MEDHARNESS: A UNIFIED BUDGETED RUNTIME FOR RELIABLE MEDICAL IMAGING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Medical imaging agents still lack a shared runtime reliability layer: under a limited verification budget, deployment must decide when to stop and accept, when to open additional evidence by failure mode, when to repair, and when to reject. Tool-using workflows often freeze a fixed evidence path or a private always-on stack per dataset; those pipelines are costly, still leave many wrong accepts, and cannot share one controller across heterogeneous imaging modalities. We present **MedHarness**, a registered multi-surface runtime reliability layer that attaches after a proposer's initial prediction: a shared control policy with surface recipe adapters under one packet–route–decision-trace contract, so a new imaging surface attaches by recipe registration to the same stop / open-evidence / repair / reject controller. We evaluate on four registered surfaces (4848 cases) under two complementary tasks: live-pool decision quality and failure-enriched stress efficiency. On a 512-case live pool, MedHarness cuts wrong accepts versus always-on by about 47% and raises accepted accuracy (83.2% vs. 72.9%). On a failure-enriched four-surface stress ranking, the same controller cuts estimated API calls/case under matched context-load accounting by about 71% versus always-on. A held-out three-surface check and an alternate LLM transfer keep the same control ordering. The code is available at `https://anonymous.4open.science/r/MedHarness-paper-release-A7F3/`.

1 INTRODUCTION

Medical vision-language models and medical agents increasingly produce plausible initial predictions for image-grounded tasks (Li et al., 2024a;b). The remaining systems problem is not only higher raw accuracy, but a runtime control process under a limited verification budget: when to stop and accept, when to open more case-specific evidence, when to repair, and when to reject (Geifman & El-Yaniv, 2017; Jiang et al., 2021). Without that process, a strong first answer can still be deployed unsafely, and a weak answer can still be over-checked at fixed cost. Clinical-impact studies likewise emphasize that predictive accuracy alone does not determine whether an individual output is safe to accept (McKinney et al., 2020; Kelly et al., 2019).

Retrieval-augmented and tool-using medical workflows improve grounding (Tang et al., 2024; Wang et al., 2026). They still often pre-commit a fixed evidence path or a private always-on stack when the surface changes (Lewis et al., 2020; Yao et al., 2023). Selective prediction and self-refine modules exist (Chow, 1970; Madaan et al., 2023), but are rarely composed as one registered, budgeted runtime contract across heterogeneous medical imaging queues—so each dataset tends to grow its own pipeline rather than sharing control decisions.

Existing medical agents typically treat verification, repair, and abstention as workflow components. **MedHarness** treats them as runtime control actions over a shared decision object—the missing runtime reliability layer under a limited verification budget. It wraps a fixed proposer with task-conditioned recipes and budgeted verification and repair under one packet–route–decision-trace contract. The process is defined by four control decisions: (i) when to stop if the current candidate is sufficient, rather than always-on checking; (ii) when to open heavier evidence on the observed failure mode, rather than a pre-fixed tool path; (iii) when to attempt intermediate repair before a terminal lock; and (iv) when to reject (abstain/escalate) under the same contract, rather than forcing an unsafe accept. A new surface attaches by recipe registration, not by rewriting a private agent workflow per dataset. The contribution is the budgeted runtime control process; it wraps a frozen proposer and starts from that initial prediction. MedHarness evaluates that shared controller on four

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.

registered surfaces with a fixed proprietary proposer. Heterogeneous surfaces attach by recipe registration under one packet/route/decision-trace interface and one control policy, so the same stop / open-evidence / repair / reject decisions apply across dermatology, retinal grading, clinical-image QA, and endoscopy without rewriting a private always-on stack per dataset.

Our contributions: (i) We propose **MedHarness**, a registered multi-surface runtime reliability layer with shared control policy π (stop / open-evidence / repair / reject) and recipe adapters under one packet/route/decision-trace contract, so heterogeneous imaging surfaces attach by registration rather than a private always-on stack per dataset; (ii) on a 512-case live pool, MedHarness cuts wrong accepts versus always-on by about 47% and raises accepted accuracy (83.2% vs. 72.9%), and on a failure-enriched four-surface stress ranking cuts estimated API calls/case under matched context-load accounting by about 71% versus always-on, with live ablations isolating repair, specialist evidence, and lock; (iii) HyperKvasir-held-out ranking (Borgli et al., 2020) and alternate-LLM transfer keep the same control ordering, and on a fixed module pool budgeted π improves wrong-accept / accepted accuracy over always-on and proposer-only while opening fewer modules (Table 3).

2 Related Work

Multi-modal medical diagnosis and medical VLMs. Medical image analysis and diagnosis research has long developed task-specific models for visual assessment, from segmentation architectures to self-configuring biomedical pipelines (Ronneberger et al., 2015; Isensee et al., 2021). Skin and retinal imaging have also served as canonical high-stakes classification settings (Esteva et al., 2017; Gulshan et al., 2016). Recent medical foundation models and multimodal assistants expand this substrate for image-grounded reasoning (Ma et al., 2024; Li et al., 2024b). Broad vision-language pretraining and instruction tuning push toward reusable multi-task systems (Radford et al., 2021; Liu et al., 2023). Clinical-language and generalist medical AI studies show a parallel pressure toward general medical reasoning (Singhal et al., 2023; Moor et al., 2023). These lines raise first-pass image-grounded answer quality, but a strong candidate is not yet a budgeted accept / open-evidence / repair / abstain decision under deployment cost and safety constraints.

Vision-language and tool-augmented agents. Large language model (LLM) and vision-language model (VLM) agents provide frameworks for planning, reflection, and multi-agent collaboration (Shinn et al., 2024; Li et al., 2023). Chain-of-thought and tree-of-thought prompting improve complex problem solving (Wei et al., 2022; Yao et al., 2024). Action-oriented and tool-learning agents make decomposition operational (Yao et al., 2023; Schick et al., 2023). Self-consistency and iterative self-feedback add correction mechanisms (Wang et al., 2023; Madaan et al., 2023). Retrieval or program augmentation exposes external evidence or computation (Lewis et al., 2020; Gao et al., 2023). Dense retrievers and token-level memory widen that interface (Karpukhin et al., 2020; Borgeaud et al., 2022), yet evidence paths are often prescribed or always-on, leaving stop / repair / reject untyped even after extra retrieval.

Medical agentic systems and runtime verification. Recent medical agentic systems use collaboration and multimodal tool use to improve medical reasoning (Tang et al., 2024; Kim et al., 2024). Adaptive multi-agent and rare-disease agents with evidence-linked traces still optimize collaboration or domain diagnosis rather than a shared imaging runtime (Tu et al., 2025; Zhao et al., 2026). Tool-using chest X-ray agents remain modality-bound, and evidence-based planners still pre-commit a workflow (Fallahpour et al., 2025; Wang et al., 2026). Process-level verification and tool-interactive critiquing catch stepwise errors that outcome scores miss (Lightman et al., 2024; Gou et al., 2024). These systems bring agent tooling into clinical-style workflows, but remain stronger on answer generation and per-surface workflow construction than on composing stop / open-evidence / repair / reject under one budgeted contract. Selective prediction and self-refine already supply pieces of that stack (Chow, 1970; Madaan et al., 2023); the systems increment is a shared controller π over typed actions with surface recipe adapters under one budgeted contract. **MedHarness** instantiates that contract and attaches each new surface by recipe registration instead of rewriting a private verification workflow per surface. That registration path is what lets the same accept/repair/reject interface travel with a new imaging surface without cloning a private always-on verification stack. We therefore read medical agent tooling as providing modules and traces that a shared budgeted controller can call, rather than as a finished substitute for stop / open-evidence / repair / reject under one contract.

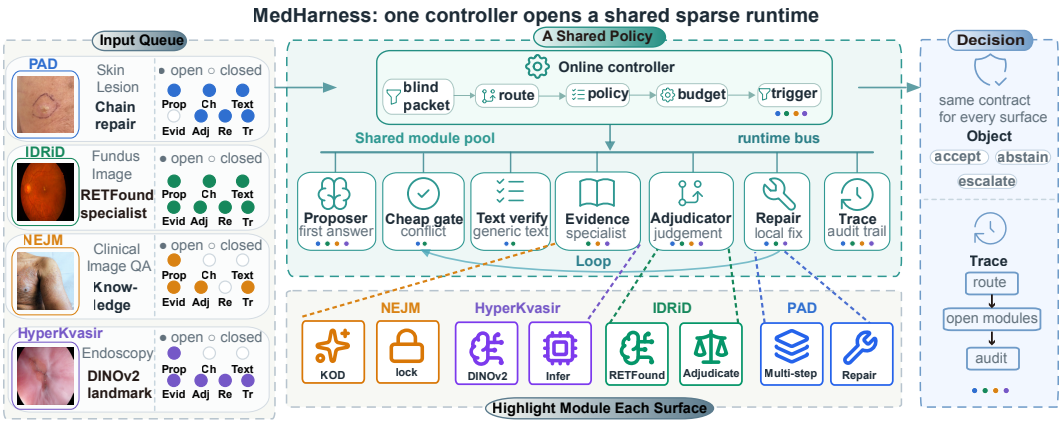


Figure 1: MedHarness shared runtime: clinical input, budgeted controller, shared module pool, and terminal decision object. Colored dots mark which surfaces use each card. Per-surface recipes list opened modules; Highlight module marks the highest-contribution module on each surface.

3 METHOD

3.1 OVERALL WORKFLOW

MedHarness is a runtime reliability layer over a fixed proposer. The systems contribution is one shared control contract—scrubbed inputs, dispatch features, stop/open/repair/lock predicates, unit module-call cost, and terminal accept/abstain/escalate reported as Correct/wrong accept / abstain (C/W/Abs.)—so heterogeneous surfaces share one packet–route–decision-trace problem. After the proposer emits an initial prediction, source-identity-scrubbed packets map by deterministic schema-aware dispatch to a registered recipe, which sparsely opens modules under budget to stop or open evidence. Intermediate repair may update the candidate; every case ends in one decision object with ordered decision-trace cost and terminal status in $\{\text{accept}, \text{abstain}, \text{escalate}\}$. Escalate covers out-of-profile / gate-failure cases and folds into Abs. as a non-accept reject in reported C/W/Abs. counts; primary scored rows keep escalate at zero. Shared objects remain the packet, route, budgeted history, and terminal decision on every registered surface.

We formalise the core objects. An incoming case is a source-identity-scrubbed packet

$$x = (v, q, o, m), \tag{1}$$

where v is the image, q the request, o optional constraints, and m deployment-visible metadata. A fixed proposer P emits the initial candidate

$$a_0 = P(x). \tag{2}$$

Deterministic dispatch selects a family–recipe route

$$r = (f, \rho) = R(x, a_0), \tag{3}$$

where R is the dispatcher and ρ the registered recipe. Under budget b,

$$H = (m_1, \ldots, m_K), \qquad C(H) = \sum_{k=1}^{K} c(m_k) \le b, \tag{4}$$

where H is the ordered history, $C(H)$ the module-call cost, and $c(m)$ the per-module charge with $c(m) = 1$ in primary comparisons. Every case terminates with a single decision object

$$d = (y, s_{\text{final}}, r, H, \tau), \qquad s_{\text{final}} \in \{\text{accept}, \text{abstain}, \text{escalate}\}, \tag{5}$$

where y is the final or withheld answer and τ is the decision-trace path. Intermediate controls (continue / open-evidence / repair) may update the candidate or history, but are never terminal outcomes. Figure 1 places the same objects end-to-end from scrubbed packet through sparse module opens to one decision object. Each registered surface attaches a short recipe over that shared pool. PAD (Pacheco et al., 2020) runs a three-stage repair-first dermatology chain; IDRiD (Porwal et al., 2018) continues to a fundus specialist after repair. NEJM (New England Journal of Medicine, 2024) opens knowledge evidence into an answer-lock when needed; HyperKvasir (Borgli et al., 2020) opens a conditional endoscopy-landmark specialist (Appendix A.1; Table 8). Those four attachments keep the same accept/repair/reject interface under one shared controller.

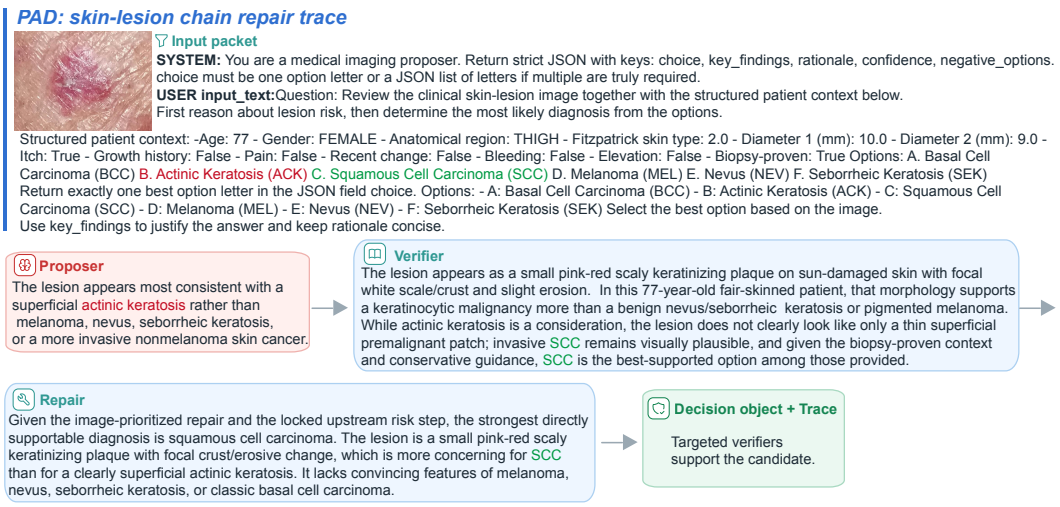


Figure 2: PAD repair running example: wrong low-risk branch corrected to squamous cell carcinoma, then accept with route and module history; illustrative single-case trace.

