CAN AGENTS ADAPT WORLDS WITHOUT BREAKING WHAT WORKS? BENCHMARKING ROBOT-CONDITIONED ENVIRONMENT AUTHORING

Anonymous authors
Paper under double-blind review

ABSTRACT

An authoring agent should adapt an environment to a changed robot or task while preserving useful functions. We introduce a benchmark for robot-conditioned environment authoring centered on paired functional adaptation. The frozen source and edited world are evaluated under the same changed contract, requiring completion of a main job and retention of an independent protected job certified after the change but before editing. Each job starts from a separate reset of the submitted initial state. Acceptance requires complete motion witnesses for each declared test condition and distinguishes certification, directly established violations, and unresolved verification. Shared tools and explicit resource accounting support fixed-backbone policy comparisons, including our dependency-aware reference authoring workflow, and separate within-backbone Base/Ours comparisons. Frozen-world and literal-translation controls test certification gains from editing and when simple shifts suffice. Nested verifier budgets assess the sensitivity of conclusions to evaluation effort. Creation and valid position variation serve as auxiliary tracks. Main-only repair isolates the gap between main completion and joint completion, making protected-job retention an explicit evaluation target.



Figure 1: **Adapt the world while retaining what works.** A 20 cm mounting shift leaves all objects fixed. An author may reposition the blue-payload task's tray and teal bin, while the amber task's fixtures remain fixed. Frozen and edited worlds face the same changed contract: complete job A and retain independent job B, each from its own reset. Layouts and robot appearance are conceptual; Figure 7 isolates the potential protection conflict.

1 INTRODUCTION

A robot environment is useful relative to what a particular robot must do. Consider a remounted arm: moving a teal bin closer may help job A, a blue-payload transfer, yet interfere with job B, an independent amber-payload transfer between fixed fixtures. Figure 1 illustrates the changed contract.

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.

B's continued executability requires checking its complete job in the edited world. This motivates our central question: *can an authoring agent restore a robot's required job after a contract change while preserving another complete job that still works?*

Recent work supplies substantial machinery for this problem. EmbodiedGen V2 combines task-driven world construction with stateful editing (Wang et al., 2026d), and SAGE couples task-consistent scene augmentation with robot motion planning (Xia et al., 2026). MUSE and AuthorBench organize requirement satisfaction and preservation-aware editing (Xu et al., 2026); Edit-As-Act plans scene edits through goal regression and validated preconditions (Noh et al., 2026). Our evaluation unit is a fixed source under a changed robot–task contract: restoring one complete job while retaining another.

We introduce a benchmark for rigid tabletop manipulation centered on *paired functional adaptation*. Each source satisfies its original contract; after one condition changes, the frozen source and edited world receive the same new-contract evaluation. In the primary suite, one independent job B is certified after this change but before editing. Case admission also requires a target certified for both jobs. We measure regression retention (RR) of B separately from main-job completion. Both jobs start from separate resets of one submitted initial state, with continuous state within each multi-transfer job. Support and edit permissions are hard constraints; the protected registry contains jobs. Creation and valid position variation are auxiliary tracks.

Independent acceptance requires all contract checks and complete motion witnesses for both jobs under every declared condition. Witnesses may differ across conditions. Direct contract violations yield R; otherwise, complete certification yields C and insufficient evidence yields U. Failed planner searches remain U until direct violation or sound exclusion evidence is available. Graded checks distinguish geometry and endpoint reachability from complete execution, while nested verifier budgets accumulate evidence on the same fixed outputs.

Controlled comparisons include strong geometry search, motion-aware search, and a generic feedback policy. We compare shared-backend Edit-As-Act-style, MUSE-style, and EmbodiedGen-style adapters with a dependency-aware reference policy. Common tools, charged probes, and matched resource caps constrain backend advantages. Frozen-world and literal-translation controls test certification gains from editing and when simple relocation suffices. The study examines when search is sufficient, whether edits targeting A reduce B's certification, and how additional verifier effort changes the interpretation.

Our contributions are threefold:

- **Benchmark.** We define paired functional adaptation with an independent protected job certified after the contract change and before editing.
- **Measurement protocol.** We specify complete-witness evaluation with shared tools, explicit resource accounting, and distinct $C/R/U$ outcomes.
- **Reference workflow and comparisons.** We combine inherited mechanisms in a dependency-aware authoring workflow, with separate fixed-backbone policy and within-backbone Base/Ours comparisons of joint completion, retention, and resource tradeoffs.

2 Related Work

Task-conditioned scene generation. Simulation environments increasingly combine scene construction with robot task resources. Holodeck generates interactive 3D environments from language prompts (Yang et al., 2024); WorldClaw plans editable open worlds, and SimWorlds adds dynamic scenes with runtime-state checks (Guo et al., 2026; Liu et al., 2026). RoboCasa supplies diverse kitchens, interactable assets, and manipulation tasks (Nasiriany et al., 2024). EmbodiedGen V2 constructs task-conditioned scene graphs, instantiates assets, and supports stateful edits (Wang et al., 2026d); MesaTask infers tabletop inventories and spatial relations (Hao et al., 2025). Interact3D uses collision-aware composition and visual correction for interacting objects (Shan et al., 2026). SAGE incorporates physical feedback and task-preserving augmentation with robot actions (Xia et al., 2026); SceneSmith combines hierarchical generation with robot-policy evaluation (Pfaff et al., 2026). These systems connect construction to downstream use. Our complementary question concerns a fixed source under a changed contract: whether editing restores one complete robot job while retaining another, measured against the frozen world under that same change.

Preservation-aware environment authoring. Preserving existing capabilities is central when environments change. Environment shaping bounds degradation of an actor's original task through source-based edits and rollback (Saisubramanian & Zilberstein, 2021); workspace co-design couples layouts, scheduling, and collision-constrained trajectories (Zeng et al., 2024). MUSE uses explicit requirements, persistent memory, and protection-aware repair; AuthorBench separates internal checks from exported-state and independent visual-semantic evaluation (Xu et al., 2026). Edit-As-Act preserves goals through regression and precondition/effect validation (Noh et al., 2026), while Embodied-BenchForge uses typed artifacts, dependencies, and provenance for selective re-execution, repair, and rollback (Jiang et al., 2026). VULCAN couples tool-based object arrangement with multi-agent verification; aDSL combines relational 3D operators with Plan–Execute–Critic repair (Kuang et al., 2025; Wang et al., 2026a). ScenePilot learns local and global layout repair, while SceneOrchestra plans full tool-call trajectories for scene synthesis (Zhang et al., 2026; He et al., 2026). Our reference builds on these mechanisms, and our evaluation measures retention of an independent complete job certified after the contract change and before editing.

Embodied benchmarks and verification. Robot benchmarks provide task protocols against which functional claims can be tested. RLBench supplies manipulation tasks, demonstrations, and task-building validators (James et al., 2020), while LIBERO studies knowledge transfer across task suites for lifelong robot learning (Liu et al., 2023). These benchmarks primarily evaluate robot policies; 3DCodeBench instead tests agent-generated procedural models (Gao et al., 2026). RoboPhys-3D measures video world models across image, geometry, state, and task levels (Wang et al., 2026c); WorldBench isolates physical concepts, and GAUGE grounds physical-fidelity tests in measured trajectories (Upadhyay et al., 2026; Wang et al., 2026b). PhyScensis combines spatial/physical predicates, physics-backed solving, and agent feedback (Wang et al., 2026e); HumanCLAW uses calibrated atomic skills and staged diagnosis to separate action decisions from motor-execution confounds (Li et al., 2026). These precedents motivate strong search controls and explicit verification boundaries. Our target is authoring robot environments: the protocol distinguishes complete-execution certification, established violations, and unresolved witness discovery.

Appendix Table 5 contrasts prior capabilities with these evaluation obligations.

3 Benchmark Design and Evaluation Protocol

Figure 2 separates authoring decisions, shared tool execution, and external acceptance. The author edits the world through the shared tools; the external evaluator retains final acceptance evidence.

3.1 Paired contracts and independent protection

An episode specifies an instruction, source world W_0, old/new contracts κ_0/κ_1, main job A, protected registry J, hard requirements H, allowed edits, asset pool $\mathcal{A}$, and resource envelope $\mathcal{B}$. Persistent IDs identify objects, geometry, poses, and physical attributes. An external intervention changes one robot, workspace, or task requirement and cannot be reversed by the author. Frozen W^- preserves the source scene; both W^- and submitted W^+ are evaluated under κ_1.

Admission requires H-compliant old sources and permitted new targets, complete $A + B$ in each, and frozen-new B including its shared obligations, across all declared conditions. Screening is independent of policy outcomes; cases needing no repair or solved by translation remain eligible. Thus $J = G = \{B\}$, where G contains protected jobs certified immediately before editing. H contains support and permission constraints; the protection count covers jobs in G.

In the six-object case, A transfers a cube from its tray into a bin; B transfers a different block between fixed fixtures. Only A's three initial poses are editable. After a mounting shift, translating these objects places the bin near B's declared grasp space; a second candidate relocates the bin away from this region (Figure 7). These are candidate layouts for subsequent contract evaluation (Appendix A.2).

3.2 Tasks and paired analysis

Main jobs comprise single transfer, ordered sorting, or staged transfer. A and B start from independent resets of the same submitted initial world. Within either job, robot and object states persist across



Figure 2: **Conceptual authoring workflow with independent acceptance.** Ours uses dependency records and public feedback from shared, charged tools; final-audit evidence never returns to the author. Frozen and submitted worlds face the same new contract. Joint certification requires all hard checks and complete witnesses per declared condition; A and B reset separately to the same submitted initial world, each continuous internally.

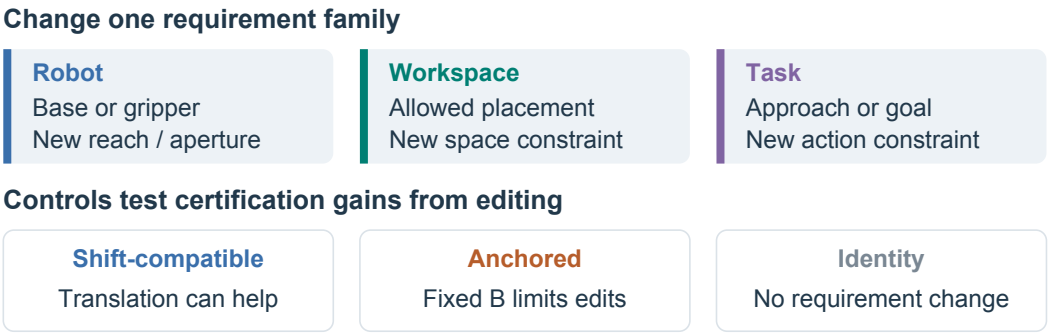

Figure 3: **Interventions and controls answer different questions.** One requirement family changes at a time. Shift-compatible cases retain simple coordinate-restoration opportunities; anchored cases constrain edits through fixed fixtures. Identity cases test unnecessary changes. The formal suite has 216 changed and 24 identity contexts across 48 task-asset bundles. Construction tags define demand archetypes; Appendix A.9 specifies allocation and split construction.

edges: no reset, teleportation, attachment shortcut, or prefix splicing is allowed. A failed prefix ends the attempt.

The allocation contains 48 bundles across three task families and two splits, with five contexts per bundle: 216 changed contexts and 24 separate identity controls (Figure 3). ID and joint-OOD label controlled demand regimes with disjoint symbolic roots; the latter adds a fixed difficulty offset. Appendix A.9 separates this stress test from geometric class/requirement holdout. The development pilot is disjoint. We average three scored seed slots within each context, then changed contexts equally. Equal weighting of changed-bundle means would alter this target because bundles contribute four or five changed contexts. Bundles retain all dependent records. The primary descriptive contrast is reference minus MUSE-style standard-budget joint certified completion (CC) on changed contexts; identity summaries are reported separately (Appendix A.9). Creation and position variation remain auxiliary tracks (Appendix A.10).

3.3 Shared Access and Charged Authoring

Every policy receives the contracts, complete scene/robot metadata, allowed edits, catalog, three fixed views, public conditions, and feedback history. Reference trajectories are withheld. The controlled-policy comparison fixes the backbone, decoding, and observation encoding; backbone controls instead pair two authoring policies within each model. Typed tools inspect, render, edit,

check, probe, restore, and submit. The standard envelope is 32 atomic calls and eight combined endpoint/full-probe credits. Each endpoint or full probe addresses one function and condition, using one call and one shared credit; S0/S1 checks use calls only. Batches expand into atomic requests; scene-JSON differences become charged per-ID edits in addition to submission. Failed requests and restoration consume resources without refunds.

Exact geometry, IK, planning, and simulation must use charged tools, not private solver copies. Cheap candidate arithmetic is permitted and timed. Shared wall-time, token, and model-call caps supplement call/credit limits; realized computation is recorded separately (Appendix A.4).

3.4 Complete witnesses and diagnostic evidence

Each transfer checks approach, bilateral grasp, lift, transport, release, and stable placement, including required retraction and contact semantics. The seven conditions are nominal, common $\pm 5\,\mathrm{mm}$ world-x/y payload offsets, and friction multipliers $0.9/1.1$. Each reset perturbs all distinct payloads once, including the inactive job's payload, but not fixed fixtures.

Let $V(W, f, z, w)$ mean that the external evaluator obtained and successfully replayed complete witness w for function f under condition z and κ_1. Certification is

$$\begin{aligned} \mathrm{Cert}(W; \mathcal{F}) \iff & H(W) = \mathrm{PASS} \\ & \wedge\ \forall f \in \mathcal{F}\ \forall z \in \mathcal{Z}\ \exists w:\ V(W, f, z, w). \end{aligned} \tag{1}$$

Witnesses may be condition-specific. Main completion uses $\mathcal{F} = \{A\}$; joint CC uses $\mathcal{F} = \{A, B\}$. Regression retention (RR) is one when B has a complete witness for every condition, regardless of unrelated hard violations. Shared obligations, including inactive-payload support, must still pass for a functional witness; hard-failure attribution follows each job's required predicates. $B = \mathrm{R}$ and $B = \mathrm{U}$ receive zero credit but remain separate. Completion and retention rates average these indicators with the context weights defined above. Every requested output remains in the denominator.

