# PERSONALARC: HARNESSES AT BOTH ENDS OF AGENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Personal-context agents must answer from task-scoped evidence while learning from prior executions. This creates two linked challenges: controlling how evidence enters a training trace, and deciding whether a learned answer should replace the base answer. We introduce `PersonalArc`, a framework that places **restricted harnesses at both ends of agent learning**. A task-conditioned harness declares stage inputs, artifact types, tool scopes, and resource limits before execution. Workspace-grounded mutations are audited and distilled into content-only stage supervision. At inference time, a back-end fusion harness governs evidence access, rubric-blind composition, and final-answer selection. It adopts the fused candidate when both comparison orders prefer it and numeric retention passes, returning the base answer exactly on rejection. Together, the two harnesses link experience acquisition and answer deployment, turning audited executions into reusable supervision while keeping final-answer adoption under explicit control. We evaluate the full evaluation sets of four benchmarks, with three generating backbones in the main comparison. Averaged across these backbones, `PersonalArc` improves Yes Rate and Coverage by **27.0%** and **18.2%** relative to Base Search, respectively, while increasing F1 by **6.77 points**. It also outperforms the evaluated aggregation baselines at every tested token ceiling. Our implementation is available in this anonymous repository.

## 1 INTRODUCTION

Personal-context agents answer questions and carry out requests using a user's files, conversations, and accumulated interaction history (Zhang et al., 2025c; Liu et al., 2025; Xu et al., 2026; Luo et al., 2026). A request may depend on a revised travel plan, a recorded expense, or a decision scattered across conversations, linking profile modeling, memory, planning, and action execution (Xu et al., 2026; Zhang et al., 2026c). Recent benchmarks make the challenge concrete. CL-bench evaluates learning from task-specific context, CL-bench Life extends this setting to fragmented everyday records, and HippoCamp places agents in personal file systems with multimodal evidence (Dou et al., 2026b;a; Yang et al., 2026). An effective agent must retrieve evidence, reason over dependencies, and preserve task-relevant facts in the final response.

Retrieval and memory systems organize information for future use (Packer et al., 2023; Zhang et al., 2025a); MemSkill learns reusable operations for extracting and revising memories (Zhang et al., 2026b). Automated workflow methods configure how models, tools, and intermediate results interact (Zhang et al., 2025b; Wang et al., 2026), while AutoHarness and Meta-Harness optimize executable control code around a model (Lou et al., 2026; Lee et al., 2026). Self-Harness, Agentic Harness Engineering, and MemoHarness further use execution experience to revise or adapt this control layer (Zhang et al., 2026a; Lin et al., 2026; Huang et al., 2026b). These methods connect learned components with the workflows that govern execution.

Reusing executions for learning introduces a specific requirement: ❶ each supervised stage must receive the task-visible evidence and declared artifacts available to that stage at deployment. Trajectory fine-tuning and synthetic teaching transfer agent experience into model parameters (Zeng et al., 2024; Mitra et al., 2024); in personal-context tasks, a stage-level evidence interface makes the transferred behavior compatible with the deployed workflow.

Answer adoption creates a second requirement. ❷ Base and student outputs can be complementary, yet revising an answer can remove correct details. For example, a synthesized response might recover the purpose of a payment while omitting its recorded amount or date. Answer refinement



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.

and tool-grounded critique provide mechanisms for revision (Madaan et al., 2023; Gou et al., 2024), but the final deployment decision still needs to specify which existing details to retain and when to return the original answer. Together, these requirements call for explicit **training-trace admission** and **final-answer selection** rules across the learning and deployment pipeline.

`PersonalArc` connects learning and answer adoption through explicit evidence interfaces. Adaptive harnesses govern execution, while workflow portfolios route queries to reusable structures (Lee et al., 2026; Yuan et al., 2026). Our BUILD stage specifies a bounded graph of stages, typed artifacts, tools, and resource limits. Static validation and runtime checks enforce these declarations within the authorized workspace, giving execution and supervision a common boundary for task-scoped evidence.

LEARN combines workspace-grounded task mutation with stage-level distillation. Answer-preserving mutations vary the request; constraint-changing mutations recompute answers and evidence from the same workspace. An auditor isolated from the task generator's hidden context checks tasks and teacher traces before their content-only stage input–output pairs train a low-rank student. The stage interface aligns supervision inputs with the evidence available during execution.