3.2 Running example: dermatology repair

The following running example is a single PAD case under the registered dermatology recipe: it shows how repair updates the candidate before the terminal lock, matching Figure 2.

The objects above are easiest to read on one case. Consider a scrubbed skin-lesion management packet x: the answer schema marks a dermatology decision, but source identity is removed. A fixed proposer P first emits a candidate a_0 from $P(x)$—for example, a wrong low-risk branch such as actinic keratosis. Deterministic dispatch maps (x, a_0) to the registered dermatology recipe ρ_{derm} as the recipe adapter. Under budget b, the recipe opens only the needed modules: a cheap acceptability gate fails, a verifier localizes the wrong intermediate branch, and intermediate repair produces a corrected candidate a_{t+1}, here squamous cell carcinoma. The decision lock then issues terminal $d = (y, s_{\mathrm{final}}, r, H, \tau)$ with accept, route r, ordered history H, and decision-trace tag τ. Cost $C(H)$ charges each opened module once per call, so repeated repair is charged repeatedly. If evidence remains insufficient, the same lock abstains instead of forcing an unsafe accept (Figure 2).

3.3 Shared objects

Shared objects are the packet, route, shared module pool, ordered history, and decision object (Algorithms 1–2; Appendix A.1). Compact runtime profile keys and the runtime gate checklist sit in Appendix Tables 8 and 10. After scrubbing, dispatch may use task content, modality, and answer schema, but not an explicit source-dataset identity (Appendix A.2). Figure 1 prints seven shared-pool cards: Proposer emits a_0, Cheap gate can stop early, TextVerif checks text, Evidence opens failure-mode evidence, Adjudicator resolves conflict, Repair updates non-terminally, and Trace writes τ. The same surface-colored dots appear on those cards and on the decision-object Trace field, so a reader can see which surfaces open each module and which surfaces emit a decision-trace. Evidence/Repair trees open those cards by failure mode; Accept / Abstain / Escalate are written by the decision lock as s_{final}. Modules expose comparable decision-facing signals so heterogeneous specialists can share one fabric (Lakshminarayanan et al., 2017; Guo et al., 2017). Cheap misclassification scores stay gates rather than the terminal policy (Hendrycks & Gimpel, 2017; Ovadia et al., 2019).

The deployed controller is recipe- and threshold-based rather than end-to-end trained. Strict / default / loose are *evidence-budget* operating points (early-stop / default / extended-evidence) on one shared acceptance policy and one shared pipeline. A known surface attaches by registration, opens only recipe-admissible modules under the shared policy, and still returns d:

$$\mathrm{Reg}(\mathrm{surface}) = (f, \rho), \qquad H = \pi(r, x, a_0, b), \;\; m_k \in \mathcal{M}_\rho. \tag{6}$$

where Reg is the registration map from a surface to (f, ρ), π is the shared policy, H is the opened history under budget b, and x is the packet, so a newly registered surface reuses the shared accept/repair/reject interface without a private always-on stack.

Table 1: Evaluation pools, surfaces, and common abbreviations as a glossary.

Pool		Surface	
Name	**Role (n)**	**Name**	**Role (n)**
4848	Attach+replay all four (n = 4848).	PAD	Skin repair-first path (n = 200).
Live512	When-to-stop 171/171/170 PAD/IDRiD/NEJM, no HK.	IDRiD	Retinal specialist (n = 259).
Mix4	Stress 128×4, 64/64, w/ HK.	NEJM	QA knowledge-lock path (n = 685).
Mix4-HKout	Specialist held-out 128×3 (n = 384).	HK	Endoscopy specialist (n = 3704).

Arms		Metrics	
Abbrev.	**Meaning**	**Abbrev.**	**Meaning**
Prop.	Proposer alone; Mods = 0.	C/W/Abs.	Correct/wrong accept / abstain.
Always-on	Always-open evidence; same lock.	Acc.	Accepted accuracy.
Mono.	Alias of Always-on arm.	Mods.	Mean opened module calls.
SC-Abs.	Self-consist. then abstain.	WA	Wrong accepts matched.
Text-check	Text-only verify gate.	R/H	Rescue / harm accept counts.
agree/conf	Agree + step-score confidence.	R-H	Rescue minus harm vs. Prop.
MH	Full sparse controller.	Cov/Rsk	Coverage / risk-among-accepts.
MH-strict	Early-stop budget.	Utility	Safety-oriented utility.
MH-default	Default budget.	HK‡	Registration-scale only.
MH-loose	Extended budget.	LOFO	Leave-one-family-out.

3.4 Sparse layers and repair under the same loop

Given route r, MedHarness does not open every sparse layer on every case. A cheap gate may already mark the candidate accept-eligible and leave heavier evidence closed; otherwise the runtime upgrades to targeted re-inspection or specialist evidence, then repair support and adjudication by observed failure mode rather than one always-on path. This is minimum-sufficient verification: spend evidence until the decision is supported, then stop. Always-on enumeration versus sparse opening under the shared policy is

$$H_{\mathrm{AO}} = \mathrm{enum}(\mathcal{M}_\rho), \qquad H_{\mathrm{MH}} = \pi(r, x, a_0, b). \tag{7}$$

where enum opens every admissible module, H_{AO} is always-on history, H_{MH} is sparse history under π, and ρ and x are the registered recipe and packet. When opened evidence e_t supports a recoverable error, intermediate repair updates the candidate inside the same loop as $a_{t+1} = \mathrm{Repair}(x, a_t, e_t)$, where Repair is the intermediate candidate update under opened evidence; the decision lock then assigns s_{final} from $(x, a_{t+1}, e_t, H_{t+1})$. A wrong intermediate branch can still become a correct accept if repair is supported; otherwise the runtime abstains under budget. Repair stays intermediate before terminal accept or abstain, and never replaces the shared decision object d.

3.5 Instantiated recipes

The four recipes share packet, router, and d: dermatology repair-first, retinal post-repair specialist, clinical-QA cheap-first knowledge lock, and endoscopy conditional specialist—adapters over one contract, not private always-on workflows. A new surface attaches by registration rather than by rewriting that shared decision object d.

4 Experiments

4.1 Setup

We evaluate MedHarness as a runtime control process—when to stop, open failure-mode evidence, repair, or reject under budget—rather than as a new predictor on a single benchmark. Experiments use one shared packet/route/decision-trace interface over known registered recipes rather than zero-shot surface discovery. The experiments form a matched policy comparison: can one registered contract attach to heterogeneous medical decisions; does budgeted control improve decision quality versus always-on verification and simple rejection on a shared queue; and do its repair, specialist, and lock decisions account for that difference?
Table 1 is the glossary for pools, surfaces, and metric abbreviations used below. Arm open/stop detail is in Appendix A.3.

Pool composition. Live512 composition is 171/171/170 PAD/IDRiD/NEJM without HyperKvasir; Mix4 and Mix4-HKout compositions follow Table 1. PAD-UFES-20 follows prior skin-surface releases (Pacheco et al., 2020; Tschandl et al., 2018). IDRiD and HyperKvasir follow their grading and endoscopy benchmarks (Porwal et al., 2018; Borgli et al., 2020). NEJM Image Chal-

Table 2: Surface attachment and Live512 when-to-stop; names in Table 1.

Surfaces (%): CA/WA; Acc/Abs					Live512			
Method	**PAD**	**IDRiD**	**NEJM**	**HK‡**	**Method**	**C/W/Abs.**	**Acc.↑**	**Mods.↓**
Cases	200	259	685	3704	Cases	512	–	–
Proposer-only	63.5/36.5 63.5/0.0	46.3/53.7 46.3/0.0	87.0/13.0 87.0/0.0	43.9/56.1 43.9/0.0	Always-on	373/139/0	0.729	4.00
MedHarness	**83.0/12.5** **86.9/4.5**	**47.9/17.4** **73.4/34.7**	**98.1/1.9** **98.1/0.0**	93.6/0.0 100.0/6.4	**MedHarness**	**367/74/71**	**0.832**$_{\uparrow+0.103}$	**3.86**$_{\downarrow-0.14}$

lenge follows the public clinical image quiz (New England Journal of Medicine, 2024). Cases enter as source-identity-scrubbed packets; proposer, scorer, module pool, and runtime remain fixed unless stated otherwise. Primary results evaluate one strong proprietary proposer **GPT-5.4**. On the same Live512, Grok-4.5 Proposer-only, Always-on, SC-Abs., and full MedHarness (Table 21) transfer the LLM proposer, verifier, repair, and lock stack while keeping specialists fixed; Appendix A.6 parks reuse diagnostics. Every main comparison freezes the proposer and changes only which evidence path the registered controller may open under that shared verification budget.

Backends and decoding. Backends: unless noted, the proposer is **GPT-5.4**; verification, repair, and lock use the same stack. Decoding: proposer temperature 0.1; verifier/repair/lock temperature 0.0. Specialists and role prompts sit in Appendix A.5. Recipes and thresholds were fixed offline before Mix4 and Live512, then held fixed for every scored arm in the Results tables below. No scored comparison below changes the proposer mid-table.

4.2 Protocol and metrics

Decision accounting compares matched proposer-only against full MedHarness under one shared score. Arm and metric abbreviations follow Table 1; the full calculus is restated in Appendix A.4. The three scored pools stay non-interchangeable under the Setup pool composition above. Table 2 uses Live512 ids; Table 4 uses Mix4. Selective matching, bootstrap intervals, and repair/specialist/lock ablations isolate levers; LOFO / five-split / reweight rows remain appendix sensitivity (Appendix A.6). Unless noted, every scored arm uses the same frozen recipe thresholds and proposer–scorer pair on its named pool, with no per-arm retuning of decoding or specialist prompts. Pool roles stay fixed for the whole Results chain below, including the Live512 quality pool and the Mix4 stress ranking with HyperKvasir.

5 Results

Results follow five subsections keyed to Table 1: Multi-surface decision quality, Same-pool activation policy, Mix4 stress efficiency, Component ablation, and Budget and results.

5.1 Multi-surface decision quality

On Live512 under a shared module set, MedHarness improves accepted decision quality versus Always-on and strengthens surface attachment and when to stop versus the proposer floor across registered surfaces (Table 2). HyperKvasir is held out of that primary evaluation; pool abbreviations are Table 1. Same-pool activation is Section 5.2 (Table 3). Appendix Tables 13 and 12 expand the same-pool controls. Escalate folds into Abs. in reported C/W/Abs. counts; default Live512 and Mix4 scored rows keep escalate at zero (Section 5.4).

The four left-hand improvements are four forms of the same runtime decision: PAD recovers a repairable branch (83.0/12.5 vs. proposer 63.5/36.5), IDRiD cuts wrong accepts on the repaired path (17.4 vs. 53.7) with explicit abstention when unsupported (34.7 Abs.; correct-accept 47.9 vs. 46.3), NEJM opens a knowledge lock only for the cases that need it (98.1/1.9 vs. 87.0/13.0), and HyperKvasir activates a specialist path (93.6/0.0 vs. 43.9/56.1) as registration-scale attachment. Failure/abstain honesty beside those surface stops is Figure 9. Thus the table establishes attachment across registered surfaces under one shared decision contract. The full 4848-case scrubbed replay is path-complete (Appendix Table 11) and supports registered routing under preserved task schemas.

Absolute surface rates are heterogeneous; the shared metric is decision quality under a fixed proposer on each registered surface. On Live512 against the same full module set, MedHarness cuts wrong accepts from 139 to 74 and raises accepted accuracy from 0.729 to 0.832 (367/74/71 at 3.86 modules vs. Always-on 373/139/0 at 4.00) under a safety-oriented utility. Self-consistency abstention and a text-check gate fall short of that Live512 quality result; matching the same wrong-accept count

Table 3: Same-pool activation policies on Live512 with HyperKvasir excluded and on Mix4-HKout. Only the activation policy changes; specialist stays in the pool. No-specialist component ablation appears in Appendix A.6.

Live512				Mix4-HKout			
Arm	**C/W/Abs.**	**Acc.↑**	**Mods.↓**	**Arm**	**C/W/Abs.**	**Acc.↑**	**Mods.↓**
Proposer-only	301/211/0	0.588	0.00	Proposer-only	192/192/0	0.500	0.00
Always-on	373/139/0	0.729	4.00	Always-on	253/131/0	0.659	4.00
Agree/conf cascade	302/153/57	0.664	2.79	Agree/conf cascade	164/179/41	0.478	1.74
MedHarness	**367/74/71**	**0.832**$_{\uparrow+0.103}$	**3.86**$_{\downarrow-0.14}$	**MedHarness**	**291/46/47**	**0.864**$_{\uparrow+0.205}$	3.87$_{\downarrow-0.13}$

Table 4: Mix4 stress ranking with bootstrap CIs; HyperKvasir included. Role/Contrast/Target/CA expansions sit in Appendix A.3.