Outcomes distinguish certified (C), established violation (R), and unresolved verification (U). For either main or joint scope, R requires a direct hard violation or sound exclusion of all permitted witnesses for a required function–condition. Otherwise, missing certification gives U. Failed grasp/IK/path attempts alone cannot establish R. Author exhaustion without a completed submission gives main/joint R for a submission-contract violation. Infrastructure preventing assessment gives U absent separate R evidence. Both no-artifact cases have $B = \mathrm{U}$. The active configuration uses direct hard predicates; functional exclusion requires a registered, scope-complete checker and payload. Unbound negative proposals remain U (Appendix A.6).

Diagnostics record S0's binding nominal serialized envelope, S1's exact initial geometry, permissions, and settling, compatible endpoints E_A/E_B with held-object geometry, and complete A/B replays (Figure 4). We cross-tabulate each layer's PASS/direct-failure/UNKNOWN against main and joint C/R/U separately. Continuous-A diagnostics condition on cumulative S1-and-E_A passes and satisfied hard checks; B-specific evidence is analyzed separately. U after a pass denotes unresolved verification. The recorded E_A/E_B labels define the endpoint diagnostic; Appendix A.5 specifies the independent endpoint-only search for obtaining them in an executable implementation.

Final verification enumerates legal grasp tuples with whole direct-interpolation and RRTConnect attempts. A 60 s function–condition cap includes IK, planning, settling, and replay; the external evaluator independently checks public successes. Nested $2\times$ verification retains the same outputs and all $1\times$ evidence, extending unresolved obligations within cumulative caps. C/R persist; this positive-search extension may convert U to C through additional witnesses. Larger authoring budgets generate new outputs; independent-planner audits are distinct from this nested extension (Appendix A.7).

3.5 Controlled authoring policies

Seven policies share these observations and resources. Strong geometry search uses support-conditioned continuous multistart proposals, local refinement, and charged S1 checks and A/B endpoint queries. Motion-aware search uses the same proposal family: on changed cases, two initial nominal endpoint checks, then complete public A/B probes, all charged to the same credit cap. For each context and authoring budget, fixed searches use effective author seed zero and reuse the output,

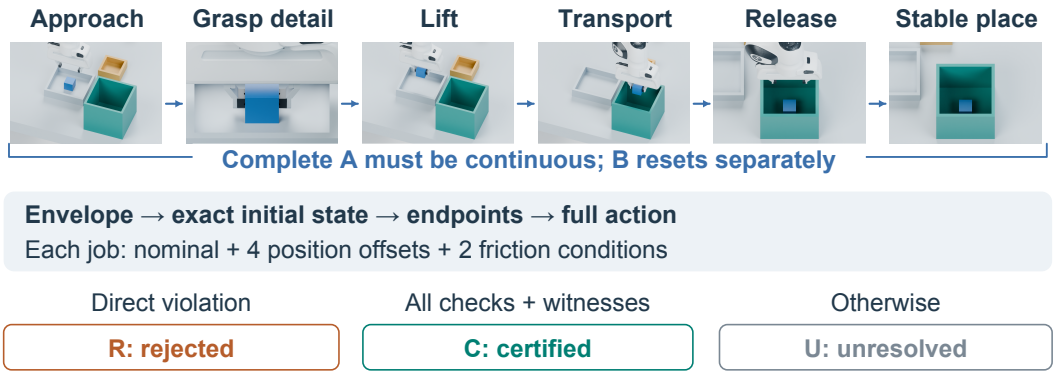

Figure 4: **Continuous functional evidence and three-way certification.** Posed, static stage illustrations show the transfer sequence; grasp is a close-up and the final two panels share a higher destination view. Composed transfers retain state, while B reloads the same submitted initial scene. Geometry and endpoint checks precede complete-action evidence. A direct required violation takes precedence (R); all hard checks and a complete witness for each job under each condition give C. Otherwise the outcome is U. Failed sampled grasp, IK or path attempts alone cannot establish R.

evaluation, and resource record across scored slots. Generic feedback chooses actions from complete history without an imposed requirement program.

Edit-As-Act-style preserves goal regression and precondition/effect validation (Noh et al., 2026); MUSE-style retains executable requirements, memory, protection, consistency checks, and repair (Xu et al., 2026); EmbodiedGen-style routes a task-role graph through typed skills (Wang et al., 2026d). These are shared-backend adapters, not native reproductions; each retains protection and functional checks.

Our authoring policy. **Ours** is a dependency-aware reference authoring workflow that tracks instance, robot-parameter and region dependencies in requirement memory. Edits invalidate affected evidence and trigger targeted rechecks, protection-first repair or restoration; reuse requires complete, unchanged pure-metadata read sets under the declared snapshot rules. This workflow builds on requirement and dependency mechanisms (Xu et al., 2026; Jiang et al., 2026). Checkpoints contain public evidence only; ablations retain the same final auditor.

Frozen and literal-translation controls test certification gains from editing and when simple relocation suffices. Translation moves only permitted objects, without clamping; invalid proposed scenes retain their violations rather than inheriting an unchanged backend fallback's score. Appendix A.8 specifies rankings, stopping rules, and adapters.

4 Experiments

Design and interpretation. The numerical results below are fixed synthetic example records; no agent, model, robot, or simulator runs produced them. Table 1 compares seven policy profiles at shared caps and context weights (Section 3.2), excluding identity controls. Summaries are descriptive and retain reuse-linked dependence (Appendix A.9).

In a paired frozen-world diagnostic, frozen joint-C is 78/648; the corresponding Ours outputs have 39 C, 13 R and 26 U (Appendix Table 13). These transitions classify joint outcomes; B retention uses separately recorded B evidence.

Performance of our authoring policy. Ours leads joint CC and ties motion-search RR; Edit-As-Act-style and EmbodiedGen-style have higher A completion (Table 1). Joint C is the primary metric. Relative to MUSE-style, Ours has 3.24 points higher joint CC, a 95% paired-bundle resampling interval of $[1.39, 5.30]$ points (Appendix A.9), and one extra mean call at unchanged credits (Table 18). Both methods have main=joint.

Table 1: Policy comparison on 216 changed contexts (648 slots), at 32 calls/eight credits. Rates are percentages; joint C/R/U partition all slots and RR includes B's support obligations. Joint C is primary. Orange identifies Ours; bold/underline mark best/second distinct A CC, joint C and RR, including ties.

Policy	A CC ↑	Joint C ↑	R	U	RR ↑
Strong geometry search	10.6	10.6	0.9	88.4	57.4
Motion-aware search	57.4	54.6	4.6	40.7	**92.1**
Generic feedback	39.2	18.8	10.5	70.7	56.3
Edit-As-Act-style	**73.9**	56.6	7.1	36.3	70.1
MUSE-style	63.3	<u>63.3</u>	8.3	28.4	<u>91.7</u>
EmbodiedGen-style	<u>72.8</u>	62.0	4.3	33.6	83.5
Dependency-aware (Ours)	66.5	**66.5**	7.9	25.6	**92.1**

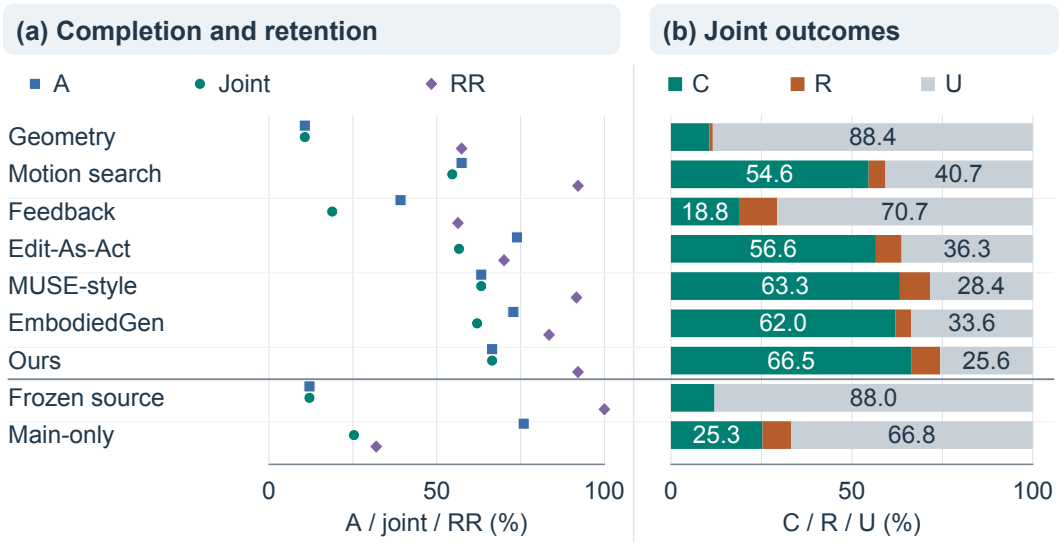

Figure 5: **Main-task completion and joint availability are distinct outcomes.** Left: certified main A, joint completion, and RR (B certification over all outputs). Right: joint C (certified), R (established violation), and U (unresolved verification). All 216 changed contexts (648 slots) remain included without static-pass filtering. Lower rows: frozen-source and main-only controls. Caps: 32 calls/eight shared probe credits. Literature names denote style adapters; exact rates appear in Tables 1, 2 and 11.

When do simple controls suffice? Translation, motion search and Ours receive joint C on all 54 slots in 18 shift-compatible contexts, using 5.00/17.61/27.61 mean calls. Separate identity controls retain both jobs without edits (Table 12). Motion search matches our RR with different B evidence (30 R/21 U versus 51 R/0 U), fewer calls, and higher workspace joint CC (80.56% versus 64.35%).

Does main-task repair preserve independent work? Main-only scheduling raises A completion but lowers joint CC (Table 2); 328/492 main-certified slots have $B = \mathrm{U}$ (Table 14). RR's denominator retains hard-R outputs; B's shared obligations still apply.

Fixed full rechecks use means of 32 calls/eight credits, compared with the reference's 30.33/6.43.

What remains after initial-state and endpoint passes? Within the $S1 \wedge E_A$ PASS subset, 107/538 Ours records are main-U because complete-function coverage remains missing (Table 4). This gate combines S1's shared-support checks across conditions with E_A passes. Figure 7 illustrates a potential B-endpoint obstruction; Table 4 reports main ($H + A$) outcomes.

Table 2: Auxiliary policies and reference ablations under the standard authoring budget. All methods retain the same contract. Rates and highlighting follow Table 1; orange identifies Ours.

Policy	A CC ↑	Joint C ↑	R	U	RR ↑
Dependency-aware (Ours)	66.5	**66.5**	7.9	25.6	**92.1**
Main-task-only scheduling	**75.9**	25.3	7.9	66.8	31.9
Fixed full rechecks	56.0	56.0	7.9	36.1	**92.1**
Without structured ledger	58.3	54.9	7.9	37.2	88.3
Without mandatory rechecks	72.7	46.5	7.9	45.7	59.0

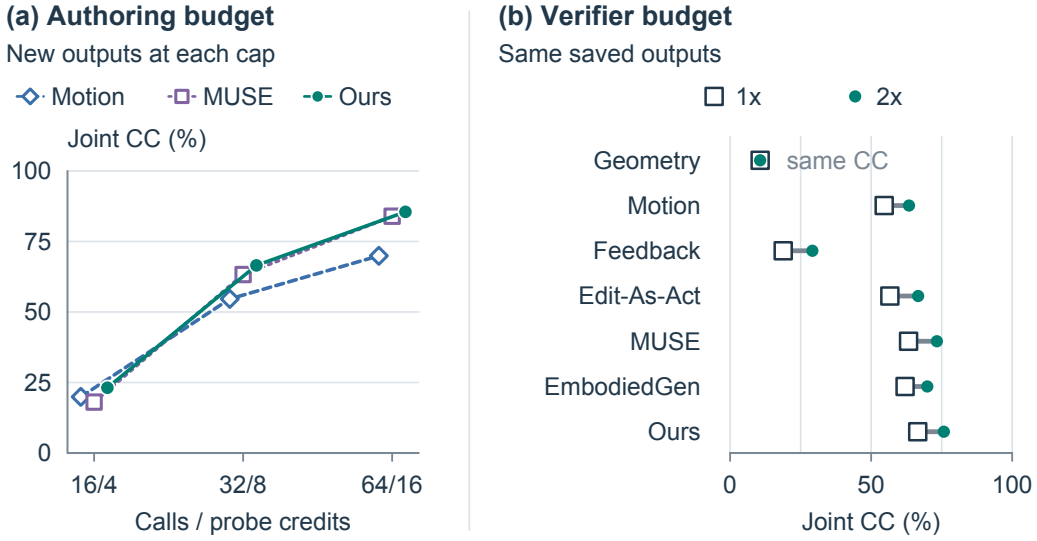

Figure 6: **Authoring effort changes worlds; verifier effort changes evidence.** Left: new outputs at three call/credit caps, with the evaluator fixed. Categories are discrete; symbols have small horizontal offsets and connecting lines guide reading. Right: nested $1\times$ /$2\times$ verification of the same standard-budget outputs retains earlier evidence. Both panels use the same 216 changed contexts. Literature names denote style adapters.

Table 3: Model-labelled Base/Ours workflow controls. Joint CC (%); ΔCC/ΔRR: Ours minus Base in pp, computed before rounding. Orange marks Ours.

