang, Hui Lin, and Weiyao Lin. Mecd: Unlocking multi-event causal discovery in video reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 92554–92580. Curran Associates, Inc., 2024. doi: 10.52202/079017-2939. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a8320b6b9d95798dc286a867c44742a1-Paper-Conference.pdf.

Tieyuan Chen, Huabin Liu, Yi Wang, Yihang Chen, Tianyao He, Chaofan Gan, Huanyu He, and Weiyao Lin. Mecd+: Unlocking event-level causal graph discovery for video reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2628–2645, 2026. doi: 10.1109/TPAMI.2025.3626549.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. doi: 10.1007/s11263-021-01531-2.

Shiying Duan, Pei Ren, Nanxiang Jiang, Zhengping Che, Jian Tang, Zhaoxin Fan, Yifan Sun, and Wenjun Wu. RoboPARA: Dual-arm robot planning with parallel allocation and recomposition across tasks. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 143933–143974, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e8aa3f5e8b5f24b830f31fd828e2a2a3-Paper-Conference.pdf.

Hongbo Gao, Zeyu Ni, Xin Wen, Siyu Xu, and Ruifeng Li. Agm: Achievement-grounded memory for closed-loop agents with frozen vla policies, 2026. URL https://arxiv.org/abs/2608.29537.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022. URL https://arxiv.org/abs/2207.05608.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2307.05973.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Gireesh Nandiraju, Yuanliang Ju, Seungjae Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. MomaGraph: State-aware

unified scene graphs with vision-language models for embodied task planning. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 96361–96386, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/9baf31febefde7bd76023c2d2f13cbd7-Paper-Conference.pdf.

Lingbai Kong, Wengen Li, Hanchen Yang, Yichao Zhang, Jihong Guan, and Shuigeng Zhou. Causalformer: An interpretable transformer for temporal causal discovery. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):102–115, 2025. doi: 10.1109/TKDE.2024.3484461. URL https://arxiv.org/abs/2406.16708.

Chenxu Li, Zixuan Chen, Yetao Li, Jiapeng Xu, Hongyu Ding, Jieqi Shi, Jing Huo, and Yang Gao. Momastage: Skill-state graph guided planning and closed-loop execution for long-horizon indoor mobile manipulation, 2026. URL https://arxiv.org/abs/2603.08383.

Jin Li, Shoujin Wang, Qi Zhang, Feng Liu, Tongliang Liu, Longbing Cao, Shui Yu, and Fang Chen. Revealing multimodal causality with large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 101384–101430, 2025. doi: 10.52202/085713-3395. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/92e9846e694ccee9c80260d47053d8b5-Paper-Conference.pdf.

Yunzhu Li, Antonio Torralba, Anima Anandkumar, Dieter Fox, and Animesh Garg. Causal discovery in physical systems from videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2007.00631.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. URL https://arxiv.org/abs/2209.07753.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M. Asano, Taco Cohen, and Efstratios Gavves. Citris: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2202.03169.

Disheng Liu, Yiran Qiao, Wuche Liu, Yiren Lu, Yunlai Zhou, Tuo Liang, Yu Yin, and Jing Ma. Causal3d: A comprehensive benchmark for causal learning from visual data. *arXiv*, 2025. URL https://arxiv.org/abs/2503.04852.

Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, Fangqiang Ding, and Ismini Lourentzou. Palm: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28096–28110, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_PALM_Progress-Aware_Policy_Learning_via_Affordance_Reasoning_for_Long-Horizon_Robotic_CVPR_2026_paper.html.

Dhia Naouali, Minghan Wu, Claudia Wong, Abhinav Puthran, and Omar G. Younis. Vlcp: Vision language control policy closed-loop code replanning for robot manipulation, 2026. URL https://arxiv.org/abs/2608.16978.

Robert Ramirez Sanchez, Heramb Nemlekar, Shahabedin Sagheb, Cara M. Nunez, and Dylan P. Losey. Recon: Reducing causal confusion with human-placed markers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. doi: 10.1109/IROS60139.2025.11246741. URL https://arxiv.org/abs/2409.13607.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Chang Song, Bin Qian, Yan Feng, and Zhijie Song. Vla-corrector: Stage-aware observable state understanding for prompt-based closed-loop recovery of vision-language-action policies, 2026. URL https://arxiv.org/abs/2609.06508.

Peter Spirtes, Clark N. Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, second edition, 2001. doi: 10.7551/mitpress/1754.001.0001.

Derek Ming Siang Tan, Shailesh Shailesh, Srikrishna Iyer, William Wei Jie Teo, Yuanliang Ju, Qiao Gu, and Guillaume Sartoretti. Rl^2-vla: Adaptive rl latent compositional steering with test-time scaling for vision-language-action models, 2026. URL https://arxiv.org/abs/2607.26991.

Wenbing Tang, Fenghua Wu, Shang wei Lin, Zuohua Ding, Jing Liu, Yang Liu, and Jifeng He. Causal deconfounding deep reinforcement learning for mobile robot motion planning. *Knowledge-Based Systems*, 303:112406, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2024.112406. URL https://www.sciencedirect.com/science/article/pii/S0950705124010402.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.04619.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1910.01442.

Fan Zhang, Seongbin Park, Baharan Mirzasoleiman, Shariar Talebi, and Nader Sehatbakhsh. Probeact: Probe-guided training-free failure recovery in vision-language-action models, 2026a. URL https://arxiv.org/abs/2606.09740.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Zhengru Fang, Yi Nie, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness vla: Steering frozen vlas into reliable manipulation primitives via memory-guided agents, 2026b. URL https://arxiv.org/abs/2607.08448v4.

Yize Zhang, Meiqi Chen, Sirui Chen, Bo Peng, Yanxi Zhang, Tianyu Li, and Chaochao Lu. Causight: Learning to supersense for visual causal discovery, 2025. URL https://arxiv.org/abs/2512.01827.

Zhiyu Zhao, Haoxuan Li, Haifeng Zhang, Jun Wang, Francesco Faccio, Jürgen Schmidhuber, and Mengyue Yang. Curious causality-seeking agents in open-ended worlds. In *Advances in Neural Information Processing Systems*, volume 38, pp. 153856–153893, 2025.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 2018.

A Dataset Construction Details

A2-CausalHOI is constructed through a three-stage pipeline (Fig. 4). **Stage 1:** strategy-guided BFS expands from a source object to compatible objects, admitting candidates only when at least one causal strategy connects them with the existing set. **Stage 2:** a Qwen-based planner emits a structured interaction plan with object roles, strategy types, and auxiliary arguments (tool, source, destination). **Stage 3:** the deterministic Strategy Registry verifies preconditions and instantiates the final ground-truth graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$, preventing annotation drift from free-form generation.