Pool ranking						Bootstrap contrast			
Method	**C/W/Abs.**	**Acc.↑**	**R/H**	**Mods.↓**	**Role**	**Contrast**	**Metric**	**Δ**	**95% CI**
Proposer-only	256/256/0	0.500	–/–	0.00	floor	MH–Prop.	CA	+154	[132, 177]
SC-Abs.	251/194/67	0.564	35/15	0.00	selective	MH–Prop.	WA	–210	[–231, –188]
Text-check	261/245/6	0.516	37/31	1.99	selective	MH–Prop.	R-H	+166	[145, 187]
Always-on	307/205/0	0.600	76/25	4.00	mono	MH–Mono.	CA	+103	[82, 124]
MedHarness	**410/46/56**	**0.899**$_{\uparrow+0.299}$	**168/2**	**3.15**$_{\downarrow-0.85}$	sparse	MH–Mono.	WA	–159	[–181, –139]
						MH–Mono.	R-H	+120	[100, 139]

Selective matched						MH R-H gap
Method	**Target**	**C/W/Abs.**	**Acc.↑**	**R/H**	**R-H**	**Gap note**
SC-Abs.	coverage	250/193/69	0.564	34/15	19	R-H 19 $\ll$ 166
SC-Abs.	WA	147/47/318	0.758	6/3	3	WA match; heavy abstain (318)
Text-check	coverage	236/220/56	0.518	33/27	6	R-H 6 $\ll$ 166
Text-check	WA	54/46/412	0.540	8/6	2	WA match; rescue collapses
Mono.	coverage	307/149/56	0.673	76/15	61	R-H 61 still $<$ MH 166
Mono.	WA	299/45/168	0.869	70/4	66	best selective; gap remains

collapses both selective arms to all-abstain (Appendix Table 15). Qualitative case panels expand the same accept/repair/abstain story (Appendix A.7). Figure 6 preserves PAD repair traces under the shared accept/repair/abstain ladder. Figure 7 keeps same-case three-arm fairness under the shared contract. Figure 8 keeps the complementary multi-surface visual question answering (VQA) gallery under the same stops. Figures 10 and 11 preserve the disease wall and PAD/IDRiD report strips on that contract. Figures 12 and 13 preserve the NEJM/HK report strips and PAD/IDRiD decision cards. Figure 14 keeps the NEJM/HK decision cards on the same accept/repair/abstain stops. Mix4-with-HyperKvasir is the complementary stress ranking for token/call efficiency under the same runtime.

5.2 Same-pool activation policy

With the proposer, scorer, and admissible module pool fixed, activation policy alone moves the Live512 quality point; specialist, repair, and lock remain scheduled modules in that pool (Table 3). Agree/conf uses disclosed step raw-score proxy confidence. Removing specialist is a component ablation only (Table 6).

On Live512, MedHarness 367/74/71 at 3.86 modules cuts wrong accepts versus always-on 373/139/0 at 4.00; agree/conf 302/153/57 fails to match that wrong-accept / accepted-accuracy result. On Mix4-HKout with 384 cases, MedHarness 291/46/47 (0.864) at 3.87 beats always-on and agree/conf; no-specialist 351/146/15 erases the Live512 wrong-accept win (Appendix A.6).

5.3 Mix4 stress efficiency

On the controlled Mix4 stress pool that includes HyperKvasir, budgeted MedHarness beats always-on and the proposer floor on the joint Acc.–cost point under one shared controller with Live512 (Table 4). The pool fixes the proposer, module pool, scorer, and family balance: four families with 128 cases each, each balanced to sixty-four proposer-correct and sixty-four proposer-wrong under the cached proposer floor (hence the 256/256/0 overall activation floor). Mix4 is intentionally failure-enriched as the complementary stress task beside Live512 under the same runtime. The floor is proposer-only (256/256/0 at 0 modules); Always-on opens every admissible branch (307/205/0 at 4.00 modules); MedHarness is the budgeted policy (410/46/56 at 3.15 modules). Competing explanations stay strong: always-on opens every available branch, whereas selective baselines can abstain without opening evidence. Matched policy arms and paired bootstrap intervals rank those

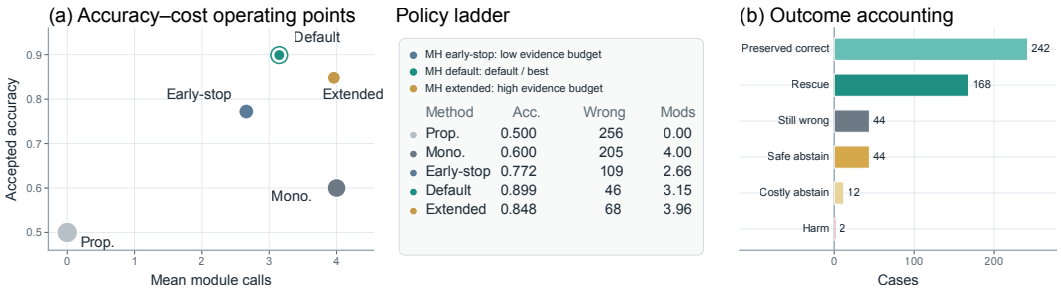


Figure 3: Accuracy–cost points, policy-ladder comparison, and outcome buckets on the Mix4 stress ranking, complementary to Live512.

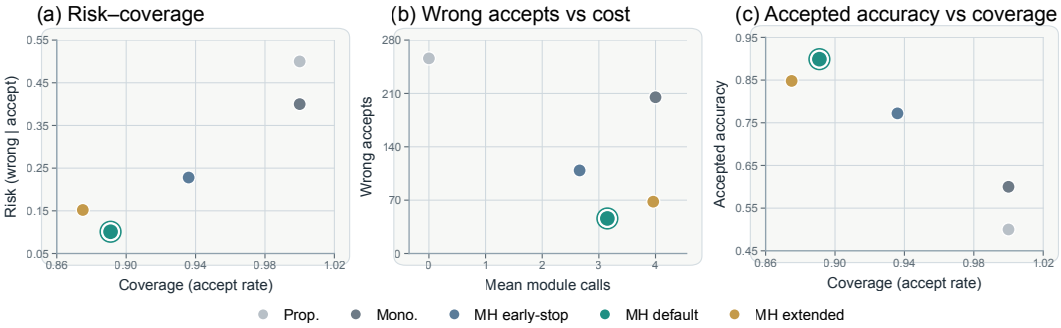


Figure 4: Risk–coverage–cost for Prop./Mono./MedHarness budget points on the Mix4 stress ranking across early-stop / default / extended-evidence.

alternatives under balanced stress. Mix4's 3.15 vs. 4.00 module column includes the HyperKvasir specialist used for this stress ranking; Live512 quality remains Table 2 / Table 3. Recipe adapters remain surface-specific under the shared control policy π. MedHarness cuts Mix4 offline token/call burden far below always-on under the same matched context-load accounting (distinct from Live512 provider billing) on Table 7; all six bootstrap CIs in Table 4 exclude 0. At matched coverage, SC-Abs. and text-check reach R-H of only 19 and 6; at matched wrong accepts they abstain on 318 and 412 while rescue collapses, and coverage-matched Mono. reaches only R-H 61. Selective abstention alone does not deliver when-to-open-evidence and when-to-repair under budget (Angelopoulos & Bates, 2023). Always-on improves on the proposer, but it opens the entire branch set and still leaves 205 wrong accepts; MedHarness instead obtains 410 correct accepts, 46 wrong accepts, and 56 explicit abstentions on this including-HyperKvasir stress pool. An independent GPT-5.4 proposer-only re-measure on the same Mix4 ids is Appendix Table 16. The grok-4.5 Live512 transfer (306/113/93 vs. always-on 341/171/0) remains in Appendix A.6 (Table 21).

Figure 3 preserves the Mix4-with-HyperKvasir stress efficiency ranking beside Live512 quality in Table 2. Same-pool activation policies remain Table 3. Panel (a) places default sparse at accepted accuracy 0.899 with 46 wrong accepts and 3.15 modules versus always-on at 0.600 with 205 wrong accepts and 4.00 modules. Panel (b) splits the Mix4 gain into 242 preserved correct accepts and 168 rescues, with harm 2, still-wrong 44, safe abstain 44, and 12 costly abstentions on this Mix4 split.

Figure 4 restates Mix4 as early-stop / default / extended-evidence points: early-stop saves calls but leaves more wrong accepts; extended-evidence spends near the always-on column without recovering the default sparse Acc./WA numbers. The useful operating point is the default budget rather than a single acceptance threshold. Together these displays remain the main-text dual-task results: Live512 when-to-stop quality, rescue-dominated Mix4 efficiency under stress, and monotonic evidence budgets. Appendix Tables 17 and 18 keep Acc. and rescue-minus-harm stable under leave-one-family holds and five-split rechecks.

5.4 Component Ablation

Component ablation removals of repair, specialist, or lock, and route-control extremes, move Mix4 accepted quality and rescue off the full sparse point (Table 5); Live512 lever removals stay on Table 6. Always-on is 307/205/0 on this stress pool; Oracle-route matches full MedHarness at 410/46/56, while Shuffled-route collapses rescue. Hierarchical adapted (Wang et al., 2026) remains far from that sparse point on the same GPT-5.4 case ids.

Table 5: Mix4 components and route extremes.

Mix4 components					Route extremes				
Method	**C/W/Abs.**	**Acc.↑**	**R/H**	**Mods.↓**	**Method**	**C/W/Abs.**	**Acc.↑**	**R/H**	**Mods.↓**
MedHarness	**410/46/56**	**0.899**↑+0.299	**168/2**	**3.15**	Always-on	307/205/0	0.600	76/25	4.00
No repair	369/134/9	0.734	123/8	2.84	Oracle-route	410/46/56	0.899	168/2	3.15
No specialist	352/113/47	0.757	111/5	2.30	Shuffled-route	184/272/56	0.404	6/66	3.15
No lock	356/100/56	0.781	114/2	3.15					
Always-on	307/205/0	0.600	76/25	4.00					

Registered recipe binding therefore still matters beyond a generic module pool under the same sparse controller. Oracle-route parity with MedHarness and Shuffled-route collapse together show that route control sets the sparse Mix4 point. Appendix Tables 19 and 22 expand hierarchy contrast. Appendix Tables 9 and 14 list arm abbreviations and offline token/call rows.

5.5 Budget and Results

Default Mix4 evidence budget preserves the joint stress operating point under the same accept/repair/reject ladder used for Live512 when-to-stop. Early-stop / default / extended-evidence ranks stay on Table 7; Live512 lever isolation on the quality pool stays on Table 6. Default Mix4 budget preserves stress-pool utility and cuts estimated offline API calls/case from 8.09 to 2.38 on Table 7, under matched context-load accounting distinct from Live512 provider billing and from Mods. module-slot counts on that same dual-task host. Mix4 default remains the stress-pool utility winner on this ladder; Live512 keeps the when-to-stop quality win on Table 2. Three-mix ordering under the same controller is Figure 5; utility weights sit in Appendix A.4. Leave-one-family / five-split / reweight stay appendix-only (Tables 17, 20). Mix4-HKout / alternate-LLM transfer and repeated-split rows are Tables 21 and 18. Figures 3 and 4 share the same Mix4 budget accounting. HyperKvasir stays registration-scale on Live512 Table 2; Mix4 including that specialist is the complementary stress ranking that keeps dual-task Results under one packet–route–decision-trace contract (Live512 when-to-stop; Mix4 budgets $2.66 < 3.15 < 3.96$).

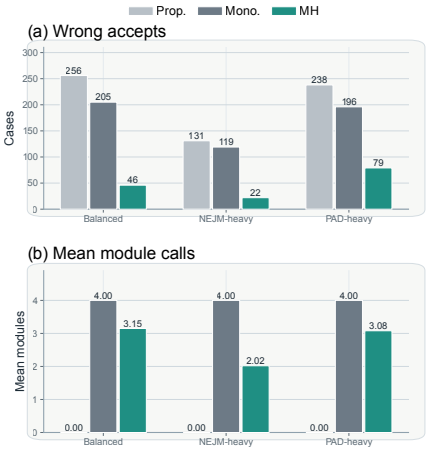


Figure 5: Three Mix4 mixes: wrong accepts and mean module calls.

Table 6: Live512 lever removals.

Method	C/W/Abs.	Acc.↑	Mods.↓
MedHarness	**367/74/71**	**0.832**↑+0.103	3.86↓−0.14
Always-on	373/139/0	0.729	4.00
Hier. adapted	254/257/1	0.497	4.72
Shuffled-route	196/179/137	0.523	3.31
No repair	312/74/126	0.808	3.83
No specialist	351/146/15	0.706	3.20
Seq. lock-off	393/119/0	0.768	3.69

Table 7: Mix4 budget ladder with Offline context-load under matched context-load accounting.

Method	C/W/Abs.	Mods.↓	Cov/Rsk	Utility↑
Prop.	256/256/0	0.00	1.00/.50	0.0
Always-on	307/205/0	4.00	1.00/.40	−102.8
MH-strict	370/109/33	2.66	.94/.23	108.3
MH-default	**410/46/56**	**3.15**↓−0.85	**.89/.10**	**174.6**↑+277.4
MH-loose	380/68/64	3.96	.88/.15	77.2
Offline context-load: est. tokens / API calls				
Prop.	1962		0.75	
Always-on	25003		8.09	
MedHarness	**7706**		**2.38**↓71%	

6 Discussion and Limitations

MedHarness shows a registered multi-surface runtime can stop, open evidence, repair, or reject under one packet–route–decision-trace contract, improving accepted quality over abstention-only and always-on baselines on a fixed proposer. Live512 when-to-stop quality and Mix4 stress-pool efficiency remain complementary under the same controller. Unseen surfaces still need a registered recipe adapter under shared π; prospective clinical use and live billing stay outside present evidence (Appendix A.8, A.9; (Liu et al., 2020; Vasey et al., 2022)). Future work should learn cross-surface recipes and runtime policies jointly under the same accept/repair/reject interface, keeping that next step inside the same registered multi-surface contract.

