# Where Reasoning Does Not Matter: Isolating Chain-of-Thought in a Modular, Agentic VLM Manipulation Pipeline

Basu Hela
*HiRo Lab, CPS*
*IISc*
Bengaluru, India

Ravi Prakash
*HiRo Lab, CPS*
*IISc*
Bengaluru, India

***Abstract*—End-to-end Vision–Language–Action (VLA) models fold semantic reasoning and physical control into one set of weights, so when they fail no component can be blamed or repaired. We instead keep a frozen multimodal VLM as an explicit, external reasoner driving a modular pipeline whose perception, grasping, and planning stages are off-the-shelf—Reason–Predict–Plan (RPP). Because the VLM is the only reasoning component, its chain-of-thought becomes a single controllable variable: we vary the model's native reasoning budget (full vs. minimal) with every other stage held identical, scoring each episode by ground truth across four Isaac Lab tasks ($n$=30 per condition, $n$=50 on Shelving). Reasoning matters in one place. It is decisive only on Shelving (52% vs. 2% success), the one task that must ground *where* to place an object into a partially observed shelf; the three tasks whose action sequence is forced once the target is named show no measurable effect, and a pilot "reasoning hurts" result dissolved under ground-truth scoring. The effect is not where intuition puts it: reasoning leaves *categorical* scene understanding untouched—the minimal model names the target and recognises the shelf identically—and a good-box counterfactual attributes most of the gain to the *metric* grounding of the destination, a correct placement box alone lifting an otherwise-minimal pipeline $1/50 \rightarrow 22/50$ with the loop's thrash a downstream symptom. The message is practical: more reasoning mostly buys latency unless a task forces a genuine metric-grounding decision, and a modular design is what lets us see which regime a task is in.**

## I. Introduction

Consider a household shelving task. A robot is asked to "put the bowl on the bottom shelf," but a mug is sitting in front of the bowl. The robot must *reason* (the mug is in the way; stash it first; the bowl should be grasped from above to clear the lip), *select a stable 6-DOF grasp* on the bowl in the cluttered pile, and then *execute* a collision-free trajectory to a placement pose on a partially observed shelf. Several different things can go wrong, and the cause of the failure is information the operator wants. Was the wrong object selected? Was the grasp geometrically infeasible? Did the trajectory clip the shelf? Even with the right object and a feasible grasp, was the destination itself—where on the shelf the bowl should go—grounded correctly? Each of these has a different repair.

End-to-end Vision–Language–Action (VLA) models [3, 11, 2] compress this entire decision chain into one set of weights. This is appealing when training data is plentiful and tasks are within distribution, but it has a known cost: when the model fails, the failure is not localised, and it is rarely clear whether more training data, a different action representation, a better camera, or a structural change to the model is the appropriate intervention. Modular pipelines [1, 12] restore localisation but typically replace VLM-driven reasoning with hand-written task scripts, sacrificing the capability that made VLAs interesting in the first place.

This paper argues that the two strengths can be combined. We present **Reason–Predict–Plan (RPP)**, a pipeline that uses a strong multimodal VLM [7] as (i) a per-step *tool selector* (in the agentic spirit of SayCan [1, 21], with a deterministic feasibility mask in place of per-skill RL value functions), (ii) a *per-grasp validator* that inspects rendered overlays of candidate gripper poses returned by an off-the-shelf 6-DOF grasp predictor [9] across a wrist camera and a fixed front camera, and (iii) a *geometry-grounded placement predictor* that derives a 6-DOF placement pose from single-view depth on open surfaces and, for receptacles, from shelf-tier geometry perceived from that same depth. Selected grasp and placement poses are routed to a GPU motion planner [17] with a voxel collision world built from depth [14]. The pipeline is fully implemented and runs end-to-end on four Isaac Lab [15] manipulation environments.

Because the VLM is the only reasoning component in this design, we can vary chain-of-thought directly instead of retraining weights. Modern multimodal models expose a native reasoning-budget control, Gemini's `thinking_level` [7], that sets how many internal reasoning tokens the model spends before answering. We use it as a clear ablation: *high* gives the model full chain-of-thought at every hook (tool selection, occlusion reasoning, per-grasp validation), while *minimal* suppresses deliberate reasoning without touching one pipeline component. The same agentic loop, validator, and planner run in both conditions; only the depth of the VLM's reasoning changes. This isolates the contribution of reasoning from the perception and planning machinery around it, the question a workshop on semantic reasoning is asking.

*a) What this paper contributes.:*

1) A modular Reason–Predict–Plan pipeline for cluttered ma-

nipulation that uses VLM reasoning at three explicit hooks — per-step tool selection, occlusion reasoning, and per-grasp overlay validation (cross-checked across a wrist and a fixed front camera) — with placement poses grounded geometrically in depth and depth-perceived receptacle geometry, and without per-skill RL. The only scripted skill—a pre-recorded-demonstration placement fallback—is exposed solely as a recovery option and proves inert (it reaches no success; App. D), so task semantics are carried by VLM reasoning rather than task-specific scripts.

2) A single-knob chain-of-thought ablation: because the VLM is the only reasoner, we vary its native reasoning budget (`thinking_level` high vs. minimal) while holding the entire pipeline constant, so the contribution of deliberate reasoning is measured directly rather than confounded with disabling pipeline stages.
3) An implementation on four Isaac Lab manipulation environments (Shelving, Kitchen-Tidying, Occluded-Assembly, Delicate-Unstacking) with a 7-DOF arm, including the agentic tool loop, the VLM grasp validator, the geometry-grounded placement predictor, and the cuRobo/nvblox planner.

*b) Scope.:* The contribution is scoped to semantic reasoning and geometric planning; learned residual control is out of scope, as is a multi-model study of whether the regime structure is model-general or specific to one VLM.

## II. Related Work

*a) LLM/VLM-driven task planning for manipulation.:* SayCan [1] grounds language in robot affordances by combining a large language model (LLM, the *Say* term) with per-skill reinforcement-learning (RL) value functions (the *Can* term); Inner Monologue [8] adds environment feedback loops, and Code-as-Policies [12] emits executable Python instead of structured tool calls. Closest to us, Yamao et al. [21] drop the RL *Can* term in a home-service planner, replacing it with rule-based skill filtering plus abstract-noun clarification. Our agentic loop follows this trajectory: a hard mask filters skill feasibility before each VLM call, eliminating per-skill RL while retaining a tool-typed action space.

*b) Chain-of-thought reasoning for embodied control.:* Chain-of-thought (CoT) prompting [20] elicits intermediate reasoning steps from LLMs. ECoT [24] embeds CoT inside a VLA, training it to emit reasoning tokens before action tokens; CoT-VLA [25] explores a similar design. We invert this architecture: where ECoT trains CoT *into* one end-to-end model, we keep the VLM frozen and use it as an external reasoner calling a separate grasp predictor and motion planner. This trades end-to-end optimisation for a clear reasoning/control separation and—the focus of this paper—a reasoning component whose depth we control directly, modulating the frozen model's native reasoning budget and observing the effect on the same downstream stages.

*c) Where, and how much, reasoning helps.:* A growing line measures *when* test-time reasoning helps an embodied policy, and often finds it does not. ECoT-Lite [5] drops embodied reasoning at test time on in-distribution tasks with little loss, and Gemini Robotics-ER [6] exposes the same native reasoning-budget knob we use, on a modular orchestrator with uneven per-task gains. Three concurrent 2026 results sharpen the question. Trinh et al. [19] corrupt a VLA's chain-of-thought and find the action decoder sensitive only to the entity *name*; RARRL [13] and Act-Think-or-Abstain [10] instead *learn* when to reason. We add the controlled measurement these methods presuppose—a sole native-budget knob in a fully modular, ground-truth-scored pipeline—plus a localisation absent from that work: in our setting *categorical* grounding is budget-invariant, and the decisive lever is the *metric* grounding of the destination, with the orchestration loop largely downstream of it (§V, App. E).

*d) 6-DOF grasp prediction and semantic affordance.:* Contact-GraspNet [18] and ZeroGrasp [9] produce geometrically stable 6-DOF grasps, ranking candidates by stability alone—treating a knife handle and blade as equivalent and leaving semantic preference and occlusion ordering to a downstream module. We treat the predictor as a candidate generator and use a VLM to validate candidates against the natural-language target by inspecting per-grasp overlay renderings: the LLM-as-judge pattern from language tasks, applied to manipulation.

*e) Open-vocabulary segmentation in manipulation.:* SAM 3 [4] gives text-promptable instance segmentation. We use it to ground the agentic loop's chosen noun phrase into a binary mask for the grasp predictor; swapping it for an alternative is a one-function change.

