GUI-EVOLVER: SCALING GUI AGENTS VIA DATA–MODEL CO-EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce GUI-Evolver, an autonomous data–model co-evolution framework for GUI agents. Student interactions update a persistent State Model Graph (SMG), which supplies continual GUI knowledge and failure-driven remedial tasks. Accumulated knowledge and admitted trajectories train the next student, selected by Validation without per-round human decisions. Teacher-assisted intervention and diagnosis-guided rerollouts provide alternative targets when teacher reasoning is unavailable. Three-evaluation means on Qwen3.5-0.8B/2B/4B/9B and Qwen3.8-27B improve over Base by 9.9–25.0 points on AndroidWorld and 4.8–13.1 on MobileWorld. Controlled comparisons support the contributions of explicit knowledge, targeted practice, and student-compatible supervision within the evaluated settings.

1 INTRODUCTION

Visual language models increasingly connect screen understanding to GUI actions (Hong et al., 2024; Cheng et al., 2024; Wu et al., 2025), but a wrong selection, overlooked prerequisite, or repeated ineffective action can derail an otherwise feasible task. AndroidWorld (Rawles et al., 2025) exposes these failures through execution outcomes. Collecting more rollouts is not itself a curriculum: a useful training pipeline must determine which operation needs practice and how to obtain supervision the student can use.

What should the student learn? Trajectory targets supervise actions but need not explicitly teach reusable page, icon, or transition knowledge. *Incremental GUI Knowledge Supervision* (IGKS) supplies these targets. Complementarily, failure-driven subtasks concentrate practice on weak operations without repeatedly rehearsing an entire task. Both data streams accumulate for joint training. Their value is a testable hypothesis: local knowledge and targeted practice must improve complete, unassisted task execution, not merely easier practice tasks.

Which supervision is compatible with the student? Qwen3.8-27B produces thousands of visible reasoning tokens, whereas the teacher exposes no corresponding trace. A requested short `think` explanation neither recovers that trace nor matches the student's response distribution. Smaller students benefit from teacher demonstrations and 27B from diagnosis-guided student rerollouts. This motivates testing supervision compatibility without attributing the difference to length or model size alone.

Closing the data–model loop. Student experience updates the SMG and both data streams; cumulative training updates the student, and Validation selects the next collector. After initial configuration, task generation, diagnosis, admission and selection require no per-round human decisions, though model teachers/judges remain involved. MobileWorld (Kong et al., 2026) evaluates transfer without curriculum feedback.

The paper is organized around three contributions:

- **SMG-mediated data–model co-evolution.** An autonomous loop couples persistent memory, incremental knowledge, and failure-directed practice with successive student updates.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

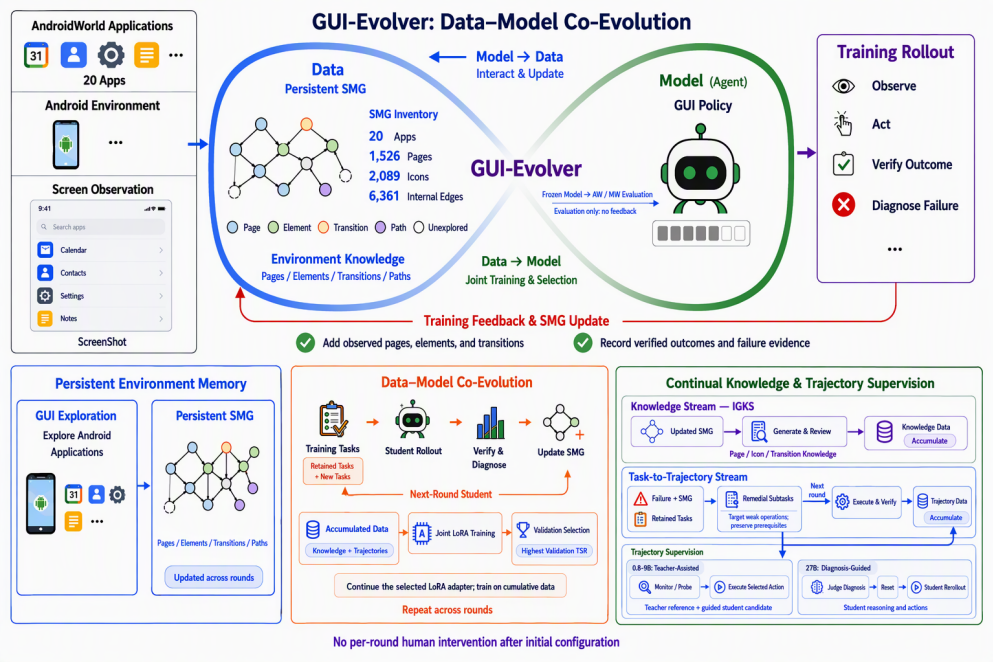


Figure 1: GUI-Evolver overview with continual knowledge and trajectory supervision. Algorithm 2 defines dataflow and evaluation boundaries. The phone screen is illustrative; inventory counts follow Table 2.

- **Student-compatible trajectory supervision.** Teacher-assisted targets and diagnosis-guided student trajectories support learning when teacher reasoning is unavailable.
- **Five-model evaluation and attribution controls.** Reported source/transfer outcomes are paired with controls designed to isolate data evolution, memory, and supervision effects.

2 Related Work

GUI perception, data, and benchmarks. CogAgent, SeeClick, Ferret-UI, UGround, and OS-ATLAS develop screen understanding and grounding (Hong et al., 2024; Cheng et al., 2024; You et al., 2024; Gou et al., 2025; Wu et al., 2025); AppAgent acquires application knowledge, while Mobile-Agent and WebVoyager support visual interaction (Zhang et al., 2025; Wang et al., 2024; He et al., 2024). AITW and Mind2Web provide human trajectories, TongUI extracts tutorial supervision, and AndroidControl studies data scale (Rawles et al., 2023; Deng et al., 2023; Zhang et al., 2026a; Li et al., 2024). Mobile infrastructure includes AndroidEnv, AndroidLab, AndroidWorld, and MobileWorld (Toyama et al., 2021; Xu et al., 2025; Rawles et al., 2025; Kong et al., 2026); WebArena/VisualWebArena and OSWorld/Windows Agent Arena extend evaluation to web and desktop (Zhou et al., 2024; Koh et al., 2024; Xie et al., 2024; Bonatti et al., 2025). We study learning within existing mobile environments, not a new benchmark.

Exploration, memory, and environment synthesis. GUI ripping, DroidBot, and Stoat use interaction models for testing (Amalfitano et al., 2012; Li et al., 2017; Su et al., 2017). OS-Genesis, GUI-explorer, Explorer, AutoPlay, and HATS connect exploration to task/knowledge synthesis or replay repair (Sun et al., 2025; Xie et al., 2025; Pahuja et al., 2025; Ramrakhya et al., 2026; Shao et al., 2026). Closely related OpenMobile combines global screen/transition memory, task synthesis, and monitored expert correction (Cheng et al., 2026); neither memory nor intervention alone is our distinction. We study their role in two persistent data streams with explicit memory and curriculum controls.

Environment synthesis offers a complementary route: AutoWebWorld and InfiniteWeb generate websites; GUI-GENESIS reconstructs verifiable GUI interactions (Wu et al., 2026b; Zhang et al.,

2026b; Cao et al., 2026). PhoneWorld and MobileGym provide phone simulations; CUA-Gym mixes desktop/Web environments and OpenComputer supplies real-software tasks (Liu et al., 2026b; Wu et al., 2026a; Wang et al., 2026a; Wei et al., 2026). ScaleEnv targets tool–database environments (Tu et al., 2026). Our graph instead records observations of existing Android applications without replacing their runtime.

Curricula and student-compatible supervision. UI-Genie iterates mobile exploration and verification; Mobile-Agent-E accumulates Tips/Shortcuts without weight updates (Xiao et al., 2025; Wang et al., 2025). WebRL uses failure-driven web RL, EvoCUA refines generated computer-use experience, and Agent-World expands tool environments from diagnoses (Qi et al., 2025; Xue et al., 2026; Dong et al., 2026). Thus neither failure-driven learning nor data–model evolution is new in isolation.

Distillation transfers teacher outputs, and DAgger addresses learner-induced distribution shift (Hinton et al., 2015; Ross et al., 2011). MiniLLM and on-policy distillation use student-generated sequences; BEPA and SGCD study expert plans/re-execution and verified skill-guided continuations (Gu et al., 2024; Agarwal et al., 2024; Wang et al., 2026b; Fan et al., 2026). We compare available target sources without claiming guided recovery itself as novel. Self-Instruct generates instructions; STaR/SPIN train on generated outputs, and LIMA emphasizes supervision quality (Wang et al., 2023; Zelikman et al., 2022; Chen et al., 2024; Zhou et al., 2023). Reflexion/Self-Refine refine attempts without weight updates (Shinn et al., 2023; Madaan et al., 2023). Judge and jury studies motivate scrutiny of model feedback (Zheng et al., 2023; Verga et al., 2024), not treating it as executable verification.

3 GUI-EVOLVER

3.1 SETTING AND PERSISTENT ENVIRONMENT MEMORY

Let π_r be the round-r student, $\mathcal{X}_r$ its training task set, and $G_r = (V_r, E_r)$ the environment graph. A node $v = (o_v, d_v, \mathcal{E}_v)$ stores a representative screenshot, semantic description, and interactive elements. Each element has a recorded location and supported action list; an edge (v_i, ϵ, a, v_j) binds an actually observed action on element ϵ to its source and target pages. Provenance links the transition to its original observation and execution.

Task-independent construction. Visual BFS initializes G_0 from a task-independent baseline, executing eligible element actions and matching screens by semantic retrieval plus visual comparison. Back or reset with checked replay restores navigation; budgets bound coverage (Appendix C). Training successes and failures then supply screenshot–action–screenshot updates. Deduplication retains evidence and actor provenance.

The SMG is observation memory, not a simulator: similar pages may differ in state, and recorded edges do not guarantee executable paths or adaptation to application changes.

3.2 INCREMENTAL GUI KNOWLEDGE SUPERVISION

IGKS turns pages, elements/icons, and observed transitions into understanding/grounding targets, excluding the legacy Single-Step action subset. The initial pool $\mathcal{K}_0$ and later increments train alongside trajectories, not in a separate pretraining stage. Reviewed new graph facts produce

$$\Delta\mathcal{K}_r = \mathrm{Review}_K(\mathrm{Generate}_K(\Delta G_r, \pi_r)), \quad \mathcal{K}_{r+1} = \mathrm{Dedup}(\mathcal{K}_r \cup \Delta\mathcal{K}_r). \tag{1}$$

Mastery checks filter known items and rejected targets receive reference-grounded revision, retaining visual evidence (Appendix D). Knowledge omission and fixed-knowledge replay controls distinguish the value of explicit facts from their continual updating.

3.3 FROM EXECUTION FAILURES TO PRACTICE TASKS

Subtasks concentrate practice on a failed operation rather than replaying mastered navigation. They retain prerequisites and state context, augment full tasks, and do not guarantee long-horizon composition. Evaluation still attempts complete goals without decomposition assistance.

A verifier distinguishes model failures from invalid tasks and environment errors. Diagnosis f binds the goal, deviation, screenshot, reason and actor; leaving one graph path alone is not failure. After updating the graph, retrieval $\mathcal{R}(G_{r+1}, f)$ returns matched pages, outgoing actions and recorded entry routes. Unsupported matches or setups cause abstention, not invented corrective edges (Appendix E).

Remedial generation. The generator conditions on the failure evidence and retrieved context:

$$q \sim \mathrm{Generate}(f, \mathcal{R}(G_{r+1}, f)) \,. \tag{2}$$

Bounded review admits at most one standalone proposal per selected failure, checking relevance, graph support, feasible task-independent setup, observable success, and answer leakage. A proposal contains neither a click-by-click solution nor the parent's hidden answer. It becomes trajectory supervision only after later execution, verification and export admission; a recorded route alone is not replay evidence.