AI USE STATEMENT

This statement discloses generative AI use in this submission. In this work, we used generative AI tools to help implement methods and experiment code, to draft and edit parts of the manuscript for readability, and to assist retrieval and discovery of related work. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or proofs, propose or refine scientific hypotheses, design the research methodology or experimental protocol, clean or reformat datasets, perform qualitative thematic analysis, or interpret experimental results; those required-disclosure tasks were performed by the authors or are not applicable. We have reviewed all AI-assisted code and text: experiment scripts were verified and executed by the authors against registered evaluation protocols; related-work candidates suggested by generative AI were read and vetted by the authors before any citation was added; and all scientific claims, numbers, and citations were checked by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This ethics statement records the evaluation scope of the present work. This work evaluates a runtime control layer on publicly released medical imaging benchmarks and educational image-challenge cases under source-identity scrubbing. No new patient recruitment, clinical intervention, or prospective bedside deployment was performed. Reported operating points are offline evaluation artifacts; prospective clinical use and live billing remain outside the present evidence, as scoped in the Discussion and in Appendix A.8 and Appendix A.9. Authors adhere to the International Conference on Learning Representations (ICLR) Code of Ethics.

REPRODUCIBILITY STATEMENT

This reproducibility statement points to the artifacts that support the reported results. We provide an anonymous code link in the abstract. Appendix A documents shared runtime objects, routing and metric contracts, coined-arm protocols, additional tables, and the reproducibility boundary that binds each scored table row to regenerable artifacts. Main Live512 / Mix4 operating points are recoverable from the released package under the same packet–route–decision-trace interfaces described in the Method.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George B. M. van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pp. 2206–2240, 2022.

Hanna Borgli, Vajira Thambawita, Pia H. Smedsrud, Steven Hicks, Debesh Jha, Sigrun L. Eskeland, Kristin R. Randel, Konstantin Pogorelov, Mathias Lux, Duc Tien Dang Nguyen, Dag Johansen, Carsten Griwodz, Hakon K. Stensland, Enrique Garcia-Ceja, Peter T. Schmidt, Hugo L. Hammer, Michael A. Riegler, Pal Halvorsen, and Thomas de Lange. Hyperkvasir, a comprehensive multiclass image and video dataset for gastrointestinal endoscopy. *Scientific Data*, 7(1):283, 2020. doi: 10.1038/s41597-020-00622-y.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017. doi: 10.1038/nature21056.

Adibvafa Fallahpour, Jun Ma, Alif Munim, Hongwei Lyu, and Bo Wang. MedRAX: Medical reasoning agent for chest x-ray. *arXiv preprint arXiv:2502.02673*, 2025.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799, 2023.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016. doi: 10.1001/jama.2016.17216.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnu-net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z.

Heinrich Jiang, Been Kim, Melody Guan, and Maya R. Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

Christopher J. Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, 17:195, 2019. doi: 10.1186/s12916-019-1426-2.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. Mdagents: An adaptive collaboration of LLMs for medical decision-making. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-2522.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, et al. Mmedagent: Learning to use medical tools with multi-modal agent. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 8745–8760, 2024a. doi: 10.18653/v1/2024.findings-emnlp.510.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36, 2024b.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for mind exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, pp. 51991–52008, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xiaoxuan Liu, Samantha C. Rivera, David Moher, Melanie J. Calvert, Alastair K. Denniston, SPIRIT-AI, and CONSORT-AI Working Group. Reporting guidelines for clinical trial reports for interventions involving artificial intelligence: The consort-ai extension. *Nature Medicine*, 26 (9):1364–1374, 2020. doi: 10.1038/s41591-020-1034-x.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024. doi: 10.1038/s41467-024-44824-z.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Scott M. McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S. Corrado, Ara Darzi, Mozziyar Etemadi, Felicity Garcia-Vicente, Fiona J. Gilbert, Mark Halling-Brown, Demis Hassabis, Sunny Jansen, Alan Karthikesalingam, Christopher J. Kelly, Dominic King, Joseph R. Ledsam, David Melnick, Hossein Mostofi, Lily Peng, Joseph J. Reicher, Bernardino Romera-Paredes, Rachel Sidebottom, Mustafa Suleyman, Daniel Tse, Kenneth C. Young, Jeffrey De Fauw, and Shravya Shetty. International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94, 2020. doi: 10.1038/s41586-019-1799-6.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M. Krumholz, Jure Leskovec, Eric J. Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023. doi: 10.1038/s41586-023-05881-4.

New England Journal of Medicine. NEJM image challenge. Online clinical image quiz, 2024. Public educational cases.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision, 2023.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Andre G. C. Pacheco, Gustavo R. Lima, Amanda S. Salomao, Breno Krohling, Igor P. Biral, Gabriel G. de Angelo, Fabio C. R. Alves, Jose G. M. Esgario, Alana C. Simora, Paulo B. C. Castro, Fabio B. Rodrigues, Paulo H. L. Frasson, and Renato A. Krohling. Pad-ufes-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. *Data in Brief*, 32:106221, 2020. doi: 10.1016/j.dib.2020.106221.

Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, and Fabrice Meriaudeau. Indian diabetic retinopathy image dataset (idrid): A database for diabetic retinopathy screening research. *Data*, 3(3):25, 2018. doi: 10.3390/data3030025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Alaa Babiker, Nathanael Scharli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL*, pp. 599–621, 2024. doi: 10.18653/v1/2024.findings-acl.33.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5: 180161, 2018. doi: 10.1038/sdata.2018.161.

Tao Tu, Mike Schaekermann, Anil Palepu, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Yong Cheng, Elahe Vedadi, Nenad Tomasev, Shekoofeh Azizi, Karan Singhal, Le Hou, Albert Webson, Kavita Kulkarni, S. Sara Mahdavi, Christopher Semturs, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S. Corrado, Yossi Matias, Alan Karthikesalingam, and Vivek Natarajan. Towards conversational diagnostic artificial intelligence. *Nature*, 642(8067): 442–450, 2025. doi: 10.1038/s41586-025-08866-7.

Blake Vasey, Myura Nagendran, Bruce Campbell, David A. Clifton, Gary S. Collins, Spiros Denaxas, Alastair K. Denniston, Livia Faes, Bart F. Geerts, Mohammed Ibrahim, Xiaoxuan Liu, Bilal A. Mateen, Piyush Mathur, Peter McCulloch, Lucy Morgan, Johan Ordish, Campbell Rogers, Suchi Saria, Daniel S. W. Ting, Peter Watkinson, Wouter Weber, Peter Wheatstone, Paul McCulloch, and DECIDE-AI Expert Group. Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: Decide-ai. *Nature Medicine*, 28(5): 924–933, 2022. doi: 10.1038/s41591-022-01772-9.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Ziyue Wang, Junde Wu, Linghan Cai, Chang Han Low, Xihong Yang, Qiaxuan Li, and Yueming Jin. Medagent-pro: Towards evidence-based multi-modal medical diagnosis via reasoning agentic workflow. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Weike Zhao, Chaoyi Wu, Yanjie Fan, Pengcheng Qiu, Xiaoman Zhang, Yuze Sun, Xiao Zhou, Shuju Zhang, Yu Peng, Yanfeng Wang, Xin Sun, Ya Zhang, Yongguo Yu, Kun Sun, and Weidi Xie. An agentic system for rare disease diagnosis with traceable reasoning. *Nature*, 651(8106):775–784, 2026. doi: 10.1038/s41586-025-10097-9.

Yukun Zhou, Mark A. Chia, Siegfried K. Wagner, Murat S. Ayhan, Dominic J. Williamson, Robbert R. Struyven, Timing Liu, Moucheng Xu, Mateo G. Lozano, Peter Woodward-Court, et al. A foundation model for generalizable disease detection from retinal images. *Nature*, 622(7981): 156–163, 2023. doi: 10.1038/s41586-023-06555-x.

Contents

The appendix contents are indexed below.

A Technical Appendix 15
A.1 Runtime objects and algorithms 15
A.2 Scrubbed routing policy 16
A.3 Coined arms and contrast baselines 16
A.4 Decision calculus and metrics 18
A.5 Component and prompt contracts 19
A.6 Additional results 19
A.7 Case studies and failure analysis 22
Qualitative case panels (Figs. 10–14) 25
A.8 Registered-surface evaluation contract 30
A.9 Reproducibility boundary 30

A Technical Appendix

This appendix collects Method notation, coined-arm protocols, routing and metrics contracts, denser sensitivity tables, and case-level mechanism traces that support the main-text Live512 / Mix4 claims. Subsections A.1–A.9 bind the tables and figures used below.

A.1 Runtime objects and algorithms

Every scored arm uses the shared Method objects below, so routing, metrics, and component contracts share one notation. The canonical decision is $d = (y, s_{\text{final}}, r, H, \tau)$ under sparse history H and budget $C(H)$; repair is intermediate control, never a terminal status (Algorithms 1–2). Notation and Algorithms 1–2 below; main-text Method keeps the PAD running example together with the compact shared objects used by every scored arm.

For each case MedHarness builds a source-identity-scrubbed packet $x = (v, q, o, m)$ whose fields are the image, task request, optional answer schema, and deployment-visible cues, with source/path/collection identifiers removed. A proposer yields $a_0 = P(x)$; the router emits a family–recipe route $r = (f, \rho) = R(x, a_0)$ from modality, task form, option schema, and signals on a_0 rather than a dataset-id switch under source-id scrubbed schema routing. Recipe ρ fixes admissible modules $\mathcal{M}_\rho$, cheap-to-heavy order, valid intermediate repairs, and lock/abstention rules. With pool $\mathcal{M}$, policy π, and budget b, the runtime records an ordered activation history $H = (m_1, \ldots, m_K) = \pi(r, x, a_0, b)$ with $m_k \in \mathcal{M}_\rho$ and cost $C(H) = \sum_{k=1}^{K} c(m_k) \le b$. Write $S = \text{supp}(H)$ only as the set of distinct opened module types; sequential control and budget use H, not set cardinality. The canonical decision object is

$$d = (y, s_{\text{final}}, r, H, \tau), \qquad s_{\text{final}} \in \{\text{accept}, \text{abstain}, \text{escalate}\},$$

where τ is the decision-trace route–trigger–evidence–repair–lock record. Repair is an intermediate control that may update the candidate inside π; it is not a terminal status.

For completeness, layered minimum-sufficient activation can be written with ordered layers $L_0, L_1, L_2 \subseteq \mathcal{M}_\rho$ for the cheap gate, targeted re-inspection/specialist, and repair support/adjudication, and sequential updates $H_0 = \emptyset$, $A_t = \pi_t(r, x, a_t, H_t, b)$, $H_{t+1} = H_t \oplus A_t$, with stop $\text{Stop}(x, a_t, H_t) = \mathbf{1}\{\text{gate}(x, a_t, H_t) = \text{pass}\}$, where Stop is the cheap-gate early-stop predicate. When evidence e_t supports repair, $a_{t+1} = \text{Repair}(x, a_t, e_t)$ and the loop continues under budget; terminal status is assigned only by the decision lock from $(x, a_{t+1}, e_t, H_{t+1})$ as s_{final}. The contrast remains always-on history $H_{\text{AO}} = \text{enum}(\mathcal{M}_\rho)$ versus sparse $H_{\text{MH}} = \pi(r, x, a_0, b)$. The trace is $\tau = (\tau^{\text{route}}, \tau^{\text{trigger}}, \tau^{\text{evidence}}, \tau^{\text{repair}}, \tau^{\text{lock}}, \tau^{\text{launch}})$. Routing/path gates with Table 2 score family, recipe, trace, and launch completeness, not final answer accuracy. The main-text Method keeps the PAD running example and compact shared objects; this appendix gives the full shared-object notation used above.

Algorithm 1: End-to-end MedHarness runtime

1. Packet $x = (v, q, o, m)$; proposer $a_0 \leftarrow P(x)$.
2. Route $r = (f, \rho) \leftarrow R(x, a_0)$; load registered recipe π, budget b, launch fields.
3. Run sparse sequential activation to obtain ordered history $H \leftarrow \pi(r, x, a_0, b)$, which may include intermediate repair updates of the candidate; write evidence into τ; charge cost $C(H)$.
4. Emit terminal decision $d = (y, s_{\text{final}}, r, H, \tau)$ with observed $s_{\text{final}} \in \{\text{accept}, \text{abstain}\}$ on scored pools.[a]

[a] API also admits escalate for out-of-profile cases; reported C/W/Abs. folds any escalate mass into Abs.

Algorithm 2: Sparse activation and decision lock

1. From a_0 and the recipe policy: if $\text{Stop} = 1$, keep a_0 accept-eligible while leaving evidence modules closed on that case.
2. On a route failure trigger, open the minimal evidence path (verification, specialist, knowledge lock); if evidence supports a recoverable error, apply intermediate Repair and recheck the updated candidate under the same budgeted loop.
3. Decision lock issues only a terminal status: accept a schema-valid, route-supported answer, else abstain. Escalate remains reserved for out-of-profile cases and is unused on scored pools. Unopened modules are unopened, not negative evidence. Repair never appears as s_{final}.

