Hecaton: One Mind, Many Arms — Live, Language-Guided Orchestration of Multi-Arm Robot Cells

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

Abstract—Personal AI agents can now plan, converse and use software tools, but they stop at the screen. Complex physical work—assembly, laboratory protocols, dressing a person in a rigid exosuit—needs many hands at once, yet cells with three or more arms are still programmed offline as fixed scripts. We present Hecaton, an agent that conducts N robot arms as one body while a person talks to it: requests, corrections and faults change the plan within seconds without stopping the cell, the agent warns about foreseen conflicts before they happen, and every action it commits is verified by deterministic tools so that the language model never drives joints. To make such claims testable we first build HecatonBench, a benchmark of scenarios with ground-truth events whose oracles read only the simulation log. [TODO: headline numbers from the sealed test split: RoboBallet 10×40 sets vs. non-LLM baselines; live-interaction success and latency; cooperative suit-up completion; 0 collisions.]

I. Introduction

Multi-arm cells unlock capabilities two arms cannot provide: hold–hold–operate fixtureless assembly, hand-over chains, large or flexible parts, and redundancy when an arm fails. Today such cells rely on hand-designed trajectories. Learned planners such as RoboBallet [5] jointly allocate, schedule and plan eight arms for reaching tasks and generalize zero-shot to new layouts, but they do not take language, cooperative holding, or a human working in the loop. Language-model planners coordinate robots through dialogue or dependency graphs [8, 3, 2], mostly for two or three arms and at the task level. [TODO: one paragraph: the gap – live, cooperative, verified orchestration of many arms with a person in the loop.]

Contributions:

- HecatonBench: scenarios as data with ground-truth utterances, faults, human motion and hazards; oracles computed from the simulation log only; lockstep and real-time clocks; reference controllers that prove the oracles have power; a sealed test split.
- Hecaton: a talker/thinker agent that searches over verified tools (propose → simulate → refine), a reflex layer that stops the cell without the language model, and a living receding-horizon plan.
- An evaluation on RoboBallet's own task sets, live-interaction scenarios, and a cooperative suit-up cell, with ablations that separate the value of the language model from the value of search.

II. Related Work

Multi-arm planning. [TODO: RoboBallet [5]; prioritized and conflict-based multi-agent planning (verify refs before adding); GPU trajectory optimization (verify).] Language models for robot planning. Code as Policies [6]; LLM+P [7]; multi-robot LLM collaboration [8]; dual-arm parallel planning [3, 2]. [TODO: position: we add many arms, verification, live human interaction, and a benchmark with oracles.]

III. HecatonBench

[TODO: scenario schema; clocks; controller interface (observe / tools / commit through a verify gate); oracles: safety, stop latency, intent satisfaction and effect latency, hazard alert precision/recall/lead time, cooperative fastening and hold drift, solved/makespan; reference controllers (null, fixed-plan, unsafe, privileged) and their predicted vs. measured outcome matrix; dev/test split.]

A. Surgical Track: Agent-Driven Multi-Arm Procedure Environments

The same orchestration problem that a suit-up cell exposes appears, sharpened, in the operating room: a surgical robot already puts four arms into one scene, procedure guidelines already describe the task sequence, and every step must be verified before it may act. We target three pillars as the surgical track of HecatonBench: (1) robot AI — multi-arm policies for two to eight cooperating arms, with VLA/LLM baselines evaluated under the same oracles; (2) high-quality surgery — 100 or more grounded procedures rebuilt at realism ≥8/10 with real anatomy and cited/calibrated physics; (3) agent-integrated — an agent environment factory that builds and gates the environments, and agents that plan and drive the arms through them. The target package date for this track is 2026-12-04. The bench also makes beyond-human measurable rather than rhetorical: every task family carries a cited human-expert baseline and its metrics report the agent-to-human-expert ratio; beyond-human is claimed only where measured, and is otherwise the stated target. The long-range scope is every standard operation: a coverage catalog built from public surgical curricula, case-log categories and procedure classifications is the denominator, and it extends beyond human surgery to laboratory-animal procedures (e.g. rodent stereotaxic