FUSE, the **back-end fusion harness**, governs candidate synthesis and final-answer selection through declared inputs, comparison stages, and fallback rules. Its rubric-blind composer merges Base and student outputs using authorized evidence. The harness adopts Fusion when both comparison orders favor it and numeric retention passes; otherwise, it returns Base verbatim. Evaluation on the original benchmark sets measures adaptation after training on reference-informed synthetic tasks. In this setting, Gemma-3-12B improves LongMemEval-S accuracy and F1 over Base Search by **7.40** and **6.70** points (Table 1).

We summarize our contributions in the following three points:

❶ ***Problem Formulation.*** We identify two linked requirements for learning from personal-context executions: preserving declared evidence boundaries when constructing supervision, and deciding when a learned answer should replace Base. A shared harness interface connects training-trace admission to final-answer selection for personal-context learning and deployment.

❷ ***Unified Harness Framework.*** We propose `PersonalArc`, a framework with restricted harnesses at both ends of agent learning. Task-conditioned execution graphs and audited workspace-grounded mutations produce content-only stage supervision. A back-end fusion harness then controls the evidence available to answer composition, executes comparison and retention checks, and routes their outcomes to the fused response or the verbatim Base fallback.

❸ ***Empirical Assessment.*** We examine adaptation using benchmark references across Life, HippoCamp, LongMemEval-S, and $\tau$-bench retail, with three generating backbones. System comparisons, component analyses, and budget sweeps characterize answer quality and inference cost under this protocol.

## 2 RELATED WORK

**Personal context and execution experience.** Personalized agents connect profiles and memory to planning and action (Xu et al., 2026; Zhang et al., 2026c). LoCoMo and LongMemEval test conversational recall (Maharana et al., 2024; Wu et al., 2025); CL-bench, Life, and HippoCamp extend evaluation to context learning and personal records (Dou et al., 2026b;a; Yang et al., 2026). Memory substrates affect factual and sequential tasks differently (Huang et al., 2026a). AgentTuning and AgentInstruct transfer execution experience through supervised trajectories and synthetic teaching (Zeng et al., 2024; Mitra et al., 2024), while MemSkill and MemoHarness learn memory routines or reuse execution diagnoses (Zhang et al., 2026b; Huang et al., 2026b). `PersonalArc` connects these concerns through audited stage input–artifact pairs: workspace-grounded mutations expand supervision while preserving declared evidence dependencies.

**Automated workflows and harnesses.** MaAS and MAS$^2$ automate workflow construction (Zhang et al., 2025b; Wang et al., 2026), and FlowBank routes queries through reusable portfolios (Yuan et al., 2026). AutoHarness and Meta-Harness optimize executable control code using environment feedback and prior traces (Lou et al., 2026; Lee et al., 2026); Self-Harness and Agentic Harness Engineering use model weaknesses and structured execution records to guide revisions (Zhang et al., 2026a; Lin et al., 2026). `PersonalArc` specifies evidence flow in a restricted graph language whose typed artifacts, context boundaries, and fallback rules connect task execution to distillation and answer fusion.