Task-set update. Let $\mathcal{Q}_r$ be reviewed remedial proposals, $\mathcal{C}_r$ optional coverage proposals from new graph observations, and $\mathcal{U}_r$ tasks excluded with recorded reasons. The next set is

$$\mathcal{X}_{r+1} = \mathrm{Dedup}[(\mathcal{X}_r \setminus \mathcal{U}_r) \cup \mathcal{Q}_r \cup \mathcal{C}_r] \,. \tag{3}$$

Retained tasks and reviewed additions are deduplicated before publication. Coverage policy is fixed within comparisons. Neither new knowledge nor tasks may derive from Validation/benchmark feedback, directly or through the SMG.

3.4 Two Supervision Configurations

Teacher-assisted intervention. A Monitor inspects student actions before execution. During intervention, the teacher reference and a guided student candidate share the same device state; Probe selects which action executes. Executed intervention responses are eligible targets, ordinary weak steps remain context, and parse-error fallbacks retain provenance. This is mixed intervention data, not teacher-only demonstrations or proof of mastery from action agreement.

Diagnostic-feedback-guided rerollout. For 27B, a judge diagnoses the failed training attempt; after reset, the student generates every action and visible reasoning target. The judge supplies feedback, not token-level answers. CoT denotes generated text, not verified internal computation. The two configurations are specified per model, not an automatic size router (Figure 1). Export checks must ensure targets remain supported after guidance removal; Appendix F details admission semantics.

Training objective. Training targets include reviewed GUI knowledge answers and admitted executed trajectory responses, each with its visual context. Both streams use the weighted objective over selected target positions T:

$$\mathcal{L}(\theta) = -\frac{1}{Z} \sum_{t \in T} w_t \log \pi_\theta(y_t \mid x, y_{<t}), \qquad w_t = 0.2 \text{ for CoT-body tokens.} \tag{4}$$

Only the current assistant target contributes loss: input and history are masked. Recognizable CoT-body tokens have weight 0.2; summaries, actions, and other answer text have weight 1.0. This does not fix CoT at 20% of total loss. LoRA (Hu et al., 2021) adapts language-model linear layers while the visual encoder and Aligner remain frozen. Training-profile details are given in Appendix H.

3.5 Autonomous Data–Model Co-Evolution

Each round freezes its tasks, student, harness and budgets. Reviewed knowledge joins current training; new tasks enter the next rollout and yield targets only after execution. With admitted trajectories accumulated as $\mathcal{A}_{r+1}$, training resumes the previous round's best-Validation LoRA adapter on $\mathcal{K}_{r+1} \cup \mathcal{A}_{r+1}$, without merging or restarting the adapter. Full historical data are replayed. The selected student's subsequent observations and failures reshape both streams, closing the loop.

For the frozen candidate set Θ_r, validation chooses

$$\theta^*_{r+1} = \arg\max_{\theta \in \Theta_r} \mathrm{TSR}_{\mathrm{val}}(\theta), \tag{5}$$

with earlier checkpoints breaking ties. Final uses the highest Validation mean across evaluated rounds, never AW/MW maxima. Benchmarks also cannot determine stopping or generate data. The full schedule is Algorithm 2 in Appendix G; stopping, lineage and recovery records remain separate from this protocol.

4 Experimental Setup

Students and environments. All five models use MobileWorld's public `general_e2e` harness and start with 2,970 AndroidWorld training tasks. Every AW score, including Base, is measured in this harness rather than copied from official model reports. These runs are not vendor-harness reproductions. Within this shared harness, 27B uses a history-bearing configuration, unlike the smaller students' single-interaction setting; cross-family effects are not size-only. AW116/MW117/Validation100 denote nominal manifest sizes, not necessarily effective evaluation denominators.

Data boundaries. Only training interactions can update knowledge, curricula or their shared SMG, including inherited data and paraphrases. AndroidWorld application overlap is not unseen-app generalization. Interpreting MobileWorld gains as transfer requires excluding its tasks from project training, tuning and selection; aggregate scores do not establish training-data lineage or base-model pretraining contents.

Graph inventory. The SMG inventory contains 20 applications, 1,526 pages, 2,089 icons, and 6,361 internal edges (Table 2). Icon counts are distinct from historical page-element entries; these counts imply neither complete coverage nor replay guarantees. Appendix B distinguishes this inventory from earlier snapshots.

Training and budgets. Training uses three epochs per round, global batch 32, sequence limit 32,768, BF16, and LoRA rank/alpha 16/32 with dropout 0.1. Learning rates are $3/3/2/2/1 \times 10^{-5}$ across the five sizes; cosine decay uses 0.05 warmup and no packing. Cost-aware comparisons (Kapoor et al., 2025) count unweighted target tokens, including replay and CoT. Budget-controlled experiments allocate 120M/240M tokens to 9B/27B, distinct from the full-data runs; equal tokens do not establish equal compute (Appendix K).

Metrics. TSR averages three evaluations of the same checkpoint, not three training seeds or Pass@3. We report one-decimal means without inferring variance or statistical significance. Evaluation uses unassisted single attempts (Appendix H).

5 Results and Analysis

5.1 Joint Learning and Transfer

Table 1 compares Base with the round checkpoint having the highest Validation mean. All five Final means exceed Base: AW gains span 9.9–25.0 points and MW gains 4.8–13.1. Differences use rounded inputs and reflect joint learning, not separate IGKS or subtask effects.

The overview in supplementary Figure 2 uses the same selected checkpoints. MobileWorld is not used for selection. Interpreting its gains as AndroidWorld-only transfer depends on the training-data boundaries above; model-family/configuration differences and the absence of repeat-level uncertainty estimates preclude size-only or statistical-significance claims.

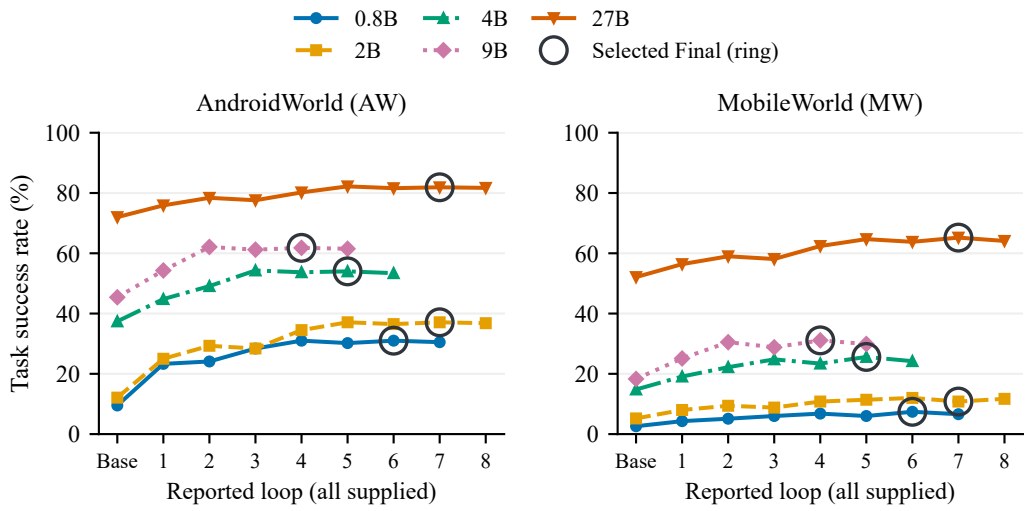
5.2 Progress Across Co-Evolution Loops

Figure 2 retains all 7/8/6/5/8 reported loops for 0.8B/2B/4B/9B/27B, including regressions and post-selection rounds. Validation selects Loops 6/7/5/4/7, not the last points or benchmark maxima (Appendix I). For 27B, Final has AW 81.9% versus 82.2% at Loop 5 because its Validation mean is

Table 1: Task success rates (%): Base versus Validation-selected Final; gains in percentage points.

Model	Base		Final		Gain	
	AW	MW	AW	MW	AW	MW
Qwen3.5-0.8B	9.5	2.6	31.0	7.4	+21.5	+4.8
Qwen3.5-2B	12.1	5.2	37.1	10.8	+25.0	+5.6
Qwen3.5-4B	37.4	14.8	54.0	25.6	+16.6	+10.8
Qwen3.5-9B	45.4	18.3	61.8	31.1	+16.4	+12.8
Qwen3.8-27B	72.0	52.1	81.9	65.2	+9.9	+13.1

AW = AndroidWorld; MW = MobileWorld. Same-checkpoint means of 3; gains from rounded means. Final is Validation-selected, not the last round. All models use general_e2e; AW results use this harness.



All models: general_e2e; 27B: Qwen3.8, others: Qwen3.5.
Mean of 3 evaluations per checkpoint.
Final: Validation-selected, not the last loop.

Figure 2: Mean TSR over three evaluations per checkpoint. Highlighted points are Validation-selected Finals; later rounds remain visible. All models use the public `general_e2e` harness. Repeat-level uncertainty is not reported; AW/MW do not select checkpoints.

higher. Task additions and admitted IGKS counts are reported separately; they neither measure trajectory volume nor establish matched compute or a scaling law.

5.3 What Do the Two Graph Outputs Contribute?

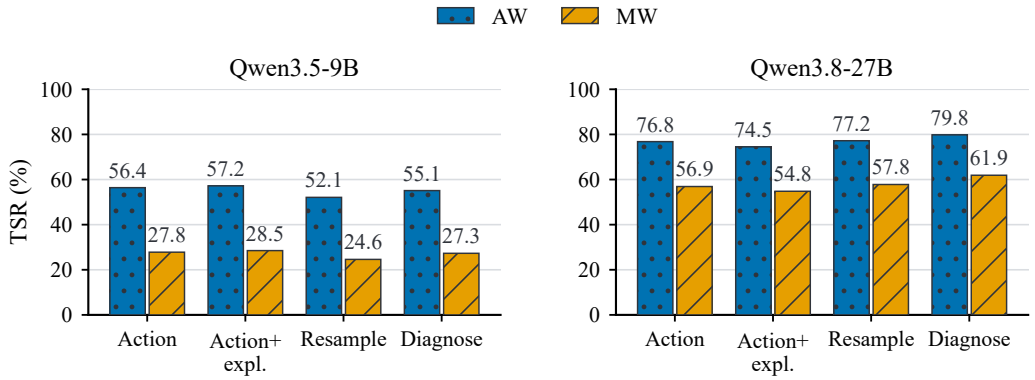
Knowledge and task increments. Joint IGKS/trajectory training exceeds trajectory-only by 3.6/3.0 AW/MW points on 9B and 1.8/2.8 on 27B (Table 14). Under equal total tokens but unequal action exposure, this comparison favors allocating budget to explicit knowledge. In the 2×2 control, both increments exceed fixed replay by 6.8/5.0 and 3.7/6.1 points, respectively (Table 15). Knowledge-only increments retain trajectory replay; task-only increments retain knowledge replay. The 27B AW interaction is -0.1 points: joint benefit need not be superadditive.

Memory and curriculum contrasts. With knowledge fixed, D–B gives 4.8/3.3 AW/MW points on 9B and 3.2/5.0 on 27B; D–E gives 2.4/1.2 and 1.2/1.9. E retains the initial graph plus current observations. Reusing C/D in Table 16, subtasks exceed original-task reruns by 4.6/2.5 and 3.0/4.0

Table 2: Curriculum ablations (TSR, %). Bold identifies the proposed configuration.

Configuration	Qwen3.5-9B		Qwen3.8-27B	
	AW	MW	AW	MW
A Flat/ordinary	52.8	24.9	–	–
B Flat/failure	55.4	26.8	77.6	58.4
C Graph/ordinary	56.7	27.4	78.2	59.1
D Persistent graph/failure	**60.2**	**30.1**	**80.8**	**63.4**
E Initial+current graph/failure	57.8	28.9	79.6	61.5

Control endpoints are separate from the selected Final. – means unallocated, not zero or a failed experiment. AW = AndroidWorld; MW = MobileWorld.



Fixed-pool controls; not online Final.

Figure 3: Supervision-source comparison on a fixed task pool. Action: teacher actions only; Action+expl.: the same actions plus requested short explanations; Resample: student retries without diagnosis; Diagnose: diagnosis-guided student retries. AW/MW values are from pure-source controls, not the online Finals.

points; ordinary synthesis slightly trails reruns on MW. These contrasts favor targeted practice rather than novelty alone; without matched task horizons and setups, attribution is limited to the acquisition policy as a whole.