A.2 Scrubbed routing policy

Scrubbed routing policy selects family and recipe from preserved task schema and modality cues rather than collection identity. Registered profiles reach perfect family/recipe correctness under preserved schemas; the route/path gate with Table 2 checks schema-routing completeness under scrubbing rather than cross-domain answer quality. Table 8, the full-corpus route/path gate on Table 2, and scrubbed checks in Table 11 support this policy.

The four families share one case-packet interface with different task schemas: PAD-UFES-20 dermatology management, IDRiD retinal grading, NEJM Image Challenge clinical image QA, and HyperKvasir endoscopy landmarks. Recipes match the main-text instantiation: ρ_{derm} for repair-first, ρ_{idrid} for post-repair specialist continuation, ρ_{nejm} for knowledge-on-demand lock, and ρ_{endo} for a conditional specialist path. Table 8 lists compact profile keys.

Table 8: Compact runtime profile keys; surrounding text states the guards.

Surface	Cue	Modules	Lock
PAD	derm. schema	prop.+text+repair	verify; no word-only
IDRiD	DR schema	+img specialist	repair+arbitrate
NEJM	MCQ image QA	+answer-lock	schema lock/abstain
HyperKvasir	full landmarks	+endo specialist	landmark-only

PAD admits only dermatology management schemas with image evidence, not skin-like wording alone. Its recipe is a three-stage clinical chain from suspicion through diagnosis to referral, with step-local propose/verify/repair under text verifiers and no specialist card. IDRiD continues after repair to a RETFound (Zhou et al., 2023) fundus specialist and abstains on unresolved image–text conflict. NEJM opens knowledge-on-demand evidence into an answer-lock and accepts only a supported schema-valid option. HyperKvasir requires endoscopy modality plus the full landmark option schema covering cecum, pylorus, z-line, and retroflex-stomach labels, and opens a conditional DINOv2 (Oquab et al., 2023) landmark specialist; endoscopy-like wording alone does not select that profile. The operational route/path gate removes source-identity tokens from the route-facing packet while preserving task content, and defines the full-corpus scrubbed schema-routing check that Table 2 reports on the Live512 quality pool.

A.3 Coined arms and contrast baselines

For each coined policy arm and contrast baseline in Table 1, the paragraphs below state which modules open or stop and how neighboring arms differ under the shared controller. On Mix4, Role tags mark floor for proposer-only, selective for abstain without full evidence, mono for Always-on/Mono., and sparse for budgeted MedHarness; MH–Prop. / MH–Mono. contrasts are MedHarness minus that arm, with CA the correct-accept count delta at a matched coverage or wrong-accept WA operating point used for that contrast.

Prop. Prop. is the proposer-only floor: the candidate a_0 is accepted with control modules closed, so Mods. stays 0 and the path stays the raw proposer emit without repair or opened evidence steps.

Always-on / Mono. Always-on is the monolithic always-open evidence arm under the same decision lock as MedHarness: it enumerates every recipe-admissible module on every case. Mono. is the table and figure alias for that Always-on arm and shares the same always-open module path and lock. It opens the full admissible module set, keeps the lock available, and charges the always-on module-call column on every scored case in the pool.

SC-Abs. SC-Abs. is a selective self-consistency abstention baseline: it draws multiple proposer samples, accepts when they agree, and abstains on disagreement; Mods. remains 0 because evidence, repair, and specialist modules stay closed. Budgeted MedHarness still opens sparse evidence and may repair under the shared controller policy.

Text-check Text-check is a selective text-only verification gate used as a contrast baseline: it opens a light text verification module, then stops or abstains from that gate signal while leaving the full MedHarness evidence–repair–adjudicator stack closed. The MedHarness TextVerif card instead runs inside the sparse controller under a shared evidence budget and can escalate into specialist or repair support when the case needs it.

agree/conf agree/conf is a selective agree-plus-confidence gate, including the Agree/conf cascade arm in activation tables: it combines agreement with a disclosed step raw-score proxy confidence threshold, optionally cascaded, and may accept or abstain while the budgeted sparse evidence ladder stays closed. MH-strict, MH-default, and MH-loose remain evidence-budget points of MedHarness.

MH MH / MedHarness is the full sparse controller: under recipe ρ and budget b it opens modules only as needed, may repair intermediately, then locks accept / abstain / escalate. It stops at the cheap gate when eligible; otherwise it opens failure-mode evidence, repair, and adjudication under $C(H) \leq b$.

MH-strict / early-stop MH-strict is the early-stop evidence-budget operating point of the same MedHarness controller: it stops after fewer module opens when the cheap gate already supports the decision, saving calls relative to MH-default while leaving more wrong accepts on Mix4. early-stop in figures denotes this MH-strict budget and keeps the same accept/repair/reject ladder as MH-default.

MH-default MH-default is the default evidence-budget operating point used for primary Mix4 result accounting: it opens the registered sparse ladder until the decision is supported under the default budget, then stops; on Mix4 it is the joint Acc.–cost comparison versus Always-on.

MH-loose / extended-evidence MH-loose is the extended-evidence budget point of the same controller: it allows more module opens toward the always-on call column while remaining sparse under π; extended-evidence in figures denotes this MH-loose budget and keeps the same accept/repair/reject ladder rather than a separate permissive threshold.

Table 9: Ablation-arm abbreviations.

Abbrev.	Meaning	Abbrev.	Meaning
No repair	Drop intermediate repair only.	No specialist	Drop specialist evidence.
No lock	Remove terminal decision lock.	Always-on	Open modules; lock kept.
Oracle-route	Gold recipe binding.	Shuffled-route	Shuffled recipe binding.
Seq. lock-off	Live512 decision lock removed.	lock-off	Live512 Seq. lock-off Mods. shorthand.
Route-only sparse	Mix4 sparse route extreme.	route-only	Route-only sparse alias.

No repair No repair drops intermediate repair modules while the sparse evidence ladder and decision lock stay available: it closes Repair updates, charges only opened evidence/specialist/adjudicator modules, and still takes terminal accept/abstain from the lock.

No specialist No specialist drops specialist evidence modules from the admissible set: it keeps proposer, cheap gate, text verify, repair, and lock, and does not open family-specialist cards. Full MedHarness may still open specialist evidence under the registered recipe.

No lock No lock removes the terminal decision lock on Mix4 component accounting: it keeps sparse module opens and repair, and does not enforce lock accept/abstain gating as in full MedHarness, so unsupported paths can still accept.

Always-on route extreme On Mix4 route extremes, Always-on is the same monolithic open-module arm as Always-on / Mono. above: 307/205/0 at 4.00 modules with the decision lock kept. Oracle-route and Shuffled-route then isolate recipe binding at matched call scale.

Oracle-route Oracle-route binds each case to the gold registered recipe under the sparse controller: it keeps MedHarness open/repair/lock policy and replaces dispatch with oracle recipe binding so route choice is not the failure mode.

Shuffled-route Shuffled-route keeps the same sparse module budget class but shuffles recipe binding away from the registered route: it opens modules under the shuffled recipe while the lock still assigns s_{final}. Oracle-route uses gold binding at matched call scale.

Route-only sparse Route-only sparse is the Mix4 offline route extreme that keeps a sparse module path without the full MedHarness evidence–repair stack: it opens only the route modules under offline Mix4 Mods. accounting and stops short of always-on module cost, so Acc./Mods. isolate route binding under a sparse offline budget.

Seq. lock-off Seq. lock-off is the Live512 lever that removes sequential decision-lock gating on the shared controller: it runs the Live512 module ladder without the terminal lock stop used by full MedHarness, so accepts can rise when the lock would have abstained. Caption shorthand lock-off refers to this arm's Mods. column.

A.4 Decision calculus and metrics

Every comparative table below uses one decision calculus and metrics set so C/W/Abs., accepted accuracy, rescue/harm, and module cost stay comparable across arms. Each scored case is exactly one terminal outcome (correct accept, wrong accept, or abstention); intermediate repair never appears as s_{final}, and escalate folds into Abs. so $C{+}W{+}Abs{=}N$. Definitions below; runtime gate checklist Table 10; primary Live512 / Mix4 rows in the main-text tables.

The decision calculus uses main-text case-level accounting (C/W/Abs., Acc., R/H, Mods.). With indicators c_i, w_i, u_i for those events, $C_h = \sum_i \mathbf{1}[c_i]$, $W_h = \sum_i \mathbf{1}[w_i]$, $A_h = \sum_i \mathbf{1}[u_i]$, $\mathrm{Rescue}(h) = \sum_i \mathbf{1}[a_0^{(i)} \neq y_i^\star \wedge c_i]$, and $\mathrm{Harm}(h) = \sum_i \mathbf{1}[a_0^{(i)} = y_i^\star \wedge w_i]$, where Rescue counts proposer-wrong cases that become correct accepts and Harm counts proposer-correct cases that become wrong accepts. Accepted accuracy is $C_h/(C_h + W_h)$ when coverage is nonzero; rescue and harm are paired against proposer-only. Coverage is $(C_h + W_h)/N$ and risk-among-accepts is $W_h/(C_h + W_h)$ when coverage is nonzero; Table 7 prints these as Cov/Rsk. Mix4 utility uses fixed author-chosen safety-oriented weights $U = C_h - W_h - 0.5A_h - 0.1\,\overline{C(H)}\,N$ with pool size $N = 512$ and $\overline{C(H)}$ the mean Mods. column; the same weights give Prop. 0.0, Always-on -102.8, MH-strict 108.3, MH-default 174.6, and MH-loose 77.2. The routing/path gate scores family, recipe, trace, and launch completeness independently of final-answer correctness. The decision lock's terminal API is $s_{\text{final}} \in \{\mathrm{accept}, \mathrm{abstain}, \mathrm{escalate}\}$: accept when $y \in o$ and route-compatible evidence supports y; abstain under conflict, schema violation, or insufficient support; escalate when the case leaves the configured profile. Reported C/W/Abs. always folds escalate into Abs. as a non-accept reject, so reported mass is exhaustive ($C{+}W{+}Abs{=}N$) on every scored arm. Primary default Live512 and Mix4 scored rows have escalate at zero; the live shuffled-route control is $196/179/137$ after folding ten PAD escalate-to-human outcomes into Abs. Intermediate repair may propose a recheckable a_1 before the lock, but repair is never a terminal status. Unopened modules do not vote; opened modules must write evidence into τ before the lock can use it.

Runtime gate checklist (inputs / rule / fail action). The four control gates under shared policy π are recorded as inputs / rule / fail-action triples. Stop/accept, open evidence, repair, and reject/abstain are recipe-adapted thresholds under one registered policy rather than private per-surface stacks. Table 10.

Table 10: Runtime gate checklist under the shared control policy (rule-based / registered). Rule cells are short labels; full wording is in the paragraph above.

Gate	Inputs	Rule	On fail
Stop / accept	score, budget	threshold accept	open evidence
Open evidence	failure mode	typed module open	skip
Repair	verifier fail	$\leq K$ repair	abstain
Reject / abstain	residual risk	unjustified $\rightarrow$ Abs.	terminal Abs.

A.5 Component and prompt contracts

Role-specific prompt interfaces sit on the normalized packet so modules stay swappable under one registration contract. Proposer, verification, repair, specialist evidence, and decision lock share the same packet schema; contracts are interfaces, not case-specific prompts. Role list below; full-corpus scrubbed route/path gate on Table 2.

Each component uses a role-specific prompt template over the normalized packet: proposer candidate; text/secondary verification; intermediate repair under the same schema; artifact specialist evidence; decision lock $d = (y, s_{\text{final}}, r, H, \tau)$. In a typical clinical image QA packet, the proposer selects an option, answer-lock evidence is opened when triggered, supported candidates accept, contradicted candidates repair-and-recheck, and unsupported cases abstain. Table 2's full-corpus scrubbed route/path gate retains deployment-visible task schema after source scrubbing and bounds the claim to registered surfaces that still expose modality and answer-schema cues after scrubbing.

A.6 Additional results

Denser sensitivity, routing-check, specialist-control, and pool-role tables support the main-text Live512 / Mix4 claims without crowding the Results narrative. Routing checks stay path-complete with zero source-id residual; same-pool activation still ranks MedHarness over always-on on wrong accepts; Live512 and Mix4 keep distinct evaluation roles. Tables 11, 12, 13, 17, 18, 19, 20; matched-selective coverages and Table 4.

Main-text Tables 2, 4, 6, and 7 and Figure 5 already report multi-surface attachment, when-to-stop, Mix4 ranking with selective baselines and bootstrap CIs, live claim-critical ablations, discrete operating points, and three-mix ranking. The matched-selective extras not printed in Table 4 are the realized coverages under cross-fitted calibration: matched-coverage arms reach 0.865 / 0.891 / 0.891 and matched-WA arms reach 0.379 / 0.195 / 0.672 for SC-Abs. / text-check / Mono. Paired bootstrap intervals remain in Table 4 (all six intervals exclude zero). Claim-critical lever isolation on Live512 is Table 6; Mix4 component and route-extreme rows are main-text Table 5. Routing replay and leakage checks:

Table 11: Routing replay and offline leakage checks (no fresh model call; not live answer quality).