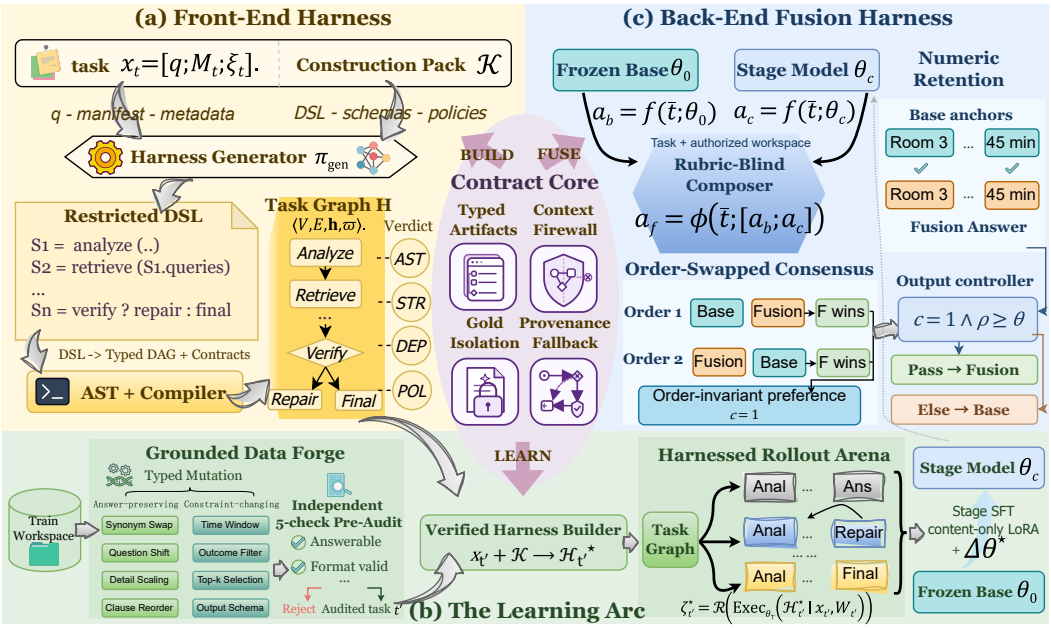



Figure 1: Overview of `PersonalArc`. We first execute audited, workspace-grounded mutations with restricted task harnesses, then distill stage trajectories into a student. At inference, the back-end fusion harness controls rubric-blind composition, order-swapped comparison, and numeric retention. It selects Fusion when both checks pass and returns Base verbatim on rejection or synthesis failure.

**Answer refinement and selection.** Self-Refine and CRITIC revise answers using feedback or tools; LLM-Blender combines pairwise ranking with fusion (Madaan et al., 2023; Gou et al., 2024; Jiang et al., 2023). `PersonalArc` places composition and selection inside a back-end fusion harness. Its contract declares authorized evidence, order-swapped comparison, numeric retention, and verbatim Base fallback. The composer generates a candidate, while the surrounding harness determines whether to deploy it through explicit checks and output control.

## 3 Methodology

Figure 1 organizes `PersonalArc` into Build, Learn, and Fuse. Build compiles the runtime-visible task and construction pack into an admitted graph. Learn audits complete mutated tasks, exposes only their runtime-visible projections during execution, and distills retained stage records into $\theta_c$. Fuse composes Base and student answers, checks order-swapped preference and numeric retention, and routes to Fusion or verbatim Base. The shared Contract Core declares artifact types, context boundaries, provenance, and fallback rules. Graph validation, content auditing, and answer selection enforce these declarations at their respective stages; Algorithm 1 summarizes the procedure.

### 3.1 Problem Setting

**Runtime-visible task and hidden contract.** For task $t$, let $q$ denote the request, $M_t$ the file manifest, $\xi_t$ the task metadata, and $W_t \subseteq W$ the authorized workspace slice. A double bar separates runtime observations from the hidden evidence and evaluation contract $C_t$:

$$t \triangleq \big\langle \underbrace{[q; M_t; \xi_t]}_{x_t},\ W_t \ \big\|\ \{c_j\}_{j=1}^{m_t} \big\rangle. \tag{1}$$

Each hidden $c_j \in C_t$ specifies an answer requirement, an evidence locator in $W_t$, and a format, numeric, or semantic check. The harness generator and stage executors observe $x_t$, but never the hidden answer, rubric, evidence list, or $C_t$; only offline construction, audits, and scoring use the latter.

**Restricted conditional harness.** A harness is a forward conditional graph $\mathcal{H} = \langle V, E, \mathbf{h}, \varpi \rangle$. Each stage contract $h_v$ specifies a goal, type, output schema, declared input and output artifacts, one allowed executor scope, a resource budget, and a bounded routing rule $\delta_v$. The final-answer contract $\varpi$ names a preferred answer artifact and an ordered fallback list. On every reachable route,

each consumed artifact must have a producer on that route or an explicit fallback. Edges enumerate allowed forward transitions, activated by observed artifacts. Search, restricted Terminal, and isolated JSON reasoning use separate tool scopes and artifact contracts.

### 3.2 Verified Task Harness

**Task-conditioned program synthesis.** The generator receives $x_t$ and a fixed construction pack $\mathcal{K}$ containing the restricted domain-specific language (DSL), graph-writing rules, safety policies, examples, pattern cards, and prior error-avoidance rules. It proposes at most $R$ programs; the abstract syntax tree (AST) parser and compiler translate each proposal into a graph whose declarations can be checked before execution:

$$p_i \sim \pi_{\text{gen}}(\cdot \mid x_t, \mathcal{K}) \xrightarrow{\mathcal{C}} \mathcal{H}_i, \qquad i \in [R]. \tag{2}$$

The generator selects stages, artifact flow, tools, budgets, routes, repair, and final-answer source within fixed syntax, whitelists, workspace boundaries, context isolation, resource ceilings, quality gates, and default-harness rules for every generated task program.