5.4 Which Supervision Source Helps Which Student?

Does supervision help because it supplies a better action, an explanation target, or useful feedback on the student's own attempt? Figure 3 compares these mechanisms on a fixed task pool. Action-only and action-plus-explanation share teacher actions; resampling and diagnostic rerollouts share retry opportunities. These are pure-source controls, not the online workflow's mixed Probe targets.

What does the explanation add? Adding short explanations improves 9B by 0.8 AW and 0.7 MW points, but reduces 27B by 2.3 and 2.1 points. Because actions are shared, the contrast targets the added explanation rather than action correctness. Yet extra rationale tokens change action exposure at a fixed token budget; it is not a length-only intervention. The pattern motivates matching available targets to the student's response distribution, not a general rejection of teacher explanations.

Diagnosis versus another attempt. Diagnostic rerollouts exceed unassisted resampling by 3.0/2.7 AW/MW points on 9B and 2.6/4.1 on 27B. In this comparison, feedback adds value beyond retry-and-filter alone. Teacher short-explanation targets still lead diagnosis by 2.1/1.2 points on 9B, while

diagnosis leads teacher action-only by 3.0/5.0 on 27B. Thus one supervision source need not suit every student.

The takeaway is *supervision compatibility*, not that parameter count or long CoT causes the reversal. Teacher quality, visited states, model family and interaction history can differ; these contrasts do not isolate reasoning length.

5.5 Comparison with Original-Paper Results

Table 3 collects original results from four relevant methods. OpenMobile is the closest structural comparator: global environment memory supports task synthesis and expert intervention. HATS uses hardness-directed synthesis/refinement, UI-MOPD studies continual on-policy distillation, and MobileForge couples feedback-grounded curricula with RL. Their backbones, training budgets and evaluation settings differ from ours.

Table 3: Original-paper results and our reported outcomes (TSR, %). Different backbones, training budgets and evaluation protocols; not a controlled ranking.

Method	Model	AW	MW
Original-paper reports (different protocols)			
OpenMobile (Cheng et al., 2026)	Qwen3-VL-8B	64.7	17.7
HATS (Shao et al., 2026)	Qwen2-VL-7B	24.35	–
UI-MOPD (Lian et al., 2026)	Qwen3-VL-8B (think)	–	12.0
MobileForge (Liu et al., 2026a)	ForgeOwl-8B	67.2	41.0
This work (general_e2e)			
GUI-Evolver	Qwen3.5-9B	61.8	31.1
GUI-Evolver	Qwen3.8-27B	81.9	65.2

External values are checked against the cited PDFs, not reproduced here. – means not reported for that method/model, not zero. Our rows use Validation-selected Finals and three-evaluation means. Dataset subsets and source locations are listed in the supplement.

OpenMobile reports 64.7/17.7 AW/MW, while MobileForge's ForgeOwl-8B reports 67.2/41.0. Our 9B results, 61.8/31.1, fall below both on AW and between them on MW, not uniformly above prior work. Our 27B outcomes are higher numerically but also change model family and capacity. HATS reports AW only; UI-MOPD reports MW after data collection that includes MW. Missing cells are not inferred.

These are source-paper results, not our reproductions. We use AW Pass@1 rather than MobileForge's 77.6 Pass@3; OpenMobile's reported spread is a three-run half-range, not standard deviation. Appendix J.1 records source pages and protocol differences. Original-paper context cannot replace same-backbone, matched-budget distillation and curriculum controls.

5.6 Does Local Practice Improve Complete Tasks?

A subtask can become easy without repairing the complete goal. Figure 4 gives a schematic example linking a failed calendar operation to graph-supported practice, executed supervision, and a fresh full-task test. It retains recurrence-related state and setup rather than clipping a trace.

Table 4 evaluates a separate cohort of 100 related full tasks per row, not AW116/MW117. On 9B, rerun-to-subtask full success rises 52% to 63%, while target errors fall 24 to 13 and operation reach rises 78 to 84. Conditional operation success rises 69.2% to 84.5%, and fewer target errors are not accompanied by reduced aggregate reach. These aggregates do not show whether the same tasks reach the operation; paired reach/error transitions are needed to exclude selective earlier failures. The 27B pattern is smaller but consistent: full success rises 76% to 83% and conditional operation success 85.6% to 92.5%.

Within this cohort, local reliability and complete-goal completion improve together. Transfer to untrained goals requires paired evidence on predeclared held-out tasks, not merely re-evaluating

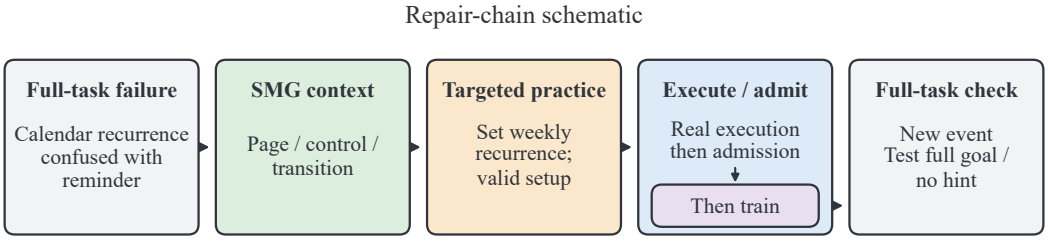


Figure 4: Schematic example of a repair chain, not a recorded execution. Practice must be executed and admitted before training; its effect is assessed on complete tasks without hints.

Table 4: Repair diagnostics on a separate related-task cohort (%).

Model	Policy	Reach %	Target fail %	Op. success %	Full TSR %
9B	Rerun	78.0	24.0	69.2	52.0
9B	Ordinary	80.0	21.0	73.8	57.0
9B	Subtask	84.0	13.0	84.5	63.0
27B	Rerun	90.0	13.0	85.6	76.0
27B	Ordinary	90.0	12.0	86.7	78.0
27B	Subtask	93.0	7.0	92.5	83.0

All entries are percentages, not counts. Reach/Target fail/Full TSR use cohort denominator 100; target failures occur among reached tasks. Op. success=(reached-target failures)/reached*100, conditional on reaching the operation. Separate from AW/MW benchmark results and the selected Final.

training parents. If reach or full-task success does not improve, the bottleneck-repair explanation weakens even when subtask success is high.

6 Limitations and Conclusion

GUI-Evolver couples cumulative knowledge and failure-directed practice through a persistent SMG. Mean TSR improves on both benchmarks, and controlled comparisons support knowledge allocation, targeted practice, and compatible supervision within the evaluated settings. State aliasing, infeasible setup, teacher errors, leakage, and proprietary teachers remain limitations. Batch/configuration differences prevent size-only attribution; equal target-token budgets do not imply equal compute. Aggregate means do not establish statistical significance, and local repair does not by itself establish transfer to untrained goals.

AI Usage Statement

We used AI assistance for manuscript writing and editing, code inspection and plotting code, and generation of the overview figure. Model use within the proposed data-generation and diagnostic pipeline is described in the method.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024.

Domenico Amalfitano, Anna Rita Fasolino, Porfirio Tramontana, Salvatore De Carmine, and Atif M. Memon. Using gui ripping for automated testing of android applications. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pp. 258–261, 2012. doi: 10.1145/2351676.2351717.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Keunho Jang, and Zheng Hui. Windows agent arena: Evaluating multi-modal OS agents at scale. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4874–4910, 2025. URL https://proceedings.mlr.press/v267/bonatti25a.html.

Yuan Cao, Dezhi Ran, Mengzhou Wu, Yuzhe Guo, Xin Chen, Ang Li, Gang Cao, Gong Zhi, Hao Yu, Linyi Li, Wei Yang, and Tao Xie. Gui-genesis: Automated synthesis of efficient environments with verifiable rewards for gui agent post-training, 2026. URL https://arxiv.org/abs/2602.14093.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *ICML*, 2024.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*, 2024.

Kanzhi Cheng, Zehao Li, Zheng Ma, Nuo Chen, Jialin Cao, Qiushi Sun, Zichen Ding, Fangzhi Xu, Hang Yan, Jiajun Chen, Anh Tuan Luu, Jianbing Zhang, Lewei Lu, and Dahua Lin. Openmobile: Building open mobile agents with task and trajectory synthesis, 2026. URL https://arxiv.org/abs/2604.15093.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28091–28114, 2023. doi: 10.52202/075280-1220. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5950bf290a1570ea401bf98882128160-Paper-Datasets_and_Benchmarks.pdf.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, Jiajie Jin, Yutao Zhu, Hanbin Wang, Fangyu Lei, Qinyu Luo, Mingyang Chen, Zehui Chen, Jiazhan Feng, Ji-Rong Wen, and Zhicheng Dou. Agent-world: Scaling real-world environment synthesis for evolving general agent intelligence, 2026. URL https://arxiv.org/abs/2604.18292.

Zhimin Fan, Hongwei Yu, Yeqing Shen, Haolong Yan, Guozhen Peng, Tianhao Peng, Yudong Zhang, Xiaowen Zhang, Kaijun Tan, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. Skill-guided continuation distillation for gui agents, 2026. URL https://arxiv.org/abs/2606.18890.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *ICLR*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *ICLR*, 2024.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an end-to-end web agent with large multimodal models. In *ACL*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. CogAgent: A visual language model for GUI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290, June 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

Quyu Kong, Xu Zhang, Zhenyu Yang, Nolan Gao, Chen Liu, Panrong Tong, Chenglin Cai, Hanzhang Zhou, Jianan Zhang, Liangyu Chen, Zhidan Liu, Steven Hoi, and Yue Wang. MobileWorld: Benchmarking autonomous mobile agents in agent-user interactive and MCP-augmented environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6142–6167, 2026. doi: 10.18653/v1/2026.acl-long.278. URL https://aclanthology.org/2026.acl-long.278/.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *NeurIPS Datasets and Benchmarks Track*, 2024.

Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. DroidBot: A lightweight UI-guided test input generator for android. In *ICSE Companion*, 2017.

Niu Lian, Alan Chen, Zhehao Yu, Chengzhen Duan, Fazhan Liu, Hui Liu, Pei Fu, Jian Luan, Yaowei Wang, Shu-Tao Xia, and Jinpeng Wang. Ui-mopd: Multi-platform on-policy distillation for continual gui agent learning, 2026. URL https://arxiv.org/abs/2607.04425.

Guangyi Liu, Pengxiang Zhao, Gao Wu, Yiwen Yin, Mading Li, Liang Liu, Congxiao Liu, Zhang Qi, Mengyan Wang, Liang Guo, and Yong Liu. MobileForge: Annotation-free adaptation for mobile GUI agents with hierarchical feedback-guided policy optimization, 2026a. Version 1, 18 June 2026.

Yuxuan Liu, Xin Lai, Junyi Li, Pengyuan Lyu, Jason, Yiduo Guo, Zhengyao Fang, Yang Ding, Yi Zhang, Weinong Wang, Huawen Shen, Xingran Zhou, Liang Wu, Fei Tang, Sunqi Fan, Shangpin Peng, Zheng Ruan, Anran Zhang, Chengquan Zhang, Han Hu, Benyou Wang, Ji-Rong Wen, Rui Yan, and Zhengyang Tang. Phoneworld: Scaling phone-use agent environments, 2026b. URL https://arxiv.org/abs/2605.29486.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, et al. Self-Refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Hassan Awadallah. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6300–6323, 2025. doi: 10.18653/v1/2025.findings-acl.326. URL https://aclanthology.org/2025.findings-acl.326/.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, pp. 79791–79821, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c66e1fcc9691aae706250638f36f681b-Paper-Conference.pdf.