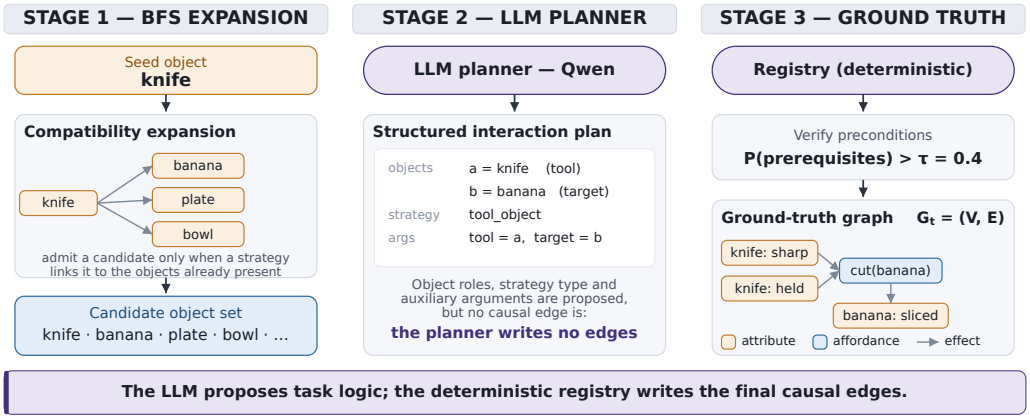


Figure 4: **Dataset construction pipeline.** Strategy-guided BFS expands from a seed object to compatible candidates; an LLM planner emits a structured interaction plan; the deterministic Strategy Registry instantiates the final ground-truth causal graph. The LLM proposes task logic but is not trusted to write final causal edges.

Recording protocol. Each task is recorded as 5 first-person videos with controlled diversity in object placement, camera angle, and lighting (251 of the 256 tasks carry exactly 5; the other five carry exactly 10 after re-shoots, for 1,305 in total). Videos are captured at 1080p/30 fps using a head-mounted GoPro; the operator's hands enter the frame during manipulation and objects remain $\geq 85\%$ within frame. Recordings are compressed via FFmpeg before storage. Strict per-video quality checks enforce hand visibility, object containment, and correct interaction ordering.

The object database covers 89 everyday categories spanning kitchenware, electronics, stationery, furniture, and tools. Category priors encode the base probability of each attribute and affordance (e.g., most knives are `sharp` with $P = 0.9$, most apples are `portable` with $P = 0.95$). These priors are combined with per-video VLM observations through the weighted fusion in Eq. (1) to handle both dynamic state changes (e.g., a sliced apple) and hard-to-observe properties (e.g., metallic material).

B Extended Task Designs for Robot Experiments

Table 4 lists the two manipulation tasks evaluated in this paper: T1 (cut-and-plate) and T2 (tea brewing). Both are reported in Sec. 6—T1 across all variants, T2 for the full harness. Extending the evaluation to additional interaction strategies such as assembly (insertion) and device activation is left for future work.

T1 causal chain (full). The complete T1 graph expands as: `pick_up(knife)` $\rightarrow$ `cut(banana)` $\rightarrow$ `pick_up(banana)` $\rightarrow$ `place_on(plate, banana)`, repeated for radish and cherry_tomato. The knife lane shows `knife.attr.held` with out-degree 3 (shared across three cut actions), while each fruit's `held` attribute has out-degree 1 (bridge to a single placement). Three fruits are ordered left-to-right by a top-camera VLM; the knife is held continuously by the left arm (gripper sentinel).

	T1: Cut-and-plate	T2: Tea brewing
Strategies	`tool_object` (cut), `pick_up`, `put_place`	`pick_up`, `transfer_content` (pour), `put_place`
Objects	knife, banana, radish, cherry tomato, plate	teapot, cup, teabag
Steps	10	5
Arms	Left: knife Right: pick/place	Right: teabag Left: teapot
Status	Reported (all variants)	Reported (full harness)

Table 4: The two tasks evaluated for CAP-Plan. T1 uses `tool_object` for cutting and T2 uses `transfer_content` for pouring; both also use `pick_up` and `put_place`, with reversed arm assignment. T1 is reported across all four variants, T2 for the full harness only.

Strategy	Score = 1.0 (`s`)	Score = 0.5 (`d`)	Score = 0.0 (`f`)
`pick_up`	Object fully lifted, gripper stable	Lifted but gripper slipping/tilted, or partial grasp	Not lifted, or dropped immediately
`put_place`	Object stable on target, gripper released	On target but unstable (rolling/half-hanging)	Not on target surface, or fell outside
`tool_object` (cut)	Visible cut mark or object split in two	Shallow mark, not fully cut through	No visible cut mark
`transfer_content`	Liquid flows steadily into target, source returns upright	Brief flow, insufficient volume, or splashing	No transfer, or all liquid spilled
`embed_insert_connect`	Part fully inserted/connected to base	Partially inserted but not fully seated	Not inserted, or detached after insertion
`turn_on_activation`	Device clearly on (light, sound)	Momentary flicker or uncertain activation	No activation signal

Table 5: Human scoring rubric per strategy (execution quality dimension). A planning decision that does not match the ground-truth graph's optimal action (retry / rollback / skip / proceed) caps the score at 0.5, and the standby arm is not evaluated.

T2 causal chain. Stage 1 (right arm): `pick_up(teabag)` → `place_in(cup, teabag)`. Stage 2 (left arm): `pick_up(teapot)` → `pour(cup)` → `place_on(table, teapot)`. The left arm adopts an anti-leak pose during transfer to prevent dripping.

Human scoring rubric. Human operators press `s`/`d`/`f` after each subtask. Scoring has two dimensions. **Execution quality** follows Table 5: whether the physical action achieved its intended effect. **Planning optimality** checks whether the current subtask aligns with the ground-truth causal graph's optimal path: i.e., retrying when the graph says retry, rolling back when the graph says rollback (reverse BFS yields an upstream affordance), skipping when precondition truncation applies, and progressing normally otherwise. If the planning decision is suboptimal (e.g., blindly retrying a `place_on` whose bridge state is lost, or skipping a chain that should be retried), the score is capped at **0.5** regardless of execution quality. T1 `pick_up(knife)` relaxes the threshold: any lift without dropping counts as at least partial success. Only the executing arm is evaluated; the standby arm's state is ignored.

C Additional Experimental Details

C.1 Baseline Descriptions

This section describes the two methods compared in the graph recovery experiments (Table 2); each is instantiated with all three VLM backbones.

- **Strategy-based (Ours, PiCausal):** Our A2-CAP-based neuro-symbolic approach, which uses a VLM to extract probabilistic attributes/affordances and constructs A2-CAP graphs via predefined causal strategies with weighted prior fusion.
- **VLM-only (zero-shot, no strategy registry):** The same VLM receives the same frames, the same object grounding and the same A2-CAP output grammar, but no strategy registry: no predefined interaction mechanisms, no coupling templates, and no LLM fallback that fills a template. In a single zero-shot pass the model must name the attribute and affordance nodes itself and propose the causal edges with confidences. This ablation isolates the contribution of the neuro-symbolic strategy layer from the contribution of the output formalism and the backbone.

Both methods use the same sampling policy (2 FPS on the video timeline, capped at 64 frames), the same image policy (longest side resized to 1024 px, JPEG quality 90), the same object grounding (identity attributes from benchmark metadata only) and the same backbone (Qwen3.8-Flash, DeepSeek-V4-Flash-Vision or GPT-5.6-Luna) at temperature 0. Every prediction stores the exact frame indices it used, so the evidence budget of a comparison can be audited video by video.

Evaluation protocol. Node and edge matching is identical for every method: the same relaxed node matcher (`relaxed=true`) normalizes verb synonyms and active/passive forms, and explicit equivalence nodes of the form $[A \equiv B]$ are expanded on both sides. No method receives a method-specific relaxation.