Model label	Joint CC (%)		ΔCC	ΔRR
	Base	**Ours**		
GPT-6 Astra	29.9	33.2	+3.2	+13.4
Claude Sonnet 5	24.7	30.9	+6.2	+15.9
Gemini 3.8 Flash	33.8	37.8	+4.0	+13.0
Qwen3.8-27B	29.8	28.9	-0.9	+10.8

Table 4: Main (H+A) counts for Ours from Table 1, within each PASS subset (n). Full diagnostics: Appendix Table 16.

PASS subset	n	C	R	U
S0	648	431	51	166
S1	597	431	0	166
$S1 \wedge E_A$	538	431	0	107

Model-labelled workflow contrasts and tradeoffs. Table 3 reports Base/Ours profiles under four model labels. Joint CC increases in three model slots and decreases by 0.9 points in the Qwen-labelled slot; Table 24 reports the paired A/cost tradeoffs.

Additional evidence or additional authoring? Figure 6 separates new worlds from added verification on fixed outputs. Positive-only $2\times$ search converts 60 Ours slots from U to C, giving 75.77% joint CC with R unchanged (Table 17); motion search exceeds Ours RR at the high authoring cap (Table 19).

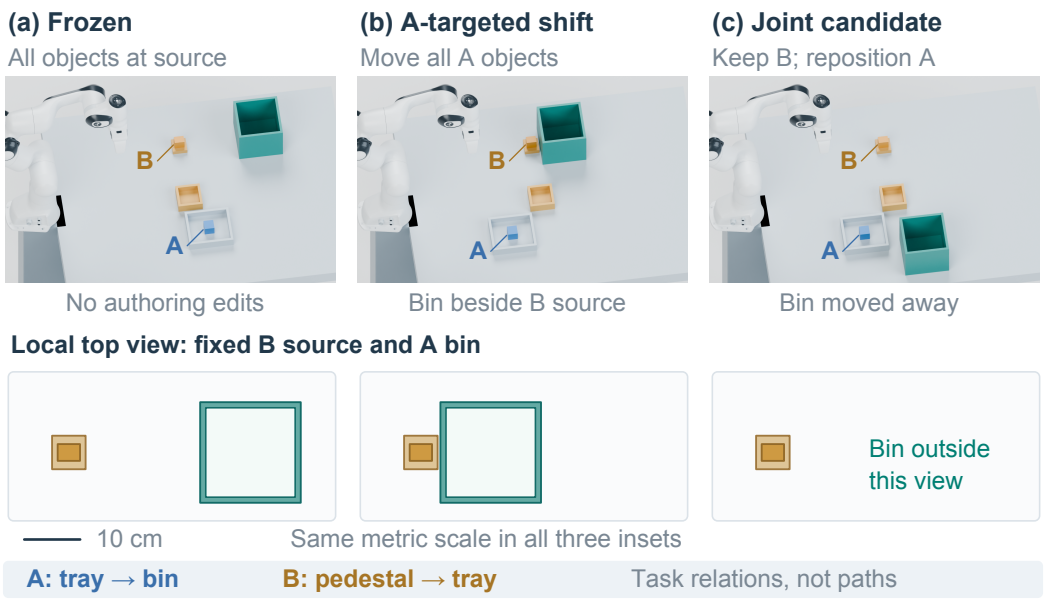

Figure 7: **Protected-job checks extend beyond fixed object poses.** Literal translation of the permitted A objects places the tall teal bin near the fixed amber source, motivating a check of B's approach space. The joint candidate relocates the bin while preserving all initial B poses. The three layouts share one camera/crop and the same changed robot; local top views use an identical metric window and scale. These schematic views specify candidate geometry; collision and execution outcomes require corresponding verifier evidence.

Auxiliary construction and position variation. Creation requires both jobs anew and reports joint C/R/U with RR=N/A (Table 20). Variation reports valid, unique-valid and complete-batch yield with missing outputs counted; position rejection screens descriptor separation only (Table 21).

Limitations. Scope is rigid tabletop contracts under a finite condition suite. A/B certification uses separate resets with shared obligations; serial execution and a condition-robust controller require separate tests. Unequal joint-U shares keep comparisons descriptive and evaluator-specific (Appendix A.7). Physical capability, efficiency, and mechanism attribution require geometry-bound execution, calibrated evaluators and controllers, measured costs, and controlled policy runs. Variation measures position distinctness.

5 Conclusion

Paired functional adaptation evaluates whether a world edited for a changed robot–task contract completes the main job while retaining an independently certified job. In the main-only control, 75.93% of submissions certify A but 25.31% certify both jobs (Table 2); 328 of 492 main-certified slots leave B unresolved (Appendix Table 14). Protected-job retention is therefore a distinct benchmark outcome, with U recording missing evidence rather than an established loss.

The diagnostic controls clarify the comparison. Translation, motion search, and the reference workflow each obtain joint C on all 54 shift-compatible slots, making simple coordinate restoration a meaningful baseline. Among 538 reference records passing initial-state and endpoint checks, 107 remain main-U pending complete witnesses (Table 4). Doubling verification on fixed submitted worlds converts 60 additional U outcomes to C (Appendix Table 17); increasing the authoring budget instead changes the worlds submitted. The benchmark, complete-witness protocol, and dependency-aware reference workflow separate these sources of variation under shared tools and budgets, providing a controlled benchmark specification for environment-authoring comparisons.

AI USE STATEMENT

Use of Large Language Models.During the preparation of this work, the authors used ChatGPT to improve language, readability, and presentation. The authors reviewed and edited all model-assisted content as needed and take full responsibility for the content of the publication.

REFERENCES

Yipeng Gao, Lei Shu, Genzhi Ye, Xi Xiong, Ameesh Makadia, Meiqi Guo, Laurent Itti, and Jindong Chen. 3DCodeBench: Benchmarking agentic procedural 3D modeling via code. *arXiv preprint arXiv:2606.01057*, 2026. URL https://arxiv.org/abs/2606.01057v1.

Chunchao Guo, Jinpeng Li, Yang Li, and Zilong Huang. WorldClaw: Agentic 3D open-world generation at scale. *arXiv preprint arXiv:2608.05248*, 2026. URL https://arxiv.org/abs/2608.05248v1.

Jinkun Hao, Naifu Liang, Zhen Luo, Xudong Xu, Weipeng Zhong, Ran Yi, Yichen Jin, Zhaoyang Lyu, Feng Zheng, Lizhuang Ma, and Jiangmiao Pang. MesaTask: Towards task-driven tabletop scene generation via 3D spatial reasoning. *arXiv preprint arXiv:2509.22281*, 2025. URL https://arxiv.org/abs/2509.22281v1.

Yun He, Kelin Yu, and Matthias Zwicker. SceneOrchestra: Efficient agentic 3D scene synthesis via full tool-call trajectory generation. *arXiv preprint arXiv:2604.19907*, 2026. URL https://arxiv.org/abs/2604.19907v2.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. doi: 10.1109/LRA.2020.2974707. URL https://sites.google.com/view/rlbench.

Baoyang Jiang, Fengchun Zhang, Leyuan Wang, Haotian Li, Yida Wang, Zhe Ji, Jinshan Lai, Xi Ren, Danyang Li, Zheng Yang, Jianwei Hu, and Qiang Ma. Embodied-BenchForge: A closed-loop agentic workflow for embodied benchmark construction. *arXiv preprint arXiv:2609.13082*, 2026. URL https://arxiv.org/abs/2609.13082v1.

Zhengfei Kuang, Rui Lin, Long Zhao, Gordon Wetzstein, Saining Xie, and Sanghyun Woo. VULCAN: Tool-augmented multi agents for iterative 3D object arrangement. *arXiv preprint arXiv:2512.22351*, 2025. URL https://arxiv.org/abs/2512.22351v2.

Siyao Li, Jiawei Gu, Shuai Liu, Kairui Hu, Zekun Li, Linjie Li, Chengcheng Tang, Po-Chen Wu, Ivan Shugurov, Lingni Ma, Michael Zollhoefer, Sizhe An, Abhay Mittal, Amy Zhao, Ranjay Krishna, Manling Li, Ziwei Liu, and Chuan Guo. HumanCLAW: Can vision-language models act through a body? *arXiv preprint arXiv:2607.27180*, 2026. URL https://arxiv.org/abs/2607.27180v2.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Chunjiang Liu, Xiaoyuan Wang, Haoyu Chen, Yizhou Zhao, Ming-Hsuan Yang, and László A. Jeni. SimWorlds: A multi-agent system for dynamic 3D scene creation. *arXiv preprint arXiv:2607.01766*, 2026. URL https://arxiv.org/abs/2607.01766v1.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of household tasks for generalist robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.050. URL https://www.roboticsproceedings.org/rss20/p050.html.

Seongrae Noh, SeungWon Seo, Gyeong-Moon Park, and HyeongYeop Kang. Edit-As-Act: Goal-regressive planning for open-vocabulary 3D indoor scene editing. *arXiv preprint arXiv:2603.17583*, 2026. URL https://arxiv.org/abs/2603.17583v1.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. SceneSmith: Agentic generation of simulation-ready indoor scenes. *arXiv preprint arXiv:2602.09153*, 2026. URL https://arxiv.org/abs/2602.09153v2.

Sandhya Saisubramanian and Shlomo Zilberstein. Mitigating negative side effects via environment shaping. *arXiv preprint arXiv:2102.07017*, 2021. URL https://arxiv.org/abs/2102.07017v1.

Hui Shan, Keyang Luo, Ming Li, Sizhe Zheng, Yanwei Fu, Zhen Chen, and Xiangru Huang. Interact3D: Compositional 3D generation of interactive objects. *arXiv preprint arXiv:2603.16085*, 2026. URL https://arxiv.org/abs/2603.16085v2.

Rishi Upadhyay, Howard Zhang, Jim Solomon, Ayush Agrawal, Yunhao Ba, Alex Wong, Celso M. de Melo, and Achuta Kadambi. WorldBench: Benchmarking physical understanding of world models by isolating physics concepts. *arXiv preprint arXiv:2601.21282*, 2026. URL https://arxiv.org/abs/2601.21282v2.

Rui-Huan Wang, Si-Tong Wei, Jia-Qi He, Heng-Yi Wei, Baoquan Chen, and Peng-Shuai Wang. aDSL: Agentic 3D creation via joint agent-program design. *arXiv preprint arXiv:2608.17975*, 2026a. URL https://arxiv.org/abs/2608.17975v1.

Shuai Wang, Yaxin Feng, Xuekun Jiang, Shihan Tian, Ningyu Yan, Xing Shen, Chaoyang Lyu, Hui Wang, Yunsong Zhou, Hanqing Wang, Jiangmiao Pang, Yang Xiang, Xing Gao, Chunhua Shen, and Weinan Zhang. GAUGE: A measurement-grounded benchmark for physical fidelity in simulation engines and video world models. *arXiv preprint arXiv:2608.05948*, 2026b. URL https://arxiv.org/abs/2608.05948v1.

Tianyi Wang, Jiazhou Chen, Yiming Xu, Xiangyu Li, Tianyi Zeng, Chih-Hsien Chou, Ning Lu, Liang Peng, Junfeng Jiao, and Christian Claudel. RoboPhys-3D: A comprehensive embodied world model evaluation via 3D reconstruction. *arXiv preprint arXiv:2608.28718*, 2026c. URL https://arxiv.org/abs/2608.28718v1.

Xinjie Wang, Liu Liu, Taojun Ding, Andrew Choi, Chaodong Huang, Mengao Zhao, Ziang Li, Jackson Jiang, Chunlei Yu, Shengxiang Liu, Wei Xu, and Zhizhong Su. EmbodiedGen V2: An agentic, simulation-ready 3D world engine for embodied AI. *arXiv preprint arXiv:2607.07459*, 2026d. URL https://arxiv.org/abs/2607.07459v2.

Yian Wang, Han Yang, Minghao Guo, Xiaowen Qiu, Tsun-Hsuan Wang, Wojciech Matusik, Joshua B. Tenenbaum, and Chuang Gan. PhyScensis: Physics-augmented LLM agents for complex physical scene arrangement. *arXiv preprint arXiv:2602.14968*, 2026e. URL https://arxiv.org/abs/2602.14968v1.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. SAGE: Scalable agentic 3D scene generation for embodied AI. *arXiv preprint arXiv:2602.10116*, 2026. URL https://arxiv.org/abs/2602.10116v2.

Ruijie Xu, Xinnan Zhu, Jiayu Ying, Daoguo Dong, Yuzhou Ji, and Xin Tan. MUSE: Agentic 3D scene authoring via memory-grounded incremental requirement satisfaction. *arXiv preprint arXiv:2606.14168*, 2026. URL https://arxiv.org/abs/2606.14168v1.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3D embodied AI environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16227–16237, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Holodeck_Language_Guided_Generation_of_3D_Embodied_AI_Environments_CVPR_2024_paper.html.

Peiyu Zeng, Yijiang Huang, Simon Huber, and Stelian Coros. Multi-robot workspace design and motion planning for package sorting. *arXiv preprint arXiv:2412.11281*, 2024. URL https://arxiv.org/abs/2412.11281v1.

Jiawei Zhang, Hongsong Wang, and Pan Zhou. ScenePilot: Grow-and-repair policy for text-driven 3D indoor scene generation. *arXiv preprint arXiv:2608.30307*, 2026. URL https://arxiv.org/abs/2608.30307v1.

A PROTOCOL DETAILS

A.1 SOURCE–CHANGE PAIRS AND INDEPENDENT PROTECTION

An adaptation context contains an old contract κ_0, a new contract κ_1, a source world W_0, a complete main job A, compatible protected functions J, allowed scene edits, hard conditions H, and fixed resource limits. The intervention changes one robot, workspace, or task requirement. The frozen world W^- retains the serialized source scene and is evaluated under κ_1, exactly as the authored world W^+ is.