Ram Ramrakhya, Andrew Szot, Omar Attia, Bogdan Mazoure, Anh Nguyen, Yuhao Yang, Zhe Gan, Harsh Agrawal, and Alexander Toshev. Scaling synthetic task generation for agents via exploration. In *International Conference on Learning Representations*, pp. 123630–123660, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c8a4dd7d9e13583d714ce8580da7bbc7-Paper-Conference.pdf.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, pp. 406–441, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/01a83bc2f2732a58e6aa731e659e7101-Paper-Conference.pdf.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. AndroidInTheWild: A large-scale dataset for android device control. In *NeurIPS Datasets and Benchmarks Track*, 2023.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

Rui Shao, Ruize Gao, Bin Xie, Yixing Li, Kaiwen Zhou, Shuai Wang, Weili Guan, and Gongwei Chen. Hats: Hardness-aware trajectory synthesis for gui agents, 2026. URL https://arxiv.org/abs/2603.12138.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Ting Su, Guozhu Meng, Yuting Chen, Ke Wu, Weiming Yang, Yao Yao, Geguang Pu, Yang Liu, and Zhendong Su. Guided, stochastic model-based GUI testing of android apps. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pp. 245–256, 2017. doi: 10.1145/3106237.3106298.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, and Zhiyong Wu. OS-genesis: Automating GUI agent trajectory construction via reverse task synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5555–5579, 2025. doi: 10.18653/v1/2025.acl-long.277. URL https://aclanthology.org/2025.acl-long.277/.

Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. AndroidEnv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*, 2021.

Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, Yi-Kai Zhang, Zhikang Xia, Yu Yang, Yueqing Sun, Xingchen Liu, Furao Shen, Qi Gu, Hui Su, and Xunliang Cai. Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training, 2026. URL https://arxiv.org/abs/2602.06820.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models, 2024. URL https://arxiv.org/abs/2404.18796.

Bowen Wang, Dunjie Lu, Junli Wang, Tianyi Bai, Shixuan Liu, Zhipeng Zhang, Haiquan Wang, Hao Hu, Tianbao Xie, Shuai Bai, Dayiheng Liu, Que Shen, Junyang Lin, and Tao Yu. Cuagym: Scaling verifiable training environments and tasks for computer-use agents, 2026a. URL https://arxiv.org/abs/2605.25624.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-Agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*, 2024.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In *ACL*, 2023.

Zezhou Wang, Ziyun Zhang, Xiaoyi Zhang, Zhuzhong Qian, and Yan Lu. From off-policy to on-policy: Enhancing gui agents via bi-level expert-to-policy assimilation, 2026b. URL https://arxiv.org/abs/2601.05787.

Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. Mobile-Agent-E: Self-evolving mobile assistant for complex tasks. *arXiv preprint arXiv:2501.11733*, 2025.

Jinbiao Wei, Qianran Ma, Yilun Zhao, Xiao Zhou, Kangqi Ni, Guo Gan, and Arman Cohan. Opencomputer: Verifiable software worlds for computer-use agents, 2026. URL https://arxiv.org/abs/2605.19769.

Dingbang Wu, Rui Hao, Haiyang Wang, Shuzhe Wu, Han Xiao, Zhenghong Li, Bojiang Zhou, Zheng Ju, Zichen Liu, Lue Fan, and Zhaoxiang Zhang. Mobilegym: A verifiable and highly parallel simulation platform for mobile gui agent research, 2026a. URL https://arxiv.org/abs/2605.26114.

Yifan Wu, Yiran Peng, Yiyu Chen, Jianhao Ruan, Zijie Zhuang, Cheng Yang, Jiayi Zhang, Man Chen, Yenchi Tseng, Zhaoyang Yu, Liang Chen, Yuyao Zhai, Bang Liu, Chenglin Wu, and Yuyu Luo. Autowebworld: Synthesizing infinite verifiable web environments via finite state machines, 2026b. URL https://arxiv.org/abs/2602.14296.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. OS-ATLAS: A foundation action model for generalist GUI agents. In *ICLR*, 2025.

Han Xiao, Guozhi Wang, Yuxiang Chai, Zimu Lu, Weifeng Lin, Hao He, Lue Fan, Liuyang Bian, Rui Hu, Liang Liu, Shuai Ren, Yafei Wen, Xiaoxin Chen, Aojun Zhou, and Hongsheng Li. UI-Genie: A self-improving approach for iteratively boosting MLLM-based mobile GUI agents. In *Advances in Neural Information Processing Systems*, volume 38, pp. 150376–150411, 2025. doi: 10.52202/085713-5028. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/dd1577afd396928ed64216f3f1fd5556-Paper-Conference.pdf.

Bin Xie, Rui Shao, Gongwei Chen, Kaiwen Zhou, Yinchuan Li, Jie Liu, Min Zhang, and Liqiang Nie. GUI-explorer: Autonomous exploration and mining of transition-aware knowledge for GUI agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5650–5667, 2025. doi: 10.18653/v1/2025.acl-long.282. URL https://aclanthology.org/2025.acl-long.282/.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *NeurIPS Datasets and Benchmarks Track*, 2024.

Yifan Xu, Xiao Liu, Xueqiao Sun, Siyi Cheng, Hao Yu, Hanyu Lai, Shudan Zhang, Dan Zhang, Jie Tang, and Yuxiao Dong. AndroidLab: Training and systematic benchmarking of android autonomous agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2144–2166, 2025. doi: 10.18653/v1/2025.acl-long.107. URL https://aclanthology.org/2025.acl-long.107/.

Taofeng Xue, Chong Peng, Mianqiu Huang, Linsen Guo, Tiancheng Han, Haozhe Wang, Jianing Wang, Xiaocheng Zhang, Xin Yang, Dengchang Zhao, Jinrui Ding, Xiandi Ma, Yuchen Xie, Peng Pei, Xunliang Cai, and Xipeng Qiu. Evocua: Evolving computer use agents via learning from scalable synthetic experience, 2026. URL https://arxiv.org/abs/2601.15876.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-UI: Grounded mobile UI understanding with multimodal LLMs. In *ECCV*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022.

Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. TongUI: Internet-scale trajectories from multimodal web tutorials for generalized GUI agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (15):12367–12375, 2026a. doi: 10.1609/aaai.v40i15.38229.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. AppAgent: Multimodal agents as smartphone users. In *CHI*, 2025.

Ziyun Zhang, Zezhou Wang, Xiaoyi Zhang, Zongyu Guo, Jiahao Li, Bin Li, and Yan Lu. Infiniteweb: Scalable web environment synthesis for gui agent training, 2026b. URL https://arxiv.org/abs/2601.04126.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, et al. Judging LLM-as-a-Judge with MT-Bench and chatbot arena. In *NeurIPS Datasets and Benchmarks Track*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55006–55021, 2023. doi: 10.52202/075280-2400. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ac662d74829e4407ce1d126477f4a03a-Paper-Conference.pdf.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *ICLR*, 2024.

GUI-EVOLVER: SCALING GUI AGENTS VIA DATA–MODEL CO-EVOLUTION SUPPLEMENTARY MATERIAL

Anonymous authors
Paper under double-blind review

A EXPERIMENTAL SOURCES AND REPORTING CONVENTIONS

This supplement documents performance, SMG inventories, controlled comparisons, and training budgets. `data/reported_results.json` records the five-model Base and 34 loop rows together with the updated SMG counts. Scores are means of three evaluations of the same checkpoint. Control results are recorded in a separate structured results artifact. Historical graph inventories and original-paper comparisons retain their distinct reporting conventions.

Experimental scope. Comparisons are interpreted with their checkpoint initialization, cumulative training-data lineage, harness configuration, and Validation selection rule. Train/test separation concerns the entire learning history, including tasks, screenshots, diagnoses, and graph facts, not only the final data increment. Trajectory-only task generation is distinguished from SMG-grounded generation.

Table 1: Experimental sources and reporting conventions. Results, historical inventories, and original-paper comparisons have distinct scopes.

Item	Source and reporting convention
SMG inventory	Page/icon/internal-edge counts; historical local JSON inventories reported separately.
Five-model performance	Means of three evaluations per checkpoint, not independent training seeds or Pass@3.
Supervision preference	Comparisons under model-specific supervision and context configurations.
CoT loss weight	CoT-body weight 0.2 for all five models.
Validation selection	Highest Validation TSR; earlier checkpoint breaks ties.
Autonomous loop	No per-round human task authoring, annotation, admission, or checkpoint selection after initial configuration.
Original-paper comparisons	OpenMobile, HATS, UI-MOPD, and MobileForge results (Appendix J.1), distinct from matched-budget controls.
Data and checkpoints	Initial 2,970 tasks per model; saved epoch/round-step and admitted IGKS increments.
Budget and uncertainty	Target-token budgets distinguish controlled comparisons from full-data runs; aggregate means do not establish statistical significance.

B ANDROIDWORLD SMG INVENTORY

The inventory in Table 2 covers 20 applications, 1,526 pages, 2,089 icons, and 6,361 internal edges. The per-application sums agree with the totals recorded in the `smg_inventory` entry of `data/reported_results.json`. Icons are a distinct metric, not the 10,696 page-element entries in the historical snapshot below. Counts alone do not establish coverage, executable paths, or which graph version each training round used.

Table 2: AndroidWorld SMG inventory. ICON counts are distinct from page-element entries; internal edges exclude separately recorded external transitions.

Application	Pages	Icons	Internal edges
Retro Music	77	83	350
Camera	14	15	55
Chrome	66	105	164
Settings	108	98	244
Expense	45	60	256
Audio Recorder	50	88	237
Broccoli	32	21	146
Contacts	56	69	174
Clock	39	28	221
Files	32	37	95
Simple Calendar	218	259	815
Simple Draw	29	64	118
Simple Gallery	81	81	417
Simple SMS Messenger	13	41	43
OpenTracks	106	158	493
Joplin	73	142	309
Markor	196	352	1,050
OsmAnd	46	80	122
Tasks	158	179	632
VLC	87	129	420
Total: 20 applications	1,526	2,089	6,361

B.1 Historical Local Snapshots and Provenance

The earlier local inventories were read on 25 September 2026 at 14:15 UTC from each package's `graph.json` and referenced page JSON. Hashes were stable during that read, with no duplicate page IDs or missing edge endpoints. These historical structural checks do not audit the new icon metric. Although the updated page and internal-edge totals match the earlier Loop-1 inventory, matching counts alone do not establish snapshot identity or per-model lineage.

Table 3: Historical local JSON inventories, not the updated ICON-count table. Element entries, self-loop-inclusive graph edges, and external records retain their original counting conventions.

Snapshot	Nodes	Elements	Edges	External records
Desktop initial candidate	1,238	8,139	4,801	0
Loop-1 working input	1,238	8,139	4,817	0
Loop-1 updated graph	1,526	10,696	6,361	134

Historical counting conventions. For these historical snapshots, page counts are records indexed by package and page ID, not globally unique visual states. All three snapshots include 87 nodes classified as `not_app`. Excluding them leaves 1,151 pages in the desktop and working-input graphs and 1,439 pages after the Loop-1 update. This still includes any retained WebView nodes. Elements are page-level list entries, not unique functions across pages; the updated graph contains 10,696 entries, including one on a `not_app` page.

The desktop graph contains 4,801 edge records, including 1,132 self-loops. The working input contains 4,817 edges, including 1,140 self-loops. The updated graph contains 6,361 edges, including 1,497 self-loops, plus 134 separately stored external-transition records. The 11,574 updated element-action records are a different object type and must not be reported as 11,574 observed transitions.

Source identity. The desktop initial candidate is the local `Desktop/SMG/Android_World/` tree. The working input is `D:/RSI-1/smg/Android_World/`, and the updated tree is `D:/RSI-1/Loop-1/03_smg/Android_World/`. A content-inventory hash is computed over

the sorted list of relative POSIX paths and their file SHA-256 values, serialized as compact UTF-8 JSON without ASCII escaping; screenshots are not included in this fingerprint.

Desktop inventory
`0306332e740429f95cf8f2fdb22a296894`
`435b7d6b99e07acc3601a4f9f75873`

Working-input inventory
`21d531a4f1d01ec2239481b61dace0053`
`a108afe1c35327daff7d40c4211bd93`

Updated inventory
`b11fdc21c743b0fe4ee6f9a4cac4cad87`
`09aed385fcc0b0f2e38cb718e1e0822`

Line breaks within each fingerprint are for typography only.

The update summary records 20 applications marked `done`, no job-level errors, and a completion timestamp of `2026-09-23T19:46:50.171945+00:00`. Its file hash is `6ae63e9258f07b2e22329c5d8769d4225aa41a75f79a0f95ce7e057616af3b85`. Its increments, 288 pages, 1,544 edges, and 134 external records, are relative to the *working input*, not the desktop tree. The local round record identifies a training Probe run as the update source. This provenance describes the update, not the entire history of the inherited graph. A successful update job alone establishes neither path executability nor benchmark independence.