C.2 CLEVRER Mapping Protocol

CLEVRER provides event annotations for synthetic collision videos. To evaluate graph recovery in the A2-CAP formalism we map those annotations deterministically: (i) an object in motion becomes an attribute node `obj.attr.moving`; (ii) each contact becomes an affordance node `a.affd.collide(b)`, where `a` is the object already moving before contact (the faster one, by the dataset's direction rule); (iii) causal edges follow the annotated event order, `moving_source` $\rightarrow$ `source.affd.collide(target)` $\rightarrow$ `target.attr.moving`. Only the `physics_collide` strategy is active; no hand–object strategy applies. All mappings were manually reviewed on the evaluated subset. The mapping is many-to-one and discards geometry, so absolute scores are not comparable to A2-CausalHOI; the CLEVRER column is reported as a cross-domain sanity check, not as a like-for-like benchmark.

C.3 Causal Strategy Registry

The strategy module is a neuro-symbolic causal instantiation layer that converts object-centric visual evidence into weighted causal graph edges. Each object is represented by a normalized name, probabilistic attributes $\{p_i\}$, and grouped affordances. A strategy $S = (\mathcal{C}_S, \mathcal{E}_S, \tau_S)$, defined in Sec. 3.3, checks the relevant evidence from $\mathcal{C}_S$, computes its joint confidence, filters on τ_S, and emits weighted edges (v_i, v_j, w_{ij}) between symbolic attribute and affordance nodes.

The strategy execution order is randomized per call (with optional seed for reproducibility) to avoid deterministic bias toward any single strategy. The LLM fallback is invoked only after all 12 predefined strategies fail to activate. Two execution entry points are provided: `apply_strategies` for pairwise matching and `execute_plan_with_strategies` for structured multi-object interaction plans.

C.4 Causal Relation Types

Table 7 lists the three relation types of Sec. 3.3 with a concrete example of each; every edge emitted by the registry or by the LLM fallback instantiates one of them.

C.5 Baseline Behavior Comparison for Robot Planning

Table 8 provides a detailed capability matrix across all five robot planning baselines.

Strategy	Mechanism	Typical causal pattern
`tool_object`	Tool acts on target	Tool state → target action → target state change (cut, hit, wipe, stir, …)
`physics_collide`	Contact dynamics	Moving object → collision → displacement / deformation
`turn_on_activation`	Device activation	Trigger → on/active; self-activation or interaction-activation branch
`attr_power_electric`	Power transfer	Power source → electric affordance → powered state
`container_content`	Containment	Capacity attr. → contain affordance → inside relation
`support_place`	Support relation	Stable surface → support affordance → on-top state
`put_place`	Placement (composite)	Portable object → place action → on/in destination; delegates to contain/support
`pick_up`	Lifting	Portable object → pick action → held/removed state
`pick_and_place`	Sequential relocation	Chains pick→place with explicit $\mathrm{Affd} \rightarrow \mathrm{Affd}$ sequence edge
`transfer_content`	Pouring / transfer	Source contains content → transfer → destination contains content
`soft_fold`	Deformation	Soft/flexible → fold action → folded shape
`embed_insert_connect`	Assembly	Compatible parts → insert/connect → assembled/connected state
`llm` (fallback)	Open-ended repair	Unmatched cases → Qwen proposes typed edges with score-augmented confidence

Table 6: Complete causal strategy registry. Each strategy maps VLM-derived attributes and affordances to a typed causal edge pattern.

Cause	Effect	Relation	Example
State change	Attribute change	$\mathrm{Attr} \rightarrow \mathrm{Attr}$	Water temperature rises, increasing a thermometer reading.
State change	Affordance change	$\mathrm{Attr} \rightarrow \mathrm{Affd}$	A small, rigid block becomes placeable on a table.
Action occurrence	Attribute change	$\mathrm{Affd}\cdot \rightarrow \mathrm{Attr}$	Picking up an apple causes it to become held/suspended.

Table 7: A2-CAP causal relation types. Relations are weighted by prerequisite confidence and instantiate mechanism-level precondition–effect structure.

C.6 Metric Definitions

The five evaluation metrics used in robot planning experiments are defined as follows. Let $\mathcal{Z} = \{z_1, \ldots, z_K\}$ be the subtask sequence for one round, and R be the number of evaluation rounds.

Task Success Rate (SR). $\mathrm{SR} = \frac{1}{R}\sum_{r=1}^{R} s_r$, where $s_r \in \{0.0, 0.5, 1.0\}$ is the overall task completion of round r, derived from the operator's per-subtask scores: 1.0 if no final subtask score falls below 1.0, 0.5 if none falls below 0.5 but some is only partially successful, and 0.0 otherwise.

Sub-task Success Rate (Sub-SR). $\text{Sub-SR} = \frac{1}{R}\sum_{r=1}^{R} \frac{|\{z_i : \mathrm{score}(z_i) \geq 0.5\}|}{|\mathcal{Z}|}$, measuring the fraction of unique subtasks that reach at least partial success. When a subtask is retried, only its final outcome is counted. The denominator is the round's own subtask list, which differs by method and by round—the causal planners record 10, 12 or 14 subtasks for the same three fruits, since a cutting subtask may be split into two halves, Direct runs 16 in nine of its ten rounds, and VLM Planner decides its own 8–12 steps (Sec. C.7)—so Sub-SR is a fraction of a per-method denominator rather than of a shared step list.

	CAP-Plan	w/o verify	VLM Planner	Direct	Online Graph
			Planning		
Planning	Causal topo-sort	Causal topo-sort	VLM online step-by-step	Fixed plan order	Online causal graph
Flexibility	Fixed (topological)	Fixed (topological)	Dynamic (VLM chooses)	Fixed (array)	Fixed (topological)
			Execution		
Precond. check	VLM targeted query	Off (always pass)	None	None	VLM targeted query
Postcond. verify	VLM targeted query	Off (always pass)	VLM free assessment	None	VLM targeted query
T1 fruit detect.	VLM online	VLM online	VLM online	Hardcoded 16-step	VLM online
Arm avoidance	Plan + deferred apply	Plan + direct apply	Plan + direct apply	Plan + direct apply	Plan + exec. guard
			Recovery		
Failure detect.	VLM targeted query	Skipped (always pass)	VLM free assessment	None (human = stop)	VLM targeted query
Retry cap	3 per subtask	3 per subtask	3 per subtask	N/A	3 per subtask
Causal replan	Reverse BFS + gate	Off (always true)	VLM re-decides	No recovery	Reverse BFS + gate
Global attempt cap	2×#subtasks	2×#subtasks	2×#subtasks	N/A	2×#subtasks
			VLM usage		
Purpose	Targeted attr. query	Fruit detection only	Free scene assess.	None	Graph + attr. query
Calls / step	2 (pre + post)	0–1 (fruit only)	1 (assess + decide)	0	2 (pre + post)
			Metrics		
RR	Available	N/A	N/A	N/A	Available

Table 8: Capability matrix across all five robot planning baselines. Rows describe each variant's current configuration. **Online Graph** is implemented with visual graph generation and the same closed-loop executor as CAP-Plan; its only archived runs predate effect verification and causal replanning, and no valid round of the current configuration exists (§C.7). Thus no reported score exercises that column.

Execution Efficiency (EE). $\mathrm{EE} = \frac{1}{R}\sum_{r=1}^{R} W_r \cdot \frac{N_r^{\text{qualified}}}{N_r^{\text{attempts}}}$, where W_r is the round's mean per-subtask score, $N_r^{\text{qualified}}$ counts subtasks scoring ≥ 0.5 and N_r^{attempts} counts all attempts including retries. The first factor keeps partial success visible, the second penalizes redundant retries.