Primary-suite admission requires complete A and B in the old source, complete B in the frozen new-contract scene, and a permitted new-contract reference world supporting both jobs. These obligations cover every declared condition. Admission requires the old source and new target to pass their respective H. Frozen B must satisfy its own and shared obligations; an unrelated new A-layout constraint may fail. Admission and replacement order are fixed independently of evaluated policy scores; cases where the frozen main job still works and cases solved by a simple translation remain in the population. The registry supersedes old A, requires new A, and retains or transforms the purpose of B under the current robot specification. Support, immutable fields, and initial validity belong to H, not to the protected-function count.

A and B are independent complete jobs, each reset to the same authored initial world. State persists across all edges of either composed job. Joint availability concerns both jobs independently available from that world. Primary admission gives $J = G = \{B\}$, where G contains protected functions certified in the frozen new-contract scene. Protected satisfaction and regression retention coincide on this suite; we report retention once. Independent jobs have distinct task objectives and separate resets; they may share safety/support predicates and have dependent outcomes.

Table 5: **Our benchmark in relation to prior work.** Each row pairs capabilities summarized in the cited discussion with our fixed evaluation obligations. Memory, rollback and robot execution are inherited capabilities.

Cited work	Source-bounded capabilities	**Our benchmark: fixed obligations**
Environment shaping (Saisubramanian & Zilberstein, 2021)	Source-based edits bound degradation of the actor's original task and support rollback.	Fix an external contract change and compare frozen/edited worlds under it.
Workspace co-design (Zeng et al., 2024)	Layout, scheduling and collision-constrained trajectories are co-designed.	Test complete A/B availability from the same initial world, with separate resets.
MUSE (Xu et al., 2026)	Requirements, persistent memory and protection-aware repair; AuthorBench separates internal, exported-state and visual-semantic evaluation.	Certify independent B after the change and before editing, then measure its retention.
Edit-As-Act (Noh et al., 2026)	Goal regression with precondition/effect validation preserves goals.	Treat the protected unit as a complete job and report main, joint and retention outcomes.
EmbodiedGen V2 (Wang et al., 2026d)	Task-conditioned world construction and stateful editing.	Compare authoring choices through common observations, tools and charged resource caps.
SAGE (Xia et al., 2026)	Task-consistent augmentation combines physical feedback with robot motion planning.	Require complete A/B witnesses under each declared condition, with separate resets.
SceneSmith (Pfaff et al., 2026)	Hierarchical scene generation connects to robot-policy evaluation.	Distinguish complete certification, established violations and unresolved witness discovery.
Embodied-BenchForge (Jiang et al., 2026)	Typed artifacts, dependencies and provenance support selective re-execution, repair and rollback.	Inherit dependency mechanisms; withhold final acceptance evidence from the author and separate verification extensions from new authoring.

A.2 A METRIC-SCALE RUNNING CASE

The running case uses a Panda parallel gripper above a fixed table of size $(1.20, 0.90, 0.06)$ m, centred at $(0.45, 0, 0.72)$ m. Its top is at $z = 0.75$ m. The mounting intervention moves the base from $(0, 0, 0.75)$ to $(-0.20, 0, 0.75)$ without rotation. Table 6 specifies six task objects. Container positions refer to outer bottom centres; payload positions refer to centres. All listed yaws are zero. Each open container has a 1 cm full-width floor and four nonoverlapping 1 cm walls. A fixed pedestal supports the second payload independently of the main tray.

Table 6: Running-case objects and source positions in metres. Fixed B poses are immutable to authoring; their payload remains dynamic during execution.

ID	Size (x, y, z)	Source position	Role
`tray_A`	$(.18, .16, .04)$	$(.46, -.18, .75)$	Editable source tray
`cube_A`	$(.04, .04, .04)$	$(.46, -.18, .78)$	Editable main payload
`bin_A`	$(.18, .18, .16)$	$(.72, .24, .75)$	Editable destination
`pedestal_B`	$(.06, .06, .02)$	$(.395, .24, .75)$	Fixed source; bottom-centre pose
`block_B`	$(.04, .03, .04)$	$(.395, .24, .79)$	Protected-job payload
`tray_B`	$(.10, .10, .04)$	$(.40, -.02, .75)$	Fixed receiving tray

A transfers `cube_A` into `bin_A`. B transfers `block_B` from its pedestal to `tray_B`, releases it, and maintains a one-second stable placement. The tasks use different payloads, sources, destinations, and completion predicates. B has a declared downward top-grasp orientation with finger-opening axis parallel to world x, yaw 0 or π within 5°, and 8 cm approach opening. This orientation constraint is part of the unchanged task purpose, not a geometric assertion that other grasps are physically impossible.

The author may translate or yaw the three A objects. Tray and bin centres obey $x \in [.25, .78]$, $y \in [-.30, .30]$, $z = .75$; cube centres obey $x \in [.20, .83]$, $y \in [-.35, .35]$, $z = .78$. Yaw lies in $[-\pi, \pi]$. Moving the tray never moves the cube implicitly. Dimensions, masses, friction, collision flags, fixed B poses, the post-change robot, and task predicates are immutable. Adaptation forbids insertion, deletion, execution-time teleportation, and kinematic payload attachment.

The cube starts within its tray with 5 mm footprint margin and bottom within 2 mm of the floor. The B payload has the same margin on the pedestal top. Fixture footprints lie at least 10 mm inside the table; bottoms lie within 1 mm of its top. Initial solids do not interpenetrate, both goals are unsatisfied, and each payload starts at least .25 m horizontally from its destination. One-second settling preserves these predicates with speed below .02 m/s and .1 rad/s. Fixed fixtures require geometric support even though they do not fall.

Literal translation of the permitted A objects by $(-.20, 0, 0)$ places the tray/cube at $(.26, -.18)$ and the bin at $(.52, .24)$. This restores their robot-relative coordinates while leaving the table and B fixtures fixed. The bin's left wall lies at $x \in [.43, .44]$, adjacent to B's declared grasp space; initial pedestal and payload solids remain separated from it. The joint candidate instead places the bin at $(.49, -.30)$, yielding an A source–destination distance of .259 m; B's distance is .260 m. These coordinates define candidate geometry. Gripper-collision and task-success labels require corresponding evidence; endpoint obstruction is diagnosed separately from continuous-motion failure.

A.3 CONTINUOUS TASK COMPOSITION AND CONTACT

A transfer edge $T(p, s, t, k, \tau)$ identifies payload, source, destination, destination kind, and stable-support dwell. Single transfer has one edge. Ordered sorting has two designated transfers in order, with the first placement maintained during the second. Staged transfer uses an intermediate tray, releases the payload for .5 s of stable support, and then regrasps it for the final transfer. The second edge starts from the actual state produced by the first, including observed object pose and velocities. Prefix failure ends the whole attempt; successful prefixes cannot be spliced across resets.

Each edge has six arm-motion segments: current state to pregrasp, descent to grasp, lift, transport to preplacement, descent to release, and retraction. Finger motion and contact/support dwells consume rollout time without adding planning segments. The semantic checks are approach, bilateral grasp,

lift, transport, release, and stable placement. Pregrasp is .10 m above the current payload centre. Bilateral finger forces reach .1 N for .1 s; lift clears the source rim or pedestal top by .05 m. Before destination descent, the payload footprint is within the opening and its bottom clears the rim by .02 m. Running-case horizontal displacement is at least .20 m. Short intermediate legs use their own geometry-derived positive displacement bounds.

Release opens the fingers by at least .05 m and ends their payload contact. For destination-local footprint half-extents (h_x, h_y), placement requires $|x| + h_x \leq W_{\text{in}}/2 - \delta$ and $|y| + h_y \leq D_{\text{in}}/2 - \delta$, where $\delta = 2$ mm for bins and 5 mm for trays. The bottom is within 3 mm of the floor, tilt is at most 5°, and no finger contact remains. Speeds stay below .02 m/s and .1 rad/s throughout the dwell. Bins require the payload top below the rim; trays permit protrusion. All non-superseded final placements hold jointly for a one-second terminal dwell.

Allowed contacts are stage-specific. Finger–active-payload contact is permitted during grasp, lift, transport, and release until contact termination; it is forbidden after release and throughout stable-placement and terminal dwells. Source support is permitted during supported grasp and must cease by completed lift. Destination support is permitted during release, placement, and dwell. Inactive payloads retain their initial support, earlier sorting payloads retain their goal support, and fixed fixtures retain table support. Forbidden penetration above 2 mm for two consecutive physics steps fails that attempt; scene-level status follows the complete evidence rules. An m-edge job permits at most $6m$ planned segments and $15m$ simulated seconds, including actuation and dwell, after one-second reset settling. Paths respect 25% of URDF velocity limits; an overlong path fails the attempt instead of being accelerated.

A.4 Public tools and resource accounting

All policies receive old/new contracts, persistent IDs, scene geometry and physics, robot limits, allowed edits, the asset catalog, three fixed 512×512 views, and the seven condition definitions. The authoring interface consists of `inspect`, `render`, `insert`, `set_pose`, `remove`, `check`, `probe`, `restore`, and `submit`. A request specifies typed IDs, fields, poses, predicates, and, where applicable, one function and one condition. Batches expand into atomic operations; pose edits never silently repair dependent objects.

Every request consumes one call. An endpoint-level check or full-function probe also consumes one credit from the *same* probe pool. S0/S1 checks use a call without an endpoint/probe credit. The low, standard, and high envelopes are 16/4, 32/8, and 64/16 calls/credits for the entire episode, including rejected proposals and all Variation outputs. Each also caps authoring at 900 s, 96k cumulative input tokens, 12k output tokens, 64 model calls, and 32k tokens per model-call context. Repeated context and images count. Search CPU time and solver work are logged; non-LLM methods report zero model calls. Equal caps do not imply equal realized computation.

An S0 request checks one serialized condition with a 1 s/10,000-query cap. S1 evaluates exact initial contacts/support and one-second settling within 10 s/20,000 geometry queries and 240 physics steps. An endpoint request covers one function, condition, and legal grasp tuple. For an m-edge job it searches at most $6m$ compatible arm-stage endpoints. Each endpoint permits at most eight IK starts, 100 iterations per start, and 4,000 collision queries, subject to a 20 s request-wide wall cap. Held-object geometry and one grasp transform persist within each edge; shared edge boundaries use compatible object and gripper states. This hypothetical endpoint chain establishes neither a connecting path nor dynamic attainability. A public probe tries one full RRTConnect job with seed 0, 2 s and 4,000 queries per segment, the same public IK limits, and a 20 s whole-request cap. Every policy can choose the same named conditions and legal grasp tuples. A public success is not the external certification label.

Checks and probes restore their input snapshot after testing. Mutating actions are transactional, and failed requests consume budget. Restoring a saved state never refunds resources. Submission accepts saved state IDs or a scene JSON; JSON differences expand into charged per-ID edits. A declared invalid artifact retains its violation rather than receiving the score of an unchanged fallback after a refused edit. All policies have this submission path. The wrapper refuses over-cap requests before execution and retains the last completed submission. A backend exception triggers transactional restoration and permits one identical retry if resources remain; both requests are charged. An evaluable earlier submission remains the submitted artifact despite a later exception. Author exhaustion without

any completed submission is a direct submission-contract violation, giving main and joint R with A/B evidence U. Infrastructure that prevents assessment gives U unless separate direct violation evidence establishes R. Every allocated case remains in the denominator, and missing worlds receive no certified-retention credit.

A.5 GRADED CHECKS AND C/R/U EVIDENCE

S0 is the required nominal serialized envelope proxy: IDs/categories, pose/size bounds, bounding support and containment, unsatisfied goals, minimum displacement, table footprints, and forbidden envelope overlap with declared support exceptions. S1 adds exact initial geometry, permissions, support and settling across conditions. E_A and E_B seek compatible endpoint configurations, including held-object geometry and one grasp transform per edge. Full A requires a continuous main-job replay; full B requires its own complete replay from an independent reset. The endpoint chain can exist without a connected or dynamically successful motion.

Each layer stores PASS, direct failure, or UNKNOWN, with condition/predicate causes. Finite IK exhaustion is UNKNOWN. A sampled endpoint collision rules out that candidate, not every allowed pose. A direct geometric impossibility certificate must bind an admitted checker and its complete premises, covering all legal poses, tolerances and grasp alternatives. A certificate name or a capacity deficit does not establish this coverage. Function evidence is C when every condition has a complete witness, R when a sound violation excludes a required function, and U otherwise. Joint C requires all hard checks and every required function; a direct hard/function violation yields joint R; missing evidence without R yields U. Main C/R/U uses H and A only. B evidence remains reportable even when an unrelated hard violation rejects the world.

Predicate scope determines whether a hard violation is unrelated to a function. Evidence records identify the predicate, affected roles or instance IDs, condition, and all jobs sharing the obligation. A complete B witness includes inactive-payload support, required fixture support, and stage-specific stability. A directly failed shared obligation invalidates a claimed witness for that condition; it cannot coexist with complete certification merely because it was stored under H. An authoring-permission violation outside B's functional obligations can instead coexist with valid B evidence. Retention denominators include both types of output.

For joint certification, let $\mathcal{F}$ denote the contract's complete set of required jobs, defined separately from the protected registry J. In Adaptation, $\mathcal{F} = \{A\} \cup J$. In the current Creation and Variation tracks, $\mathcal{F} = \{A, B\}$: Creation has $J = G = \varnothing$ but still requires both newly constructed jobs. For finite conditions $\mathcal{Z}$, certification has the following quantifier structure:

$$H(W) = \text{PASS} \quad \wedge \quad \forall f \in \mathcal{F}\, \forall z \in \mathcal{Z}\, \exists w : \text{ReplayValid}(W, f, z, w).$$

Each condition may use a different witness. A failed grasp, attempted-path collision, heuristic reach estimate, or exhausted planner remains U until direct violation or sound exclusion evidence establishes R.