*f) VLM-driven placement.:* Recent methods use a frozen VLM to emit a 2-D placement point or region, then lift to 3-D via depth back-projection, with visual-prompting techniques (Set-of-Mark [22], PIVOT [16]) providing the primitives. Our placement is lightweight by design: the VLM and SAM 3 decide *which* receptacle is the target, and the 6-DOF pose is grounded geometrically (single-view depth on open surfaces, depth-perceived tier geometry for shelves, a re-grounded mask centroid for cup-like containers) rather than learned or VLM-pointed. This keeps placement failures strictly geometric (Stage 2b/3) and lets the VLM own only the semantic decision.

*g) Modular vs. end-to-end manipulation.:* The modular/end-to-end trade-off is well documented [3, 11, 2]. Our position is about *attribution*: when reasoning lives in one frozen, swappable component, its contribution can be isolated by varying the model's reasoning budget while the rest of the pipeline is held constant. This gives semantic reasoning one stage to live in and a single knob to be evaluated against.

## III. The Reason–Predict–Plan Pipeline

### A. Architecture overview

Given a natural-language instruction $\ell$ and an initial RGB-D observation $o_0$ from two calibrated cameras — one wrist camera (`camera_0`) and a fixed front camera (`camera_1`) placed in front of the workspace and aimed at the arm — RPP produces a sequence of end-effector commands $a_t \in \mathbb{R}^7$

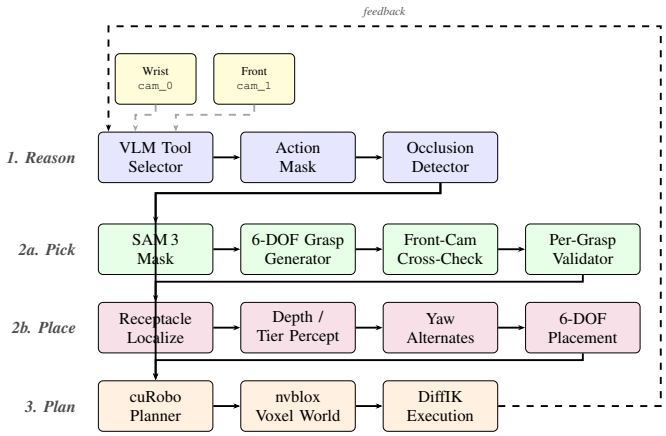


Fig. 1. **Reason–Predict–Plan (RPP).** Two calibrated cameras (one wrist + one fixed front camera) feed the VLM tool selector in Stage 1. The agentic loop branches the predict stage between a *pick* path (SAM 3 mask → 6-DOF grasp generator → front-camera cross-check → per-grasp overlay validator) and a *place* path (receptacle localization → depth-perceived tier geometry → yaw-alternate selection → 6-DOF placement pose). Accepted poses are routed to cuRobo with an nvblox voxel-fused collision world (Stage 3). The dashed loop-back is the per-step feedback that the agentic loop consumes before its next call.

(6-D pose plus binary gripper, distinct from the arm's seven joint DOFs, onto which a differential-IK term maps it). Each episode runs in two interleaved phases (Fig. 1):

**Phase 1: Deliberation** (per sub-task). A short geometric pre-pass back-projects depth from both cameras on the GPU and estimates table height, hover/lift clearances, and per-object PCA extents. The VLM — Gemini 3.5 Flash [7] here, though treated as a swappable frozen black box — is then queried for the next tool call in an agentic loop, capped at 15 calls per episode. On `pick_object`, the predict stage segments the target with SAM 3, generates 6-DOF grasp candidates, cross-checks the wrist-camera candidate against the front camera, and validates the pose via per-grasp VLM overlay inspection on both views. On `place_object` for a receptacle-like target (e.g. a shelf), the place stage perceives the shelf tiers from front-camera depth (or samples a point from single-view depth on open surfaces), derives a 6-DOF placement pose on the lowest reachable tier, and composes its orientation from the validated grasp. The motion planner is then called for the first accepted pose.

**Phase 2: Execution** (∼physics rate). The robot follows the planner's trajectory via the simulator's differential-IK action term. The execution loop has no *learned* components in the present work; learned closed-loop residual correction is a separable extension out of scope here. Its one closed-loop element is geometric, not learned: the deep-receptacle insertion path (§III-D) applies a forward-kinematics tip correction with no trained parameters.

The decomposition is standard; our contribution is to concentrate all of the pipeline's reasoning in one frozen component, so the contribution of chain-of-thought can be isolated by varying that one component's reasoning budget (§IV) while every other stage is held constant.

### *B. Stage 1: Reason — agentic tool loop*

The VLM is wrapped in a per-step agentic loop, capped at 15 tool calls (steps) per episode. At every step it receives: the task name, the user instruction, the noun phrase currently identified as the target, the gripper state (*empty* vs. *holding*), a running history of prior tool calls and their outcomes (success / failure / failure reason), and a *contextually masked* list of allowed tools. The VLM is required to respond in a strict ReAct-style [23] format:

```
<think>...</think>
<tool>ToolName</tool>
<args>{"k":"v"}</args>
```

A response that fails to parse is rejected and the model is re-queried with an explicit error message.

*a) Tool set.:* Nine tools cover the loop: `estimate_workspace`, `detect_occlusion`, `pick_object(target)`, `place_object(location)`, `stash_object(location)`, `place_using_human_demo`, `return_home`, `ask_user_for_clarification(question)`, and `task_complete`. Each tool either changes the world state (pick/place/stash/demo/home), emits a structured observation back into the loop (workspace, occlusion), or terminates the episode (complete). `place_using_human_demo` replays a pre-recorded demonstration trajectory as a placement fallback—the one scripted skill in the action space, exposed only as a recovery option and inert in our runs (it converts no failure into a success, App. D). `ask_user_for_clarification` is included for the long tail of ambiguous instructions ("the cup" when there are three).

*b) Action masking as the* Can *module.:* Following the rule-based skill filtering of Yamao et al. [21], we replace SayCan's per-skill RL value function with a deterministic feasibility mask. Before each VLM call, a hard filter exposes only the feasible tools given gripper state and prior history (e.g., `place_object` is masked when the hand is empty; `pick_object` when already holding), eliminating a class of trivially infeasible calls without per-skill training.

*c) Repeat-failure detection.:* Agentic loops commonly re-issue a just-failed call. Our loop fingerprints recent (tool, args, outcome) triples; on the third consecutive identical call the mask removes the tool for one turn and injects a "you have just tried this and it failed" annotation, converting an oscillation into a recovery decision.

### *C. Stage 2a: Predict — pick (segment, generate, cross-validate, validate)*

When the VLM invokes `pick_object(target)`, the predict stage runs four sub-stages.

**(2a.i) SAM 3 segmentation.** The noun phrase plus each camera's RGB is fed to SAM 3 [4], returning a binary target mask per view. If no mask is returned (e.g., a fully occluded object), the stage emits a structured "target not visible" failure;

the loop typically responds with `detect_occlusion` then `stash_object` on the occluder.

**(2a.ii) 6-DOF grasp generation.** The wrist camera's masked depth/RGB pair is passed to ZeroGrasp [9], returning a ranked list of scored 6-DOF candidates. These are re-ordered by Euclidean distance from the current end-effector to bias toward easy-to-plan grasps.

**(2a.iii) Front-camera cross-check.** The wrist proposal is cross-checked against the front camera: ZeroGrasp is re-run on the front view with that view's mask, and the resulting pose is projected into the wrist frame. The accepted pose minimises the Cartesian disagreement between views; if the front camera returns no usable pose, the wrist proposal is kept. This second viewpoint suppresses monocular-depth artefacts (transparent or reflective surfaces, mask-boundary noise) before any VLM call.

**(2a.iv) Per-grasp VLM overlay validation.** Each candidate is rendered as a coloured gripper overlay (centroid disc and two finger pads, a Set-of-Mark-style visual prompt [22, 16]) on *both* views, tiled into one image; the VLM is asked whether the pose is "a stable grasp on the *[target]*, free of collisions with the table or surrounding objects." Candidates are validated in rank order, and the first yes advances to the planner. If none is approved within the top $N{=}20$, the stage emits a "no acceptable grasp" failure, which the loop typically follows with `ask_user_for_clarification` or `detect_occlusion`.

**The role of the validator.** The validator is the only point in Stage 2a where the agent's semantic intent enters the geometric grasp pipeline. Without it, the pipeline takes the closest geometrically scored grasp and never asks whether that grasp picks the right object the right way; with it, the agent can refuse a grasp that would close on the wrong handle or whose overlay shows the fingers passing through the table.

### *D. Stage 2b: Predict — place*

The pick path returns a 6-DOF pose for an object already in the scene. The place path must invent a 6-DOF pose at a location that may lack a clear depth signal, candidate masks, and any off-the-shelf predictor. The VLM has already named the target receptacle through the agentic loop and SAM 3; we keep its role purely semantic and ground the 6-DOF pose geometrically. Invoked on `place_object`, this stage branches by target type: an open surface (2b.i), a tiered shelf (2b.ii), or a cup-like container such as a bowl or mug (2b.iv); orientation composition (2b.iii) applies across all three.