Recovery Rate (RR). $\mathrm{RR} = \frac{1}{R'}\sum_{r:\, n_r^{\text{retried}}>0} \frac{|\{\text{retried result rows that eventually score } \geq 0.5\}|}{|\{\text{retried result rows}\}|}$, i.e. the mean over the R' rounds that retried at least once; rounds without a retry are reported as not applicable (–) rather than as 0. The unit is the retried result row rather than the unique subtask, so a round that rolls back into the same subtask index more than once contributes one row per attempt to the denominator. Defined for methods with explicit failure detection and recovery (CAP-Plan and Online Graph); a precondition truncation ends the round without a retry and therefore does not enter the denominator.

Attempt Count (Att). Mean total attempts (including retries) per round. Lower is better; reflects recovery overhead. $\mathrm{Att} = \frac{1}{R}\sum_{r=1}^{R} N_r^{\text{attempts}}$.

C.7 Evaluation Coverage and Excluded Runs

Table 9 lists, per variant and task, how many rounds the main table aggregates. Only rounds the run record marks valid enter Table 3. The rule is registration-based: a round is 'valid' when it was registered in the original run log, and every unregistered round is left out of the headline numbers whatever its score. For T1, eight unregistered harness rounds nevertheless ran the full plan to the same subtask count as the valid ones, so we report the inclusive reading alongside the registered one: completing the task in $3/16$ T1 rounds (mean score 31.2 ± 40.3), against $2/8$ for the registered rounds and $0/10$ for each baseline variant. No equivalent inclusive reading is available for T2: its unregistered harness rounds all stopped part-way through the five-subtask plan (0–3 subtasks), and because the scoring scheme gives a stopped round no credit for the subtasks it never reached, including them would score an aborted run as a failure rather than as missing data. The T2 candidate metric JSONs were pruned before the record audit and cannot be re-scored; the T1 candidates and per-round ledger are released.

The empty and reduced cells are accounted for as follows. **Online graph** is out of scope on T1 by design. Its current real-robot mode generates a graph from a captured scene frame and passes the resulting plan to the closed-loop executor, but has no valid evaluation data. On T2, eight archived runs stopped after 0–4 of the 5 subtasks and used an earlier harness configuration without effect verification or causal replanning; they are excluded rather than re-scored. **VLM Planner** has 11

Variant	T1	T2
A2-CAP + $\pi 0.5$ (Ours)	8 valid (all recorded)	11 valid
w/o verification	10 valid (all recorded)	not run
VLM Planner + $\pi 0.5$	10 valid of 21 recorded	not run
$\pi 0.5$ Direct	10 valid (all recorded)	not run
A2-CAP w/ online graph	0 valid	0 valid

Table 9: Evaluation coverage behind Table 3.

further T1 runs left undecided because they executed only part of the plan. **Ours** has 8 further T1 rounds that ran the full plan without being registered; the T2 record carries no unregistered round that ran the full plan. No T2 baseline variant was run. The 11 T2 rounds come from two sessions—nine from the original benchmark record and two repeated after the camera and venue change—so we checked that pooling them is not what carries the result: the nine earlier rounds alone give 55.6 ± 52.7 task SR, against 59.1 ± 49.1 for all eleven, and both readings sit inside the same interval width.

What the valid rounds exercised. Replaying each round's archived subtask outcomes through the scoring code reproduces every stored value, so the main tables can be regenerated from the record rather than transcribed. Two properties of that record bound what the metrics support.

- **No valid round contains a rollback.** Only one episode in the whole archive triggers the reverse-BFS rollback (`replan_count` $= 2$, recovery depth 1.0, both on the same subtask), and it is a non-valid round; the valid set's replan counts and recovery depths are 0 everywhere. The two recoveries reported for T1 are therefore local retries after a failed attempt, not graph-guided rollbacks. The rollback path itself is exercised by that recorded episode, whose subtask outcomes are archived with the rest of the record; its benefit is not quantified by the aggregate table.

- **Round length is a per-method, per-round denominator.** For the same three fruits, the causal planners record 10, 12, or 14 subtasks (a cutting subtask may be split into two halves), Direct records 16 in nine of its ten rounds and 10 in one, and the VLM Planner decides its own 8–12 steps. Sub-SR is a fraction of that round's own subtask list, computed after de-duplicating retried subtasks so that each unique subtask counts once.

D CAP-Plan Recovery Details

D.1 Effect Failure: Causal Replan via Reverse BFS

When a verified effect $v_{\text{fail}} \in \text{Eff}(a_i)$ is absent after execution, the robot performed the action but the expected outcome did not materialize (e.g., the knife moved but the fruit is not sliced). The recovery procedure (Algorithm 1) runs a reverse BFS from v_{fail} with the out-degree gate $d^+(v) = \big|\{u : (v, u) \in \mathcal{E}\}\big|$:

Failed subtask	a_{skip}	BFS path	Result
`place_on(plate)`	`place_on`	`on` → `place_on`(penetrate) → `held`(d$^+$=1, penetrate) → `pick_up`	Rollback
`cut(banana)`	`cut`	`sliced` → `cut`(penetrate) → `held(knife)` (d$^+$=3, stop)	Retry
`cut(tomato)`	`cut`	`sliced` → `cut`(penetrate) → `held(knife)` (d$^+$=3, stop)	Retry
`pick_up(knife)`	`pick_up`	`held` → `pick_up`(penetrate) → `portable`/`on`(d$^+$ >1, stop)	Retry

Table 10: Reverse BFS recovery scenarios. The out-degree gate distinguishes bridge states ($d^+ = 1$, rollback) from stable states ($d^+ > 1$, retry).

Algorithm 1 Causal-Graph-Guided Recovery

Require: Causal graph $\mathcal{G}$, failed effect attribute v_{fail}, current subtask z_i, maximum depth $L_{\max}$
Ensure: Restart subtask index, or -1 for local retry

$a_{\text{skip}} \leftarrow \text{action}(z_i)$
$\mathcal{Q} \leftarrow [(v_{\text{fail}}, 0)]$, $\mathcal{S} \leftarrow \emptyset$
while $\mathcal{Q} \neq \emptyset$ **do**
  $(v, \ell) \leftarrow \text{pop_front}(\mathcal{Q})$
  if $v \in \mathcal{S}$ or $\ell > L_{\max}$ **then**
    continue
  end if
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$
  for each $p \in \text{Pred}_{\mathcal{G}}(v)$ **do**
    if $p \in \mathcal{V}_{\text{affd}}$ and $p = a_{\text{skip}}$ **then**
      push $(p, \ell + 1)$ to $\mathcal{Q}$ {penetrate failed action}
    end if
    if $p \in \mathcal{V}_{\text{affd}}$ and $p \neq a_{\text{skip}}$ **then**
      return $\text{index}(p)$ {upstream root cause}
    end if
    if $p \in \mathcal{V}_{\text{attr}}$ and $d^+(p) = 1$ **then**
      push $(p, \ell + 1)$ to $\mathcal{Q}$ {bridge attribute}
    end if
  end for
end while
return -1 {stable attribute or no upstream cause; retry locally}

Table 10 illustrates the mechanism across representative failure cases:

The key insight is that rollback is a consequence of topology, not of strategy type. In the `place_on` failure, `held(fruit)` has out-degree 1—it only leads to `place_on`, so losing this bridge state makes retrying placement meaningless and forces a return to `pick_up`. In the `cut` failure, `held(knife)` and `sharp(knife)` each have out-degree 3 (the knife is still held and sharp, usable for other cuts), so a local retry of `cut` is sufficient.