Check	n	Result
Full routing gate	4848	path 1.000/1.000, 1.000/1.000; 0 source-id residual
512-case replay	512	path 1.000/1.000, 1.000/1.000; 0 source-id residual
Raw metadata check	512	leak hits 512/512
Scrubbed task-text check	512	0 family-token residual; 0/512 text-only hits
Metadata-only baseline	512	512/512 route hits
Majority baseline	512	128/512 route hits
Det. random baseline	512	139/512 route hits

HyperKvasir specialist / cascade controls. Because HyperKvasir uses a very strong specialist, the following HK-128 block is registration-scale attachment evidence only and is excluded from primary when-to-stop ranking; main-text Table 2 uses Live512.

Table 12: HK-128 specialist / cascade controls under live GPT-5.4. Registration-scale evaluation only; excluded from primary when-to-stop rank.

Control	C/W/Abs.	Acc.↑	Role	n
Proposer-only	64/64/0	0.50	floor	128
Prop.+spec. cascade	121/0/7	1.00	cascade	128
Always-on	54/74/0	0.42	mono	128
Specialist-only	128/0/0	1.00	oracle-like	128
MedHarness	119/0/9	$1.00_{\uparrow +0.58}$	sparse	128

Same-pool activation policies with specialist kept in the pool. Activation policies are ranked on a fixed proposer, scorer, and module pool (specialist included) so fairness gaps come from when-to-stop rather than a private tool set. On Live512, MedHarness 367/74/71 cuts wrong accepts versus always-on 373/139/0; on Mix4-HKout, MedHarness 291/46/47 (Acc. 0.864) beats always-on 253/131/0 (0.659). Removing specialist erases the Live512 WA win (351/146/15). Table 13; main-text Table 3.
Specialist/repair/lock modules are admissible scheduled components. Table 13 lists proposer-only, Always-on, MedHarness sparse, and a deployable agree/conf cascade whose non-HK confidence equals the disclosed step raw-score proxy. Removing specialist is a component ablation only.

Table 13: Same-pool activation policies on Live512 (HyperKvasir excluded) and Mix4-HKout. Agree/conf uses disclosed step raw-score proxy confidence.

Pool	Arm	C/W/Abs.	Acc.↑	Mods.↓	Evid.
Live512	Proposer-only	301/211/0	0.588	0.00	filled
Live512	Always-on	373/139/0	0.729	4.00	live
Live512	MedHarness	367/74/71	0.832$_{\uparrow+0.103}$	3.86$_{\downarrow-0.14}$	live
Live512	Agree/conf cascade	302/153/57	0.664	2.79	offline
Live512	No specialist	351/146/15	0.706	3.20	live
Mix4-HKout	Proposer-only	192/192/0	0.500	0.00	cached
Mix4-HKout	Always-on	253/131/0	0.659	4.00	live
Mix4-HKout	MedHarness	291/46/47	0.864$_{\uparrow+0.205}$	3.87$_{\downarrow-0.13}$	live
Mix4-HKout	Agree/conf cascade	164/179/41	0.478	1.74	offline

Live512 proposer-only is now the filled n=512 GPT-5.4 floor (301/211/0; Acc. 0.588), matching the other Live512 arms rather than the earlier half-pool cache.

Why Live512 $\neq$ Mix4 (evaluation roles). Live512 and Mix4 stay non-substitutable so quality and stress-efficiency claims do not trade places. Live512 (PAD/IDRiD/NEJM 171/171/170; HyperKvasir excluded) is the primary live when-to-stop quality pool; Mix4 is the complementary failure-enriched stress ranking for budgeted efficiency under the same runtime. Live512 wrong-accept / accepted-accuracy gains versus always-on in Table 2; Mix4 offline token/call burden in Table 14.

Table 14: Mix4 offline context-load token/call burden under estimated context-load accounting, distinct from Live512 provider billing. Same HyperKvasir-inclusive stress pool as Table 4.

Method	Avg est. tokens↓	API calls/case↓
Proposer-only	1962	0.75
Always-on	25003	8.09
MedHarness	**7706**	**2.38**

Table 15: Live512 selective extremes under the same full module set as Table 2.

Arm	C/W/Abs.	Acc.↑
SC-Abs.	258/209/45	0.552
Text-check	277/234/1	0.542
SC-Abs.	0/0/512	–
Text-check	0/0/512	–

Protocol, split role, light contamination, and discrete utility. Protocol note: the 512-case Mix4 pool is a fixed balanced compare set (128 cases per family); surface recipes and operating-point thresholds were fixed offline before evaluating that pool. The five-split mixed256 table (Table 18) is a controlled-overlap stability fallback under the same offline accounting protocol. Light contamination check finds no duplicate source case ids and scrubbed routing family accuracy 0.996 with zero residual explicit family tokens on the 512-case pool. Table 7 lists discrete risk–coverage / utility-surface points aligned with main-text Figures 3 and 4; under the Mix4 utility weights in Appendix A.4, MedHarness-default utility is 174.6 versus always-on -102.8, with MH-strict 108.3 and MH-loose 77.2 on the same ladder.

Proposer re-measure under GPT-5.4. An independent GPT-5.4 proposer-only pass on the full 512-case pool is Table 16 versus the cached proposer floor and MedHarness on the same case ids. This is a same-backbone full-n re-measure of the proposer floor, distinct from the alternate-provider full MedHarness transfer already reported in the transfer section above.

Table 16: GPT-5.4 proposer-only re-measure on Mix4 512-case ids as same-backbone floor check.

Method	C/W/Abs.	Acc.↑
Proposer-only	256/256/0	0.500
Proposer-only	274/238/0	0.535
MedHarness	**410/46/56**	**0.899**

Mix4 component and route extremes. Main-text Table 5 reports Mix4 components and route extremes. Under the same offline accounting, Route-only sparse is a single-arm extreme recorded in prose only: C/W/Abs. $260/252/0$, Acc. 0.508, and Mods. 0.85, without a printed comparison table. Live512 lever isolation remains Table 6. Shuffled-route on Live512 is the 196/179/137 row there; Mix4 shuffled-route is the main-text right-panel 184/272/56 row.

Leave-one-family, five-split, and reweight diagnostics. Main-text Table 7 keeps only discrete operating points and utility. The denser LOFO / five-split / reweight diagnostics are appendix-only sensitivity under the same Mix4 accounting. *Three-mix distribution robustness* remains in main-text Figure 5 (Balanced / NEJM-heavy / PAD-heavy 512-case mixes).

Table 17: Leave-one-family re-aggregation on the 512-case pool (n=384 remaining; sensitivity only).

Out	MH C/W/Abs.	Acc.↑	Mods.↓
PAD	308/25/51	.925	3.15
IDRiD	339/31/14	.916	1.53
NEJM	292/36/56	.890	4.06
HK	291/46/47	.864	3.87

Table 18: Five-split mixed256 stability (C/W/Abs. and MH R-H; sensitivity only).

Sp.	Prop.	Mono.	MH	R-H
00	128/128/0	150/106/0	200/28/28	80
01	128/128/0	153/103/0	199/29/28	76
02	128/128/0	157/99/0	207/23/26	87
03	128/128/0	160/96/0	207/25/24	85
04	128/128/0	149/107/0	202/27/27	78

Table 19: Same-pool hierarchical diagnostics on non-endoscopy 512. Main-text from-scratch is Table 6; 297/215/0 is reuse only. Both arms are hierarchical plan→execute baselines adapted from MedAgent-Pro (Wang et al., 2026) under our packet on the same GPT-5.4 live ids (published MedAgent-Pro used GPT-4o). From-scratch mean modules are 4.72; the reuse arm reports no matched module-count field.

Method	C/W/Abs.	Acc.↑	n	Evidence
MedHarness (default)	**367/74/71**	**0.832**	512	live
Hier. from-scratch (Wang et al., 2026)	254/257/1	0.497	512	live
Hier. reuse (F6) (Wang et al., 2026)	297/215/0	0.580	512	live+reuse

Specialist and fairness controls. Main-text Table 4 ranks matched-policy fairness controls, and Table 12 holds the HyperKvasir-128 specialist evidence modes: always-on+lock is the observed monolithic arm; sparse-no-lock force-accepts unlocked candidates on the default sparse trace (offline / non-causal sensitivity only); always-specialist is a heterogeneous proxy (true DINOv2 (Oquab et al., 2023) artifact head on HyperKvasir; other families use available specialist-like evidence).
Leave-one-family re-aggregation. Table 17 re-aggregates the 512-case pool after holding out one family (n=384 remaining) and checks ranking stability when one family is removed from the mixed pool. Mods. (P/M/MH) under hold-outs: PAD 0.00/4.00/3.15; IDRiD 0.00/4.00/1.53; NEJM 0.00/4.00/4.06; HyperKvasir 0.00/4.00/3.87.

Table 20: Stress vs. reweight wrong-accept/accepted-accuracy metrics (sensitivity-only under Mix4 four-surface accounting).

View	Mono. WA/Acc.	MH WA/Acc.	n/ESS	Mods.↓
Stress	.40/.60	.09/.90	512	3.15
Nat. rewt.	.36/.64	.07/.92	ESS457	–
Full+nat.	.55/.45	.02/.98	ESS186	1.42

Main-text three-mix bars already show that wrong-accept mass moves with queue composition across Balanced / NEJM-heavy / PAD-heavy mixes (Figure 5) while the MH-over-always-on ranking holds. Reweight rows keep the balanced 512-case pool as the stress pool and reweight the same observed traces: (i) natural proposer-error rates with equal family share; (ii) full-surface family counts times natural proposer rates (HyperKvasir 3704, IDRiD 259, NEJM 384, PAD 200 as prevalence proxies, not external epidemiology tables). ESS shrinks under importance weights.

External baseline, alternate-provider transfer, scrubbed routing check, and reproducibility artifact. Hierarchical baseline contrast, alternate-LLM transfer, scrubbed routing check, and the reproducibility artifact stay under one packet so external rows remain diagnostic rather than a second primary claim. Hierarchical from-scratch on the same GPT-5.4 Live512 ids is 254/257/1 (Acc. 0.497); reuse is 297/215/0 (Acc. 0.580); scrubbed routing replay stays path-complete with zero residual family tokens. Tables 19, 6, 21, 11; Appendix A.7.
Table 19 keeps the prior hierarchical reuse row (192 reused + 320 residual): 297/215/0 Acc. 0.580. The main-text from-scratch hierarchical run on the same GPT-5.4 Live512 ids is 254/257/1 Acc. 0.497 at 4.72 modules (Table 6), not this reuse row and not published GPT-4o. Protocol detail for the adapted hierarchy and the alternate-LLM Live512 transfer follow Appendix A.7 and Table 21. Scrubbed routing replay on the same 512-case ids remains path-complete with zero residual explicit family tokens (Table 11). The dual-panel main-text component table already absorbs the Mix4 route-extreme rows under the same offline accounting.

A.7 Case studies and failure analysis

Case studies and failure analysis show stop / open / repair / abstain under one shared controller across registered surfaces. MedHarness recovers SCC accept after step-local repair on PAD and abstains on IDRiD fundus instead of forcing a wrong grade; always-on still wrong-accepts on the honesty panel. Figures 6–9; Tables 21, 22.

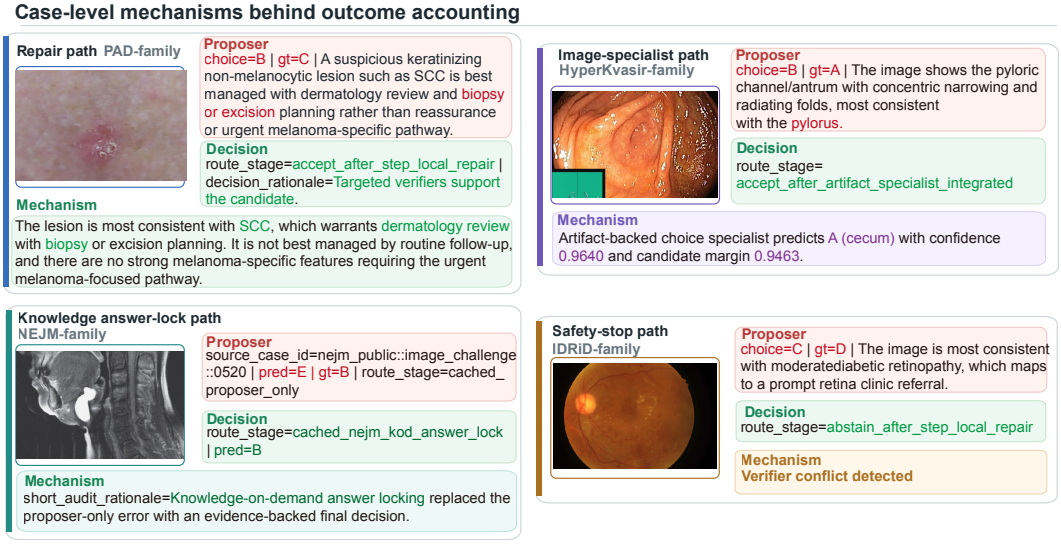


Figure 6: Four traces from the 512-case pool showing PAD repair, HyperKvasir image specialist, NEJM knowledge lock, and IDRiD safety-stop abstention.

Figure 6 keeps the four mechanism traces on the same accept/repair/abstain ladder used by the Live512 accounting tables above, so the page records stop/open/repair/abstain before the same-case three-arm strips that follow.