**(2b.i) Open-surface placement (depth-frustum sampling).** For an open tabletop target, candidate points are sampled from the front camera's back-projected depth within the named region, filtered to flat reachable free space, and lifted by an object-height clearance from the workspace pre-pass. This is the default path whenever the target is not a deep receptacle.

**(2b.ii) Shelf placement (depth-perceived tier geometry).** A shelf interior has weak, self-occluded depth from any external view, so rather than perceive a point *inside* it we perceive the shelf's horizontal *tiers* from front-camera depth, anchored on the receptacle region the agentic loop and SAM 3 have already named. The masked depth is back-projected to the robot base frame; horizontal tier surfaces appear as dense peaks in the resulting height histogram (nearby peaks merged so each physical board yields one surface), and each tier's front edge and lateral extent are measured from the points at that height. These *perceived* tiers are installed as the shelf's collision cuboids in the planner's world, so a sole percept drives both target selection and collision checking with no hand-specified dimensions. The object is placed top-down onto the lowest full-width reachable tier, a fixed margin behind *that tier's* perceived front edge; tiers with insufficient clearance for a (near-)vertical insertion are detected as planner failures and skipped (Limitation vi, §VI). The placement height accounts for the gripper length and held object's extent so the object clears the surface during insertion, then lowers onto it before the gripper opens and retracts vertically.

**(2b.iii) 6-DOF orientation composition.** With no off-the-shelf learned pose head, the placement orientation is composed geometrically from the current grasp orientation, preserving the validated grip. The planner selects a reachable yaw alternate $\{0°, \pm15°, \pm30°\}$ for the staging pose in front of the shelf, and the in-shelf insertion is (near-)vertical—a small forward tilt is admitted only when a purely top-down insertion is unreachable. A vertical gripper holds the object level (a steep tilt slips a rim grip) and makes the shelf-insertion depth equal the commanded end-effector $y$, so insertion depth is set against the perceived front edge from proprioception alone, with no object-pose feedback. We leave the gross approach axis vertical: re-orienting a held object mid-air is its own dexterity problem, out of scope here.

**(2b.iv) Deep-receptacle insertion (cup-like containers).** A narrow open container (the bowl in Kitchen-Tidying, the mug in Occluded-Assembly) is neither an open surface nor a tiered shelf: it presents one small mouth and weak interior depth. The receptacle is re-grounded with SAM 3, and the placement anchor is the robust *median* of the back-projected mask point cloud (a single-pixel back-projection biases onto the near rim). The insertion is diagonal, not vertical: a top-down wrist pose over a deep container sits at a differential-IK wrist singularity, where we observed the tracked tip diverge by $\sim$0.5 m. The held object is lowered along a moderate $\sim$35–40° "pen-in-cup" lean, its tip wedged toward the inner-near wall, and a closed-loop tip correction (a forward-kinematics tip estimate followed by a pure-translation nudge) settles the final pose. This path is a per-task extension exercised by Kitchen-Tidying and Occluded-Assembly, included for fidelity rather than claimed as a general primitive; the (near-)vertical claims above apply to the shelf path.

**Failure semantics.** If the planner finds no reachable insertion at any perceived tier, Stage 2b emits a *no-reachable-placement* failure the loop can react to (re-grounding the receptacle or escalating to clarification). Choosing the wrong receptacle or tier is, by contrast, a *reasoning* failure owned by Stage 1 — the distinction the reasoning-budget ablation is

designed to expose.

### E. Stage 3: Plan — collision-free motion

Each accepted pose $P^*$ from Stage 2a (grasp) or Stage 2b (placement) is sent to cuRobo [17], producing a collision-free joint-space trajectory $\tau = \{q_1, \ldots, q_T\}$ for the 7-DOF arm. cuRobo is initialised once at startup with the standard robot YAML and re-used throughout the episode; CUDA graphs are disabled for determinism across seeds. We use `plan_single` with `max_attempts=10` (graph mode) and `max_attempts=1` (direct mode, fast-fail).

**World representation.** The collision world is a live *nvblox voxel map* [14] (1 cm voxels, occupancy integrator) fused from both depth streams and attached to cuRobo via `BloxMap`; a 0.20 m hand-exclusion sphere keeps the robot's own body from registering as an obstacle. It is refreshed before every transport plan, and again after any robot motion from the optional wrist-camera hover step in Stage 2b, so the planner sees the *current* configuration.

**Why nvblox lives in Stage 3.** The voxel world changes only which trajectories are feasible, leaving grasp and placement pose selection untouched. This keeps responsibilities clear: a trajectory infeasible only because of unobserved background clutter is a *planning* failure owned by Stage 3, distinct from a wrong-approach decision owned by Stage 1's reasoning.

## IV. Implementation and Evaluation Setup

*a) Simulator and robot.:* Isaac Lab [15] on top of Isaac Sim, with a 7-DOF Franka-style arm and a parallel-jaw gripper. Each environment uses two calibrated cameras at $1280 \times 1024$: one wrist camera (`camera_0`) mounted on the hand and one fixed front camera (`camera_1`) about 2.5 m in front of the base at 1.0 m height, aimed at the arm and workspace. Differential inverse kinematics is used as the action term. We currently run with `num_envs=1` per process; GPU-parallel batched evaluation is a planned engineering extension.

*b) Environments.:* Four manipulation environments are registered:

- **Shelving** (primary): a tabletop bowl must be placed onto a shelf tier. We report the *de-occluded* variant—a clear workspace, isolating Stage 2a (grasping the bowl by its rim for a top-down lift) and Stage 2b (depth-perceived shelf placement, since the shelf's off-table tiers must be perceived rather than sampled from on-surface depth). An occluded variant (a mug in front of the bowl, requiring a Stage 1 stash) is the registered design target but lacks a verified run. To make top-down insertion feasible for a fixed-base 7-DOF arm, the shelf's vertical scale was raised from default ($0.8 \rightarrow 1.4$), giving $\sim$0.42 m compartments (taller than the arm's wrist–forearm stack); at default spacing every top-down plan was IK-infeasible (§VI, Limitation vi).
- **Kitchen-Tidying**: a countertop holds four mixed objects (bowl, mug, marker, banana) and the instruction is to place the banana into the bowl. The distractors make this a cluttered grasp-and-place scene; the bowl is re-grounded with SAM 3 at placement time (§III-D). Stresses instruction following among distractors and Stage 2a grasp selection in clutter.
- **Occluded-Assembly**: a marker peg must be inserted into a cup-like receptacle. The environment registers a bowl (intended receptacle) behind a mug (intended occluder); the verified run inserts the peg into the mug without clearing the occluder. The stress is the insertion—a thin peg into a narrow mouth along a feasible approach (Stage 3 and the deep-receptacle path of §III-D).
- **Delicate-Unstacking**: a banana resting on a bowl must be removed without toppling the bowl. Stresses Stage 2a grasp-role selection and the Stage 3 retraction trajectory.

*c) VLM backend.:* A frontier multimodal VLM [7] is queried via HTTP. The same endpoint serves the per-step tool call (including the workspace target/receptacle identification) and the per-grasp validation; only the prompt template differs. Per-call latency in our setup is $\sim$1.5–1.8 s, dominating wall-clock cost in deliberation.

*d) The chain-of-thought ablation.:* We ablate reasoning at its source. Gemini's native `thinking_level` [7] governs how many internal reasoning tokens the model spends before answering; our *CoT* condition sets it to *high* (the default), our *no-CoT* condition to *minimal*. Nothing else changes—the agentic loop, action mask, `detect_occlusion`, the per-grasp validator, the placement geometry, and the cuRobo/nvblox planner are identical across conditions—so the only variable is how deeply the VLM reasons. This is a stronger isolation than disabling pipeline stages, which conflates "less reasoning" with "fewer capabilities": the model is asked identical questions with the same tools, and we measure only whether more reasoning helps. (The native budget is the control; the `<think>` scaffold our prompts request is unchanged, see Limitation iii.) Placement is deterministic and held constant. We run $n$=30 episodes per task per condition (extended to $n$=50 on Shelving to resolve the box-vs-loop split, App. E) with object poses *matched across conditions*, so outcomes are paired by seed (Table I); within-cell variation comes from pose randomisation and VLM sampling (temperature=0.6, equal in both). The 15-call cap (§III-B) is shared and never binding, so reasoning depth is the only systematic difference between cells.