D.2 Precondition Failure: Causal Truncation via Forward BFS

When a precondition attribute $v_{\text{fail}} \in \text{Pre}(a_i)$ is absent before execution, the action cannot even be attempted—the environment does not satisfy its requirements (e.g., the fruit is not on the cutting board, so cutting is impossible). Unlike effect failures, precondition failures are not recoverable by retrying the same action; the correct response is to skip all subtasks that transitively depend on the missing state.

The procedure `find_downstream_affordances` runs a forward BFS from v_{fail}:

1. Push v_{fail} into a queue; initialize a blocked set $B = \emptyset$.
2. While the queue is non-empty, pop the front node u:
   - If $u \in \mathcal{V}_{\text{affd}}$, add u to B and enqueue all its successors.

	Effect FAIL (Replan)	**Precondition FAIL (Truncation)**
Trigger	After execution; verified effect absent	Before execution; precondition absent
Semantics	Action attempted but did not succeed	Action cannot be attempted at all
Recoverability	Recoverable—retry upstream cause	Irrecoverable—skip dependent chain
Graph traversal	Reverse BFS + out-degree gate (Alg. 1)	Forward BFS (downstream propagation)
Outcome	Rollback to root-cause subtask or local retry	Skip blocked subtasks, continue

Table 11: Two complementary failure-recovery paths in CAP-Plan. Both derive their decisions from graph topology rather than strategy-specific heuristics.

- If $u \in \mathcal{V}_{\text{attr}}$, enqueue all its successors (transitively propagate the failure).

3. Every subtask z_j whose action a_j matches an affordance in B is marked `precondition_ok` $=$ false and skipped.

This respects causal topology precisely. For a failed `banana.attr.on(cutting_board)`, the forward BFS reaches `cut(banana)` $\rightarrow$ `banana.attr.sliced` $\rightarrow$ `pick_up(banana)` $\rightarrow$ `banana.attr.held` $\rightarrow$ `place_on(plate)`, blocking all three downstream subtasks. For a failed `banana.attr.sliced` discovered after `cut(banana)` has already executed, the BFS only reaches `pick_up(banana)` $\rightarrow$ `place_on(plate)`—the already-completed cut is not penalized.

D.3 Knife Attribute Exception

Knife attributes (`knife.attr.sharp`, `knife.attr.held`) are pragmatically excluded from precondition-driven truncation. The VLM's detection of knife properties is systematically unreliable under dual-arm robotic occlusion and varying camera viewpoints, producing false-negative precondition failures that would incorrectly truncate long action chains. Instead of truncating, knife precondition failures emit a warning and execution proceeds normally.

D.4 Two-Path Comparison

Table 11 summarizes the two complementary recovery paths. The core design principle is that the causal graph alone encodes planning, effect verification, and both recovery modalities: reverse BFS with the out-degree gate determines *whether* to roll back, and forward BFS determines *what* to skip.

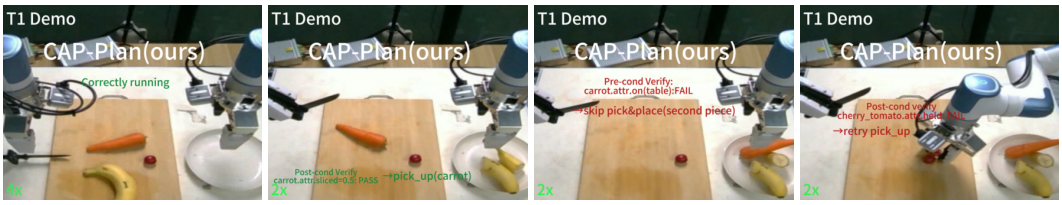
E Additional Results

E.1 Qualitative Execution Demo on T1

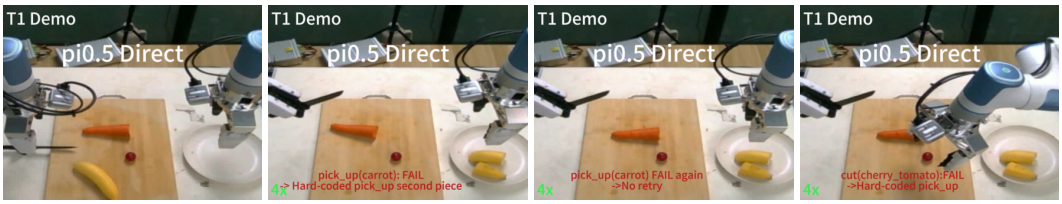
Figure 5 shows key execution moments from four methods on the T1 cut-and-plate task, captured from overhead and wrist cameras. Each row is one method; each column advances in time. **Red text** marks anomalies and failure-handling events. The video accompanying this submission plays the same four runs end to end.

E.2 Graph-Recovery Sample: Representativeness and Stability

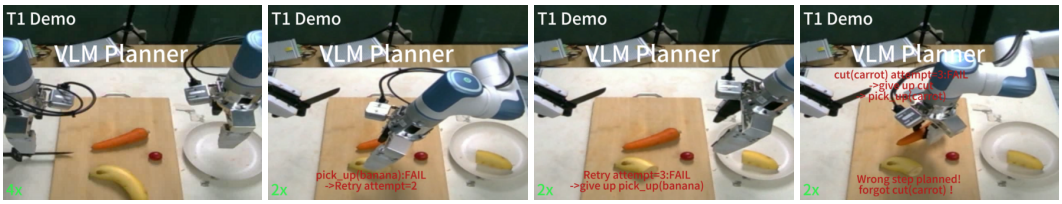
The graph-recovery tables are computed on a fixed sample of 100 clips drawn from 98 of the 256 A2-CausalHOI interaction tasks (two tasks contribute two clips each). The sample is a duration-stratified draw: its duration quartiles track the corpus closely (25th 5.6 vs. 5.5 s, median 7.4 vs. 7.5 s, 75th 9.6 vs. 9.6 s), it covers 98 of the 256 interaction tasks (38%), and it matches the corpus on the content axes that drive graph recovery—object inventory (3.77 vs. 3.61 candidate objects per task), ground-truth graph size (4.45 vs. 4.34 edges) and step count (2.47 vs. 2.45), with the share of each of the 12 strategies deviating by at most 1.8 percentage points. Its one measurable deficit is the upper tail of clip length: the 95th percentile is 11.4 s against 13.3 s corpus-wide, since the longest clips are under-represented.



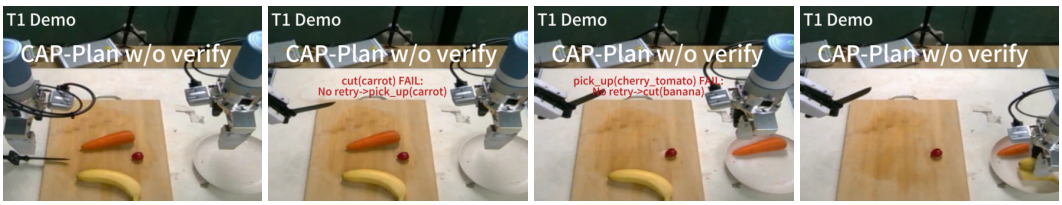
CAP-Plan + π0.5 (Ours) — detects failures via VLM postcondition checks; recovered through causal replan (rollback / retry).



π0.5 Direct — no failure detection; blindly advances after every step regardless of outcome.



VLM Planner + π0.5 — VLM re-assesses and re-chooses, but without causal constraints wastes attempts on hopeless retries.