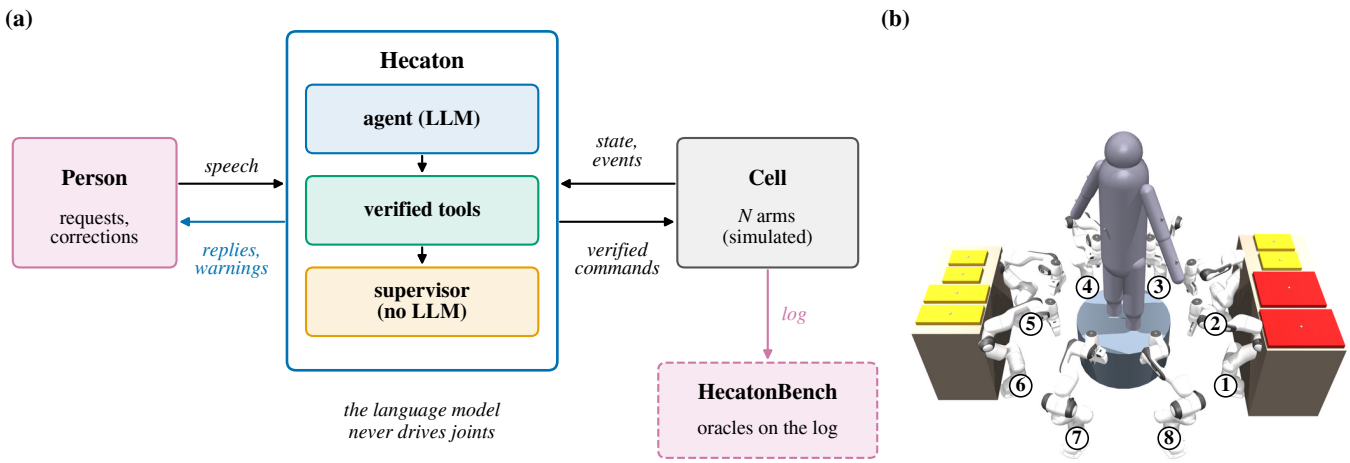


Fig. 1. (a) Hecaton overview. A person talks to the agent; the agent plans by calling verified tools; a supervisor without a language model executes and can stop the cell; HecatonBench scores runs from the simulation log only. (b) The suit-up cell in MuJoCo: eight Franka Panda arms (bases numbered 1–8), eight armor parts on two racks, and a mannequin (scene geometry only; no results shown).

and implant surgery), large-animal training models and standard veterinary operations (planned).

HecatonBench surgical builds on environments the field already runs, ported behind one interface and extended, rather than rebuilt from scratch: ORBIT-Surgical [12] and i4h-physics [9] on Isaac/Newton, LapGym [10] and SurRoL [11] task logic, and the 17 annotated tasks of ImitateCholec [4] as step ground truth for clip-and-cut cholecystectomy. The plan compresses to four gates:

- G1 scale: at least 100 gated procedure environments and at least 1,000 variants (task × patient anatomy × arm count × port layout × difficulty); both are floors, not caps.
- G2 quality: an environment counts toward G1 only at realism ≥8/10 under cross-model frame review, with CT/atlas-sourced anatomy at life scale and cited or calibrated physics parameters.
- G3 multi-robot: at least 15 tasks that need two to eight arms to cooperate — handover, retract-plus-dissect, clip-and-cut with three to four arms, and five-to-eight-arm cells.
- G4 agent-driven: an agent environment factory generates, gates and registers new environments with no human step; agent baselines (LLM planner, VLA, scripted) are evaluated over the whole bench with receipts.

Every environment enters the bench through a fail-closed gate, never by assertion: headless reset/step, a deterministic seed, a success and failure oracle shown able to fail, a measured step rate, and a realism score confirmed by cross-model frame review. Environments below the realism bar stay gated-not-counted with an explicit gap list, so the counted table reports measured quality rather than aspirational coverage.