*e) Measuring success.:* Tool returns are self-reported: `place_object` returns success after transport, release, and retract with *no* landing check (the `_log_*_groundtruth` hooks record landing geometry but do not gate any result), so a run's `summary.json` reflects the agent's belief. Every number in Table I is instead scored offline from logged object poses by a per-task ground-truth rule—object on the correct shelf tier; inside the named receptacle (cup-like tasks: horizontal distance to its centroid $< 7$ cm and $z \geq 4.5$ cm above its base); or on the table and separated—spot-checked against rendered video. This matters: at $n$=30 self-report still overcounts on the deep-insertion tasks. On Occluded-Assembly (high), 2 "successes" were markers resting beside, not in, the mug; Delicate-Unstacking contributes 2 (high) and 4 (minimal) such phantoms (the $\mathrm{FP}_{\mathrm{CoT}}$/$\mathrm{FP}_{\mathrm{min}}$ columns). These

false positives sit where the pilot "reasoning hurts" effect lived—part of why it dissolved.

*f) Logging.:* Each episode emits a JSONL trace (`temp/debug_traces/...`) of per-tool calls, per-step poses, the outcome, and—key for this study—the full chain-of-thought at every hook, plus the placement-stage debug images (front/wrist views, the SAM 3 receptacle mask, the perceived-tier target). The offline scoring and the high-vs-minimal comparison read from these.

## V. Results and Observations

We report the high-vs-minimal reasoning-budget sweep across all four environments ($n$=30 episodes per task per condition, $n$=50 on Shelving; Table I), then three qualitative observations on the design choices that concentrate reasoning in one stage (Observations 1–3) and a synthesis of *where* that reasoning proved to matter (Observation 4).

*a) The headline result: where reasoning changes the outcome.:* At $n$=30 per cell ($n$=50 on Shelving), the tasks separate into two regimes. On **Shelving**, chain-of-thought is decisive: the high condition succeeds 26/50 while the minimal condition succeeds 1/50 (Holm-corrected Fisher $p=2.6\times10^{-8}$, Cohen's $h=+1.33$). The failure is upstream of any manipulation: both budgets name the target and recognise the shelf, but the minimal budget emits an *off-frame* destination box and almost never completes (1/50). A good-box counterfactual (App. E) pins the cause there—supplying a correct box recovers most of the gap—so the decisive factor is the *metric* destination grounding, not categorical perception (invariant) and not primarily the loop, whose thrash is largely downstream. On the other three tasks we see no detectable difference at $n$=30 (28/30 vs. 27/30 on Kitchen-Tidying, 22/30 vs. 25/30 on Occluded-Assembly, 28/30 vs. 26/30 on Delicate-Unstacking; every $p_{\text{Holm}}=1$, every CI on the difference spanning zero). Once the target is named the tool sequence is essentially forced and there is little for deliberation to improve; we cannot rule out a small effect that $n$=30 is underpowered to detect, but none is visible.

*b) A "reasoning hurts" effect that did not survive.:* Occluded-Assembly is the cautionary case, and shows why ground-truth scoring at adequate $n$ matters. In a pilot at $n$=10 a full budget looked sharply counterproductive ($6/10$ vs. $10/10$, and $8/10$ by the agent's own self-report): with a full budget the model sometimes "notices" the task-irrelevant bowl behind the mug and manufactures occlusion-handling detours (`detect_occlusion`, `return_home`) that the minimal condition never takes. Firming the contrast to $n$=30 shrank the gap from $-40$ to $-10$ pts, and the apparent effect did not survive (22/30 vs. 25/30, CI [-30,+10] spanning zero; $p_{\text{Holm}}=1$; the pilot's own two-sided contrast was already not significant under Holm correction). Both conditions moved as $n$ grew—high from 60% to 73%, minimal from 100% to 83%—so the pilot's perfect $10/10$ was itself optimistic, not only the high condition's $6/10$. The over-elaboration is real and still visible in the traces, but at $n$=30 it costs deliberation latency far more reliably than it costs success. We keep this cell in the paper rather than dropping it: a small-$n$, self-report-scored ablation would have published a vivid "chain-of-thought hurts manipulation" headline that proper power and ground-truth scoring do not support.

*c) Localizing the effect: the metric lookup, with the loop downstream.: Where* inside the pipeline does the Shelving gap live? A decision-locus probe asks the minimal-budget VLM the grounding question directly ($K$=8): categorical naming, pixel-pointing, and the high-vs-minimal pointing divergence ($\leq 0.013$) are all flat on every task—*categorical* grounding is budget-invariant. Yet mining all $n$=50 traces (Fig. 2), the minimal budget on Shelving emits an *off-frame* destination box and thrashes to the 15-step cap (98% of episodes), while the high budget finishes in $\sim$8 clean steps; the three inert tasks are flat. Is the box or the loop the cause? A *good-box counterfactual* settles it (App. E): supplying the high-budget box to an otherwise-minimal pipeline recovers most of the gap and collapses the thrash, so the dominant driver is the *metric* destination grounding, with the loop thrash largely a downstream symptom. *Categorical* grounding stays invariant, and every win is a clean first placement (App. D).

*d) Reading the result.:* Before returning to *why* the effect concentrates where it does (Observation 4), we record three design choices—action masking, per-grasp validation, and the voxel collision world—that keep any reasoning effect attributable to Stage 1 rather than leaking from the surrounding machinery (Observations 1–3).

*e) Observation 1: VLM tool selection without action masking is brittle.:* Early debug episodes without the action-mask filter showed the VLM repeatedly emitting `place_object` on an empty hand, or `pick_object(``the bowl'')` after `detect_occlusion` reported the bowl hidden—each wasting a turn and a VLM round-trip on a tool the pipeline had to refuse. The hard mask (Yamao et al. [21]'s move in dropping the RL *Can* term) removed this class of trivially infeasible calls and converged the loop in fewer steps per episode.

*f) Observation 2: Per-grasp VLM validation is most useful in cluttered scenes.:* On uncluttered scenes the closest-EE ZeroGrasp candidate is usually fine, so the $\sim$1.7 s per-candidate validation is pure overhead. On multi-object scenes (Kitchen-Tidying, Occluded-Assembly) the closest candidate is frequently a grasp on a neighbour, or on the target from a side that later interferes with the receptacle; the validator rejects these and the loop falls back to the next candidate or, after $N$ rejections, escalates to `detect_occlusion`. This realises the architecture's intent: the same Stage 1 component that decides *what to do next* also judges whether a Stage 2 candidate is acceptable.

*g) Observation 3: The voxel world helps Stage 3 in specifically the environments where the cuboid world under-approximates clutter.:* On Kitchen-Tidying, the cuboid world (table + bowl) captures the relevant collisions, so adding nvblox is roughly a wash. On Occluded-Assembly, the cuboid world misses the receptacle's interior geometry and cuRobo

TABLE I

**GROUND-TRUTH TASK SUCCESS, COT (THINKING_LEVEL=HIGH) VS. NO-COT (MINIMAL),** AT $n$=30 PER CELL ($n$=50 ON SHELVING). EACH CELL IS SUCCESSES / EPISODES, SCORED OFFLINE FROM LOGGED OBJECT POSES (NOT THE AGENT'S SELF-REPORT). $\Delta$ IS THE COT−NO-COT SUCCESS-RATE DIFFERENCE IN PERCENTAGE POINTS WITH ITS 95% PAIRED-BOOTSTRAP CONFIDENCE INTERVAL (PAIRED BY SEED ACROSS CONDITIONS). $p_{\text{HOLM}}$ IS THE HOLM-CORRECTED FISHER EXACT TWO-SIDED $p$ ACROSS THE FOUR CONTRASTS (AN EXACT MCNEMAR TEST GIVES THE SAME SIGNIFICANCE PATTERN). $h$ IS COHEN'S $h$. $\text{FP}_{\text{CoT}}/\text{FP}_{\text{MIN}}$ COUNT SELF-REPORTED SUCCESSES THAT GROUND TRUTH REFUTED.

| Task | CoT | no-CoT | $\Delta$ [95% CI] | $p_{\text{Holm}}$ | $h$ | $\text{FP}_{\text{CoT}}$ | $\text{FP}_{\text{min}}$ |
|---|---|---|---|---|---|---|---|
| Shelving | 26/50 | 1/50 | $+50$ $[+36, +64]$ | $2.6 \times 10^{-8}$ | $+1.33$ | 0 | 0 |
| Kitchen-Tidying | 28/30 | 27/30 | $+3$ $[-7, +13]$ | 1.000 | $+0.12$ | 0 | 0 |
| Occluded-Assembly | 22/30 | 25/30 | $-10$ $[-30, +10]$ | 1.000 | $-0.24$ | 2 | 0 |
| Delicate-Unstacking | 28/30 | 26/30 | $+7$ $[-10, +23]$ | 1.000 | $+0.23$ | 2 | 4 |