CAP-Plan w/o verification — causal plan exists but no VLM postcondition checks; failures silently pass.

Figure 5: **Qualitative execution comparison on T1 (cut-and-plate).** Four methods share the same trained π0.5 policy. Only full CAP-Plan (row 1) detects failures and recovers by causal replan; Direct (row 2) and w/o verify (row 4) proceed blindly, and VLM Planner (row 3) detects failures but retries states that cannot be fixed. Red annotations mark the anomaly at each step.

The sample size is adequate for the paired comparison it supports. Resampling the 100 paired per-clip F1 differences 200 times at each of $n \in \{20, 30, 50, 70\}$ leaves the estimate unchanged (ΔF1 $+0.37$ for Qwen3.8-Flash and DeepSeek-V4-Flash-Vision, $+0.47$ for GPT-5.6-Luna) and the 95% interval excluding zero in 100% of draws; at the full n=100 the paired gains are $+0.370$ ($[+0.312, +0.427]$, $84/100$ clips won) with Qwen3.8-Flash, $+0.386$ ($[+0.328, +0.446]$, $84/100$) with DeepSeek-V4-Flash-Vision and $+0.468$ ($[+0.414, +0.519]$, $90/100$) with GPT-5.6-Luna. The comparison is therefore limited by clip coverage, not by the number of clips.

Method	Backbone	Frames	ΔF1 (2→4 FPS)	95% CI	W/L
PiCausal	DeepSeek-V4-Flash-Vision	16.7→32.8	$+0.030$	$[-0.100, +0.178]$	4/2
PiCausal	Qwen3.8-Flash	16.7→32.8	$+0.005$	$[-0.063, +0.076]$	3/5
VLM-only	DeepSeek-V4-Flash-Vision	16.7→32.8	-0.041	$[-0.096, +0.012]$	4/4
VLM-only	Qwen3.8-Flash	16.7→32.8	-0.000	$[-0.069, +0.063]$	4/4

Table 12: **Frame sampling sensitivity (paired, same videos).** Raising the budget from 2 to 4 FPS doubles the frames and the image payload ($1.29 \rightarrow 2.52$ MB per video), yet every confidence interval on ΔF1 includes zero, whereas the PiCausal–VLM-only gap (≈ 0.44 F1) persists across sampling rates and backbones. "W/L" counts per-video F1 wins/losses.

E.3 Frame Sampling Sensitivity

The main table fixes one evidence budget (2 FPS on the video timeline, at most 64 frames). We verify that this budget is not the limiting factor for either method. On a stratified 10-video subset of A2-CausalHOI (2.4–17.3 s, i.e. 5–70 sampled frames depending on duration) we swept the sampling rate over 2, 3 and 4 FPS with no per-video cap, and evaluated PiCausal and the VLM-only baseline with both original backbones on exactly the same clips, using the same object grounding and the same output vocabulary as the main runs (Table 12).

E.4 EPIC-KITCHENS-100 Protocol, Ground Truth and Object Lab

The sample is drawn from the official EPIC-KITCHENS-100 validation split Damen et al. (2022) (seed `20260807`, n=80) under three pre-registered constraints: the segment's object must exist in a closed 137-name vocabulary (derived from the official EPIC noun classes), clips last 1.5–10 s, and no participant or video contributes more than four or two segments. Clips are fetched with ± 1 s of context. Annotation follows the same three-role design as the main study—five jointly calibrated segments (which also lie in the blind-first block and are excluded from any agreement statistic), twenty blind-first segments (the annotator writes the interaction steps from the clip before opening the draft, which is what makes the anchoring measurement possible), and eleven double-annotated segments, drawn from the non-blind remainder so that no segment needs three passes, for which we report the primary admissible reading (the alternative moves the Table 2 metrics by less than 0.02 F1). The remaining 44 segments are shared across the blocks. Of the 80, 53 carry a reviewed graph and 50 a non-empty target; the EPIC column is scored on those 50, and the three empty ones (two whose draft was rejected and left no steps, one whose single `llm` step carries no `chain`) are reported as coverage rather than averaged in as zeros, since an empty target forces $F1{=}0$ for every method. Both methods receive the same per-clip object inventory the benchmark supplies, and are scored on the same edge-level target and with the same matching rule as the two in-domain benchmarks.

The blind-first subset gives a first read on anchoring: among the 51 segments that carry ground truth and a non-empty draft, the 11 blind-first ones are edited at a median rate of 3.0, identical to the 3.0 of the other 40 (means 4.65 vs. 3.09; direct acceptance 0.11 vs. 0.16). This anchoring count is a different partition from the scored blind arm of Table 13, which needs a reviewed graph and a prediction as well (14 blind against 32 draft-assisted). With $n{=}11$ we report the anchoring read as inconclusive rather than as evidence.

Five protocol details matter for reading the EPIC column. The first three are protocol differences from the in-domain setup; the last two are contract and template defects that affected every benchmark and were corrected before scoring.

(i) Ground truth = template expansion of the steps. On A2-CausalHOI the stored graph is exactly what the 13 registered strategies produce from the annotated steps (the 12 templates plus the LLM fallback): feeding each scene's own steps back through the strategy registry reproduces $174/211$ of the stored graphs exactly (mean edge-F1 0.944; the remaining 37 differ by one edge direction or one missing precondition), with 4.52 template edges against 4.45 stored edges per clip. The EPIC bundles were instead written as free-hand node strings, and the same regeneration reproduced $1/56$ of them (edge-F1 0.149; scored against the preserved hand-written graphs of the reviewed segments, $2/42$ exact at edge-F1 0.386, App. E.5). Because PiCausal writes template nodes, even a perfectly recognised interaction often failed to match, which capped the edge-level F1 near 0.15 whatever the

perception quality. The EPIC ground truth is therefore regenerated as the template expansion of the steps (the regeneration is released with the code); the hand-written graphs are preserved verbatim under `gt.causal_graph_handwritten`.

(ii) The object lab had to be made template-compatible. Every template gates on physical attributes read from the lab (`small`, `portable`, `light`, `edged`, `electric`, ...). The in-domain lab covers 2.01 of the 19 gate attributes per object and leaves only 3% of its objects without any; the EPIC lab covered 0.23 and left 82% of its objects without a single one (`portable` on 18% of objects against 90% in-domain, `small` on 1% against 65%). With the gates closed the registry cannot even select the annotated strategy: a correct `pick_up(banana, cupboard)` step was instantiated as a `hold/place` chain. We therefore derive the missing attributes from information the EPIC lab already carries—its affordance lists and the noun-class keywords—using an explicit rule table over the affordance lists and noun-class keywords it already carries, and force fixed installations such as `worktop`, `shelf`, `hob` or `cupboard` to stay non-mobile. Coverage rises to 6.45/19 with 0% empty objects, the templates then honour the annotated strategy, and the number of hand-written graphs the regeneration reproduces rises sharply (the template-edge density under this repair is reported under (v), where it is measured on the final pool).

(iii) Object candidates. Both methods receive the segment's object inventory as a per-segment candidate list; the interactions themselves are never provided. The lists are longer than in-domain—17.0 names per segment against 3.8 on A2-CausalHOI, because the EPIC inventory comes from an independent object-proposal pass rather than from a constructed scene—which is a protocol difference rather than a property of the domain. The four segments whose candidate list fell back to the full 137-name vocabulary are exactly the four that carry no usable target, and passing the full vocabulary instead of the per-segment list costs precision.