Physics stays pluggable. The bench runs i4h/Newton [9] and SOFA behind one scene/step/observe interface — GPU-parallel dynamics for bleeding, clipping and thermal systems, and FEM as the validated tissue oracle — choosing the best backend per physics type by measured realism, the solver/contact/parameter-swappable pattern of RoboSimStudio [1]. A task keeps one scene description; the backend is a gate-time choice with a receipt, not a fork of the bench.

Table I reports the measured status of the surgical track today. The base reproduces, the first own patch on the artery cut works end to end, and the grounding pieces (step graph, virtual patient, tissue parameters) are in place; realism scores sit far below the G2 bar, which is the point of reporting them: the gate is doing its job while the track is young.

Roadmap (planned): W1 brings the environment registry and unified API online with at least 30 environments green through the gate; W2 adds the multi-robot tasks (at least 15) and the variant generator; W3 scales to at least 100 environments and at least 1,000 variants while realism upgrades land; W4 freezes on 2026-12-01 and ships the paper package by 2026-12-04. These numbers are the plan, not measured results.

IV. Hecaton

[TODO: layers and rates; tool set (IK, allocation, space-time multi-arm planner, co-hold planner, evaluate); propose–simulate–refine search; reflex supervisor; living plan; event handling; budgets.]

V. Experiments

[TODO: setups: (A) RoboBallet 10×40, 8 arms, their referee; (B) live-interaction scenarios; (C) suit-up cooperative cell; (D) held-out cell run once. Baselines: greedy, OR-Tools, OR-Tools + equal-budget random search, fixed-plan agent, Hecaton without search. ≥3 runs per seed; cost and latency per episode.]