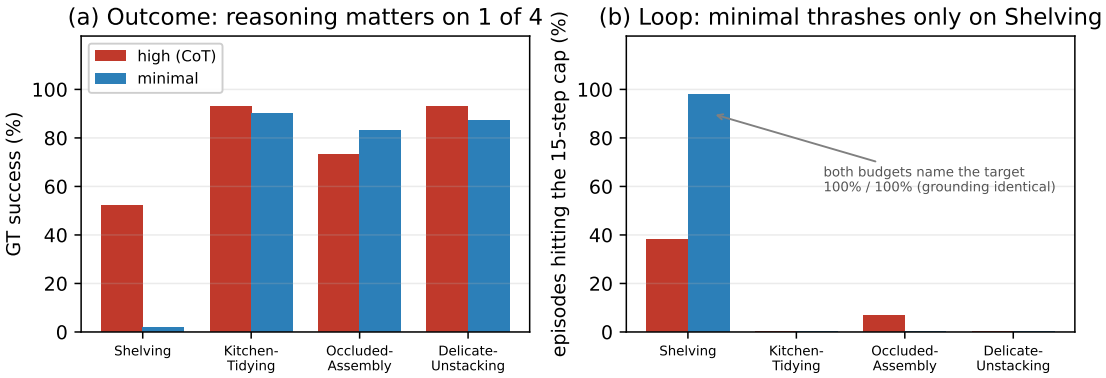


Fig. 2. **The reasoning effect: decisive on one task, and not categorical perception** ($n$=30/cell, $n$=50 on Shelving, ground-truth scored, from the trace mining of §V). **(a)** Outcome: the budget is decisive only on Shelving (52% vs. 2%); the other three tasks are flat. **(b)** The minimal budget hits the 15-step cap on 98% of Shelving episodes (vs. 38% at high) and never on the inert tasks—yet both conditions *name the target* with near-identical consistency (87–100%, annotation), so the gap is not categorical perception. A good-box counterfactual (App. E) traces it to the *metric* destination box, the cap blow-up being largely its downstream symptom.

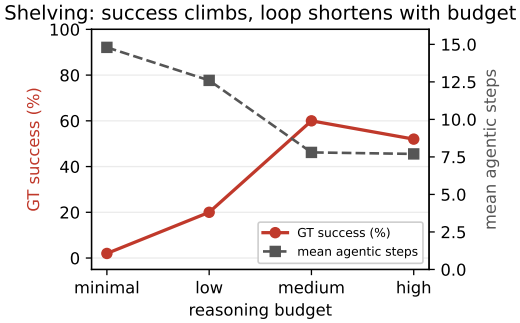


Fig. 3. **Shelving reasoning-budget dose-response.** Success climbs and the agentic loop shortens with the budget ($n$=50 at the endpoints, $n$=5 interior; the noisy $n$=5 medium point aside); a no-scaffold pilot ($n$=5) is consistent with the native budget (Limitation iii). The three inert tasks are flat across the budget and are omitted.

clips the wall during insertion; nvblox eliminates these wall-clipping collisions at a small per-call depth-fusion cost, leaving grasp selection and reasoning unchanged — the collision world affects only which trajectories are feasible, not which poses are chosen.

*h) Observation 4: Reasoning depth shows up where decisions are ambiguous, and nowhere else.:* The sweep confirms, and sharpens, a simple expectation: reasoning budget matters only where a task poses a genuine grounding decision. The one task that must precisely ground *where* to place into a partially observed shelf is the one task where reasoning is decisive (52% vs. 2%). The three tasks whose tool sequence is forced once the target is named show no significant effect (Kitchen-Tidying, Delicate-Unstacking, and, once firmed to $n$=30, Occluded-Assembly). The effect is not *categorical* perception even where it is decisive: the probe shows naming and shelf-recognition are budget-invariant. A good-box counterfactual (App. E) then localises the gain causally: supplying the high-budget destination box to an otherwise-minimal pipeline recovers most of the gap ($1/50 \rightarrow 22/50$, $\sim$84%), so the dominant driver is the *metric* destination grounding—the one signal the shelf placement depends on, where the other tasks re-ground or tolerate it—with the loop thrash largely its downstream symptom and the residual orchestration share small and not significant ($p$=0.55). This sharpens the opening vignette: of its four failure modes, the reasoning-sensitive one is grounding the *destination*—where on the shelf the bowl goes. Object selection is budget-invariant, and the grasp and trajectory are geometric. The design corollary, which we develop in §VII, is that a constant, global reasoning budget is the wrong choice: deliberation should be spent per-decision, at the hooks that pose a judgement.

*i) Scope of the claim.:* The numbers are from one VLM (Gemini 3.5 Flash) at $n$=30 on one simulator—a large effect on one task, its absence on three, not cross-model generality. Minimal reasoning is not generally adequate (it fails Shelving, 1/50), but, having watched our pilot "reasoning hurts" effect dissolve, we claim no general harm: at $n$=30 no task shows a significant negative effect. We also do not fully separate the native budget from the prompt-level `<think>` scaffold (held on in both); a no-scaffold pilot ($n$=5) still shows the effect (3/5 high vs. 0/5 minimal), consistent with it tracking the native budget (Fig. 3, Limitation iii).

## VI. Discussion

*a) Limitations.:* (i) The per-grasp validator can take up to ∼40 s per pick (up to 20 sequential validations at ∼1.7 s); other tool calls take ∼1.5–2 s. This rules out dynamic environments and reactive avoidance; closed-loop replanning is a natural extension. (ii) The VLM is a frozen black box; we have not studied fine-tuning it for validation, or how other models' thinking-level controls compare. (iii) The ablation varies native thinking depth but does not remove the `<think>` scaffold the prompts request; disentangling the two is future work, as is learned closed-loop control. (iv) Real-robot validation is future work: the motion stack ports to a Franka, but the sim-only stereo-depth pre-pass would swap for a real RGB-D sensor. (v) The sweep is $n$=30 per cell with paired-bootstrap CIs and Holm-corrected tests, but on one VLM and one simulator—it establishes the per-task effect, not its generality. (vi) Top-down shelf insertion needs each tier taller than the arm's wrist–forearm stack; too-shallow tiers admit no collision-free vertical insertion and are skipped as planner failures (we raised the shelf asset's vertical scale to meet this, §IV). Lifting this (in-hand re-grasping, more wrist clearance) is future work.

*b) Reproducibility.:* The pipeline is built from off-the-shelf components: a multimodal VLM, the cuRobo motion planner with nvblox collision avoidance, and Isaac Lab environments around a Franka-style 7-DOF arm. The VLM is accessed through a documented HTTP interface and can be swapped to an open-weights checkpoint with one configuration change. Section III specifies the per-stage prompts, tool signatures, and parameter choices in enough detail for an independent reimplementation.

## VII. Future Work

The result points to a concrete research program rather than a lone follow-up. We highlight the three directions that follow most directly from it.

*a) Two causal tests the trace analysis scopes but cannot settle.:* Appendices D and E scope two causal tests; we ran the first, and one remains. The *good-box counterfactual* (App. E) attributes most of the Shelving gain to the metric destination grounding, leaving only a small, non-significant residual to full-high. The open complement is a *recovery ablation*: giving the minimal condition a stronger scripted retry would test whether even that small residue is brittle recovery or a genuine first-attempt benefit.

*b) An adaptive, per-decision reasoning controller.:* Observation 4's corollary—that a constant, global reasoning budget is the wrong design—points directly at a controller that allocates budget per decision: a high budget on the multi-step decisions Observation 4 flags as reasoning-sensitive and a minimal budget on forced or purely geometric steps, so the latency cost of deliberation is paid only where it changes the outcome. The supervision for such a policy already exists: the same traces that drive the high-vs-minimal comparison (§IV) record, per hook, whether the high-budget condition changed its decision relative to minimal. Concurrent work already *learns* such allocation (RARRL [13]; Act-Think-or-Abstain [10]); what our measurement adds is the design rule it needs. Our decision-locus probe shows single-shot grounding is budget-invariant, so it is the *wrong* gating signal; a controller should instead gate on loop-level uncertainty—self-consistency of the next tool call, or a just-failed step—escalating in the orchestration loop rather than at the perceptual lookup.

*c) Generality across models, budgets, and robots.:* Our ablation is binary (high vs. minimal) on one model in simulation; three external generalisations would sharpen the claim. First, the full reasoning-budget→success *curve* per task type, rather than two endpoints, would map the shape between them. Second, a cross-model study would test whether the decisive/inert structure is model-general: most open-weights VLMs lack a *native* budget knob like `thinking_level`, so it must vary chain-of-thought at the prompt level, a different mechanism (Limitation iii). Third, real-robot replication on a physical Franka (Limitation iv), with attribution against end-to-end VLAs that entangle reasoning in the weights, would test whether keeping reasoning in one swappable component is what makes its contribution localisable and adjustable.