C Task-Independent Graph Construction

The construction loop follows GUI ripping and UI-guided model-based exploration (Amalfitano et al., 2012; Li et al., 2017), using a visual model to interpret observations rather than treating breadth-first traversal itself as a new contribution. The underlying state is external to the graph: recorded navigation is checked against the actual device.

Algorithm 1 Bounded visual BFS construction of an observation graph

Require: Application, task-independent reset, visual perception/matching, interaction budget
Reset and launch; observe and register the entry page; initialize queue Q
while Q is nonempty and the interaction budget remains **do**
  Select a queued page v and navigate to it using Back or reset plus checked replay
  if navigation cannot establish the target page **then**
    Record unresolved/unreachable status and continue
  end if
  Select an eligible unexplored element action on v
  if no eligible action remains **then**
    Remove v from Q and continue
  end if
  Execute the action; save the actual resulting screenshot
  Retrieve semantic page candidates and compare visual observations
  Match an existing page or register a newly observed page v'
  Record (v, ϵ, a, v') and its observation provenance
  if v' is a new interactive in-app page retained by the exploration policy **then**
    Append v' to Q
  end if
  Attempt a checked return to v; persist graph and progress
end while
return Observed graph, retained evidence, and incomplete/unreachable status

The reference driver explores the first proposed action of each extracted element, prioritizing non-swipe actions and then screen position. Training observations can add further actions to an element. Connectivity guards can skip actions that would disconnect the device; such skips must not be presented as executed transitions. Matching a page label alone does not establish equivalence, especially when messages, files, permissions, or selected items differ. Native hierarchy dumps are optional recorded device evidence, not trees invented from a screenshot.

D INCREMENTAL GUI KNOWLEDGE SUPERVISION

Incremental GUI Knowledge Supervision (IGKS) is the persistent graph-to-supervision stream, updated alongside trajectory supervision throughout the loop. The initial graph supplies the starting knowledge pool, not a separately required knowledge-pretraining checkpoint. New graph facts yield further reviewed knowledge targets; new eligible executions yield trajectory targets. Both stores accumulate and jointly train the student at each round.

Table 4: Knowledge targets produced from the SMG, initially and after new training observations. These are knowledge examples, not executable remedial task instructions.

Graph source	Supervision	Increment trigger
Page	Identity, stable purpose, principal layout regions	Newly observed page in a committed graph delta
Icon/element	Identity, functional meaning, visual grounding	New element/icon on a new or existing page
Transition edge	Action and observed source-to-target relation	Newly recorded supported action transition

Each example binds its screenshot, graph object, source version, question, target and review outcome. Stable page descriptions should not encode a transient clock, record count or user-created title as an invariant. Transition descriptions remain tied to observed state and prerequisites. A change of description or screenshot is not automatically a new capability, and duplicate object supervision must retain its lineage. The revised knowledge stream excludes the legacy Single-Step action-training subset. Knowledge of an observed action–transition relation remains a graph-derived target, distinct from that removed subset. Executed action responses still belong to the trajectory stream; excluding Single-Step does not remove action supervision from the method.

The configured mastery procedure obtains five student answers and requests a judgment for each answer from two configured judges. An item is skipped only when every answer passes both judges. The exact generation configuration is recorded: five calls do not imply five independent stochastic samples if decoding is deterministic. For an unmastered item, the student receives the Gold reference and rejection reasons and can revise up to three times. Each revised answer undergoes quality review. An enabled teacher fallback produces a candidate only after the revision budget is exhausted, and that candidate still requires review; unresolved examples do not become correct merely because retries ended. Deterministic coordinate targets are handled by their subtype-specific checks.

Teacher-provided references and their use during data generation are part of the sample lineage. A later training input without that reference must still support the target from its own image and question. Initial task generation similarly combines observed operations into a user goal but does not assume that paths are executable merely because edges exist. A proposed task specifies its applications, setup, observable goal, and supporting graph observations; successful device execution is a separate fact. Proposal review and successful execution remain separate admission stages.

Across rounds, the knowledge store $\mathcal{K}_r$ and trajectory store $\mathcal{A}_r$ accumulate separately with source-aware deduplication before being mixed for SFT. After graph update, $\Delta\mathcal{K}_r$ can join the current training round; new task proposals enter the next task-set rollout and only produce trajectory supervision after execution and admission. Repeated exposure still counts toward token budgets. Within a backbone, controls share the same starting model and initial knowledge/task pools; the factorial comparison varies subsequent knowledge and task increments. Neither stream is confined to a one-off warm-up. Each admitted knowledge export excludes the legacy Single-Step subset.

E GRAPH GROUNDING AND TASK ADMISSION

The graph-grounded generation entry point associates a selected failure with a committed graph version and application package. It attempts exact event provenance, unique screenshot hash, and then strict visual matching against existing pages. An unresolved match blocks the proposal rather than manufacturing a new matching page during retrieval.

The retrieved context contains the source page, its outgoing recorded edges and target pages, and a breadth-first recorded route from page zero that excludes back edges in route planning. Outgoing back and self edges remain possible local candidates. The generator must select a supplied edge reference and explain a task that can be attempted from a task-independent baseline. No automatic replay or restoration of the failed task state is implied.

Review is not execution. The generator receives the pre-action failure image and structured evidence. The independent proposal reviewer checks the textual candidate against the supplied evidence and graph metadata; this check does not itself execute the task. Accepted outputs retain `executability=not_evaluated` and an independent training-eligibility status. The observed graph edge, reviewed task proposal, verified outcome, exported sample, and admitted training sample are distinct records.

Trajectory-only comparator. Trajectory-only task generation does not consult the SMG. It can propose practice from recovered deviations as well as failed outcomes, but is distinct from the graph-grounded procedure.

Task-set mechanics. The merger prioritizes remedial tasks before coverage tasks, performs normalized-text and semantic deduplication, and can retain approximately one third of coverage tasks using application-combination and single/cross-app strata. The Train-only source policy applies to input manifests and graph provenance. The next task inventory is frozen before another training rollout, and removals retain explicit reasons.

F Supervision and Export Semantics

Monitor and Probe. The Monitor inspects a student candidate before execution. During intervention, a teacher reference and a guided student candidate can be produced from the same device state. The Probe judges the proposed actions with their visual context and selects which one is executed. A teacher reference is not automatically the executed target, and a discarded student proposal is not a training target. Parsing failures use a separately marked recovery path. The relevant run profile records intervention limits, continuation steps, and fallback attempts.

Diagnostic rerollouts. A new feedback-guided attempt initializes the environment before adding the corrective feedback to its instruction. The reset's actual semantics depend on the environment adapter; a lightweight local reset is not interchangeable with restoring a complete benchmark snapshot. The saved initial-state profile and attempt identity must therefore accompany the trajectory. Feedback validity and leakage checks cannot be replaced by merely calling the output a reason rather than an answer.

Target selection. The ordinary successful-attempt exporter selects the latest eligible attempt, not an earlier successful attempt after a later failure. For Monitor trajectories, selected executed intervention steps contribute targets; ordinary weak steps remain historical context. For successful non-Monitor rerollouts, the eligible full trajectory can contribute targets under its own export rules. A specialized verified-success override is not a blanket rule admitting all platform-Failed trajectories.

Reasoning-target compatibility. Visible Qwen3.8-27B reasoning outputs can reach thousands of tokens; this qualitative observation is not a measured mean or percentile. The available teacher exposes action responses but not the corresponding reasoning trace. Asking it to fill a `think` field yields a generated explanation; its presence does not establish correspondence to the teacher's hidden computation. The motivation is therefore an observable target-distribution mismatch in length, style, and reasoning/action organization, not privileged access to either model's internal reasoning.

The 27B rerollout uses a judge for failure diagnosis and outcome assessment while the student authors all executed actions and visible reasoning targets. It is assisted during data acquisition but autonomous with respect to action generation, and evaluation remains free of feedback. Teacher-action-only targets remain a valid comparator. Comparing the same teacher actions with and without

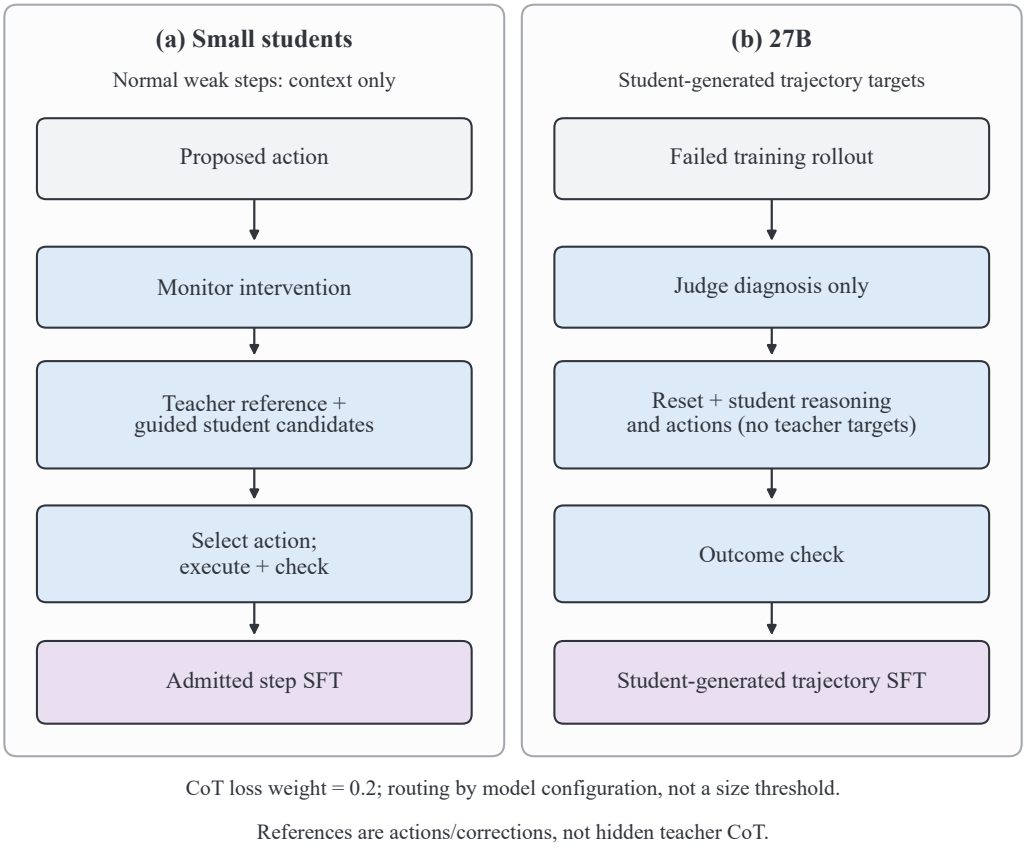


Figure 1: Two supervision configurations. The small-student path selects an action within an ongoing rollout; the 27B path diagnoses a training attempt and starts a new initialized student attempt. The configurations are not an automatic size-threshold router. Admission and actual execution are distinct from the candidate-selection decision.

requested short explanations isolates a narrower hypothesis than comparing unrelated full trajectories from different actors. Adding explanations changes the budget, so action-target exposure and total target tokens are distinct quantities. These comparisons do not establish that teacher explanations are generally incorrect or that longer CoT is inherently better.

Reasoning and guidance. All five students use a CoT-body loss weight of 0.2. Context-only assistant messages are masked from the loss. The loss mask excludes input and history and weights current-target summaries, actions, and other answer text at 1.0. Target reasoning and any feedback-dependent rewrite retain separate provenance. Removing guidance from input text can create an unsupported target if its reasoning still assumes the guidance; text removal alone does not establish independence.

G Round Schedule and Data Admission

Algorithm 2 makes the round boundary explicit: current observations produce reviewed knowledge for current training, while new tasks require next-round execution before yielding trajectory targets. The loop retains both cumulative data stores and the same LoRA adapter. Independent benchmarks are outside the feedback path.