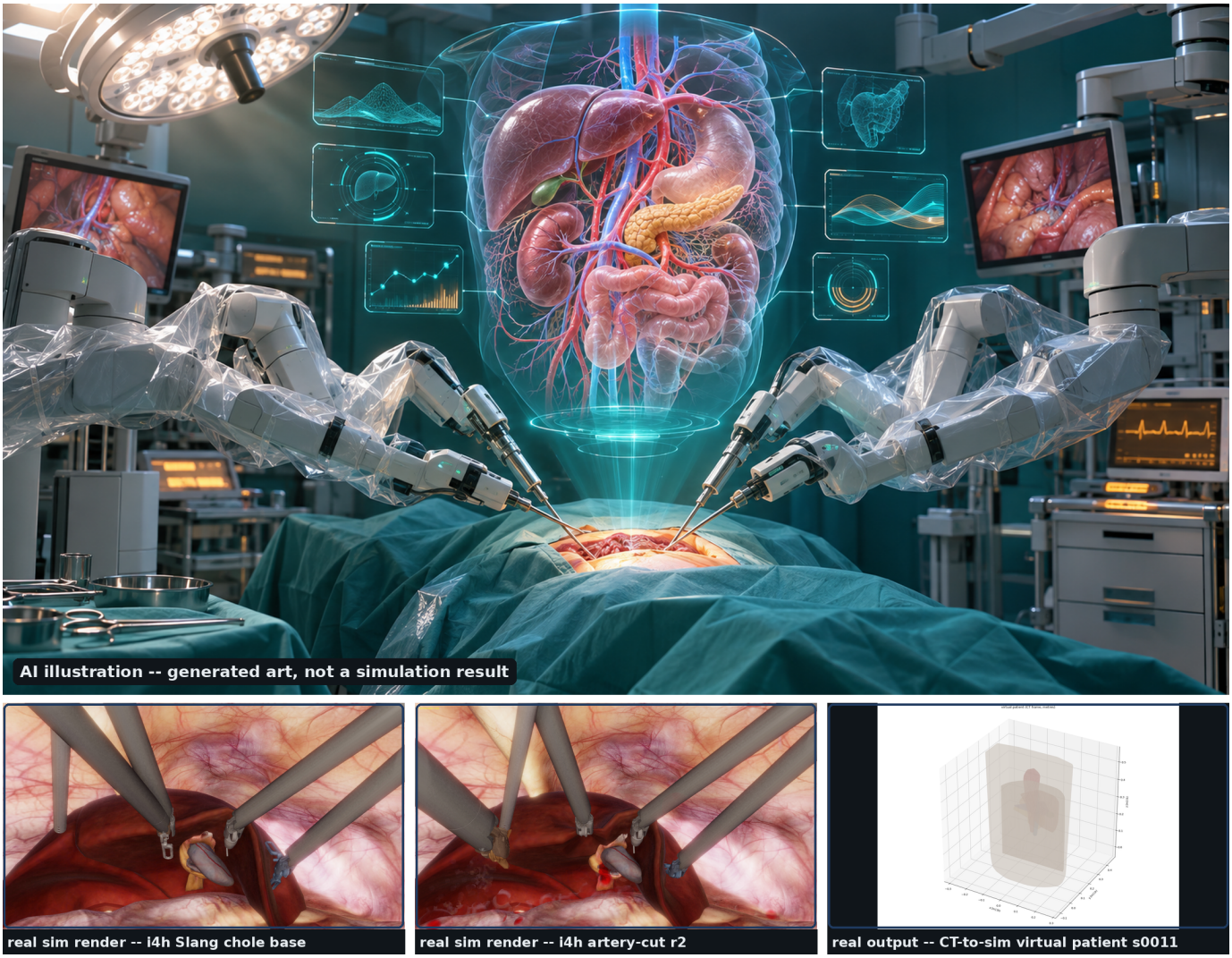


Fig. 2. Surgical track at a glance. Top: AI illustration of the target multi-arm operating room (generated art, labelled illustration; not a simulation output and not measured data). Bottom: real output of this bench's own stacks — i4h Slang base render of the cholecystectomy scene, scripted driver (cross-model realism 5/10; receipt artifacts/verify/i4h-slang-render-20261106.md); i4h artery-cut r2 with visible bleeding at the cystic pedicle (realism 4/10; receipt t2/artifacts/i4h-artery-r2/RESULT.md); CT-to-sim virtual patient s0011 (receipt artifacts/verify/s1a-pr50-lead-verify-20261105.md). Receipt paths are under the 20261104-surgery-multiarm task run.

VI. Limitations and Stage-2 Extension Plan

[TODO: simulation only; weld-based attach; language-model latency; planned extension: surgery-pod and laboratory cells on the same core, 16 arms, real-arm pilot.]

References

[1] Kaiyuan Chen, Shuangyu Xie, Uksang Yoo, Xinyue Zhu, Cael Magner, Alvin Zhu, Daehwa Kim, Eric Heiden, Erwin Coumans, Miles Macklin, Spencer Huang, and Ken Goldberg. Robosimstudio: An interactive toolkit for human and agent rapid real2sim of robot manipulation tasks, 2026. Project website: https://robosimstudio.github.io.

[2] S. Duan, P. Ren, N. Jiang, Z. Che, J. Tang, et al. Robopara: Dual-arm robot planning with parallel allocation and recomposition across tasks. In International Conference on Learning Representations (ICLR), 2026.

[3] Zeyu Gao, Yao Mu, Jinye Qu, Mengkang Hu, Shijia Peng, Chengkai Hou, Lingyue Guo, Ping Luo, Shanghang Zhang, and Yanfeng Lu. Dag-plan: Generating directed acyclic dependency graphs for dual-arm cooperative planning. In IEEE International Conference on Robotics and Automation (ICRA), 2026. arXiv:2406.09953.