We retain each layer's complete contingency with main and joint C/R/U. For layer L, report $n_{L=\text{PASS},R}/n_{L=\text{PASS}}$ and $n_{L=\text{PASS},U}/n_{L=\text{PASS}}$ separately, with N/A for zero denominators. Also retain cumulative gates $K_A = (S1 = \text{PASS}) \wedge (E_A = \text{PASS})$ and $K_{AB} = K_A \wedge (E_B = \text{PASS})$; standalone endpoint passes do not imply these conjunctions. Report hard status and main outcomes within K_A, and joint outcomes within K_{AB}. Attribution beyond A endpoints requires the remaining hard checks to pass and uses main-task evidence, not B-only violations or uncertainty. B outcomes remain separately attributable in joint diagnostics. U after any pass is missing evidence, not an established physical false acceptance. All intersections are recorded jointly, never reconstructed from marginals. Because S0 remains binding, certified outputs imply S0 PASS.

External endpoint diagnostic scope. The E_A/E_B labels aggregate endpoint status over the seven conditions. The 4/8 search-unit allowances govern complete-function witness coverage; endpoint runtime is specified separately below. In the scored records, complete certification implies endpoint PASS. The labels, K_A and K_{AB}, remain fixed at $1\times/2\times$. The 107/538 U records quantify missing complete-function coverage within the $S1 \wedge E_A$ gate. The endpoint-only specification below defines an independent geometry-bound diagnostic whose output is not imported into these records.

Independent endpoint-only search. Write $E_f^{\mathrm{ind}}(W, z)$ for a separately collected endpoint diagnostic for function f and condition z. Process A and B separately, each in condition order: nominal, $x+$, $x-$, $y+$, $y-$, low friction, high friction. Reload the submitted snapshot and home robot, zero velocities, apply the condition once to every distinct payload, and perform the declared reset settling. Fixed fixtures do not move, and a staged payload is not perturbed again between edges. Enumerate the Cartesian product of the contract's legal edge-yaw lists in lexicographic contract order, normally four choices per A edge and two for the running-case B. Search each tuple using the public endpoint kernel, not its full-motion probe. Its contract-derived target-pose constructor, candidate order, solver and random-generator versions must be frozen before outputs are evaluated. At each stage, use the incoming arm configuration as the first IK start, followed by seed-0 joint-limit-bounded uniform starts in generated order. No admission witness, public solution, or full-verifier endpoint, seed or trajectory is imported.

Compatible multi-edge endpoint chains. For an m-edge job, search the at most $6m$ arm-stage endpoints in edge order, backtracking over the kernel's candidate poses and IK solutions. Check joint limits, exact geometry, stage-appropriate contact permissions and held-object geometry, with one declared grasp transform within each edge. A candidate chain carries the complete hypothetical object state, arm configuration and gripper mode at every shared edge boundary. The selected release and retraction state supplies the next edge's incoming state; a staged regrasp uses that same selected payload pose. Earlier sorting placements and inactive payload support remain represented throughout the chain. Independently feasible edges with incompatible boundary states cannot be combined. These are hypothetical endpoint states: no connecting path is planned, and the endpoint diagnostic does not certify dynamic grasp, transport or dwell success.

Endpoint stopping and resources. The proposed external endpoint cap is 20 s in total per function–condition, across all tuples, including reset, settling, candidate construction, IK, geometry and compatibility checks. For each stage of a tuple, permit at most eight IK starts, 100 iterations per start and 4,000 collision queries in total, including backtracking. Stop positive search at the first checked complete compatible chain, or when candidates or any applicable allowance are exhausted. Neither a retry nor a tuple change renews the condition clock. This external cap is separately declared: it does not change the public 20 s single-tuple request cap or the full-verifier 60 s function–condition cap, and it is not a task deadline.

Endpoint outcomes and aggregation. A checked compatible chain gives condition-level PASS. Finite IK exhaustion, sampled collisions, incompatible sampled chains, untried alternatives and resource or infrastructure limits give UNKNOWN, not direct failure. Missing artifacts or geometry bindings likewise supply no endpoint refutation. Direct endpoint failure requires an admitted, binding-checked certificate excluding every legal compatible endpoint chain for the named function–condition, including all permitted poses, tolerances and grasp alternatives. Exhausting the finite yaw list does not prove such exclusion. A motion-only refutation does not by itself refute endpoint feasibility. Record every condition; suite-level E_f^{ind} is PASS only when all seven conditions pass, direct failure when any has a sound endpoint exclusion, and UNKNOWN otherwise. A sound endpoint exclusion can support R for its required function–condition. Existing hard, shared-predicate, missing-artifact and infrastructure rules retain their own scopes; an unrelated hard R is not automatically an endpoint failure.

Complete witnesses and endpoint evidence. Let $\mathrm{EndFeas}(W, f, z)$ denote existence of a compatible endpoint chain under the same geometric predicates and tolerances, with path connectivity and dynamic execution requirements relaxed. This relation is not restricted to the finite target poses or IK starts sampled by the endpoint search. Its implementation must admit the stage projection of every valid complete witness: $\exists w : \mathrm{ReplayValid}(W, f, z, w) \Rightarrow \mathrm{EndFeas}(W, f, z)$. The converse does not hold. We do not admit full-witness endpoints as evidence in the independent endpoint-search ledger: a full witness implies endpoint feasibility, not that the independently budgeted search found a chain. Consequently, future full certification can coexist with $E_f^{\mathrm{ind}} = \mathrm{UNKNOWN}$; independent endpoint PASS is not an additional acceptance requirement. A sound endpoint exclusion and a valid full witness with identical bindings would instead signal an evaluator consistency error. The implication for the scored labels is therefore not a completeness guarantee for independent search.

Table 7: Admitted negative-evidence rules. Every item binds its subject and condition; a direct H violation and a failed attempted motion have different scopes.

Rule	Required finite evidence or premises	Result scope
Hard contract	Serialized required-ID/permission/bounds mismatch, or a direct initial support/settling predicate failure with object, condition and tolerance.	H violation; propagate to a job only when its obligations include that predicate.
Shared predicate	Registered predicate-to-job map and the failed condition's direct fact; active/inactive payload and fixture support remain explicit.	Remove the affected A/B condition witness; no unrelated-H exemption.
Missing artifact	Required slot absent after author exhaustion; completed-submission history contains no usable earlier artifact.	Main/joint R; A/B evidence U. Infrastructure alone supplies no R.
Function refutation	Registered checker returns `REFUTED` on bound contract, snapshot, condition and complete alternative-covering premises.	The named required function–condition only; `NO_PROOF` supplies no negative evidence.

Separate endpoint accounting. Log endpoint work under its own world, contract, function and condition bindings, recording kernel and generator versions, tuple order, attempted stages, IK starts and iterations, collision queries, compatibility outcomes, wall/CPU time, stopping cause, and any certificate binding. Include failed attempts and setup work. Do not debit or credit this work to authoring calls, public probe credits, or the full-verifier allowance; report it separately without importing any of its proposed costs into the present resource tables. For a future nested full-verifier extension, retain the saved independent endpoint results without rerunning or backfilling them. Any extension of the endpoint budget must be separately versioned and reported.

A.6 Admitted negative evidence and an analytic example

R evidence records a rule/version, tested output and snapshot, contract, subject objects or roles, condition, affected jobs, predicate tolerance, and evidence source. Table 7 fixes its scope. A universal functional refutation also requires the checker version, full premise payload, and a successful check result. The names `endpoint-set-empty` and `complete-motion-cut` without those bindings provide no negative evidence. Finite RRT failure, unsuccessful IK, or a collision on an attempted trajectory only leaves that attempt without a witness; none proves that all permitted alternatives fail.

An executable example uses a necessary distance bound, rather than planner failure. Let S contain every valid initial payload centre of mass after reset and G every allowed completion centre, expanded by all positional tolerances. A valid trajectory has an absolutely continuous COM path, no unclocked COM translation, and a hard Euclidean COM-speed bound v at all times and in every mode, including grasped motion, release, free flight and regrasp. Completion is required by a *contract* deadline T; v and T include their acceptance tolerances. For axis-aligned boxes, define

$$g_i = \max(0, G_i^{\mathrm{lo}} - S_i^{\mathrm{hi}}, S_i^{\mathrm{lo}} - G_i^{\mathrm{hi}}), \qquad d_{\min}^2 = \sum_{i=1}^{3} g_i^2.$$

For completion time $t_f \le T$, every valid path satisfies $d_{\min} \le \int_0^{t_f} \|\dot{p}(t)\|\, dt \le v t_f \le vT$. Thus $d_{\min}^2 > (vT)^2$ refutes the bound function–condition. Equality or a nonviolated inequality provides no proof of feasibility. This necessary condition covers arbitrary allowed orientations, grasps and reorientations because it relaxes those choices while preserving the complete start/goal sets and global COM bound. Missing premises cannot be replaced by a sampled speed estimate.

For a standalone example, the initial centre is $(0, 0, .78)$ m and the goal box is $[.40, .42] \times [-.01, .01] \times [.78, .80]$ m. Expanding both by 5 mm gives $d_{\min} = .39$ m. A 2 s hard task deadline with .1 s tolerance and global speed bound .1 m/s permits at most .21 m of travel. The exact squared comparison is $1521/10000 > 441/10000$. This protocol arithmetic example is separate from the result records; it assigns no certificate to the running case or any existing output. The 60 s planner cap and $15m$ s rollout cutoff are evaluator resources, *not* such task deadlines.

The standard-library checker verifies canonical contract/snapshot/condition and checker-source hashes against an independently supplied expected binding. It accepts one payload, required job and condition, at most 64 KiB of input, and bounded exact-decimal rational values under a one-second check cap. Missing fields, unsupported scope, mismatched bindings or exhausted checking return `NO_PROOF`; a matching self-declared hash alone does not authenticate a world. An author-side request costs one common check call and no endpoint/probe credit. The final checker is deterministic and separate from positive motion search.

A.7 Finite verification and nested budget extension

The seven conditions are nominal, common ± 5 mm world-x/y payload offsets, and friction multipliers .9/1.1. Each reset applies its condition once to all distinct payloads, including the inactive job's payload. Fixed fixtures do not move. A reused staged payload is not perturbed again at the intermediate step. These conditions are a finite suite, not independent Cartesian perturbations of every object.

For each function and condition, enumerate legal edge-yaw tuples in contract order. Main A normally has four payload-relative choices per edge; the running B has two world-relative choices. Each tuple receives a whole direct-interpolation attempt, then whole RRTConnect attempts with seeds 17 and 29. Every attempt reloads the exact snapshot and home robot, zeros velocities, applies the condition, and settles. RRT segments receive 5 s and 10,000 queries; direct interpolation checks joint steps no larger than .01 rad with the same query limit. IK permits 16 starts and 200 iterations per start. A 60 s function–condition cap includes IK, planning, settling and replay. Untried candidates supply no witness. This bounds A+B functional work by 840 s per world, excluding bounded hard checks and I/O; actual work is reported separately.

The $2\times$ analysis reuses exactly the same output identities and retains all $1\times$ evidence. It extends unresolved obligations with cumulative 120 s function–condition caps, 10 s/20,000-query segment limits, and twice the IK allowances. Previously spent work counts toward these cumulative limits; retries do not receive an uncharged fresh budget. Solver state is resumed when available; otherwise unresolved candidates are revisited in the same order with remaining resources. Task tolerances and rollout horizons are unchanged. In the current instance, hard/shared/missing-artifact facts and any admitted deterministic refutation prepass are fixed before positive search. The $2\times$ extension adds positive witnesses only: C→C and R→R persist, U may become C or remain U, and U→R is zero. Longer unsuccessful search cannot create a certificate. A general evidence extension could permit U→R only with a newly admitted negative item and its checker/source/premises; that is a different mode from this table. We report all transitions on fixed outputs. Larger authoring budgets instead produce new output identities.

Independent audits use a different planner or separately authored replay; a separate process running the same planner does not remove shared bias. Uniform samples of $\min(10, n)$ U cases per method–task family receive tenfold planner allowance and at most 15 minutes of expert trajectory authoring; $\min(5, n)$ C and R cases receive separate audits. Recovered/attempted counts and stratum weights summarize recovery under the stated audit budget. U above 20%, method gaps above ten points, or unstable paired effects restrict interpretation to the declared suite until diagnosis. Audit cases are unique recorded output identities: scored slots sharing the same saved output count once. Report coverage and recovery for selected K_A-passing main-U and main-C/B-U outputs, including B endpoint status in the latter group. State zero coverage explicitly; these tags change neither selection probabilities nor weights, and uniform sampling guarantees neither subgroup's coverage. Independent audit changes remain separate from nested-budget transitions.

A.8 Seven policies and simple controls

Strong geometry search is a non-LLM multistart search with support-conditioned continuous proposals and local coordinate refinement. It seeds the source, permitted-object translation and six random layouts, screens up to 256 cheap arithmetic proposals per round, and retains eight by contract residual, edit cost, and stable ID order. Exact S1 and A/B endpoint checks use charged calls and credits. Local steps begin at .05 m and $\pi/8$, halve after a nonimproving cycle, and stop at .00625 m and $\pi/64$. Source/payload coupled proposals still expand into separate charged edits. No private collision solver or uncharged IK ranks candidates. On changed cases, motion-aware search first requests

nominal E_A and E_B on the frozen initial snapshot, each consuming one call and one combined credit. These are the two initialization endpoint queries in the resource model. Candidate testing then uses charged S1 and complete A/B public probes from independent resets, only while individual requests fit the remaining budget. Standard initialization leaves at most six full-probe credits; an A request can consume the last one, leaving B explicitly untested. Stage progress guides search but is not a certification label. The identity no-edit branch instead uses S1, two full probes and submission: four calls/two credits and no endpoint query for motion search.