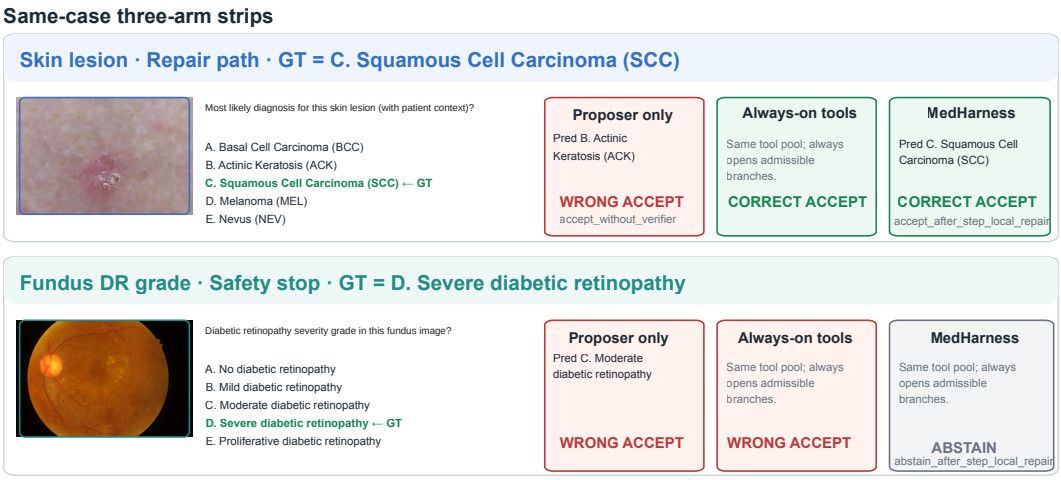


Figure 7: Same-case three-arm strips on the live pool: proposer-only, always-on tools, and MedHarness under one shared tool pool (PAD repair rescue and IDRiD safety-stop abstention shown).

Figure 6 shows mechanism traces; Figure 7 is the same-case fairness comparison for Table 13: proposer-only, always-on tools, and MedHarness share one tool pool, so gaps come from when to stop, open, repair, or abstain rather than from a private module set (Chow, 1970; Geifman & El-Yaniv, 2017; Madaan et al., 2023).

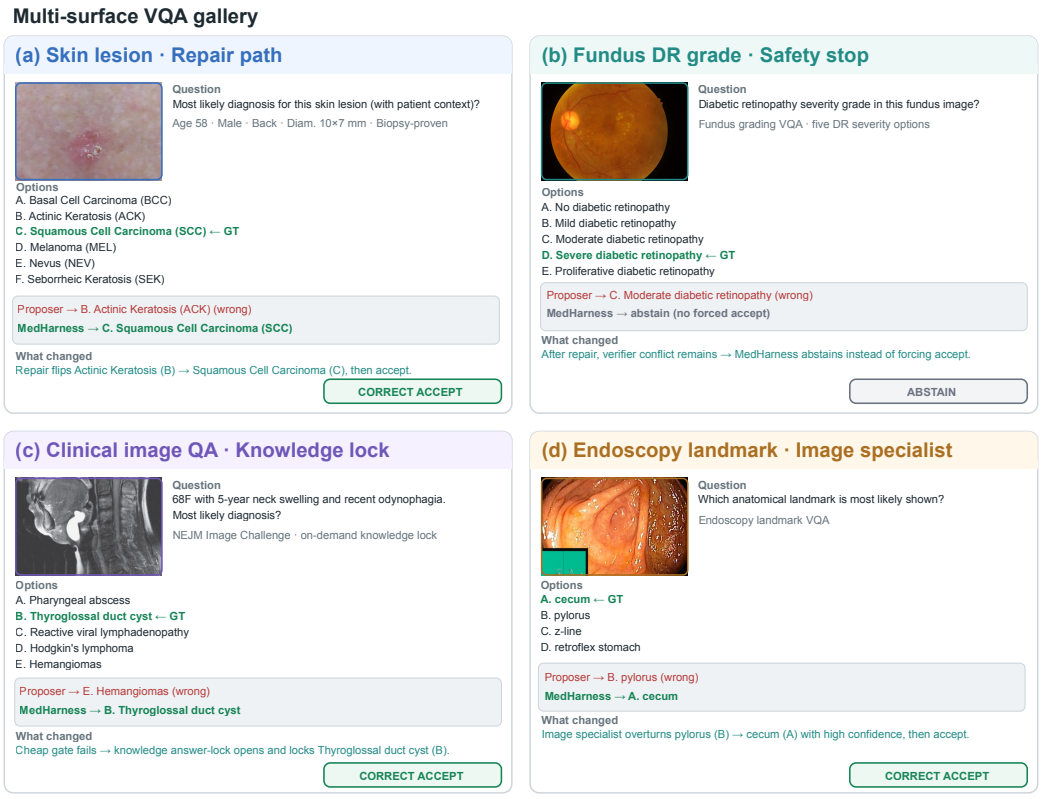


Figure 8: Multi-surface VQA gallery on the Live512 pool with endoscopy holdout specialist and HyperKvasir excluded from primary live when-to-stop: clinical image, full GT-marked options, proposer versus MedHarness decision, and each panel's accept/abstain control change.

Figure 8 is the complementary multi-surface VQA gallery: each panel keeps the clinical image, full options with the ground-truth marked, the proposer's wrong first answer, the MedHarness accept/abstain decision, and a short control-change note across PAD repair, IDRiD safety-stop abstention, NEJM knowledge lock, and HyperKvasir landmark correction. Surfaces register adapters

under one shared policy π; unseen surfaces still need an explicit recipe. Pool roles for Live512 versus Mix4 sit in Appendix A.6.

Table 21: Alternate-LLM Live512 transfer with proposer/verifier/repair/lock swapped and specialists fixed. HyperKvasir-held-out activation rank remains main-text Table 3.

Method	C/W/Abs.	Acc.↑	Mods.↓
Proposer-only	303/209/0	0.592	0.00
SC-Abs.	288/191/33	0.601	0.00
Always-on	341/171/0	0.666	4.00
MedHarness	**306/113/93**	**0.730**$_{\uparrow+0.064}$	**3.29**$_{\downarrow-0.71}$

Table 22: Capability contrast versus a MedAgent-Pro hierarchical baseline and always-on stacks with matched MedHarness packet schema.

Capability	MedAgent-Pro (Wang et al., 2026)	Always-on	MH
Plan / tools / reflect	yes	partial	upstream
Budgeted stop	n/a	no	yes
Failure-typed open	n/a	always-on	sparse
Repair before reject	n/a	weak/none	yes
Shared registration	n/a	private	π+adapters

Tables 21 and 22 give the matching alternate-LLM transfer and capability contrast under the same shared controller that closes this case-study block.

Failure / abstain honesty

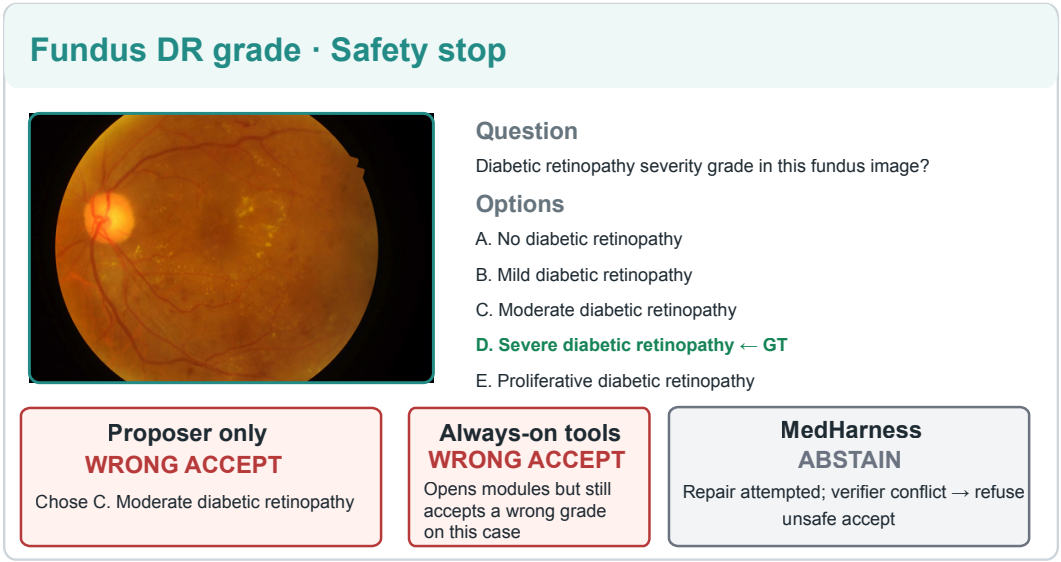


Figure 9: IDRiD fundus honesty panel: proposer-only and always-on tools wrong-accept; MedHarness abstains after repair on this case.

Figure 9 is the case-level when-to-stop case for the Live512 safety-stop claim: on this IDRiD fundus grade, MedHarness abstains after repair while always-on tools still wrong-accept. Abstain is a first-class budgeted outcome under the shared controller, and the same stop / open / repair / reject contract governs the transfer row and the hierarchical diagnostic arms below.

Adapted hierarchy protocol. The adapted hierarchy protocol used in Table 19 and Table 22 is bounded here. Both arms are *adapted* plan→execute runs under MedHarness evaluation objects on the same GPT-5.4 Live512 ids with matched module accounting and packet schema; published MedAgent-Pro (Wang et al., 2026) used GPT-4o with its own toolset/RAG. Primary activation fairness stays on the same-pool table (Table 13); the hierarchy rows are baseline diagnostics only. A prior hierarchy support run on 256 cases remains a half-pool diagnostic in Table 19. Grok-4.5 same-proposer Live512 is the alternate-LLM transfer (Table 21; MH Acc. 0.730 vs proposer-only 0.592). The source-identity-scrubbed routing check stays near-ceiling on the 512-case pool (Table 11). The

public artifact comprises 365 files with an expected case-level flip bound at most 2% under matched model settings for main operating-point recheck.

Qualitative case panels. Multi-surface qualitative panels show the same registered control contract that Live512 measures quantitatively. Figures 10–14 show accept/repair/abstain and step-local repair under one shared policy π, complementing the mechanism traces in Figures 6–9. The wall, image-to-report strips, and decision-card panels keep family-local thumbs with the same accept/repair/abstain labels used in the Live512 accounting tables. Each panel is disjoint from the mechanism and honesty cases above, so the qualitative pages illustrate the same stop contract without reusing those traces. Figures 10–14 then open as full-page panels under that shared contract. HyperKvasir rows remain registration-scale / Mix4 coverage; Live512 remains the primary when-to-stop pool (Geifman & El-Yaniv, 2017; Madaan et al., 2023).

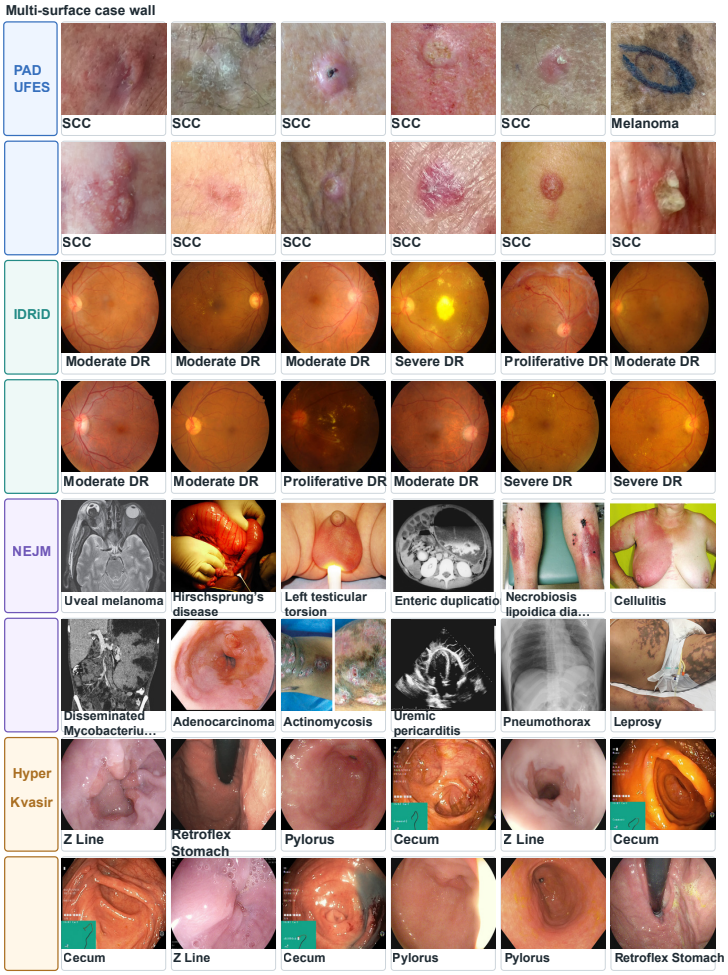


Figure 10: Four-family disease/modality wall on the registered pool: PAD-UFES-20 dermatology, IDRiD fundus grades, NEJM Image Challenge diagnoses, and HyperKvasir landmarks with short per-tile labels. Cases are disjoint from the mechanism and failure/abstain panels above.