[4] Pascal Hansen, Ji Woong Brian Kim, Antony Goldenberg, Juo Tung Chen, Yuanzhe Amos Li, Anton Deguet, Brandon White, De Ru Tsai, Richard Cha, Jeffrey Jopling, Paul Maria Scheikl, and Axel Krieger. Imitatecholec: A multimodal dataset for long-horizon imitation learning in robotic cholecys-

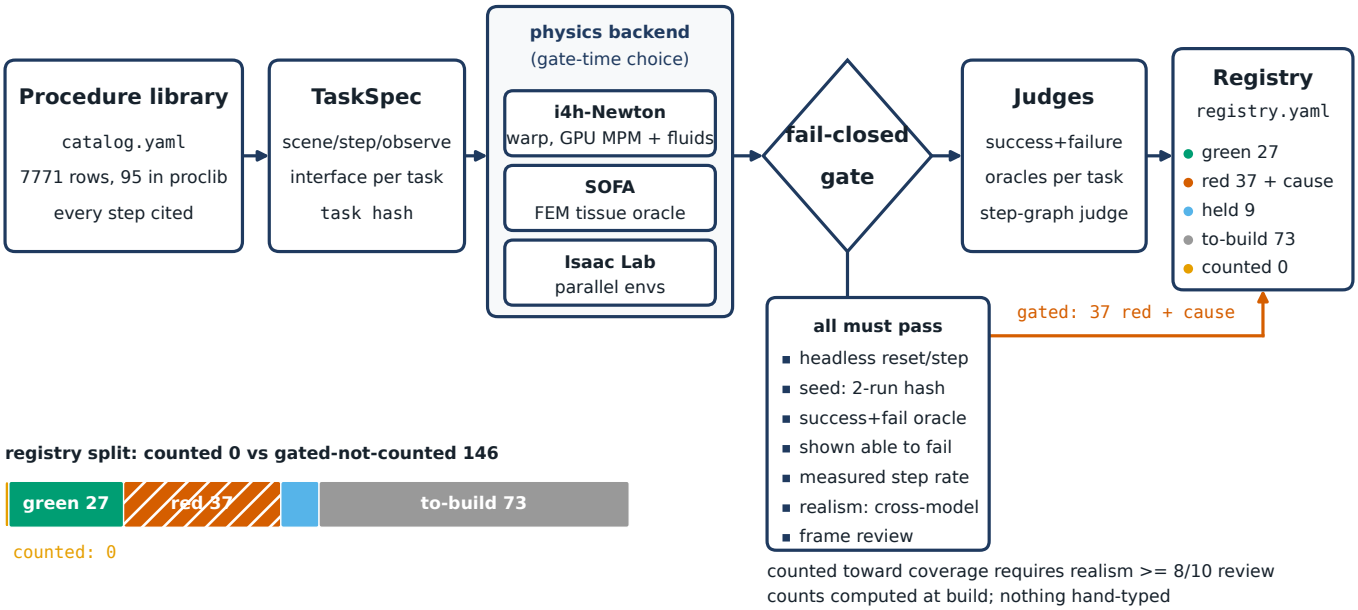


Fig. 3. Agent environment factory. Procedure-library entries become TaskSpecs behind one scene/step/observe interface, run on a gate-time-chosen physics backend (i4h-Newton, SOFA, Isaac Lab), and enter the registry only through the fail-closed gate: headless reset/step, two-run seed-state hash, a success-and-failure oracle shown able to fail, a measured step rate, and cross-model realism review; environments that fail stay gated-not-counted with a recorded cause. All counts are computed at build time from the merged environment registry (registry.yaml on main) and bench/proclib-v2/catalog.yaml (script: figsrc/surgical/fig2_pipeline.py); counted toward coverage requires realism ≥8/10.

tectomy. Scientific Data, 13(1):210, 2026. doi: 10.1038/s41597-025-06526-z.

[5] Matthew Lai, Keegan Go, Zhibin Li, Torsten Kroger, Stefan Schaal, Kelsey Allen, and Jonathan Scholz. Roboballet: Planning for multi-robot reaching with graph neural networks and reinforcement learning. Science Robotics, 10(106), 2025.