For proposal index a and public context index b_{ctx}, the proposal seed is $100000b_{\mathrm{ctx}} + 1000s_{\mathrm{eff}} + a$. Both fixed search policies use $s_{\mathrm{eff}} = 0$. For each context and authoring-budget setting, the three scored slots reuse the same output, evaluation, and resource records with explicit reuse links. They are not three independent searches. Public endpoint or probe success is not required for submission: the best encountered candidate is submitted under the declared ranking and stopping rule.

Generic feedback selects actions from the same observations and full raw history without an external requirement-program or dependency scheduler. It can still infer dependencies and check B. The Edit-As-Act-style adapter regresses symbolic A/B goals through declared preconditions/effects, checks applicability, validates effects and repairs failed plans. The MUSE-style adapter retains an explicit requirement program, persistent scene/requirement memory, protection, consistency checks and local repair. The EmbodiedGen-style adapter uses a task-role graph and typed skill routing, with transactional state changes and failure-driven local repair. Every adapter retains its stated source mechanism, access to endpoint/full-task checks, and protection (Noh et al., 2026; Xu et al., 2026; Wang et al., 2026d).

The reference adds explicit predicate–instance/robot dependencies and evidence invalidation to requirement memory. Dependencies also cover affected geometric regions, not only named object IDs and robot parameters. Each evidence item records its tested snapshot and condition; only unaffected public evidence may be reused after an edit. An edit marks affected evidence stale and schedules the corresponding hard/function checks before accepting a checkpoint. Protected regression triggers repair or restoration. The comparison unit is the complete dependency-and-scheduling intervention. Source-to-adapter mappings, changed predicates, and shared-tool expansions distinguish these controlled adapters from native-system reproductions.

Operational schedule. Table 8 makes the state and check-selection differences explicit. Fixed searches test nominal A then B, except that a known protected regression gives B priority; failed attempts advance to the next legal grasp tuple. When additional conditions are requested, equal-priority checks follow dx+, dx−, dy+, dy−, then low/high friction. An adaptive override names the condition and reason in the public action record. Nominal success does not certify other conditions. After a worse candidate, restore the saved incumbent before further mutation when necessary; a saved state can instead be submitted directly.

Each policy reserves submission capacity and stops at its declared local completion, proposal limit, or insufficient resources for another atomic request. It submits the best encountered checkpoint even if public evidence remains unresolved. A three-object translation costs three edit calls; moving a source fixture and compensating its payload costs two. S1 and restore each cost one call; every E_f or P_f also uses one shared credit. Multi-function/condition requests and typed skills expand into these operations, with failed and repeated requests charged. Public-success filtering is not silently imposed on auxiliary position rejection sampling.

Frozen and literal translation are controls. Translation applies the base displacement only to permitted objects, checks bounds without clamping, and retains invalid outcomes. Fixed tables/fixtures remain anchored, so we describe cases as shift-compatible or anchored without claiming exact equivariance. Identity controls measure unnecessary edits and function loss. Auxiliary one-shot, open-loop and legacy static-grid policies remain separate. Main-task-only scheduling retains B inputs/tools but omits mandatory B rechecks; fixed-full-rechecks, no-ledger and no-mandatory-rechecks retain freed resources under the same caps.

Evidence records. A public evidence record stores its call/tool version, tested snapshot and contract, predicate or job, condition, grasp tuple, outcome, dependency fields/region, coverage and current/stale status. Source records are immutable. Staleness concerns the new snapshot; saved-source results remain bound to their original snapshot.

Table 8: Policy-to-operation mapping. E_f and P_f each address one function and condition; all exact checks and repairs use the charged common interface.

Policy	Initialized state / next objective	Checks, B protection, recovery and incumbent
Geometry search	Fixed-seed starts and coordinate refinement.	Candidate S1, E_A, E_B; rank hard violations, missing endpoints, diagnostics, edit cost and ID; restore a better saved layout as needed.
Motion search	Initial nominal E_A/E_B, then the same proposal family.	Candidate S1, P_A, P_B from remaining credits; prioritize known B regression; rank hard violations, missing full-job successes, stage progress, cost and ID.
Generic feedback	Raw observations/history; model names the unresolved obligation.	Choose common checks/edits from feedback; retain B tools and repair known regression. Use observed hard/A/B evidence to select a checkpoint; no external ledger is added.
Edit-As-Act-style	A/B goals and protected commitments; regress goals through preconditions/effects.	Check the earliest unmet precondition, execute an applicable operator, verify effects/protection, then repair or regress again; retain full probes.
MUSE-style	Requirement program plus persistent scene/requirement memory.	Select a violated/unknown clause; edit, run consistency/protection checks and update memory. Repair or restore B regression; rank program satisfaction and edit cost.
EmbodiedGen-style	Task-role graph and typed skill/failure-routing state.	Route the next unsatisfied role to charged skills; validate deltas and repair the implicated subgraph, retaining protected-role and full-function checks.
Reference	Requirement memory with region/instance/robot dependencies.	Invalidate affected snapshot/condition evidence; recheck affected hard predicates/jobs, prioritize stale B, reuse only unaffected evidence and repair/restore regression.

Regions and conservative invalidation. Regions are axis-aligned unions of supplied object/support/goal boxes, inflated only by published tolerances and condition offsets. An edit box covers the old and new bounds. Named dependency changes, region overlap and shared-predicate edges invalidate evidence. These boxes provide neither a swept robot volume nor an independence proof: uncertain coverage conservatively invalidates the affected jobs, both A/B if its scope is unknown. Current endpoint/full-probe responses promise no complete interaction read set, so physical edits stale their results by default. Only complete, unchanged pure-metadata read sets permit reuse. ID joins and box arithmetic consume timed CPU/state tokens; exact geometry, IK and replay require charged common tools.

Queue and checkpoints. The local gate is fresh nominal S1 and public full-A/B success, not final seven-condition certification. Queue missing hard checks, previously passing but stale nominal B, missing nominal A, then remaining B. Explicit extra conditions follow. Ties use condition order (nominal, $x+$, $x-$, $y+$, $y-$, low/high friction), function, sorted predicate/instance IDs, grasp-tuple index, then call index. Failed/UNKNOWN E/P attempts advance only to untried legal tuples; they never produce R. Hard violation or incomplete S1 stops candidate probing and returns its diagnostic for repair; incomplete S1 is not R. Saved checkpoints minimize, lexicographically, hard violations, unresolved hard checks, missing fresh nominal B, missing fresh nominal A, edit cost and creation index. Missing/stale evidence cannot pass a gate. At the cap, submit the best checkpoint even if unresolved; a restore before another mutation is charged. Admission witnesses are never public checkpoint evidence.

```
committed edit:  invalidate; build stable missing-obligation queue
while affordable, reserving submit:  call next shared tool; log result
hard violation/incomplete S1:  stop; E/P UNKNOWN: next tuple, never R
compare checkpoint; then submit, propose repair, or charged restore
```

Variant	Retained state, required checks and checkpoint objective
Main-only	Ledger/invalidation and B tools retained; target/rank/gate use H+A. After edits check S1 and A; no automatic B check, B-first repair or B veto.
Full rechecks	Ledger and H+A+B objective retained; each edit triggers charged nominal S1, full A, full B, regardless of cache. Stop between requests if capped; this is two nominal probes, not fourteen function–condition checks.
No ledger	H+A+B objective and raw logs retained; no consolidated ledger, graph or cross-snapshot reuse. After S1, the model selects checks from history; local completion still needs fresh active-snapshot A+B evidence.
No mandatory rechecks	Ledger, invalidation and H+A+B ranking retained; after S1, the model chooses functional checks. Remove automatic recheck/B-first repair; stale evidence remains missing. Unlike main-only, B still affects ranking/gate.

All variants preserve inputs, final auditing and caps. Freed requests remain within the shared caps and may fund proposals or optional A/B checks. Common snapshot/call logs remain even without the policy ledger.

Priced bin-edit example. Suppose an already paid public checkpoint W_c has nominal S1/A/B successes. Moving only `bin_A` near B invalidates A/B; no final labels are consulted. If S1 passes and B returns UNKNOWN, a permitted restore branch is:

Atomic operation	Calls	Credits
`set_pose(bin_A, pose)`	1	0
Metadata invalidation / queue (timed)	0	0
`check(S1, predicates, nominal)`	1	0
`probe(B, nominal, yaw)`	1	1
`restore(`W_c`)`	1	0
`submit(`W_c`)`	1	0
Incremental total	5	1

Check/probe restores its input automatically. Directly submitting saved W_c omits the explicit restore (4 calls/1 credit). Trying another tuple costs another call/credit; no A probe is bundled. Prior checkpoint costs remain in the episode ledger. The incremental charges for this protocol branch are shown above.

Common scripted-response contrast. Both policies start from the same already-paid W_c, contract, and sufficient remaining budget, then receive the same permitted `bin_A` pose proposal. For edited snapshot W_e, inject nominal S1 PASS, public full-A SUCCESS, and public full-B UNKNOWN. A common response stub is keyed by snapshot, tool, function, condition, and legal grasp tuple, not request order; it reveals each response only when requested. The physical edit stales affected S1 and A/B records for W_e, while earlier records remain bound to W_c.

The columns show two permitted continuations; parentheses are incremental calls/credits after the paid checkpoint.

Reference: early restore	Fixed full rechecks
`set_pose(bin_A, pose)` (1/0)	`set_pose(bin_A, pose)` (1/0)
Invalidate / queue (0/0; timed)	Invalidate / queue (0/0; timed)
`check(S1, predicates, nominal)` (1/0)	`check(S1, predicates, nominal)` (1/0)
`probe(B, nominal, yaw)` (1/1)	`probe(A, nominal, yaw)` (1/1)
—	`probe(B, nominal, yaw)` (1/1)
`restore(`W_c`)` (1/0)	`restore(`W_c`)` (1/0)
`submit(`W_c`)` (1/0)	`submit(`W_c`)` (1/0)
Incremental total: 5 calls/1 credit	Incremental total: 6 calls/2 credits

The reference uses the permitted immediate-restore branch after B's fresh UNKNOWN; full rechecks request A before B. B is unresolved, not refuted, so both submit W_c. Automatic check/probe restoration returns W_e, not W_c; explicit restoration is charged. Direct submission of the saved W_c

omits one restore call in either column. If both policies instead make the same atomic requests, their charges tie regardless of order. W_e's physical checks use fresh queries. This scripted contrast specifies scheduling and accounting; Appendix A.9 separately defines the resource means reported for Table 2.

A.9 Allocation, estimands and retention

The formal suite has 48 task-asset bundles and five contexts per bundle. Each of six task-by-split strata has eight bundles and 40 contexts: four identity and 36 changed contexts, with twelve robot, twelve workspace and twelve task changes. Identity occupies the first slot of bundles 1–4; the remaining slots receive the three change types cyclically in bundle/slot order, with a fixed task-specific phase shared across splits. Totals are 216 changed and 24 identity contexts. Three scored seed slots give 648 and 72 observations per policy. Reused deterministic outputs do not create independent trials. A separate eight-bundle, four-condition development pilot has 32 pairs and contributes no formal outcomes.

Every descendant of a declared root identifier remains in one reporting bundle. Robot, table and recipe infrastructure are shared inputs. In a geometry-bound instantiation, repeated physical roots must merge clusters. All method, seed, budget, track and verifier descendants remain paired within their bundle; the current demand-regime labels and geometric holdout requirements are distinguished below.

Primary results average three seed slots within a changed context and then weight the 216 changed contexts equally. For stratum h, bundle b, changed-context set C_{hb}^{+}, and context score $\bar{Y}_{hbc}$,

$$\widehat{\mu} = \frac{1}{6}\sum_{h=1}^{6} \frac{\sum_b \sum_{c\in C_{hb}^{+}} \bar{Y}_{hbc}}{\sum_b |C_{hb}^{+}|}.$$

Four bundles per stratum have four changed contexts and four have five. Equal weighting of their changed-bundle means would alter this target. Paired contrasts use aligned context differences in the same expression. Identity and intervention-specific results have their own denominators. The primary contrast is reference minus MUSE-style standard-budget joint CC; all policy outcomes remain visible. Estimates remain descriptive, without significance tests. The primary contrast alone receives the whole-bundle resampling summary below. Any other optional resampling must likewise copy whole bundles within strata and recompute complete conditional/eligible denominators, rather than resampling seeds or method rows independently.

Whole-bundle resampling. For this contrast, retain changed Adaptation contexts at the standard authoring budget and the $1\times$ evaluator. Let $\bar{D}_{hbc} = \bar{Y}_{hbc}^{\mathrm{ref}} - \bar{Y}_{hbc}^{\mathrm{MUSE}}$, where each $\bar{Y}$ averages the three aligned scored-slot indicators of joint C. R and U both contribute zero to this certification indicator, without changing their distinct evidence meanings. No initial-state, endpoint, main-C or other outcome-based pass filter is applied.

Within each of the six task-by-split strata, sample eight of its eight complete bundles uniformly with replacement. Copy all dependent descendants together, retaining method pairing, the three scored slots and any reuse links. Identity contexts and other tracks, authoring budgets or verifier levels do not enter this contrast. For draw r, let $M_{hb}^{(r)}$ be the number of copies of bundle b, with $\sum_{b=1}^{8} M_{hb}^{(r)} = 8$. Recompute the original context-weighted contrast in percentage points:

$$\widehat{\Delta}_{\mathrm{pp}}^{*(r)} = \frac{100}{6}\sum_{h=1}^{6} \frac{\sum_{b=1}^{8} M_{hb}^{(r)} \sum_{c\in C_{hb}^{+}} \bar{D}_{hbc}}{\sum_{b=1}^{8} M_{hb}^{(r)} |C_{hb}^{+}|}.$$

Thus repeated bundles retain their multiplicities, each stratum's changed-context denominator is recomputed, and the six stratum weights remain $1/6$. Neither equal weighting of changed-bundle means nor pooling all sampled contexts across strata replaces this expression.