## VIII. Conclusion

We presented **Reason–Predict–Plan**, a modular pipeline in which a sole frozen VLM supplies all the reasoning (tool selection, occlusion ordering, per-grasp validation) while perception, grasping, and planning are off-the-shelf. Concentrating reasoning in one component buys a clear handle: varying the native reasoning budget while holding everything else constant measures how much competence is chain-of-thought rather than machinery. At $n$=30 per cell ($n$=50 on Shelving), ground-truth-scored, the answer is task-dependent but one-directional. Reasoning pays off only where a task poses a genuine multi-step decision—removing the budget drops shelf-placement success from 52% to 2% (Cohen's $h$=+1.33)—and a good-box counterfactual (App. E) places that effect in the *metric* grounding of the destination, not object selection or geometry. Once the sequence is forced, the same deliberation is inert. A modular design is what lets us see which regime a task is in, and spend deliberation only where it pays.

## References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, et al.

Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025. arXiv:2410.24164.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, et al. SAM 3: Segment anything with concepts, 2025. arXiv:2511.16719.

[5] William Chen, Suneel Belkhale, Suvir Mirchandani, Oier Mees, Danny Driess, Karl Pertsch, and Sergey Levine. Training strategies for efficient embodied reasoning. In *Conference on Robot Learning (CoRL)*, 2025. arXiv:2505.08243.

[6] Gemini Robotics Team, Google DeepMind. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer, 2025. arXiv:2510.03342.

[7] Gemini Team, Google DeepMind. Gemini 3.5 Flash model card. https://deepmind.google/models/model-cards/gemini-3-5-flash/, 2026. Model Card, Google DeepMind.

[8] 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.

[9] Shun Iwase, Muhammad Zubair Irshad, Katherine Liu, Vitor Guizilini, Robert Lee, Takuya Ikeda, Ayako Amma, Koichi Nishiwaki, Kris Kitani, Rares Ambrus, and Sergey Zakharov. ZeroGrasp: Zero-shot shape reconstruction enabled robotic grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[10] Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. Act, think or abstain: Complexity-aware adaptive inference for vision-language-action models, 2026. arXiv:2603.05147.

[11] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

[12] Jacky Liang, Wenlong Huang, Fei Xia, et al. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[13] Jun Liu, Pu Zhao, Zhenglun Kong, Xuan Shen, et al. When should a robot think? resource-aware reasoning via reinforcement learning for embodied robotic decision-making, 2026. arXiv:2603.16673.

[14] Alexander Millane, Helen Oleynikova, Emilie Wirbel, et al. nvblox: GPU-accelerated incremental signed distance field mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[15] Mayank Mittal, Calvin Yu, Qinxi Yu, et al. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

[16] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xu, Danny Lu, Andy Zeng, Tsang-Wei Edward Lee, Eric Jang, Brian Zhang, et al. PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.07872.

[17] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. cuRobo: Parallelized collision-free robot motion generation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[18] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[19] Tuan Duong Trinh, Naveed Akhtar, and Basim Azam. Altered thoughts, altered actions: Probing chain-of-thought vulnerabilities in VLA robotic manipulation, 2026. arXiv:2603.12717.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-Thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Kosei Yamao, Daiju Kanaoka, Kosei Isomoto, Akinobu Mizutani, Yuichiro Tanaka, and Hakaru Tamukoh. Development of a SayCan-based task planning system capable of handling abstract nouns. In *Proceedings of the International Conference on Artificial Life and Robotics (ICAROB)*, pages 430–433, 2024.

[22] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V, 2023. arXiv:2310.11441.

[23] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.03629.

[24] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning (CoRL)*, 2024. arXiv:2407.08693.

[25] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu,

Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, 2025. arXiv:2503.22020.

## APPENDIX A
## TOOL-CALL SCHEMA

The nine tools exposed to the VLM in Stage 1 are listed below with a one-line description of each. Argument schemas are validated by a strict JSON parser before the tool is dispatched; an unparseable response causes a re-query with the parse error included in the prompt.

- `estimate_workspace()` — runs the GPU stereo pre-pass; returns table height, hover and lift clearances, and per-object PCA extents.
- `detect_occlusion(target)` — returns the noun phrase of the occluding object if any, otherwise `null`.
- `pick_object(target)` — the predict + plan pipeline; returns success, failure phase, and failure reason.
- `place_object(location)` — plans and executes a place trajectory. Receives one of a small named-locations vocabulary or a free-text spatial phrase. When the phrase resolves to a tiered shelf, the tiers are perceived from the front-camera depth and the placement pose is the lowest reachable tier, executed by a (near-)vertical top-down insertion (§III-D), with orientation composed from the validated grasp plus yaw alternates. When it resolves to a cup-like container (bowl, mug), the receptacle is re-grounded with SAM 3 and the held object is lowered through the mouth along a moderate diagonal lean with a closed-loop tip correction (§III-D). Otherwise a depth-frustum sampler over the front-camera depth handles open surfaces.
- `stash_object(location)` — like place, but to a temporary out-of-the-way location chosen by the agent.
- `place_using_human_demo()` — a recovery fallback that places the held object by replaying a pre-recorded demonstration trajectory for the task. It is the one scripted skill in the action space, exposed only as a recovery option; in our evaluation it is inert, converting no failure into a success and reaching none of the successes scored (every win is a clean first attempt, App. D).
- `return_home()` — safe joint configuration with the gripper open.
- `ask_user_for_clarification(question)` — in evaluation, the simulator returns a deterministic "ground-truth" answer; in deployment this would surface to a human.
- `task_complete()` — terminates the episode. Tool returns are *self-reported*: for non-shelf placement, `place_object` returns success after transport, release, and retract without an automatic landing check, so `task_complete` reflects the agent's belief rather than a simulator-scored outcome. Ground-truth success in this paper was verified offline from logged object poses and rendered video (the `_log_*_groundtruth` hooks record landing geometry but do *not* gate the tool result); this offline check caught false positives such as an object dropped beside, not in, its receptacle.

## APPENDIX B
## PER-GRASP OVERLAY VALIDATOR: PROMPT SKETCH

The per-grasp validator is queried per candidate, in the order returned by ZeroGrasp (re-ordered by EE distance), until the first candidate is approved or the top $N{=}20$ are exhausted. The prompt template is:

> You are evaluating a candidate grasp on the object *[target]*. The tiled image shows both views (the wrist camera and the fixed front camera) with the candidate gripper rendered as a coloured centroid disc and two finger pads. Answer YES if (i) the fingers enclose the target without intersecting the table or another object, (ii) the wrist is roughly normal to the surface that supports the target, and (iii) the candidate is consistent with the natural-language instruction. Otherwise answer NO. Respond with a single token.

## APPENDIX C
## PER-COMPONENT LATENCY NOTES

Approximate per-call latencies in our setup (single-GPU workstation): VLM tool-select query $\sim$1.8 s; SAM 3 segmentation $\sim$0.12 s; ZeroGrasp inference 0.4–1.2 s, with the front-camera cross-check adding one further ZeroGrasp call (0.4–1.2 s) per pick; per-grasp VLM validation on the tiled two-view overlay $\sim$1.7 s $\times$ up to 20 candidates; the place stage is geometric (depth-perceived tier geometry or depth-frustum sampling) and adds no VLM call beyond the receptacle identification already issued by the agentic loop. cuRobo `plan_single` is 0.05–0.5 s. Execution runs at the simulator's physics rate; differential-IK update is $\sim$8 ms per step. Deliberation dominates wall-clock cost; the architecture is built for off-line and quasi-static manipulation, with reactive control left to future work.

## APPENDIX D
## TRACE-LEVEL TESTS OF THE LOOKUP–LOOP HYPOTHESIS

The body localises the Shelving effect to the *metric* destination grounding, with the agentic loop largely downstream (§V). This appendix substantiates that localisation directly from the sweep traces ($n{=}30$ per condition, $n{=}50$ on Shelving) with four offline analyses (`experiments/analyze_traces_hypothesis.py`; no simulator, services, or VLM calls—it re-parses the on-disk `trace.jsonl` files), presented in order: **H1**, categorical grounding is budget-invariant; **H2**, the surfaced high-budget reasoning is spent on the orchestration loop, not target naming; **H3**, the Shelving win is failure-recovery rather than a better first attempt; and **H4**, the loop metrics diverge only on the decisive task. We report what the traces can and cannot decide.