(iv) The symbolic contract had to be aligned with the ground truth. Before the repair list above, the EPIC graph stood at 0.42 F1, and the reason was not perception: the templates read the object state twice—attribute products gate a strategy, and the argmax affordance verb *names the node*—while the ground truth is instantiated from the object lab (§E.4(i)) and the prediction was expanded from the perception-fused state. Wherever the two disagreed, a correctly planned interaction produced a differently named node (`place` vs. `put`) or lost a whole edge to the weight threshold; 22% of the planned interactions had their strategy silently rewritten by the registry fallback against 5% in-domain, and the edge precision of an interaction whose object pair was right was 0.49 against 0.70 in-domain. The symbolic stage now gates on the lab for every benchmark, which is the contract the appendix already described for the gates. The change is worth +0.03 F1 in-domain and +0.12 on EPIC.

(v) Two template bugs that silently dropped annotations. The registry guarded every strategy call with `if b_obj is not None`, so a step without a partner object could never be instantiated even by `turn_on_activation` and `soft_fold`, whose templates take the subject alone; 25 of the 117 EPIC steps are of that form, and two segments regenerated to an entirely empty graph. The `turn_on_activation` template also routed exclusively through a *functional* verb (`light`, `heat`, `cook`, ...) and therefore returned nothing for a stateful appliance such as a tap or a fridge, whereas the annotation schema and `scripts/epic/steps_to_graph.py` both define the strategy as `a.attr.closed`→`a.affd.turn_on`→`a.attr.open`; it now falls back to that chain for objects that carry the `open`/`closed` pair. Together the two fixes lift the regenerated target from 3.63 to 3.96 template edges per clip against 4.64 hand-written edges, and take the steps that yield no edge at all from 25% to 15%.

What remains is annotation-side: 18% of the EPIC steps use the free-form `llm` strategy without an instantiable `chain`, 14% are `transfer_content` steps that omit their `content`, and seven segments therefore expand to fewer edges than their annotators drew (4.06 vs. 4.64 per clip). Completing those fields is what would bring the EPIC ground truth to in-domain density.

Two implementation notes matter for reproducibility. The hyperparameter replay is offline for cached perception outputs, but segments whose perception output contains an explicitly `llm`-labelled interaction may re-query the backbone for that interaction in each configuration; the sweep is paired, while exact replay requires caching those expansions. Caching changes the cost, not the ordering, which is stable across every configuration of the sweep. Second, EPIC instance suffixes are normalised before matching, so the EPIC numbers use the same relaxed node matching as the in-domain tables.

Backbone	Method	Blind (14)	Draft-assisted (32)	Hand-written (43)	In-domain
Qwen3.8-Flash	PiCausal	0.438	0.636	0.264	0.610
Qwen3.8-Flash	VLM-only	0.075	0.081	0.141	0.241
DeepSeek-V4-FV	PiCausal	0.365	0.497	0.247	0.593
DeepSeek-V4-FV	VLM-only	0.037	0.044	0.092	0.207
GPT-5.6-Luna	PiCausal	0.301	0.560	0.228	0.709
GPT-5.6-Luna	VLM-only	0.109	0.095	0.172	0.241

Table 13: **EPIC-KITCHENS edge-F1 under three annotation regimes**, using the main-table predictions and matching rule. "Blind" segments had their graph written before the annotator opened the model draft; "draft-assisted" covers 32 other segments. Four additional segments in the 50-segment EPIC main column come from separate rater bundles. "Hand-written" scores against the free-form graphs preserved under `gt.causal_graph_handwritten` (available for 43 segments). The final column repeats the in-domain A2-CausalHOI numbers for reference. Segment counts are per arm and identical across backbones.

E.5 Oracle Control and Annotation-Robustness Arms

Because a released target graph is the projection of that scene's annotated steps and the prediction passes through the same templates, two controls bound how the graph-recovery numbers should be read. Both are computed by `scripts/make_schema_controls.py`, which reuses the registry path and the metric code of the main table; all numbers below are produced by that script rather than transcribed.

Oracle: what step inference can buy. Feeding each scene's *annotated* steps back through the registry—the same code PiCausal runs, on the same object lab—reproduces the stored graphs exactly for 174 of the 211 A2-CausalHOI scenes (mean edge-F1 0.944; 4.52 template edges against 4.45 stored) and for *all* 42 reviewed EPIC segments that carry steps (edge-F1 1.000). The EPIC figure is by construction: that target was regenerated as exactly this expansion (§E.4(i)). The oracle is therefore an upper bound that no perception improvement can exceed, and the gap between it and the reported F1 is what step inference costs. Scored instead against the EPIC graphs the annotators wrote in free form, the same expansion reaches only edge-F1 0.386 ($2/42$ exact), which is the size of the vocabulary mismatch the regeneration was introduced to remove.

Annotation-robustness arms. Twenty EPIC segments were pre-registered as blind: the annotator wrote the interaction steps and the graph from the clip alone, before the model draft was opened. Fourteen of them carry a reviewed graph and a prediction. Table 13 scores the current main-table predictions on that arm and on 32 draft-assisted segments, and against the preserved hand-written graphs where available. The registry advantage survives the independent annotation with all three backbones ($+0.19$ to $+0.36$ F1 there, against $+0.45$ to $+0.56$ on the draft-assisted segments), although PiCausal's blind-arm F1 is lower (0.30–0.44 against 0.50–0.64). Four other scored main-table segments come from the separate rater bundles and are not assigned to either arm. We read the blind arm as the more conservative transfer estimate; its small sample limits precision.

Annotator agreement. On the 11 double-annotated EPIC segments, four have both raters' bundles filled. Treating each segment's edge and node universe as the union of the two raters' items and scoring absence as well as presence, agreement is below chance: pooled edge $\kappa = -0.47$ ($p_o = 0.29$, $p_e = 0.51$, 21 items) and node $\kappa = -0.52$; the macro edge F1 between the two raters is 0.56. Two annotators handed the same schema and the same clip therefore agree on roughly half the edges, which is the ceiling any method can be asked to reach on this task, and it is comparable to what PiCausal scores against the stored graph. With four segments this is an indication rather than a measurement, and we report it as such; the remaining seven double bundles were not included in this release, and the script that computes the statistic (`scripts/epic/iaa_agreement.py`) is included.

	A2-CausalHOI		CLEVRER-200		EPIC-KITCHENS	
	Qwen	DeepSeek	Qwen	DeepSeek	Qwen	DeepSeek
(a) activation threshold τ, lab contract (F1 ↑)						
0.2	**0.610**	**0.593**	**0.537**	**0.405**	**0.610**	**0.523**
0.3	**0.610**	**0.593**	**0.537**	**0.405**	**0.610**	**0.523**
0.4 *(default)*	**0.610**	**0.593**	**0.537**	**0.405**	**0.610**	**0.523**
0.5	**0.610**	**0.593**	**0.537**	**0.405**	0.605	0.520
0.6	0.469	0.445	**0.537**	**0.405**	0.568	0.479
(b) prior–observation fusion (w_p, w_o) at w_h=0.8, fused variant						
$(0.1, 0.9)$	**0.581**	**0.559**	**0.539**	**0.405**	0.508	0.425
$(0.3, 0.7)$ *(default)*	0.577	0.551	0.537	**0.405**	0.504	0.424
$(0.5, 0.5)$	0.577	0.546	0.537	**0.405**	**0.511**	**0.425**
$(0.7, 0.3)$	0.577	0.541	0.537	**0.405**	0.494	0.423
(c) hidden-state prior decay w_h at $(0.3, 0.7)$, fused variant						
0.6	0.508	0.488	0.500	0.302	0.436	0.370
0.8 *(default)*	0.577	0.551	**0.537**	**0.405**	0.504	0.424
1.0	**0.600**	**0.585**	**0.537**	**0.405**	**0.522**	**0.471**

Table 14: **Hyperparameter sensitivity of the PiCausal symbolic layer.** Every configuration is replayed on the published perception pass, so each row is paired with the default on identical clips; bold marks the best value in each column. Panel (a) uses the paper's contract—the templates gate on the object lab—and its τ=0.4 row reproduces the main table exactly on all six columns. Panels (b) and (c) are measured with the alternative `fused` object state, because under the lab contract the fusion weights never reach the templates: establishing that invariance is the point of those two panels. "Qwen"/"DeepSeek" abbreviate the two backbones swept here; GPT-5.6-Luna is not part of this sweep.