[6] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In IEEE International Conference on Robotics and Automation (ICRA), 2023. arXiv:2209.07753.

[7] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency. arXiv preprint arXiv:2304.11477, 2023.

[8] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. arXiv preprint arXiv:2307.04738, 2023.

[9] NVIDIA Corporation. i4h-physics-simulation: Isaac for healthcare physics simulation. https://github.com/isaac-for-healthcare/i4h-physics-simulation, 2026. Version v0.9.0, commit 86e2fd851e303d03043188ea7d56789e9577322a, Apache-2.0 license, accessed 2026.

[10] Paul Maria Scheikl, Balázs Gyenes, Rayan Younis, Christoph Haas, Gerhard Neumann, Martin Wagner, and Franziska Mathis-Ullrich. LapGym – an open source framework for reinforcement learning in robot-assisted laparoscopic surgery, 2023.

[11] Jiaqi Xu, Bin Li, Bo Lu, Yun-Hui Liu, Qi Dou, and Pheng-Ann Heng. SurRoL: An open-source reinforcement learning centered and dVRK compatible platform for surgical robot learning. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1821–1828, 2021. doi: 10.1109/IROS51168.2021.9635867.

[12] Qinxi Yu, Masoud Moghani, Karthik Dharmarajan, Vincent Schorp, William Chung-Ho Panitch, Jingzhou Liu, Kush Hari, Huang Huang, Mayank Mittal, Ken Goldberg, and Animesh Garg. ORBIT-Surgical: An open-simulation framework for learning surgical augmented dexterity. In arXiv preprint arXiv:2404.16027, 2024.