Algorithm 2 Autonomous SMG-mediated data–model co-evolution

Require: Base student π_0, graph G_0, tasks $\mathcal{X}_0$, initial knowledge $\mathcal{K}_0$, trajectory store $\mathcal{A}_0 = \emptyset$, validation $\mathcal{V}$, budgets

for $r = 0, \ldots, R-1$ **do**
  Freeze π_r, $\mathcal{X}_r$, harness, and round budgets
  Collect training interactions $\mathcal{T}_r$ with the configured supervision procedure
  Verify outcomes and extract evidence-supported failures $\mathcal{F}_r$
  $(G_{r+1}, \Delta G_r) \leftarrow \mathrm{Update}(G_r, \mathcal{T}_r)$
  Generate/review $\Delta\mathcal{K}_r$; deduplicate into cumulative $\mathcal{K}_{r+1}$
  $\mathcal{Q}_r \leftarrow \mathrm{Review}(\mathrm{Generate}(\mathcal{F}_r, G_{r+1}))$
  Form $\mathcal{X}_{r+1}$ from retained, remedial and coverage tasks; freeze provenance
  $\mathcal{A}_{r+1} \leftarrow \mathrm{Accumulate}(\mathcal{A}_r, \mathrm{Admit}(\mathcal{T}_r))$
  Continue the same adapter from π_r on $\mathcal{K}_{r+1} \cup \mathcal{A}_{r+1}$
  Select π_{r+1} by highest Validation TSR; earlier checkpoint breaks ties
end for
Evaluate the frozen selected model on independent benchmarks; do not feed results back

H Training and Evaluation Profiles

Table 5 records the training hyperparameters; Table 6 gives shared settings. Supplemental optimizer and evaluation configurations and controlled token budgets are described in Appendix K.

Table 5: Training configuration (LoRA SFT).

Model	LR	Rank/alpha	Dropout	Epochs	Batch	Max seq.
Qwen3.5-0.8B	3e-05	16/32	0.1	3	32	32,768
Qwen3.5-2B	3e-05	16/32	0.1	3	32	32,768
Qwen3.5-4B	2e-05	16/32	0.1	3	32	32,768
Qwen3.5-9B	2e-05	16/32	0.1	3	32	32,768
Qwen3.8-27B	1e-05	16/32	0.1	3	32	32,768

Epochs per round; batch is global. Language-model all-linear; visual encoder and Aligner frozen; BF16; Cosine schedule with warmup ratio 0.05. Loss: Current assistant target only; input and history masked; CoT body weight 0.2, other answer weight 1. Selection uses highest Validation, not benchmark scores.

Inference infrastructure. Experiments use two servers with 16 NVIDIA H20 GPUs in total. The reference execution stack supports vLLM serving (Kwon et al., 2023). Its version, tensor-parallel configuration, context budget, and image preprocessing belong to the archived run profile, not to the learning algorithm. Infrastructure throughput should not be attributed to the curriculum without separately accounting for the serving configuration.

Common harness and self-measured baselines. All models use MobileWorld's public `general_e2e` harness. All Qwen3.5 and Qwen3.8 AndroidWorld scores, including Base, are measured in our setup; none is taken from official model benchmark tables. No reproduction of vendor benchmark scores is claimed, since the corresponding vendor evaluation setup is unavailable. This does not concern the availability of AndroidWorld's public benchmark. A common implementation does not imply equal model-specific context exposure: Qwen3.5-0.8–9B uses a single-interaction setting, whereas Qwen3.8-27B uses a history-bearing conversational setting. These are configurations within `general_e2e`, not different harness implementations. Consequently, cross-family supervision preferences cannot be attributed solely to parameter count or reasoning length.

MobileWorld manifest version. The GUI-only evaluation manifest contains 117 tasks. The public MobileWorld paper describes 116 GUI-only tasks within its 201-task suite (Kong et al., 2026). Results using the 117-task manifest are not directly interchangeable with scores using the 116-task publication list.

Table 6: Shared training and evaluation protocol.

Field	Configuration
Students	Qwen3.5-0.8B/2B/4B/9B and Qwen3.8-27B.
Training domain	AndroidWorld; 2,970 initial tasks per model. MobileWorld is excluded from training sources.
Harness	Public MobileWorld `general_e2e` for every model, with model-specific history, prompt, and context settings.
Starting model	Base for the first round; subsequent rounds use the selected checkpoint. Both accumulated data streams train every round.
Loss mask/weights	Current assistant target only; input/history masked. Recognizable CoT body: 0.2; summary, actions and other answers: 1.0.
Selection	Highest Validation TSR; earlier checkpoint wins a tie.
Teacher/judge	Role-specific configurations; the reference small-student configuration uses Gemini-3.8-flash.
Adaptation	Language-model all-linear LoRA; freeze vision encoder and Aligner. Continue the selected adapter without merging.
Numerics	BF16; gradient checkpointing on; packing off.
Schedule	Three epochs per round; cosine decay, warmup ratio 0.05; save each epoch.
Token handling	Maximum sequence length 32,768; no loss on input/history.
Round budget	7/8/6/5/8 recorded rounds. Full-data training differs from matched-token controls; steps are not token counts.
Independent evaluation	No Monitor/Hint; fixed revision, task manifest, reset, retry policy, verifier, and repetitions.

Outcome aggregation. For a frozen set of n tasks with R repeated single-attempt evaluations, let s_{ir} be the verified binary outcome. The mean single-attempt success rate is

$$\mathrm{TSR} = \frac{100}{nR} \sum_{i=1}^{n} \sum_{r=1}^{R} s_{ir}. \tag{1}$$

Reported aggregates use $R = 3$ evaluations of the same checkpoint, not independent training seeds or Pass@3. The formula assumes the same denominator in each repeat; evaluation manifests are AW116/MW117/Validation100. Rounded means alone do not determine standard deviations, confidence intervals, paired significance, or integer success counts. Differences are computed from the displayed rounded means rather than reconstructed integer counts.

Autonomy scope. The learning protocol requires no per-round human task authoring, annotation, diagnosis, sample admission, or checkpoint selection after initial configuration. Researchers specify applications, starting models, supervision profiles, budgets, and split/selection rules in advance. Teachers, judges, and automated checks make subsequent learning decisions; this is not a teacher-free system or an automatic choice between supervision configurations. Initial configuration and retrospective scientific auditing are distinct from intervening in data collection or training decisions. Bounded automatic retries and abstention handle recoverable or unresolved stages; a blocked round is not counted as completed.

I Main and Per-Round Results

Final selection and scope. Table 7 applies the highest-Validation rule to the reported round checkpoints, selecting Loops 6/7/5/4/7 for 0.8B/2B/4B/9B/27B; earlier checkpoints break ties. The preceding round's best-Validation checkpoint initializes the next training round. Final identifies the reporting checkpoint, not a stopping rule. All reported rounds, including those after Final, are retained. No benchmark maximum is substituted for Validation selection.

Task and knowledge increments. Each model begins with 2,970 tasks. Task increments count deduplicated remedial subtasks plus SMG coverage tasks scheduled for the next round, not trajectories already executed in the current round. Summing additions without accounting for removals

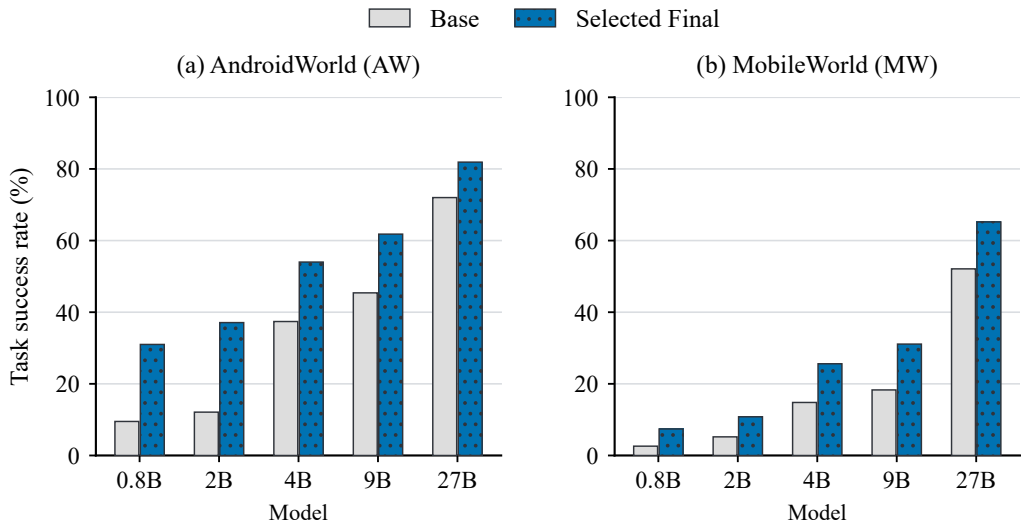


Figure 2: Base versus Validation-selected Final means over three evaluations per checkpoint. All models share the public `general_e2e` harness; model-specific configurations and family differences prevent attributing differences to parameter count alone.

Table 7: Initial IGKS pools and Validation-selected checkpoints.

Model	Initial IGKS	Rounds	Final	E/round step	Val (%)
Qwen3.5-0.8B	10,276	7	L6	E1 / 5,888	52.0
Qwen3.5-2B	8,057	8	L7	E3 / 16,050	62.0
Qwen3.5-4B	7,012	6	L5	E2 / 5,252	71.0
Qwen3.5-9B	8,926	5	L4	E1 / 1,835	78.0
Qwen3.8-27B	4,132	8	L7	E2 / 2,556	90.0

Final maximizes Validation (%) across rounds; cross-round ties choose the earlier round. E = epoch; step is local to the selected round, not cumulative. Validation is a same-checkpoint mean of 3. Training continues the preceding round's selected LoRA adapter. Initial IGKS excludes trajectories/Single-Step; 9B uses a different generation batch.

does not establish the active task-set size, and a last-round task addition does not imply another rollout. Knowledge increments count admitted IGKS samples from new pages, elements, and transitions across seven sample types, excluding trajectory SFT and legacy Single-Step items. Initial seven-type IGKS pools contain 10,276/8,057/7,012/8,926/4,132 samples for 0.8B/2B/4B/9B/27B (Table 7). The 9B pool comes from a different generation batch; batch differences preclude attributing sample-count differences to student capability alone. Initial pool sizes are not Base training results, round increments, SMG object counts, or token budgets. Aggregate SMG counts and historical snapshots do not by themselves identify each run's graph version.

LoRA continuation and accumulated data. Each training round resumes the preceding round's checkpoint with the highest Validation TSR. The original backbone is not replaced by a merged model between rounds: training continues the same LoRA adapter at the selected checkpoint, rather

than merging its weights or initializing a new adapter. The full cumulative IGKS and admitted trajectory datasets are retained and used again, including historical and new samples; this is neither increment-only training nor a fresh run from Base. The supplemental configuration resets the optimizer and cosine schedule each round while continuing the selected adapter; adapter continuation therefore does not imply restoration of optimizer, scheduler, or RNG state.

Saved checkpoints and differences. Epoch/step values identify saved checkpoints; steps restart within each training round and do not specify the total training budget, batch size, or exact parent artifact. Base has no round increment, and Final repeats a selected row without adding tasks or samples again. Table 13 computes adjacent-loop differences from reported one-decimal means; its final column is selected Final minus Base, not last round minus Base. Missing later rounds are shown as unavailable, never extrapolated. These summaries do not establish statistical significance or matched-budget mechanism effects.

Table 8: Qwen3.5-0.8B rounds with general_e2e; initial pools: 2,970 tasks and 10,276 IGKS samples.

Stage	Task delta	IGKS delta	Epoch/step	Val	AW	MW
Base			–	18.0	9.5	2.6
Loop 1	1,992	3,602	E2 / 1,998	37.0	23.3	4.3
Loop 2	1,910	2,161	E3 / 5,550	41.0	24.1	5.1
Loop 3	1,834	1,441	E1 / 2,837	47.0	28.4	6.0
Loop 4	1,700	901	E2 / 7,748	51.0	31.0	6.8
Loop 5	1,524	540	E3 / 14,709	50.0	30.2	6.0
Loop 6	1,331	324	E1 / 5,888	52.0	31.0	7.4
Loop 7	1,134	180	E2 / 13,628	51.0	30.5	6.6
Final (L6)			E1 / 5,888	52.0	31.0	7.4

Scores (%): same-checkpoint means of 3. Val = Validation. Task delta: deduplicated subtasks + coverage tasks for next-round execution. IGKS delta: admitted knowledge, not trajectories or Single-Step samples. Epoch/step is round-local. Base blanks are unavailable, not zero; Final repeats the Validation-selected row without repeating increments.