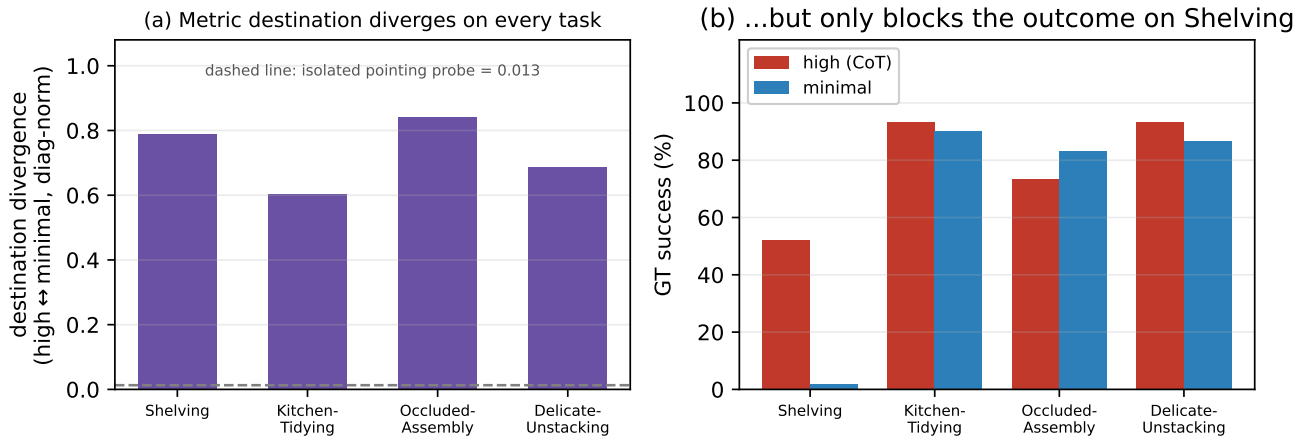


Fig. 4. **H1: categorical grounding is invariant; the metric destination box diverges on every task and, on Shelving, is causal (App. E). (a)** The high-vs-minimal divergence of the emitted destination box (diagonal-normalised centroid distance) is 0.60–0.84 on *all* four tasks—46–65× the 0.013 of the isolated single-shot pointing probe (§V)—while the categorical grounding (pick-target head-noun and shelf-recognition) agrees in 97–100% of episodes. **(b)** The same divergence is present where reasoning is inert: the three inert tasks succeed 83–90% under the minimal budget despite equally displaced destination boxes, so the divergence does not by itself predict failure.

*a) H1: categorical grounding is invariant; the metric destination is not, and on Shelving its divergence is causal (Fig. 4, App. E).:* Across all four tasks the grounded pick target agrees between conditions (modal head-noun identical, per-episode consistency 97–100%) and shelf-recognition (`is_shelf_placement`) is identical (gap $\leq 0.04$): the *categorical* grounding is budget-free, corroborating the isolated decision-locus probe (§V) inside the full pipeline. The one grounding signal that moves is the *metric* destination box the VLM emits at the place step: under the minimal budget its centroid leaves the $1280 \times 1024$ front-camera frame on every task (in-frame fraction 0–30%, vs 88–100% at high budget), a high-vs-minimal divergence of 0.60–0.84 diagonal-normalised. This is the trace-level form of the "off-frame destination box" of §V, and it is large—more than an order of magnitude above the isolated pointing probe. Critically it is *not* confined to the decisive task. The inert tasks emit equally displaced boxes yet succeed 83–90% under the minimal budget (Fig. 4b), because their cup-like or open-surface placements are re-grounded with SAM 3 or are geometrically forgiving, whereas the shelf placement anchors a depth percept on the named region and cannot absorb a bad box. A grounding signal that diverges identically where reasoning is inert does not by itself explain the decisive-task failure—it tracks the budget, not the outcome—so its *magnitude* alone does not flag the decisive task. Whether the box *causes* the Shelving failure is a separate question the traces cannot answer; the good-box counterfactual (App. E) supplies the missing counterfactual—a minimal run with a correct box—and resolves it: the box is the decisive driver there.

*b) H2: the surfaced reasoning is spent on the loop, not naming (Fig. 5).:* Every VLM hook logs its `<think>` scaffold and its length. *Caveat, stated plainly:* only that scaffold is recorded; Gemini's native thinking budget—the actual control variable—is not in the trace, so this analysis measures *surfaced* reasoning, an under-count of the high-budget deliberation, not the deliberation itself. With that caveat, two things are visible. First, the minimal condition surfaces almost no scaffold (median 0 characters at every hook), while the high condition surfaces reasoning whose *volume* concentrates in the orchestration loop: 60–88% of surfaced characters are at the per-step tool-selection hook, the remainder at grounding (substantial only on Shelving and Occluded-Assembly, the two tasks with a scene to parse; $\approx 0$ on the trivially-grounded inert tasks) and a little at occlusion-checking (Fig. 5a). Second, a coarse keyword tally of the orchestration reasoning shows its *content* is about placement/destination and recovery/sequencing, with target naming a small minority (Fig. 5b)—consistent with naming being settled cheaply and deliberation going to the multi-step decision.

*c) H3: recovery never converts a failure; every win is a first attempt (Fig. 6).:* We classify each *successful* episode as first-attempt (exactly one `place_object`, no `return_home` or `place_using_human_demo`, and no planner failure before the landing) or recovery-mediated. On Shelving, 25 of 26 high-budget successes are first-attempt (the lone exception carries $\approx 0$ recovery calls); recovery tools fire only in *failing* episodes (7.2 calls/episode at high, 9.0 at minimal), and the minimal condition recovers none of its 49 failures (Fig. 6a). The pattern holds on all four tasks: successful episodes carry $\approx 0$ recovery calls and essentially every success is first-attempt (Fig. 6b). This scopes the recovery-confound question *offline*: in this pipeline the decisive gap is a better *first* placement, not better recovery, and recovery as implemented is uniformly unproductive. What the traces cannot decide is the counterfactual the body leaves to future work—whether a *fundamentally stronger* scripted retry, given to the minimal condition, could close the gap; they show only that the *existing* recovery path does not, in either condition.

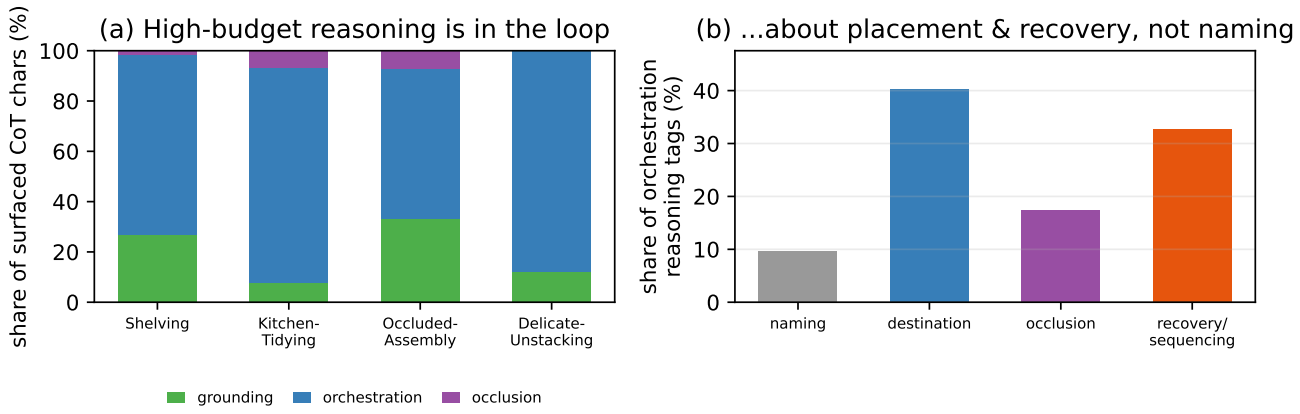


Fig. 5. **H2: the high-budget reasoning that is surfaced goes to the loop, not naming. (a)** Of the surfaced `<think>` characters in the high condition, 60–88% are at the per-step tool-selection (orchestration) hook on every task; grounding reasoning is substantial only on the scene-heavy Shelving/Occluded tasks and $\approx 0$ on the trivially-grounded inert tasks; the minimal condition surfaces $\approx 0$ characters at every hook (not shown). **(b)** A coarse keyword tally of the orchestration reasoning (pooled, high condition) is dominated by placement/destination and recovery/sequencing, with target naming a small minority.