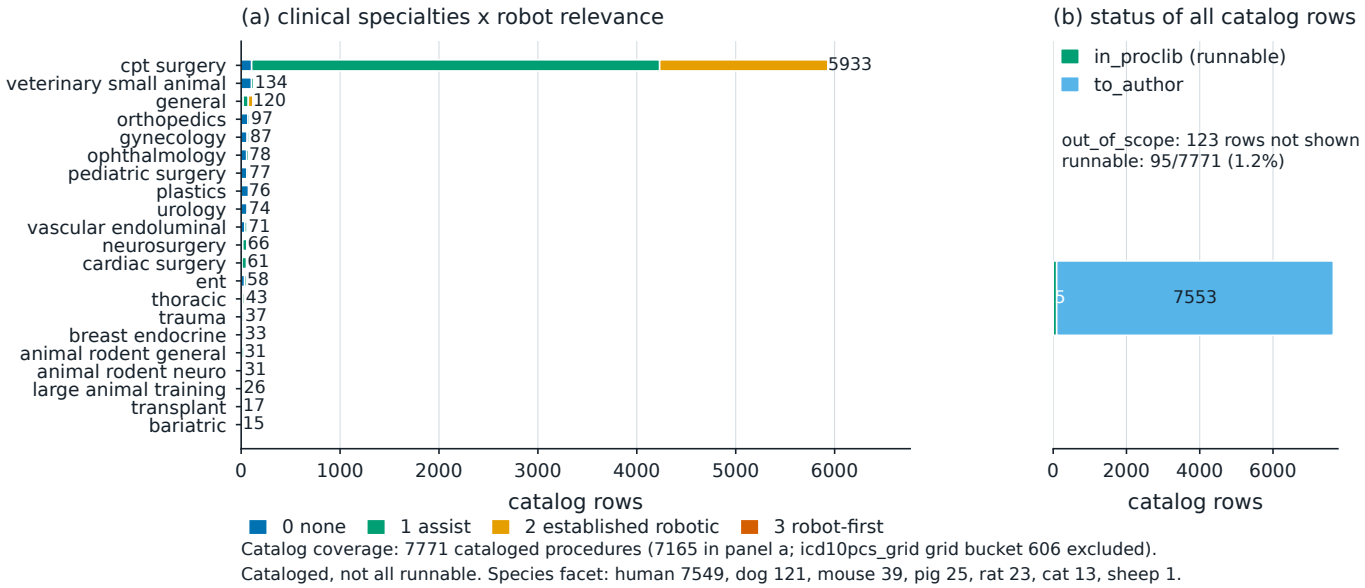


Fig. 4. Procedure coverage catalog, computed at build time from bench/proclib-v2/catalog.yaml (script: figsrc/surgical/fig3_coverage.py). (a) Clinical specialties stacked by the robot-relevance rubric (0 none / 1 assist / 2 established robotic / 3 robot-first); the ICD-10-PCS grid cross-check rows are excluded from the bars and named with their count in the footnote. (b) Procedures mapped into the procedure library vs. cataloged awaiting authoring; out-of-scope rows are reported as text.

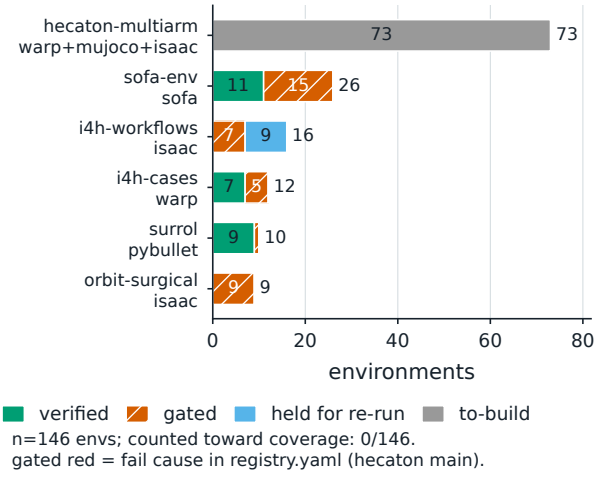


Fig. 5. Registered environments per source, stacked by measured gate status (verified green; gated vermillion, fail cause recorded in the registry; held for re-run blue; to-build gray); the gold edge marks environments counted toward coverage (0 at build time). Computed at build time from the merged registry.yaml on main (script: figsrc/surgical/fig4_envstatus.py).

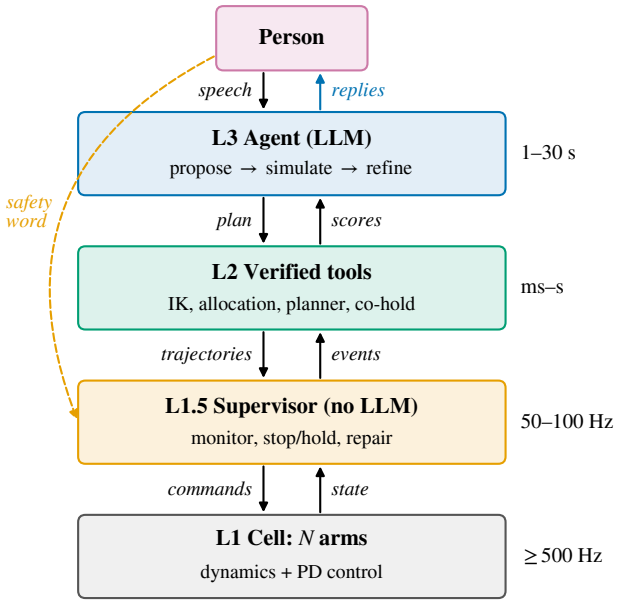
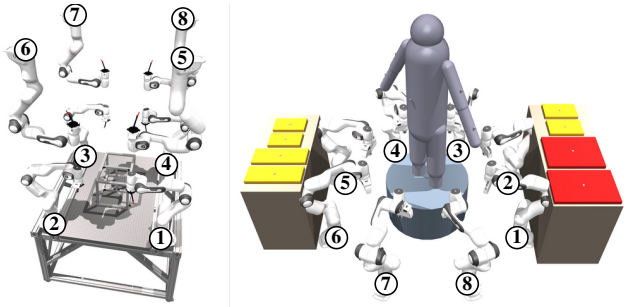


Fig. 6. Control layers and their design rates. The language model (L3) plans and explains; it never commands joints; it runs event-driven. Safety words go straight to the supervisor (L1.5). Rates are design targets, not measurements.