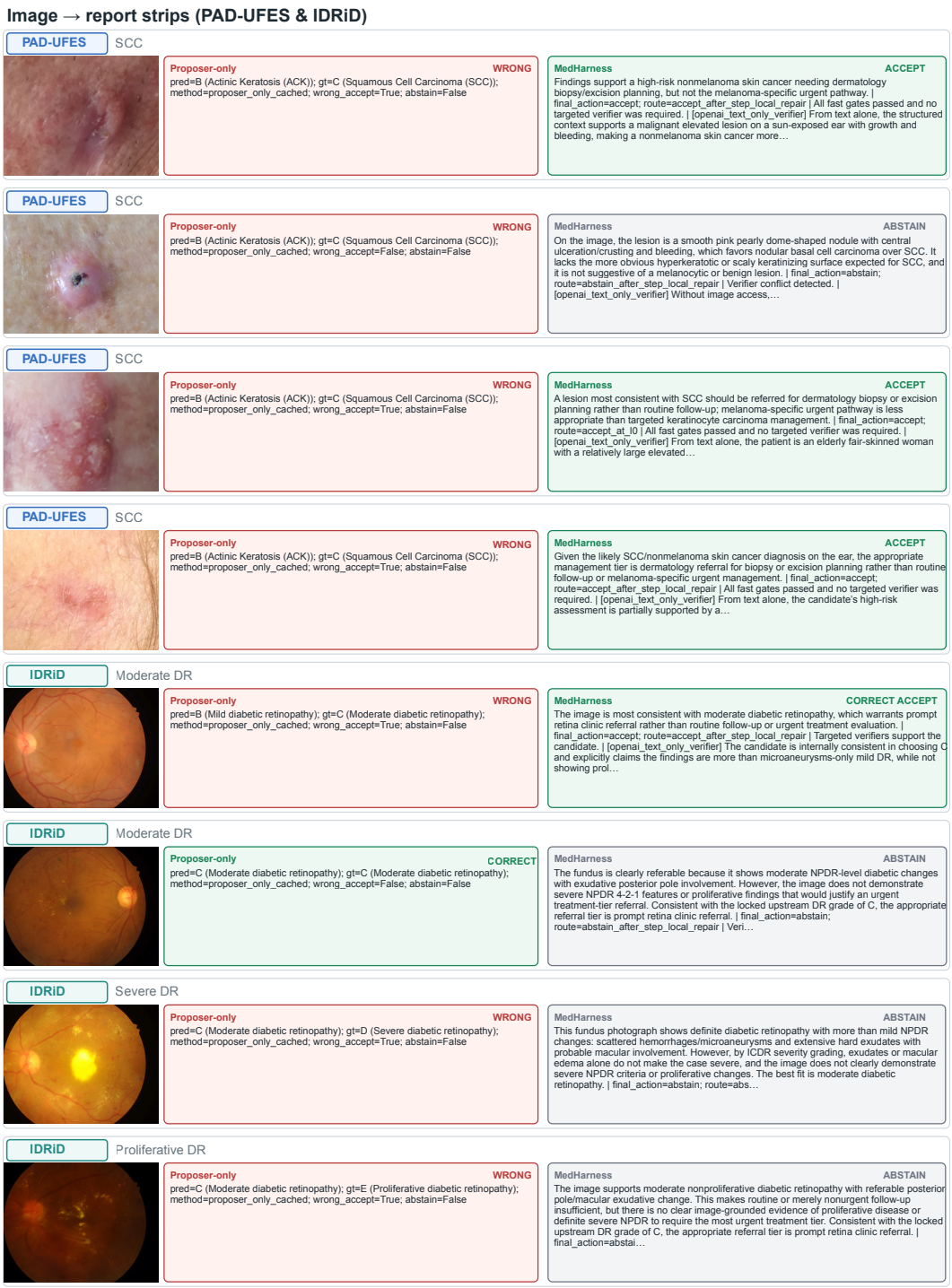


Figure 11: Image→report strips for PAD-UFES-20 and IDRiD: original clinical image beside the MedHarness final narrative, with green highlights marking accepted findings after repair rescue or safety-stop abstain under the shared budgeted controller.

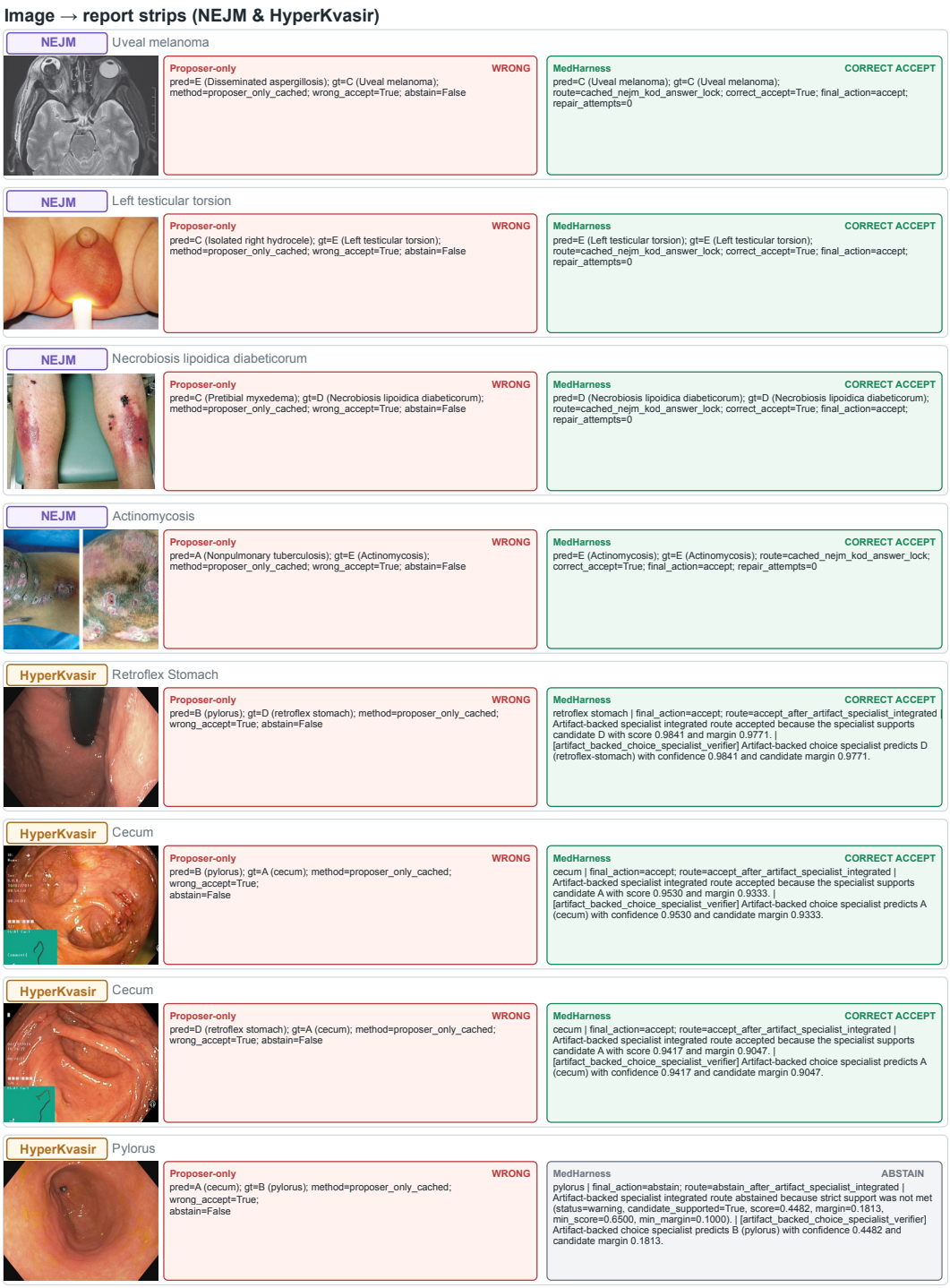


Figure 12: Image→report strips for NEJM Image Challenge and HyperKvasir: original image beside the MedHarness final narrative, with green highlights marking knowledge-lock and endoscopy-specialist corrections under the registered recipes.

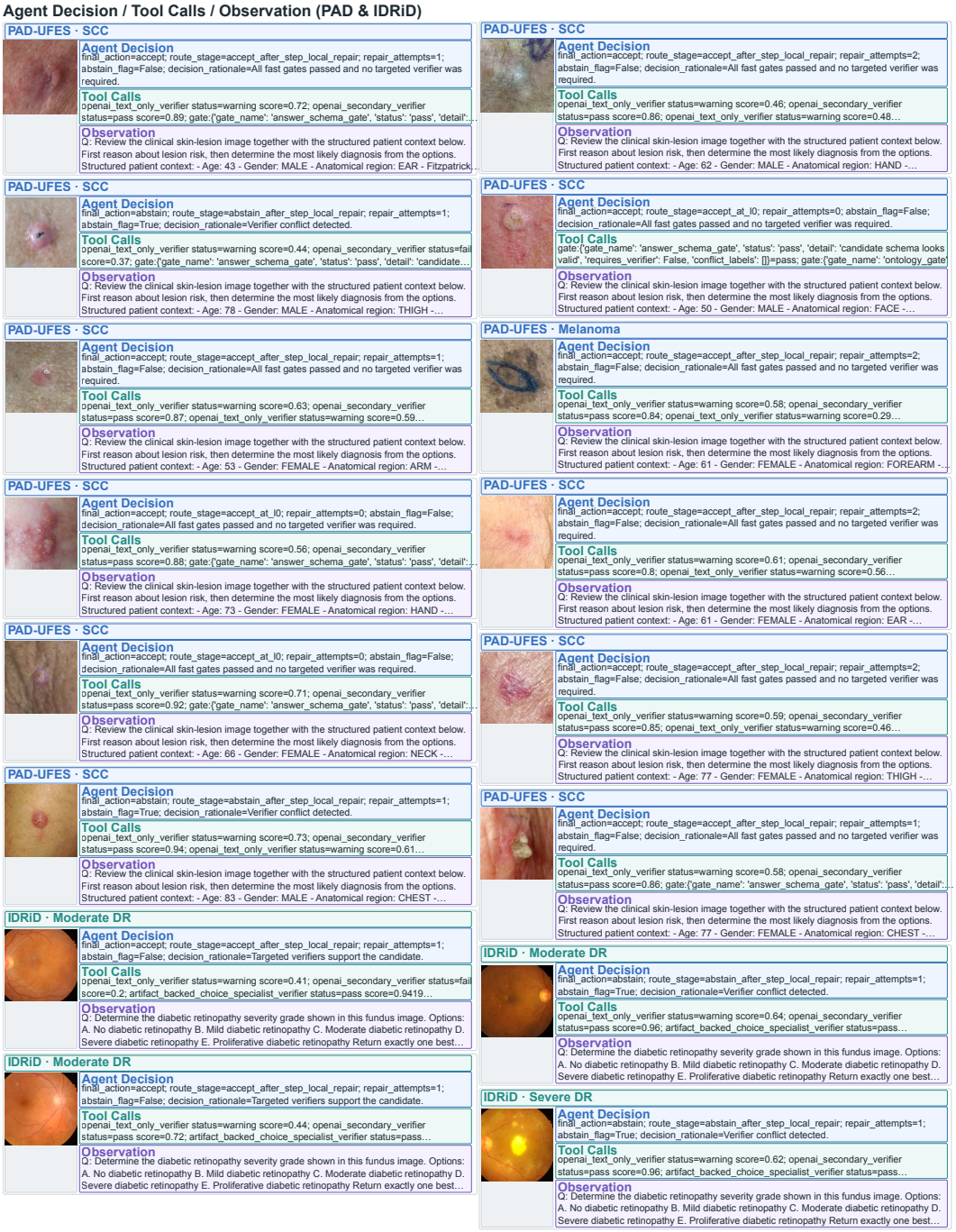


Figure 13: Agent decision / tool / observation cards for PAD-UFES-20 and IDRiD: each card keeps a clinical thumb with Decision, Tool call, and truncated Observation fields taken from the registered live decision trace for that case.



Figure 14: Agent decision / tool / observation cards for NEJM Image Challenge and HyperKvasir: each card keeps a clinical thumb with Decision, Tool call, and truncated Observation fields taken from the registered live decision trace for that case.

Figures 10–14 close the qualitative case evidence for the mechanism and failure/abstain studies above (Wang et al., 2026; Borgli et al., 2020): the wall shows real-image coverage across PAD-UFES-20 and IDRiD (Pacheco et al., 2020; Porwal et al., 2018), the report strips give readable finals beside the VQA option panels (Figure 8), and the decision cards keep decision-trace-faithful tool traces under one shared control policy π.

A.8 REGISTERED-SURFACE EVALUATION CONTRACT (EXPANDED)

The registered-surface evaluation contract extends beyond a hierarchical plan/tool/reflect baseline and a private always-on stack under the same packet–route–decision-trace contract. MedHarness composes with upstream plan/tool/reflect and adds budgeted stop, sparse failure-mode evidence open, repair-before-reject, and multi-surface shared registration. Table 22 and Figure 9 bind that contrast; the qualitative panels earlier in this appendix supply the case-level view.

Capability contrast versus MedAgent-Pro. Beyond a MedAgent-Pro hierarchical plan/tool/reflect stack (Wang et al., 2026) and a private always-on stack, MedHarness adds budgeted stop, sparse failure-mode evidence open, repair-before-reject, and multi-surface shared registration under one packet–route–decision-trace contract while composing with upstream plan/tool/reflect. Table 22 and Figure 9 bind that capability contrast for the registry row and the case-level honesty panel.

A.9 REPRODUCIBILITY BOUNDARY

Every reported table row binds to a regenerable artifact package under the same packet, route, and decision-lock interfaces used in the main text. Implementation, runtime manifests, result summaries, and regeneration notes use anonymized packet handles rather than machine-local paths, so offline accounting stays matched to the scored arms. Anonymous code URL in the abstract; artifact file count and flip bound in Appendix A.7 (365 files; $\leq 2\%$ case-level flip under matched settings).