Table 9: Qwen3.5-2B rounds with general_e2e; initial pools: 2,970 tasks and 8,057 IGKS samples.

Stage	Task delta	IGKS delta	Epoch/step	Val	AW	MW
Base			–	23.0	12.1	5.2
Loop 1	1,878	2,417	E2 / 1,616	42.0	25.0	8.0
Loop 2	1,715	1,450	E1 / 1,502	48.0	29.3	9.4
Loop 3	1,577	967	E3 / 6,885	46.0	28.4	8.8
Loop 4	1,409	604	E2 / 6,222	56.0	34.5	10.8
Loop 5	1,227	363	E2 / 7,812	61.0	37.1	11.4
Loop 6	1,044	218	E1 / 4,656	60.0	36.5	12.0
Loop 7	871	121	E3 / 16,050	62.0	37.1	10.8
Loop 8	717	73	E2 / 11,984	60.0	36.8	11.7
Final (L7)			E3 / 16,050	62.0	37.1	10.8

Definitions as in the 0.8B table. Scores (%): mean of 3; Task delta is for the next round, IGKS delta is admitted knowledge. Epoch/step is round-local; Base blanks are unavailable. Final is Validation-selected, not the last round.

Table 10: Qwen3.5-4B rounds with general_e2e; initial pools: 2,970 tasks and 7,012 IGKS samples.

Stage	Task delta	IGKS delta	Epoch/step	Val	AW	MW
Base			–	45.0	37.4	14.8
Loop 1	1,712	1,753	E1 / 613	60.0	44.8	19.1
Loop 2	1,460	1,052	E2 / 2,174	65.0	49.1	22.2
Loop 3	1,261	701	E2 / 3,220	70.0	54.3	24.8
Loop 4	1,068	438	E1 / 2,131	68.0	53.7	23.4
Loop 5	889	263	E2 / 5,252	71.0	54.0	25.6
Loop 6	729	158	E1 / 3,081	69.0	53.4	24.2
Final (L5)			E2 / 5,252	71.0	54.0	25.6

Definitions as in the 0.8B table. Scores (%): mean of 3; Task delta is for the next round, IGKS delta is admitted knowledge. Epoch/step is round-local; Base blanks are unavailable. Final is Validation-selected, not the last round.

Table 11: Qwen3.5-9B rounds with general_e2e; initial pools: 2,970 tasks and 8,926 IGKS samples.

Stage	Task delta	IGKS delta	Epoch/step	Val	AW	MW
Base			–	55.0	45.4	18.3
Loop 1	1,599	1,785	E2 / 1,236	70.0	54.3	25.1
Loop 2	1,291	1,071	E1 / 1,009	77.0	62.1	30.5
Loop 3	1,057	714	E2 / 2,856	76.0	61.2	28.8
Loop 4	857	446	E1 / 1,835	78.0	61.8	31.1
Loop 5	686	268	E2 / 4,424	77.0	61.5	29.9
Final (L4)			E1 / 1,835	78.0	61.8	31.1

Definitions as in the 0.8B table. Scores (%): mean of 3; Task delta is for the next round, IGKS delta is admitted knowledge. Epoch/step is round-local; Base blanks are unavailable. Final is Validation-selected, not the last round.

Table 12: Qwen3.8-27B rounds with general_e2e; initial pools: 2,970 tasks and 4,132 IGKS samples.

Stage	Task delta	IGKS delta	Epoch/step	Val	AW	MW
Base			–	78.0	72.0	52.1
Loop 1	1,432	600	E1 / 285	82.0	75.9	56.4
Loop 2	1,068	360	E2 / 936	85.0	78.4	59.0
Loop 3	808	240	E1 / 660	84.0	77.6	58.1
Loop 4	609	150	E2 / 1,682	87.0	80.2	62.4
Loop 5	459	90	E2 / 2,012	89.0	82.2	64.7
Loop 6	345	54	E1 / 1,151	88.0	81.6	63.8
Loop 7	260	30	E2 / 2,556	90.0	81.9	65.2
Loop 8	196	18	E1 / 1,391	89.0	81.7	64.1
Final (L7)			E2 / 2,556	90.0	81.9	65.2

Definitions as in the 0.8B table. Scores (%): mean of 3; Task delta is for the next round, IGKS delta is admitted knowledge. Epoch/step is round-local; Base blanks are unavailable. Final is Validation-selected, not the last round.

Table 13: Per-loop changes (percentage points), derived from rounded means.

Model	Metric	L1	L2	L3	L4	L5	L6	L7	L8	Final–Base
0.8B	AW	+13.8	+0.8	+4.3	+2.6	-0.8	+0.8	-0.5	–	+21.5
0.8B	MW	+1.7	+0.8	+0.9	+0.8	-0.8	+1.4	-0.8	–	+4.8
2B	AW	+12.9	+4.3	-0.9	+6.1	+2.6	-0.6	+0.6	-0.3	+25.0
2B	MW	+2.8	+1.4	-0.6	+2.0	+0.6	+0.6	-1.2	+0.9	+5.6
4B	AW	+7.4	+4.3	+5.2	-0.6	+0.3	-0.6	–	–	+16.6
4B	MW	+4.3	+3.1	+2.6	-1.4	+2.2	-1.4	–	–	+10.8
9B	AW	+8.9	+7.8	-0.9	+0.6	-0.3	–	–	–	+16.4
9B	MW	+6.8	+5.4	-1.7	+2.3	-1.2	–	–	–	+12.8
27B	AW	+3.9	+2.5	-0.8	+2.6	+2.0	-0.6	+0.3	-0.2	+9.9
27B	MW	+4.3	+2.6	-0.9	+4.3	+2.3	-0.9	+1.4	-1.1	+13.1

L1 is relative to Base; later loops are relative to the preceding supplied loop. Final–Base uses the Validation-selected round, not the last loop or the sum of all columns. – means no supplied round; no interpolation. AW = AndroidWorld; MW = MobileWorld. Scores are same-checkpoint means of 3. All models use MobileWorld's public general_e2e harness; AW values are evaluated with this harness, not official model scores.

J Controlled Comparisons

Does knowledge supervision help at all? The comparison pairs trajectory-only training with joint IGKS and trajectory training on 9B and 27B. Both start from the same Base and use the same frozen eligible trajectory pool, source policy, task curriculum, supervision configuration, and Validation rule. Trajectory-only excludes all knowledge targets, including the initial pool; it is not teacher-only distillation, which changes the target source. At a matched total target-token budget, the trajectory-only arm fills the knowledge allocation with additional replay from that same trajectory pool. Trajectory exposure therefore differs between arms, even though total target tokens are matched. Table 14 tests whether allocating part of a fixed budget to knowledge helps, not the universal necessity of IGKS.

Comparison scope and budgets. The controls examine reusable interface knowledge, incremental knowledge updates, and failure-directed practice. Table 18 specifies a target-token budget B of 120M for 9B and 240M for 27B, with four rounds, 20% knowledge/80% trajectory quotas, and collection caps. Matching target-token budgets does not imply equal action exposure or total multimodal compute. The comparisons concern these configurations rather than a universal ranking across model families.

Do knowledge and remedial-task increments help? The 2×2 comparison in Table 15 shares Base, initial knowledge and admitted trajectory pools, and frozen training-only observations. The factors are *knowledge increments* and *remedial-task increments*, not arbitrary new trajectories. Without knowledge increments, the model replays initial knowledge. Without remedial-task increments, it replays the frozen initial trajectory pool rather than recollecting original-task trajectories. With that factor enabled, reviewed subtasks are independently executed and their trajectories admitted before training. Both factors off still trains both streams; only knowledge increments enabled does not mean knowledge-only training. Increments are precomputed with one frozen collection policy and mastery/review configuration: knowledge-enabled cells share the same knowledge pools, and task-enabled cells share the same admitted subtask pools. Fork-specific actions or mastery checks do not change these inputs. Per-stream token quotas and supervision are fixed; auxiliary coverage additions are disabled. Replay, rejected proposals, and collection costs are included in the accounting. This comparison evaluates incremental data value, whereas the trajectory-only contrast removes knowledge supervision entirely. Controlled forks do not replace online runs in which trained policies change future observations.

Curriculum matrix. On 9B, the comparison includes flat/ordinary (A), flat/failure-directed (B), persistent-graph/ordinary (C), persistent-graph/failure-directed (D), and initial-plus-current-graph/failure-directed (E). On 27B, it retains B–E. These comparisons freeze a common knowledge-

Table 14: Knowledge-presence contrast (full-task TSR, %).

Training data	Qwen3.5-9B		Qwen3.8-27B	
	AW	MW	AW	MW
Trajectory-only	56.4	27.0	78.7	60.2
IGKS + trajectories	60.0	30.0	80.5	63.0

Same total target-token budget. Trajectory-only uses extra replay; action exposure differs. The joint row reuses the factorial both-on control, not the selected Final.

Table 15: IGKS increments versus remedial-task increments (full-task TSR, %).

Knowledge	Trajectories	Qwen3.5-9B		Qwen3.8-27B	
		AW	MW	AW	MW
Fixed replay	Fixed replay	53.2	25.0	76.8	56.9
New + replay	Fixed replay	55.8	26.9	78.1	59.0
Fixed replay	+ Subtasks	57.1	27.7	79.3	60.8
New + replay	+ Subtasks	60.0	30.0	80.5	63.0

Control endpoints are separate from the selected Final. Both streams remain active. Fixed replay uses frozen initial data; + Subtasks adds only executed, admitted trajectories. Matched per-stream token quotas, no extra coverage tasks.

supervision stream, allocation, and source observation history within each backbone. Flat memory retains the same raw observations and access to semantic annotations supplied to the graph arm, with matched retrieval/context budgets. Results concern the representation and retrieval pipeline rather than graph structure in isolation. The controlled D row is distinct from the endpoint of the fully adaptive two-stream system. The full method evolves both outputs, with future observations depending on the updated policy.

Why subtasks rather than full-task reruns? The acquisition comparison uses three choices from a common frozen set of training failures: recollect complete original tasks without adding goals; generate ordinary SMG tasks without conditioning on those failures; or generate failure-directed subtasks. The latter two reuse C and D under the same starting checkpoint, graph observations, knowledge stream, source failures, supervision, and budgets. Rerunning the original tasks collects new trajectories and is distinct from replaying a fixed trajectory pool in the factorial experiment. It also differs from teacher-only distillation: the rerun arm retains the same student/teacher intervention or diagnostic configuration as the subtask arm. Each arm augments the same retained full-task trajectory pool with its allotted acquisition data.

The three arms share target-token budgets, retry rules, collection caps, and task-independent reset policy; auxiliary coverage additions are disabled. Costs include setup, navigation, rejected generations, and failed attempts. Training-side matching uses application, horizon, and setup strata for ordinary and remedial tasks, but does not guarantee identical task difficulty. Operation reach and admitted supervision per budget help distinguish focused practice from merely easier goals. Residual distribution differences limit attribution to failure conditioning alone. Local supervised practice never entails resuming from an inaccessible failed state or exposing the parent task's hidden solution.

Results of the controlled contrasts. Joint training attains higher TSR than trajectory-only training, while the factorial comparison shows gains from both increments. These are not independent confirmations: the joint row is reused from the factorial both-on row under the same controlled-fork specification. On 27B AW, the combined increment gain is slightly less than the sum of the separate gains; joint benefit does not imply superadditivity. The acquisition comparison reuses C/D. Ordinary synthesis slightly trails original-task reruns on MW for both models, so adding novel tasks does not uniformly improve performance. These descriptive differences do not establish statistical significance or a model-size-only explanation.

Table 16: Acquisition policies (full-task TSR, %).