TABLE I
Surgical track: measured status

item	measured result	status
i4h chole SOTA base	34.6–55.6 Hz median step rate over 3 runs (0.55–0.88× realtime), 2400 steps, all 9 physiology systems enabled (7 fire); bleeding and suction never fire under the stock scripted driver	verified
i4h Slang render	48.8 Hz with render+capture; renderer 59.7 fps; 0 of 400 frames dropped; cross-model realism verdict 5/10 as a labelled SOTA base reproduction	verified
artery-cut patch r2	tissue deletion 205 tets (unpatched baseline 128); first bleeding step 255; peak blood particles 4096 (pool cap, lower bound); gallbladder intact in all 401 frames; visible bleeding at the cystic pedicle from ∼10 s; cross-model realism 4/10	verified
SOFA LapGym base	ReachEnv as-is: 180–223 steps/s over 3 runs; deterministic reset and step-600 end-effector pose at seed 42	verified
Isaac Lab 32-env sanity	68–148 step FPS on Isaac-Cartpole-v0, load-dependent; identical warm-up outliers in both runs	verified
chole step graph	17 nodes from SAGES steps + ImitateCholec 17 tasks: 24 seeded runs give 20 complete, 4 safe bailouts; 0 arm-arm contacts, 0 critical-structure injuries; CVS safety gate shown able to fail; deterministic step-log hashes	verified
CT-to-sim patient	virtual patient s0011: 10 organs, all watertight; Hausdorff95 0.358–0.498 mm vs. a 2 mm gate; max organ volume diff 1.203% vs. a 3% gate	verified
tissue parameters	11 tissues, 98 literature-cited entries (value + unit + citation), spot-checked against PubMed abstracts	verified
procedure library	113 procedures, 1,236 steps, every step cited; 113 of 113 mapped to an environment family and to a human-expert baseline entry (measured for 7 families, target-only for 8); validator shown able to fail	verified
multi-arm task set	23 tasks for 2–8 arms (2:4, 3:9, 4:7, 5:1, 6:1, 8:1); validator 11 of 11 checks, shown able to fail; 6 known-fail controls measured; 138 variants on disk; none counted toward G1 yet	specified
backend swap (SOFA / i4h)	one scene/step/observe interface, same task hash on both backends; grasp-retract trajectory RMSE 4.14 mm (final 13.69 mm, max 23.20 mm) between backends; gravity and contact still off	verified, not real
env registry + gate	146 registered environments: 27 green (SOFA 11, PyBullet 9, Warp 7; 18.7–2833.6 Hz), 37 red with measured cause, 9 held for re-run, 73 to build; green = headless, 2-run deterministic with non-empty state hashes, an oracle that executed, known-fail control passed; 0 counted (realism not yet at the bar)	merged
build4 physiology on i4h	four self-built systems layered on i4h: clip mechanics, respiration/pulse, thermal dose, surface blood flow with clotting; each review finding fixed test-first (red then green), 67 tests pass on CPU	merged
agent env factory v1	5 chole-family procedures to 24 phase-level environments with no human step: 17 green, 7 red with measured cause; stills + judge verdicts for every green env; best realism 2/10, 6 of 79 events judged visible, 0 counted; single judge family so far	in review

Numbers measured on our host: single RTX 5090, loaded shared



(a) RoboBallet cell (b) Suit-up cell

Fig. 7. Simulated cells, MuJoCo renders with arm bases numbered. (a) The RoboBallet evaluation cell [5] (vendored scene; four ceiling- and four table-mounted arms; its 40 reach targets per task set are not drawn). (b) Our suit-up cell: eight floor-mounted arms around the mannequin.

TABLE II
[TODO: main results – filled only from sealed-test runs]

controller	solved	collisions	makespan	live success
[TODO: rows]				