E.6 Hyperparameter Sensitivity of the Symbolic Layer

The graph-recovery result depends on three symbolic-layer hyperparameters: the strategy activation threshold τ, the prior/observation fusion weights (w_p, w_o), and the hidden-state prior decay w_h. We sweep them to check that the reported advantage is not an artefact of one tuned setting. The VLM perception pass is the only expensive step and it consumes none of the three numbers, so we replay the sweep offline on the *published* perception pass (identical frames, object grounding and VLM outputs) and every configuration is paired with the default on identical clips. Pre-fusion observations are recovered exactly by inverting the default fusion (verified to reproduce all 26,743 published attribute/affordance values with zero mismatch, and independently re-checked against a fresh perception pass). The registry's LLM fallback is disabled during the sweep to keep it deterministic; switching it off leaves the default configuration of the two in-domain benchmarks bit-identical to the main table, and on EPIC it moves F1 by 0.01. Because the templates gate on the object lab (§E.4(iv)), τ is swept under that contract—the one the main table uses—while the two fusion panels are swept under the alternative `fused` object state, which is the only mode in which w_p, w_o and w_h can reach the templates at all.

Three observations follow. **(i) Under the paper's contract the symbolic layer is invariant to τ, not merely insensitive.** Because the templates gate on the object lab, whose attributes all carry the same prior confidence, every configuration with $\tau \in [0.2, 0.4]$ selects the same strategies and keeps the same edges: F1 is *bit-identical* on A2-CausalHOI, CLEVRER and EPIC for the two backbones swept here, and at τ=0.5 it moves by at most 0.005 (two EPIC columns) with nothing else changing. The threshold only starts to prune prerequisite edges at τ=0.6, where it costs 0.14–0.15 F1 on A2-CausalHOI, 0.04 on EPIC and nothing on CLEVRER (every emitted edge there is a high-confidence `physics_collide`). The default τ=0.4 therefore sits in the middle of a plateau rather than on a tuned peak. **(ii) The consensus weights are not load-bearing.** Measured on the alternative `fused` object state—the only mode in which they can reach the templates—moving (w_p, w_o) from $(0.1, 0.9)$ to $(0.7, 0.3)$ changes F1 by at most 0.018 in any cell (A2-CausalHOI and EPIC, DeepSeek) and by under 0.01 in the others. **(iii) Only the hidden-state decay is sensitive, and only in that mode.** With

w_h=0.6 F1 drops by 0.037–0.125 in all six cells; setting w_h=1.0 helps on A2-CausalHOI (+0.023 and +0.034) and EPIC (+0.018 and +0.047) and is neutral on CLEVRER. Crucially, even the largest hyperparameter effect (0.14 F1 at the extreme τ=0.6) is far smaller than the registry effect in the main table (+0.370 to +0.468 F1 in-domain, +0.43 to +0.52 on EPIC, always over the zero-shot baseline), and no setting of any of the three numbers changes the graph-recovery ordering.

F Detailed Limitations

Coverage bound. The registry fixes the space of expressible mechanisms, so a correct plan is only expressible for interactions the 12 strategies cover; graph quality is bounded by that coverage and by VLM perception reliability, and the harness inherits both. Only 6 of the 12 predefined registry strategies carry policy training data, and the two scored robot tasks instantiate 4 of those 6, so the manipulation evidence does not exercise the whole registry.

VLM reliability. CAP-Plan's precondition and effect checks read attributes from a VLM; occlusion and arm viewpoints cause systematic false negatives, which is why the knife attributes are exempted from truncation (§D.3). A false negative upstream of a bridge state can therefore trigger a rollback that was not needed, and the reverse error silently skips a needed one.

Round selection. The headline robot numbers aggregate the rounds the original run log registered. For T1, eight further completed rounds exist and the inclusive reading is reported alongside the curated one (§C.7). For T2 the unregistered rounds stopped part-way through the five-subtask plan and their metric records are incomplete, so no inclusive reading can be scored.

Recovery evidence. The valid rounds exercise retry, not rollback: their replay logs contain no plan rebuild, so the backward causal tracing of §D is exercised by recorded rollback episodes rather than measured inside this evaluation, and its benefit is not quantified. Coverage of further strategies such as assembly or device activation is likewise future work.

G Training and Implementation Details

PiCausal configuration. The VLM perception module uses the configured backbone (Qwen3.8-Flash or DeepSeek-V4-Flash-Vision via the DashScope/DeepSeek APIs, or GPT-5.6-Luna over an OpenAI-compatible gateway) with temperature 0. The object-prior database covers 89 everyday categories; priors are hand-annotated once and reused across all videos. The fusion formula (Eq. 1 in the main paper) uses weights 0.3 (prior) and 0.7 (observation) when both signals are positive, preventing stale priors from dominating dynamic visual evidence. The strategy activation threshold is $\tau = 0.4$ across all strategies. All three numbers are swept in App. E.6 (Table 14); the graph-recovery conclusion is invariant to $\tau \in [0.2, 0.4]$—0.005 at most up to τ=0.5—and stable across the full consensus-weight range.

Graph recovery baselines. Both methods are evaluated with the same sampling budget (2 FPS on the video timeline, at most 64 frames) and the same image policy (longest side 1024 px, JPEG quality 90); the VLM-only baseline sees the identical frames, the identical object grounding and the identical output grammar, and differs only in that no strategy registry is available. Evaluation uses the same relaxed node matching (`relaxed=true`) for every method.

Robot policy training. The π0.5 VLA policy is trained on 600 skill-level episodes across 6 strategies (`pick_up`: 162, `put_place`: 108, `tool_object`: 180, `transfer_content`: 60, `embed_insert_connect`: 60, `turn_on_activation`: 30). Training uses 2×H200 GPUs with full-weight fine-tuning on the LeRobot-format dataset. Each episode is a single skill with a natural-language instruction prompt and standardized start/end states. The policy checkpoint is step 43,000 from the π0.5 base model and is kept frozen for all planning experiments; the A2-CAP harness performs no policy update. At test time, CAP-Plan dispatches strategy-specific instructions (e.g., `"Cut the banana in half using the knife"`) with arm assignment decided per strategy and task context.

Platform. All robot experiments run on the dual-arm Xtrainer platform with two 6-DOF arms, each equipped with a parallel-jaw gripper. Three RealSense cameras (left wrist, right wrist, overhead) provide visual input. The VLM verification queries are sent to Qwen-VL via DashScope API with per-attribute prompts constructed from the causal graph's effect and precondition nodes.