Applying this resampling to the frozen records with 20,000 draws and seed 20260926 gives the 3.24-percentage-point contrast and a 95% percentile interval of $[1.39, 5.30]$ percentage points, using the 2.5th and 97.5th percentiles of the resampled contrasts. The interval summarizes sensitivity to bundle allocation under the fixed labels and numerical recipe. It is a descriptive resampling summary; the evaluator-specific interpretation limits in Appendix A.7 continue to apply.

Table 9: Current context-tag allocation. Counts are contexts, before the three scored seed slots. Archetypes set demand parameters; they are not a per-context inventory of instantiated physical changes.

Tag	Archetypes and counts	Total
Robot	Shift-compatible (18), anchored conflict (18), aperture change (36)	72
Workspace	Weak geometry (24), motion bottleneck (24), anchored conflict (24)	72
Task	Approach change (24), continuous uncertainty (24), verifier-limited (24)	72
Identity	Zero-change control (24)	24

For general registries, protected satisfaction averages $|J|^{-1}\sum_{j\in J}\mathbf{1}[j=\mathrm{C}]$ within episodes, and regression retention replaces J by the pre-edit-certified subset G. Empty sets are N/A with eligible counts; empty required strata do not silently change aggregation weights. The primary suite has $J=G=\{B\}$, so RR is the fraction with B certified. Outputs rejected for unrelated hard conditions stay in this denominator. B=U receives no certified-retention credit but is distinguished from an established B violation.

Demand profiles and allocation. Each context in the archetype–demand design has a task tag, split tag, bundle/root identifier, intervention tag, and a five-component demand vector for geometry, endpoints, motion, semantic requirements, and protection. The components are dimensionless demand parameters. The T1/T2/T3 job semantics are defined above; their indices determine demand offsets and allocation phases; separate mesh catalogs and task graphs are not specified for every context. Table 9 makes the implemented tag allocation explicit. Archetypes also select evidence rules, such as extra witness-discovery demand in verifier-limited cases and shared-support failure semantics in weak-geometry cases; these clauses do not supply the missing geometric episode specifications.

Scoring rule and resource accounting. The scoring rule uses fixed capacity parameters. In geometry/endpoint/motion/semantics/protection order, reference and MUSE profiles are $(.85,.86,.82,.85,.92)$ and $(.84,.86,.80,.86,.93)$. Changed-Adaptation capacity is $a_\ell = p_{m\ell} + b + .14(u_\ell - .5) + \epsilon_\ell$, with offsets $b=-.14/0/.06$ for caps $(16,4)/(32,8)/(64,16)$. Only reference protection on `anchored_conflict` receives $\epsilon=.05$; the offset is a specified scoring parameter. Deficits are $\Delta_A=\max(d_e-a_e, d_s-a_s, d_m-a_m)$ and $\Delta_B=d_p-a_p$. Per-condition discovery cost is $q_z=\lceil 1+5\max(0,\Delta+.15)+v_z\rceil$, plus two on `verifier_limited`; u,v are fixed paired hash values. When $\Delta>.15$, one hash-selected condition has no witness. Finite costs at most four/eight supply $1\times/2\times$ coverage; shared violations also remove affected witnesses. Missing coverage is U absent supported direct evidence. The hard-failure rule applies when $a_g < d_g - .035$. For ordinary changed cases, $e=\frac{1}{5}\sum_\ell d_\ell$ determines credits $P=\min(Q,\max(2,\lceil Q(.45+.45e)\rceil))$ and calls $N=\min(T,5+h_m+2P+\lceil 3e\rceil)$, for caps (T,Q); $h_{\mathrm{ref}}=10$, $h_{\mathrm{MUSE}}=9$. These resource formulas define the accounting; full-recheck instead uses $P=Q$.

For the existing ID/T1/B01/C05 reference slot 0 at standard budget, $d_p=.860997$, $u_p=.397172$ give $a_p=.955604$ and $\Delta_B=-.094607$. With $v_{\mathrm{nominal}}=.509371$, $q_{\mathrm{nominal}}=\lceil 1.786334\rceil=2$. Its $e=.628777$ gives six credits and 29 calls: two endpoints and four full probes. All stored A/B condition costs are two or three, and H/shared predicates pass, so this $1\times$ slot receives joint C. This example traces the scoring arithmetic; operational workflow behavior is specified separately in Appendix A.8.

Split labels and asset-family scope. For a non-identity context c and demand component ℓ, the implemented rule is

$$d_{c\ell}=\mathrm{clip}_{[0,1]}\Big(d^{(0)}_{a(c),\ell}+0.035\,t_c+0.05\,\mathbf{1}[s_c=\text{joint-OOD}]+0.16(u_{c\ell}-0.5)\Big), \tag{2}$$

where $a(c)$ is the archetype, $t_c\in\{0,1,2\}$ the task index, and $u_{c\ell}$ a deterministic hash-derived value using the declared seed. Identity demands are zero. Both splits use the same archetype set and allocation; the split tag also enters the context identifier used by the hash. The 48 `asset_root_id` values identify paired reporting bundles. They are not asset-family names or mesh references: the

Table 10: Field-level examples for a proposed geometric instantiation. Only the mount example has an existing independent geometric specification; the other rows specify interface fields and do not enter the scored records.

Family	Single changed field	Preserved fields / meaning
Robot	Base position $(0, 0, .75) \to (-.20, 0, .75)$ m	Object poses, robot geometry/limits, task and edit permissions stay fixed.
Workspace	Proposed A tray/bin centre region: intersect $x \in [.25, .78]$ with $x \leq .65$ m	Physical geometry and mutable IDs/operations stay fixed; this is an initial-state constraint, not an obstacle.
Task	Proposed A grasp-yaw set: $\{0, \pi/2, \pi, 3\pi/2\} \to \{0, \pi\}$	Robot, scene, hard initial-state rules, edit permissions and B task stay fixed.

current records contain no bound `asset_family_id` catalog or requirement-combination holdout list. Thus the current ID/joint-OOD contrast describes these two demand regimes. Geometric class-held-out evaluation requires a separate versioned manifest, with development/ID family classes and held-out family/requirement combinations declared before policy outputs. New root names alone do not establish that split.

Workspace is a typed contract field. A submitted-layout region constrains initial object poses and is checked by H; an edit permission specifies mutable IDs, attributes, or operations; a physical obstacle changes geometry used by endpoint and motion checks. These are different interventions. A legal-region boundary is not a wall and does not, by itself, forbid a robot path from crossing it. The current workspace-tagged records contain demand vectors rather than a per-context old/new region, permission, or obstacle specification. The geometric running case independently specifies a mount change; it does not instantiate all 240 numerical contexts.

Versioned geometric instantiation. Table 10 gives one concrete field-level example per family for a proposed geometry-instantiation version. The mount values already occur in the running case and in 36 robot-tagged displacement fields; the region and yaw changes are proposed examples and do not enter the scored records. Each instance must record old/new values, changed JSON pointers, unchanged-field hashes, geometry-family/source bindings, and the common admission evidence before any policy is evaluated. All descendants remain in their declared bundle; admission or replacement cannot depend on a preferred policy's score.

The example region excludes the old bin centre at $x = .72$ m and includes the joint-candidate tray/bin centres at $x = .26/.49$ m; this is coordinate arithmetic, not a complete A/B witness. A physical-obstacle subtype would instead require explicit obstacle geometry while leaving regions and permissions unchanged. Aperture variants likewise need explicit old/new limits and a compatible B contract: the running case's 0.08 m approach opening cannot simply be retained under a smaller opening cap. These geometry bindings are not inferred from the archetype labels or from the observed table ordering.

A.10 Auxiliary Creation and Valid Variation

Creation starts with the robot and table and constructs the approved A/B objects. Both jobs are newly required, so it reports joint C/R/U without a retention claim. Variation starts from an admitted new-contract reference world and requests four jointly valid layouts under one complete-episode budget. Motion search, MUSE-style, EmbodiedGen-style and reference provide the auxiliary comparisons; Variation also includes independent regeneration and position rejection sampling. Regeneration sees the original source/contract at each invocation without previous candidates or feedback. Position rejection sampling filters role-position descriptors only for separation from earlier retained candidates; it does not apply a physical-feasibility gate or consult final labels. Public checks and probes remain callable through the same charged API, but PASS is not a prerequisite for submission. All requested outputs remain in the yield denominator; external H+A+B evidence determines validity.

Variation distance uses initial role-matched A-object centres in the current robot-base frame, divided by $L = 1$ m. It is the maximum role displacement, minimized over declared interchangeable roles. Fixed B coordinates contribute zero; texture, camera, yaw and planner-dependent clearance are

Table 11: Frozen-source and literal-translation controls under the changed contract. Out-of-bounds translations remain in the denominator. Rates are percentages.

Policy	A CC	Joint C	R	U	RR
Frozen source	12.0	12.0	0.0	88.0	100.0
Literal translation	25.9	19.9	2.3	77.8	91.7

Table 12: Identity diagnostic on 24 unchanged contexts (72 scored seed slots). Joint completion and B retention reveal unnecessary-edit regressions. Rates are percentages.

Policy	A CC	Joint C	R	U	RR
Strong geometry search	100.0	100.0	0.0	0.0	100.0
Motion-aware search	100.0	100.0	0.0	0.0	100.0
Generic feedback	100.0	100.0	0.0	0.0	100.0
Edit-As-Act-style	100.0	100.0	0.0	0.0	100.0
MUSE-style	100.0	100.0	0.0	0.0	100.0
EmbodiedGen-style	100.0	100.0	0.0	0.0	100.0
Dependency-aware (Ours)	100.0	100.0	0.0	0.0	100.0
Frozen source	100.0	100.0	0.0	0.0	100.0
Literal translation	100.0	100.0	0.0	0.0	100.0
Main-task-only scheduling	100.0	100.0	0.0	0.0	100.0
Fixed full rechecks	100.0	100.0	0.0	0.0	100.0
Without structured ledger	100.0	100.0	0.0	0.0	100.0
Without mandatory rechecks	100.0	100.0	0.0	0.0	100.0

excluded. At $\epsilon = .10$, enumerate all 16 subsets of four outputs and select the largest certified pairwise-separated subset of size M. Valid yield counts every certified output divided by four; unique-valid yield is $M/4$; completion requires $M = 4$. The distinctness measure is task-position separation. Four valid copies A, A, B, B yield $M = 2$ when the two positions are separated. Candidate screening and retries share the total call/probe budget; four A+B screenings already consume the standard eight endpoint/probe credits.

B Detailed Results and Diagnostic Counts

The primary comparison and ablations appear in main-text Tables 1 and 2. Table 4 summarizes reference-layer counts; full paired and other diagnostics follow here. Backbone controls (Table 3) are specified separately in Appendix C. All rates use the frozen allocation and definitions in Appendix A.9. Changed-context comparisons contain 216 contexts and 648 scored slots; identity controls are separate. Joint C/R/U exhaust every requested submission, and retention eligibility never depends on an author's success. Repeated deterministic outputs remain linked to the same output identity.

B.1 Auxiliary construction and position variation

Creation uses all 240 contexts, including the 24 identity contexts, with three scored slots per context: 720 slots per policy, rather than the 648 changed-only Adaptation slots. Table 20 reports main-C and joint C/R/U counts against this denominator. Both functions are newly required, so there is no retention denominator and RR is N/A. Variation retains the fixed four-output denominator even when a budget or proposal cap prevents submission; position rejection sampling filters only descriptor separation. Independent acceptance determines valid yield, and an exact subset calculation determines distinct valid yield.

Table 13: Frozen and edited worlds evaluated under the identical changed contract. Frozen/edited joint CC are percentages of all 648 scored slots; the difference is in percentage points. The final columns retain only the frozen joint-C row: edited C/R/U counts sum to its distinct denominator $n_{F=C}$. They distinguish retained certification, established rejection and unresolved outcomes among previously certified frozen worlds.

Policy	Frozen CC	Edited CC	Δ pp	$C \to C$	$C \to R$	$C \to U$	$n_{F=C}$
Strong geometry search	12.0	10.6	-1.4	3	3	72	78
Motion-aware search	12.0	54.6	+42.6	27	9	42	78
Generic feedback	12.0	18.8	+6.8	10	15	53	78
Edit-As-Act-style	12.0	56.6	+44.6	37	13	28	78
MUSE-style	12.0	63.3	+51.2	35	14	29	78
EmbodiedGen-style	12.0	62.0	+50.0	41	8	29	78
Dependency-aware (Ours)	12.0	66.5	+54.5	39	13	26	78
Frozen source	12.0	12.0	+0.0	78	0	0	78
Literal translation	12.0	19.9	+7.9	75	0	3	78
Main-task-only scheduling	12.0	25.3	+13.3	15	13	50	78
Fixed full rechecks	12.0	56.0	+44.0	33	13	32	78
Without structured ledger	12.0	54.9	+42.9	30	13	35	78
Without mandatory rechecks	12.0	46.5	+34.4	31	13	34	78

Table 14: Main-by-B contingency for the seven primary policies and four distinct reference ablations. Each policy's 3-by-3 matrix has rows main (H+A) and columns B, both ordered C/R/U. The three blocks list the main C, R and U rows; paired headers give (main,B) and cells are counts. All nine cells sum to 648. The main-C/B-C cell equals joint certification.