**Pre-execution admission.** Admission checks syntax, graph structure, dependencies, and task policies. Let $\Lambda = \{\text{ast}, \text{str}, \text{dep}, \text{pol}\}$ index the restricted syntax, forward graph structure, artifact/evidence dependencies, and task-conditioned tool, budget, and quality policies. Reserving $\mathcal{H}_{R+1} \equiv \mathcal{H}_0$ for the frozen valid Search-first default, validation and retry select the first proposal to pass every check, with the default providing the final fallback:

$$\mathcal{H}_t^\star = \mathcal{H}_{i_t^\star}, \quad i_t^\star = \min\big\{i \in [R+1] \;\big|\; \textstyle\prod_{\ell \in \Lambda} V_\ell(\mathcal{H}_i \mid x_t) = 1\big\}. \tag{3}$$

Syntax and structure checks enforce the restricted grammar and bounded forward routing. Dependency and policy checks validate artifact producers, evidence inputs, tool scopes, and resource declarations. Rejection returns a stage-level reason for regeneration. Runtime checks enforce access and resource limits, while offline scoring assesses evidential support and answer quality. See the admission rules in Appendix B and recorded diagnostics in Appendix D.

**Context firewall and conditional execution.** The construction pack and generator conversation leave the context after compilation. At stage $v_k$, execution follows the schema-constrained transition

$$(x_t, W_t, h_{v_k}, \mathbf{r}_{v_k}^-) \xrightarrow{\mathsf{Step}_{\theta_\mathrm{T}}} r_k \models \sigma_{v_k} \xrightarrow{\delta_{v_k}} v_{k+1}. \tag{4}$$

The executor sees only $x_t$, task-scoped access to $W_t$, $h_{v_k}$, and declared artifacts $\mathbf{r}_{v_k}^-$ from executed stages. Schema-valid results activate $\delta_{v_k}$; repeated transitions form the bounded trace $\zeta_t^g$. Search restricts itself to files declared for the current task; Terminal exposes only bounded file listing, matching, and reading; and the runtime denies cross-task reads, future or undeclared artifacts, illegal tool combinations, and budget violations during each stage execution.

**Evidence-conditioned repair and runtime recovery.** Verification emits a slot checklist and evidence-linked instructions for missing requirements or violated constraints. Repair edits only identified, supported defects. Final audit normalizes format and removes unsupported content without new retrieval or deletion of supported slots; trace auditing checks compliance.

After the generated route ends, the runtime applies the artifact fallback order $\varpi$, a rubric-blind post-execution quality gate $Q_{\text{post}}$, and, when justified, a bounded short-answer rescue. If the run remains unreliable, it executes the default Search-first harness and compares the two runs under frozen operational criteria. Recovery maps the admitted harness and generated trace to the retained execution:

$$\mathcal{H}_t^\star \xrightarrow{\mathsf{Exec}_{\theta_\mathrm{T}}(\cdot \mid x_t, W_t)} \zeta_t^g \xrightarrow{\mathcal{R}_{\mathcal{H}_0}^{\varpi, Q_{\text{post}}}} \zeta_t^\star. \tag{5}$$

The selected trace $\zeta_t^\star$ retains whether its answer came from the generated route, an earlier answer artifact, short-answer rescue, or the default harness, together with all stage, tool, quality-gate, and fallback provenance. The post-execution gate rejects execution errors, empty or refusal-like outputs, abnormal shortness, empty derived evidence, insufficient file coverage, and inconsistent verification.

### 3.3 AUDITED CORPUS DISTILLATION

**Workspace-grounded mutation.** Let $\mathcal{Q}_{\text{src}}$ denote the original benchmark tasks used for synthetic-data construction. An operator $\mu_o$ drawn from the fixed operator set $\mathcal{O}$ rewrites the request while retaining the same authorized workspace. Answer-preserving paraphrases must preserve entities, numbers, and constraints. Constraint-changing operators, such as modifying a time window, filter, cutoff, comparison axis, or output format, must recompute the answer and rebuild the evidence contract from $W_t$. Teacher execution admits synthetic tasks that stay within one source, supply sufficient evidence, have unambiguous answers, and rely solely on declared runtime-visible evidence.