Acquisition	Qwen3.5-9B		Qwen3.8-27B	
	AW	MW	AW	MW
Original-task reruns	55.6	27.6	77.8	59.4
Ordinary SMG generation (C)	56.7	27.4	78.2	59.1
Failure-directed subtasks (D)	60.2	30.1	80.8	63.4

Shared IGKS stream and matched target-token budgets. C/D reuse the ablation rows exactly, not the selected Final. Reruns acquire new trajectories, unlike fixed replay.

Table 17: Fixed-pool supervision comparison (TSR, %), separate from full curriculum evolution.

Supervision	Qwen3.5-9B		Qwen3.8-27B	
	AW	MW	AW	MW
Teacher actions only	56.4	27.8	76.8	56.9
Teacher short explanation + actions	57.2	28.5	74.5	54.8
Unassisted resampling	52.1	24.6	77.2	57.8
Diagnostic rerollout	55.1	27.3	79.8	61.9

Control endpoints are separate from the selected Final. AW = AndroidWorld; MW = MobileWorld. The training-task pool is fixed across supervision conditions.

Evidence of transfer back to complete tasks. The diagnostic comparison fixes operation/error categories and task groups from training-side information before evaluation. It uses identical full-task resets and verifiers without Hint, Monitor, extra retries, or test-time decomposition. The independent benchmark and the related-task cohort are distinct evaluation populations; different IDs alone do not establish absence of semantic or instance overlap. Re-evaluation of parent training tasks is a separate repair diagnostic, not held-out generalization. Independent benchmark records remain read-only and never generate new practice.

The fixed-denominator diagnostic records targeted failures, operation reach, and conditional success among reached attempts. An error disappearing because the student fails earlier is not repair. New failures and unresolved diagnoses remain part of the result. High subtask success without full-task gains supports easier practice, not compositional transfer. Improved full-task success without the expected operation-level change supports a policy benefit but not the proposed bottleneck mechanism. Table 19 reports a separate cohort with explicit denominators and reach/failure counts; its outcomes are not inferred from the AW/MW TSR rows.

Supervision comparison. The fixed-pool comparison includes teacher action-only targets, the same teacher actions with requested short explanations, unassisted resampling, and diagnostic student rerollouts on 9B and 27B. Task generation is fixed. The first contrast varies explanation supervision; rerollout contrasts also change the source policy and potentially visited states. Retry opportunities and primary budgets are matched. Action exposure and CoT composition differ across target sources; pure teacher targets are not mixed Probe targets.

Format and cost controls. There is no separate format-only stage in the loop. Excluding Single-Step from IGKS does not eliminate format-learning effects: trajectory targets still supervise executed actions. Gains therefore cannot be attributed entirely to graph knowledge on that basis. Comparisons share the same Base checkpoint and specified initial data pools. Primary budgets count actual, unweighted target tokens including repeated exposure; collection budgets include rejected samples, verification, and retries. Equal example counts or the CoT weight of 0.2 do not establish equal compute.

J.1 Original-Paper Result Sources

The main manuscript's original-paper comparison reports the selected methods' own trained-model results, checked against local PDFs rather than reproduced in our harness. Source identities, SHA-256 fingerprints, page/table locations and metric definitions are recorded in `data/literature_results.json`. Original-paper observations cannot replace same-backbone, matched-budget teacher-only or curriculum controls.

OpenMobile. Table 1 on PDF page 7 of Cheng et al. (2026) reports the Qwen3-VL-8B-based Ours-8B SFT model: AW Pass@1 64.7 ± 3.2 and MW GUI-only Pass@1 17.7 ± 2.2. We show means only in the common table. Appendix B.1, PDF page 19, defines the spread as half-range over three runs, not standard deviation. AW has 116 tasks; the evaluated MW subset size is not explicit. The approximately 2,800 instructions and 34K action steps are used for expert-step SFT with learner history; Gemini-3.1-Pro-Preview provides expert supervision. Its Qwen3-VL action format and evaluation profile are not established as identical to our `general_e2e` setup.

HATS. Table 1 on PDF page 5 of Shao et al. (2026) reports overall AW success of 24.35% for its Qwen2-VL-7B variant. We retain the source precision. The main protocol describes 116 tasks with screenshots and accessibility trees, 1,000 synthesized trajectories, and full-parameter SFT rather than our LoRA setup. The 50.8% component-analysis figure is from a 20-task subset and is not substituted for the main result. MW is not reported; repeat count is unspecified.

UI-MOPD. Table 1 on PDF page 8 of Lian et al. (2026) reports MW task success of 12.0% for Qwen3-VL-8B-Thinking, trained by RL plus multi-teacher on-policy distillation. The evaluated MW set has 117 tasks; AW is not reported. Mobile data collection includes both AW and MW (PDF page 17), so this is not an unseen-MW transfer setting. Inference includes the current and four historical screenshots plus full text action history (page 19); no teacher is deployed at inference. The repetition count is unspecified.

MobileForge. Table 1 on PDF page 11 of Liu et al. (2026) gives ForgeOwl-8B, adapted from GUI-Owl-1.5-8B on 900 generated tasks: AW Pass@1 $78/116 = 67.2\%$. Its Pass@3 is 77.6% and is not mixed into the single-attempt column. Table 2 on page 12 gives MW GUI-only117 success of 41.0%, with MW excluded from adaptation by the authors' protocol. Training uses hint-contextualized step-level GRPO, not our repeated SFT. The exact MW repeat count and benchmark-time hint/harness configuration remain insufficiently specified for a matched unassisted comparison.

All four scores are original-paper reports, not independently validated raw outcomes. Backbone, observation input, training exposure and evaluation differences limit interpretation to method context; unavailable benchmark cells remain unavailable rather than receiving estimated scores.

K Training Budgets and Supplemental Configuration

This section specifies the controlled target-token budgets and supplemental training and evaluation settings. Equal target-token budgets do not imply equal compute or measured runtime efficiency.

The controlled-budget runs use four scheduled rounds and fixed target-token allocations, whereas the main runs use cumulative full datasets for three epochs over their recorded round counts. They are distinct schedules; equal epochs alone cannot establish budget matching. Knowledge omission reallocates its token quota to trajectory replay. Within-factorial comparisons retain both quotas; disabled increments replay initial pools. Entire responses are retained; final-example budget deviations count toward token accounting rather than being truncated.

Reported supplemental configuration. Training uses AdamW with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01, gradient clipping 1.0 and seed 42. The optimizer and cosine schedule reset each round while the selected adapter continues. Summed weighted target-token losses are normalized by the number of unmasked current-target tokens. Evaluation uses AW116/MW117/Validation100, at most 100 actions, one attempt without Monitor/Hint, and at most one infrastructure-only retry; model parse failures remain failures. Decoding uses temperature 0, top-$p = 1$, and an 8,192-token output cap, with aspect-ratio-preserving images up to a 1,280-pixel longest edge without upscaling.

Table 18: Controlled-protocol target-token budgets, separate from full cumulative training.

Model	Target tokens (M)	K quota %	Trajectory quota %	Rounds
9B	120	20	80	4
27B	240	20	80	4

Target-token budgets are cumulative across the scheduled rounds, not per round; K=knowledge. Stream quotas are token fractions, not sample fractions. Per-protocol caps: 200,000 environment steps, 30,000 teacher/judge calls, and 3 attempts per goal. Nominal 3 epochs per round with quotas/replay to match scheduled unweighted target-token exposure within each backbone while retaining complete responses. Distinct from full-cumulative three-epoch training; matching target-token budgets does not imply equal compute.

Table 19: Repair diagnostics on a separate related-task cohort.

Model	Acquisition	Reached	Target fails	Op. success %	Full TSR %
9B	Original-task reruns	78	24	69.2	52.0
9B	Ordinary SMG tasks	80	21	73.8	57.0
9B	Remedial subtasks	84	13	84.5	63.0
27B	Original-task reruns	90	13	85.6	76.0
27B	Ordinary SMG tasks	90	12	86.7	78.0
27B	Remedial subtasks	93	7	92.5	83.0

Each row uses 100 complete related tasks, separate from AW/MW benchmark denominators and subsets. Reached and target failures are counts with this fixed cohort denominator; target failures occur among reached attempts. Conditional operation success=(reached-target failures)/reached*100. Full TSR=full successes/cohort size*100. Full successes cannot exceed successful target operations. These control results are distinct from the selected Final.

Accounting includes the recorded 7/8/6/5/8 rounds; these counts do not imply a performance-based stopping policy.

K.1 Local Repair, Generalization, and Negative Cases

Diagnostic cohort results. All six rows describe a separate cohort of 100 complete tasks, not the AW/MW evaluation. The target operation is necessary within this cohort, so full successes never exceed reached attempts minus target failures. On 9B, conditional operation success increases from $54/78 = 69.2\%$ under reruns to $71/84 = 84.5\%$ under remedial practice, alongside an 11-point full-task gain. On 27B it increases from $77/90 = 85.6\%$ to $86/93 = 92.5\%$, alongside a 7-point full-task gain. These changes are consistent with operation repair within the cohort, but do not alone establish causal attribution or generalization to independent task families.

Schematic repair example. Consider a calendar task that requires a weekly recurrence. In this hypothetical example, the student navigates to the editor but confuses the recurrence option with a notification interval. Relevant page, element, and transition evidence grounds a remedial task that practices setting recurrence from an independently constructible baseline; a subsequent full-task evaluation uses a different event and goal instance. Improvement in both unassisted recurrence handling and complete-task success would support a reusable operation rather than answer memorization. This is a schematic example, not an observed trace or screenshot; it supplies no graph edge ID, success receipt, or training sample.

Limitations. The controls characterize the specified data pools, acquisition policies, and budgets; they do not isolate every representation, supervision, or model-family effect. Equal target-token budgets do not imply equal compute or measured runtime efficiency. Aggregate means and the schematic repair example do not provide repeat-level uncertainty or establish a causal repair mechanism. Original-paper comparisons use different protocols and do not establish a controlled ranking. Validation remains the sole checkpoint-selection criterion.

REFERENCES

Domenico Amalfitano, Anna Rita Fasolino, Porfirio Tramontana, Salvatore De Carmine, and Atif M. Memon. Using gui ripping for automated testing of android applications. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pp. 258–261, 2012. doi: 10.1145/2351676.2351717.

Kanzhi Cheng, Zehao Li, Zheng Ma, Nuo Chen, Jialin Cao, Qiushi Sun, Zichen Ding, Fangzhi Xu, Hang Yan, Jiajun Chen, Anh Tuan Luu, Jianbing Zhang, Lewei Lu, and Dahua Lin. Openmobile: Building open mobile agents with task and trajectory synthesis, 2026. URL https://arxiv.org/abs/2604.15093.

Quyu Kong, Xu Zhang, Zhenyu Yang, Nolan Gao, Chen Liu, Panrong Tong, Chenglin Cai, Hanzhang Zhou, Jianan Zhang, Liangyu Chen, Zhidan Liu, Steven Hoi, and Yue Wang. MobileWorld: Benchmarking autonomous mobile agents in agent-user interactive and MCP-augmented environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6142–6167, 2026. doi: 10.18653/v1/2026.acl-long.278. URL https://aclanthology.org/2026.acl-long.278/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, et al. Efficient memory management for large language model serving with PagedAttention. In *SOSP*, 2023.

Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. DroidBot: A lightweight UI-guided test input generator for android. In *ICSE Companion*, 2017.

Niu Lian, Alan Chen, Zhehao Yu, Chengzhen Duan, Fazhan Liu, Hui Liu, Pei Fu, Jian Luan, Yaowei Wang, Shu-Tao Xia, and Jinpeng Wang. Ui-mopd: Multi-platform on-policy distillation for continual gui agent learning, 2026. URL https://arxiv.org/abs/2607.04425.

Guangyi Liu, Pengxiang Zhao, Gao Wu, Yiwen Yin, Mading Li, Liang Liu, Congxiao Liu, Zhang Qi, Mengyan Wang, Liang Guo, and Yong Liu. MobileForge: Annotation-free adaptation for mobile GUI agents with hierarchical feedback-guided policy optimization, 2026. Version 1, 18 June 2026.

Rui Shao, Ruize Gao, Bin Xie, Yixing Li, Kaiwen Zhou, Shuai Wang, Weili Guan, and Gongwei Chen. Hats: Hardness-aware trajectory synthesis for gui agents, 2026. URL https://arxiv.org/abs/2603.12138.