Policy	Main C (H+A)			Main R (H+A)			Main U (H+A)		
	(C,C)	(C,R)	(C,U)	(R,C)	(R,R)	(R,U)	(U,C)	(U,R)	(U,U)
Strong geometry search	69	0	0	0	6	0	303	0	270
Motion-aware search	354	0	18	0	30	0	243	0	3
Generic feedback	122	0	132	0	68	0	243	0	83
Edit-As-Act-style	367	0	112	0	46	0	87	0	36
MUSE-style	410	0	0	0	54	0	184	0	0
EmbodiedGen-style	402	0	70	0	28	0	139	0	9
Dependency-aware (Ours)	431	0	0	0	51	0	166	0	0
Main-task-only scheduling	164	0	328	0	51	0	43	0	62
Fixed full rechecks	363	0	0	0	51	0	234	0	0
Without structured ledger	356	0	22	0	51	0	216	0	3
Without mandatory rechecks	301	0	170	0	51	0	81	0	45

Table 15: B outcomes conditional on certified main (H+A), using the main-C row of the preceding contingency. The R/U counts are its B-R/B-U cells; R/U percentages divide those counts by the displayed main-C denominator, not by 648. A zero conditional denominator is shown as a dash.

Policy	Main-C n	R count	U count	R (%)	U (%)
Strong geometry search	69	0	0	0.0	0.0
Motion-aware search	372	0	18	0.0	4.8
Generic feedback	254	0	132	0.0	52.0
Edit-As-Act-style	479	0	112	0.0	23.4
MUSE-style	410	0	0	0.0	0.0
EmbodiedGen-style	472	0	70	0.0	14.8
Dependency-aware (Ours)	431	0	0	0.0	0.0
Main-task-only scheduling	492	0	328	0.0	66.7
Fixed full rechecks	363	0	0	0.0	0.0
Without structured ledger	378	0	22	0.0	5.8
Without mandatory rechecks	471	0	170	0.0	36.1

Table 16: Separate main and joint diagnostics: $K_A = S1 \wedge E_A$ is crossed with main (H+A) outcomes, whereas $K_{AB} = S1 \wedge E_A \wedge E_B$ is crossed with joint (H+A+B) outcomes. Each block reports its own PASS count and R/U percentages conditional on that count. R is established rejection and U is unresolved; compatible endpoints alone do not certify continuous execution. The seven primary policies and four reference ablations are shown separately.

Policy	K_A against main (H+A)			K_{AB} against joint (H+A+B)		
	PASS n	R (%)	U (%)	PASS n	R (%)	U (%)
Strong geometry search	471	0.0	85.4	345	0.0	80.0
Motion-aware search	444	0.0	16.2	426	0.0	16.9
Generic feedback	430	0.0	40.9	307	0.0	60.3
Edit-As-Act-style	586	0.0	18.3	482	0.0	23.9
MUSE-style	544	0.0	24.6	544	0.0	24.6
EmbodiedGen-style	543	0.0	13.1	476	0.0	15.5
Dependency-aware (Ours)	538	0.0	19.9	538	0.0	19.9
Main-task-only scheduling	538	0.0	8.6	412	0.0	60.2
Fixed full rechecks	538	0.0	32.5	538	0.0	32.5
Without structured ledger	465	0.0	18.7	444	0.0	19.8
Without mandatory rechecks	538	0.0	12.5	415	0.0	27.5

Table 17: Nested verifier extension on identical submitted outputs. Joint completion and RR are percentages; $U \to C$ and $U \to R$ are scored-slot counts, not distinct-output counts. Previously resolved outcomes persist.

Policy	CC 1×	CC 2×	RR 1×	RR 2×	$U \to C$	$U \to R$
Strong geometry search	10.6	10.6	57.4	68.1	0	0
Motion-aware search	54.6	63.4	92.1	92.6	57	0
Generic feedback	18.8	29.2	56.3	65.0	67	0
Edit-As-Act-style	56.6	66.7	70.1	75.6	65	0
MUSE-style	63.3	73.3	91.7	91.7	65	0
EmbodiedGen-style	62.0	69.9	83.5	85.3	51	0
Dependency-aware (Ours)	66.5	75.8	92.1	92.1	60	0
Frozen source	12.0	12.0	100.0	100.0	0	0
Literal translation	19.9	19.9	91.7	91.7	0	0
Main-task-only scheduling	25.3	30.9	31.9	37.5	36	0
Fixed full rechecks	56.0	66.5	92.1	92.1	68	0
Without structured ledger	54.9	63.7	88.3	88.9	57	0
Without mandatory rechecks	46.5	56.5	59.0	67.6	65	0

Table 18: Mean authoring resource use per scored slot at the standard cap. Calls, probe credits, endpoint queries, full-function probes and model calls describe distinct accounting categories. External evaluator use is excluded.

Policy	Calls	Credits	Endpoint	Full	Model
Strong geometry search	20.3	6.4	6.4	0.0	0.0
Motion-aware search	20.3	6.4	2.0	4.4	0.0
Generic feedback	26.3	6.4	2.0	4.4	6.9
Edit-As-Act-style	27.3	6.4	3.0	3.4	7.4
MUSE-style	29.3	6.4	2.0	4.4	7.5
EmbodiedGen-style	27.3	6.4	2.0	4.4	7.4
Dependency-aware (Ours)	30.3	6.4	2.0	4.4	7.9

Table 19: Authoring-budget diagnostic with separately generated outputs at each budget. Low, standard and high caps are 16/4, 32/8 and 64/16 calls/probe credits, respectively. Joint completion and retention are percentages; calls and credits are means.

Policy	Budget	Joint CC	RR	Calls	Credits
Motion-aware search	low	19.9	67.1	14.4	3.5
Motion-aware search	standard	54.6	92.1	20.3	6.4
Motion-aware search	high	69.9	99.1	32.2	12.4
MUSE-style	low	18.1	73.3	16.0	3.5
MUSE-style	standard	63.3	91.7	29.3	6.4
MUSE-style	high	84.0	95.1	41.2	12.4
Dependency-aware (Ours)	low	23.1	78.9	16.0	3.5
Dependency-aware (Ours)	standard	66.5	92.1	30.3	6.4
Dependency-aware (Ours)	high	85.5	95.1	42.2	12.4

Table 20: Auxiliary Creation uses all 240 contexts, including the 24 identity contexts, with three scored slots per context (720 per policy). A CC measures main (H+A) certification; joint C additionally requires B. Rates are percentages; RR is N/A. In row order, counts written as main C; joint (C,R,U) are 426; $(390, 57, 273)$, 476; $(476, 68, 176)$, 529; $(460, 47, 213)$, and 485; $(485, 68, 167)$. Each joint partition exhausts all 720 requested slots.

Policy	A CC	Joint C	R	U
Motion-aware search	59.2	54.2	7.9	37.9
MUSE-style	66.1	66.1	9.4	24.4
EmbodiedGen-style	73.5	63.9	6.5	29.6
Dependency-aware (Ours)	67.4	67.4	9.4	23.2

Table 21: Auxiliary Valid Variation uses 240 contexts, including 24 identity controls. Three scored slots per context give 720 batches per policy, with four requested outputs each (2,880 output slots). VY divides the number of certified output slots by 2,880; UY uses the summed sizes of the largest position-distinct certified subsets within batches over the same denominator. Batch CC divides the number of batches with four distinct certified worlds by 720. Ours has 14 complete batches and 906 unique-valid output slots. Missing outputs remain in the denominator. Rates are percentages.

Policy	VY	UY	Batch CC
Motion-aware search	42.5	28.9	1.2
MUSE-style	43.7	29.8	1.9
EmbodiedGen-style	38.9	26.7	1.9
Dependency-aware (Ours)	47.6	31.5	1.9
Independent regeneration	14.2	10.4	0.0
Position rejection sampling	12.3	12.3	0.0

C MODEL-LABELLED WORKFLOW COMPARISONS

Comparison unit and model identities. Each selected model identity labels two workflow conditions: generic feedback (Base) and the dependency-aware reference wrapper (Ours). The two conditions use the same 216 changed contexts and three scored slots per context, observation/interface definition, and 32-call/eight-credit caps. Within each pair, the backbone label and context-slot assignment are fixed while the workflow condition varies. Registry order is fixed before outcomes are inspected. Table 22 identifies the selected public endpoints or weights. The roster spans different tiers and sizes, including a Flash model and a 27B open-weight model; each selected identity defines a paired Base/Ours row group.

Table 22: Selected public model identities and source links (checked 26 September 2026).

Model	Public identifier	Version note / source
GPT-6 Astra	`gpt-6-astra`	Current flagship ID; dated snapshot unverified. Official source.
Claude Sonnet 5	`claude-sonnet-5`	Pinned model ID; serving infrastructure can vary. Official source.
Gemini 3.8 Flash	`gemini-3.8-flash`	Stable Flash ID; not comparable by tier to Pro. Official source.
Qwen3.8-27B	`Qwen/Qwen3.8-27B`	Official weight repository; revision pinned below. Official source.

Qwen weights are identified at revision `1d4bf0f2ff6012fd82039f2fa52739d0dd7c60c0`.

Model-labelled scoring specification. The model-labelled controls use a separate frozen scoring rule from the seven-policy results. We analyze the two data families separately. Seed 20260926 and SHA-256 keyed uniforms bind four preselected anonymous model slots to contexts and scored slots. Every model slot has the same priors: ability $a_i \sim \mathcal{U}[-.08, .08)$ and wrapper interaction $r_i \sim \mathcal{U}[-1, 1)$. Shared context-slot noise is $\eta \sim \mathcal{U}[-.05, .05)$. These identical priors govern the anonymous slots, while the workflow condition varies within each paired slot.

Let g, e, m, s, p denote the existing geometry, endpoint, motion, semantic and protection demands, each clipped to $[0, 1]$. Define

$$d_A = .25g + .25e + .35m + .15s,$$
$$d_B = .35g + .20e + .15m + .30p, \qquad d_H = .60g + .40e.$$

Base receives declared credits $q = 4 + \lfloor 5u_q \rfloor$ and calls $t = 18 + \lfloor 10u_t \rfloor + \lfloor 2d_A \rfloor$. Ours receives $q' = \min(8, q + 1)$ and $t' = \min(32, t + 2 + \lfloor 3u_o \rfloor)$; calls and credits follow the scoring rule. With $\Delta q = q' - q$ and all probabilities clipped to $[.02, .98]$, the coverage construction is

$$P_A = .45 + .05q - .25d_A + a_i + \eta,$$
$$P_B = .40 + .05q - .30d_B + .5a_i + \eta,$$
$$P'_A = P_A + .015\Delta q - .06 - .04d_A + .04r_i,$$
$$P'_B = P_B + .020\Delta q + .08 + .06d_B + .04r_i.$$

For the displayed pre-clipping expressions, write $\Delta P_f = P'_f - P_f$. The declared domains give $q \in \{4, \ldots, 8\}$, $\Delta q \in \{0, 1\}$, $d_A, d_B \in [0, 1]$ and $r_i \in [-1, 1)$, hence

$$\Delta P_A = .015\Delta q - .06 - .04d_A + .04r_i < .015 - .06 + .04 = -.005 < 0,$$
$$\Delta P_B = .020\Delta q + .08 + .06d_B + .04r_i \geq .08 - .04 = .04 > 0.$$

With $c(x) = \mathrm{clip}(x, .02, .98)$, monotone clipping gives $c(P'_A) \leq c(P_A)$ and $c(P'_B) \geq c(P_B)$. Each pair shares its coverage uniforms, so the A-coverage indicator is nonincreasing and the B-coverage indicator is nondecreasing; discrete indicators may tie. The coverage rule yields these

A/B directions, while joint CC reflects their overlap under hard PASS. Hard-failure probability is $\mathrm{clip}(.04 + .07d_H - .05a_i, .01, .20)$, identical within each pair. The hard-failure rule targets shared support with probability .30 and main layout otherwise. Only these explicit hard facts produce R; absent complete coverage is U. Shared-support failure precludes valid A/B certification; a main-layout failure still permits B certification where its obligations hold. Joint C requires hard PASS and both jobs certified. All 648 slots remain in the retention denominator, including main-layout hard-R outputs.

Full outcomes and paired differences. Table 23 reports all eight model–wrapper rows. Table 24 gives Ours minus Base within each model; rates are averaged over slots within contexts, then contexts equally. All selected slots, zero or negative gains, and costs are retained. All paired differences describe the fixed slot outcomes. The registry, frozen recipe, paired records and manifests are separately versioned in `data/v10_model_controls/`; the earlier result files remain unchanged.

Table 23: Full model-labelled outcomes: 216 changed contexts/648 slots per row, 32 calls/eight credits. Rates are percentages; joint C/R/U partition each row. Calls and credits are mean declared resources.

Model label	Policy	A CC	Joint C	R	U	RR	Calls	Credits
GPT-6 Astra	Base	54.8	29.9	9.3	60.8	50.0	23.3	6.1
GPT-6 Astra	**Ours**	49.4	33.2	9.3	57.6	63.4	26.3	6.9
Claude Sonnet 5	Base	50.2	24.7	7.6	67.7	49.1	23.3	6.1
Claude Sonnet 5	**Ours**	45.7	30.9	7.6	61.6	65.0	26.2	6.9
Gemini 3.8 Flash	Base	61.1	33.8	6.8	59.4	52.5	23.2	6.1
Gemini 3.8 Flash	**Ours**	55.1	37.8	6.8	55.4	65.4	26.2	6.8
Qwen3.8-27B	Base	55.4	29.8	7.4	62.8	51.1	23.5	6.0
Qwen3.8-27B	**Ours**	45.1	28.9	7.4	63.7	61.9	26.4	6.8

Table 24: Paired differences, Ours minus Base. Quality changes are percentage points; positive cost changes indicate additional declared resource use. Every model pair is retained.

Model label	Δ A CC	Δ Joint CC	Δ RR	Δ Calls	Δ Credits
GPT-6 Astra	-5.40	+3.24	+13.43	+2.96	+0.79
Claude Sonnet 5	-4.48	+6.17	+15.90	+2.99	+0.77
Gemini 3.8 Flash	-6.02	+4.01	+12.96	+2.98	+0.78
Qwen3.8-27B	-10.34	-0.93	+10.80	+2.94	+0.79