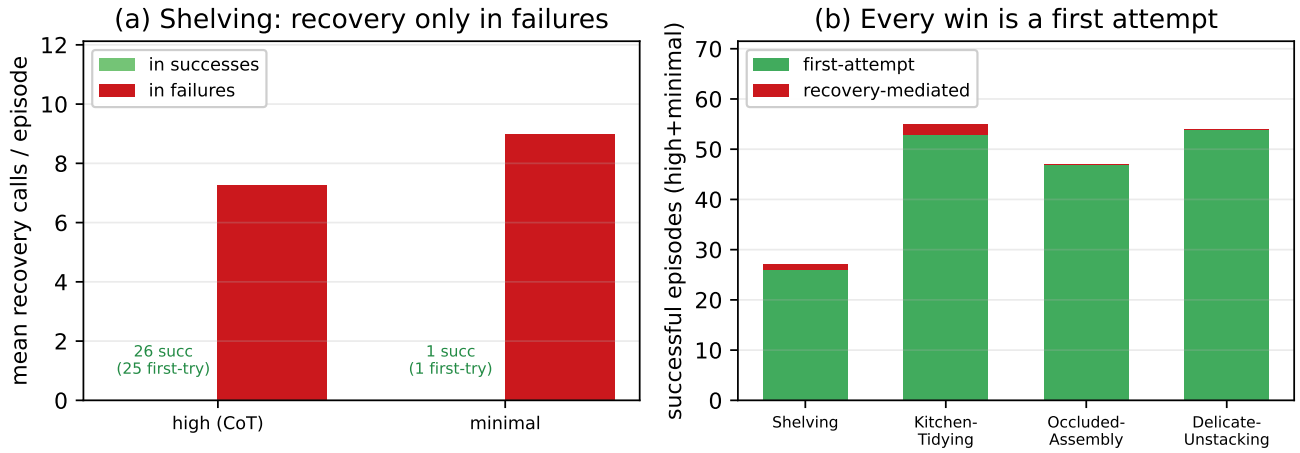


Fig. 6. **H3: recovery never converts a failure into a success; every win is a first attempt. (a)** On Shelving, recovery tools (`return_home`, `place_using_human_demo`) fire only in *failing* episodes (7.2/episode at high, 9.0 at minimal) and $\approx 0$ in successes; the minimal condition recovers none of its 49 failures. **(b)** Across all four tasks essentially every successful episode (pooled high+minimal) is a clean first attempt; recovery-mediated successes are $\approx 0$.

*d) H4: the loop metrics diverge only on Shelving (Fig. 7).:* All four loop signals (per-episode means across all episodes) separate the conditions sharply on Shelving—7.7 vs 14.8 agentic steps, 38% vs 98% step-cap rate, 2.5 vs 4.9 place-retries, 3.5 vs 8.8 recovery calls—and are flat on the three inert tasks (Occluded-Assembly shows only the mild high-budget over-elaboration discussed in §V, not a success gap). Joined with H1, H3, and the good-box counterfactual (App. E): *categorical* grounding is invariant; the *metric* destination box is the dominant, causally-tested driver of the Shelving gain; the loop divergence here is largely its downstream symptom (a small residual orchestration share remains, not significant at $n$=50); and every win is a clean first placement. What these cannot settle is cross-model generality, the native-vs-scaffold decomposition (Limitation iii), or whether the small residual loop share is real (the recovery ablation, §VII).

## Appendix E
## The Good-Box Counterfactual (A1b)

The trace tests (App. D) localise the Shelving effect correlationally but cannot say whether the divergent destination box *causes* the failure or merely accompanies it. We answer by intervention. A one-flag change (`--good-box`) forces, under the minimal budget, *only* the scene/destination grounding hook to a high `thinking_level`—so the destination box is the one a high budget emits—while the agentic loop and every other hook stay minimal; this isolates a high-budget *lookup* with a minimal *loop*. We re-run Shelving at $n$=50 (seeds matched to the main cells) and score by ground truth.

The result is decisive (Fig. 8). A correct destination box alone lifts success from 1/50 (minimal) to 22/50 (44%), most of the way to the 26/50 (52%) of the full budget—$\sim$84% of the gap. Against minimal this is highly significant (Fisher $p$=$3.7\times10^{-7}$); against the full budget the residual is small and

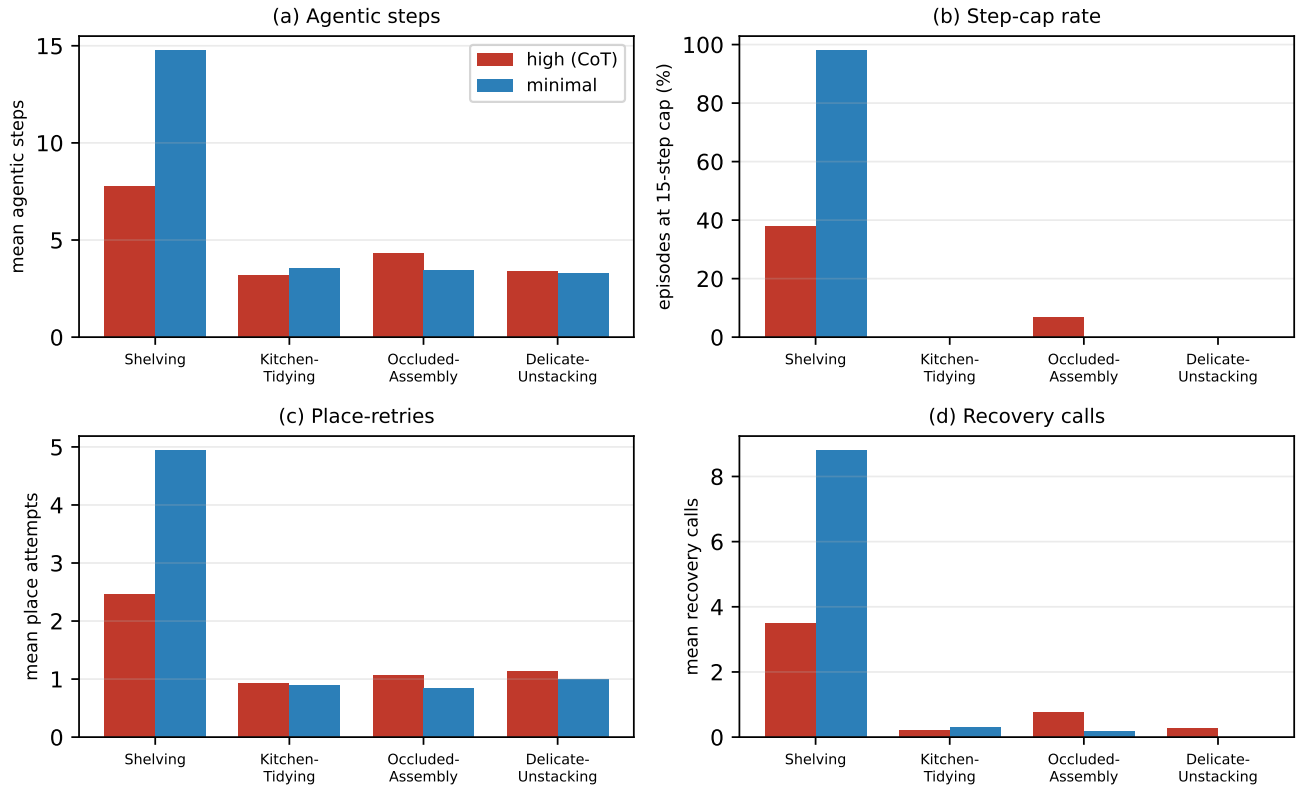


Fig. 7. **H4: the loop metrics diverge only on Shelving.** Mean agentic steps, 15-step-cap rate, place-retries, and recovery calls all separate the conditions sharply on Shelving and are flat on the three inert tasks (Occluded-Assembly shows only the mild high-budget over-elaboration of §V). Where it exists, the reasoning effect surfaces as loop-level divergence; but that divergence is largely a downstream symptom of the metric destination box, the causal driver (App. E).

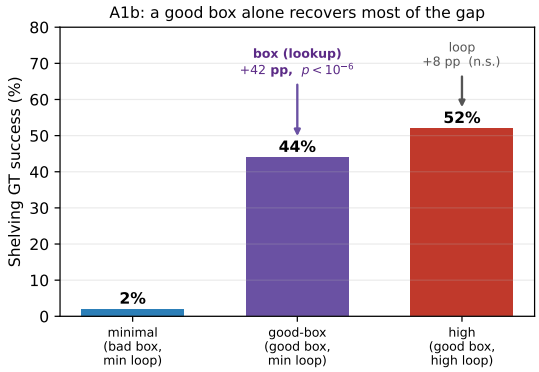


Fig. 8. **A1b: a correct destination box alone recovers most of the Shelving gain.** Under the minimal budget, forcing *only* the destination-grounding hook to a high budget (a good box, minimal loop) lifts ground-truth success from 1/50 to 22/50, most of the way to the 26/50 of the full budget. The box (lookup) step is decisive ($p$=3.7 × $10^{-7}$ vs. minimal); the loop residual to full-high is not significant ($p$=0.55 at $n$=50).

not significant ($p$=0.55). Two corollaries follow: fixing the box alone collapses the loop thrash (step-cap 98%→44%, mean recovery calls 8.8→4.2), so most of the loop divergence H4 reports is a *downstream symptom* of the bad box; and all but one of the 22/50 successes are clean first attempts, consistent with H3.

**Interpretation.** The dominant, causally-tested driver of the Shelving gain is the *metric* destination grounding—a lookup the cheap model gets wrong *in the full pipeline* (an off-frame box) even though its isolated pointing is flat (App. D). The decisive signal is thus the *metric* lookup, with the loop largely downstream; *categorical* grounding (which object, is-it-a-shelf) stays budget-invariant throughout. At $n$=50 the residual (44→52%) is small and not significant, so the loop adds no resolvable contribution beyond fixing the box; whether even that 8-point residue is real is what the recovery ablation (§VII) would settle.