**Independent audit and successful trace selection.** An auditor that shares no hidden generator context evaluates answerability, evidence sufficiency, reference faithfulness, consistency with the declared mutation, and format validity through binary verdicts $z_1, \ldots, z_5$. Infrastructure failures are tracked separately and never count as semantic passes or rejections. For an audited task $t'$ and the runtime-selected teacher trace $\zeta^\star$, the three necessary admission layers are

$$V_{\text{keep}}(t', \zeta^\star) = \underbrace{\prod_{j=1}^{5} z_j(t')}_{\text{content audit}} \underbrace{V_{\text{run}}(\zeta^\star, t')}_{\text{valid trace}} \underbrace{\mathbb{1}[s(\zeta^\star) \geq \eta]}_{\text{quality gate}}. \tag{6}$$

Here $V_{\text{run}} = 1$ requires source and workspace validity, no runtime error or reference leak, a nonempty parseable answer, valid evidence-dependent artifacts, and a replayable evidence-bound trace after recovery; $s$ is the frozen offline trajectory score. Pairs enter the corpus only if $V_{\text{keep}} = 1$.

**Stage-level content supervision.** Let $\mathcal{Z}$ contain the retained task–trace pairs, and let $\mathcal{L}_\zeta$ be the LLM calls retained by the trace builder. Each input contains the runtime-visible task, the isolated stage contract, and only the artifacts declared for that call; the target is the recorded JSON or DSL content with private reasoning removed. The resulting content-only corpus is

$$\mathcal{D} = \bigcup_{(t',\zeta)\in\mathcal{Z}} \left\{ \left( [x_{t'}; h_v; \mathbf{r}_v^-], [r_v]_{\text{c}} \right) \mid v \in \mathcal{L}_\zeta \right\}. \tag{7}$$

Here $[r_v]_{\text{c}}$ is the content projection. Metadata retains task, source, partition, operator, stage, score, and provenance; source, stage, operator, and length caps prevent a small number of workspaces or stage types from dominating the corpus. Supervised fine-tuning (SFT) uses low-rank adaptation (LoRA) (Hu et al., 2022) to train the student on retained content-token targets:

$$\Delta\theta^\star = \arg\min_{\Delta\theta\in\mathcal{A}_\kappa} \mathbb{E}_{(x,y)\sim\mathcal{D}}[-\log P_{\theta_0+\Delta\theta}(y \mid x)]. \tag{8}$$

Here $\mathcal{A}_\kappa$ applies the rank bound $\kappa$ to the LoRA update of each targeted weight matrix. We denote the resulting learned stage model parameters by $\theta_c = \theta_0 + \Delta\theta^\star$.

**Matched control.** To compare original-trace and mutation-stage supervision, the Control corpus $\mathcal{D}_{\text{c}}$ uses successful original traces, whereas the mutation corpus $\mathcal{D}_{\text{m}}$ uses audited mutations from task sources. Let $\mathcal{S}_i$ be the source set, $\mathbf{b}_i = (|\mathcal{D}_i|, U_i)$ collect row count and optimizer updates, and $\Delta_T = |T_{\text{c}} - T_{\text{m}}| / \max\{T_{\text{c}}, T_{\text{m}}\}$ measure the relative loss-token gap. Matching requires

$$V_{\text{m}} = \mathbb{1}[\mathcal{S}_{\text{c}} = \mathcal{S}_{\text{m}} \wedge \mathbf{b}_{\text{c}} = \mathbf{b}_{\text{m}} \wedge \Delta_T \leq \epsilon]. \tag{9}$$

A pair is $\epsilon$-matched when $V_{\text{m}} = 1$. Appendix B.2 gives available corpus statistics and training settings.

### 3.4 BACK-END FUSION HARNESS

The back-end fusion harness composes paired answers under a declared evidence interface and selects the final response through comparison and numeric-retention checks.

**Rubric-blind candidate synthesis.** Let $f(\cdot;\theta)$ denote the fixed task-harness runtime instantiated with stage model $\theta$. We write $f_b = f(\cdot;\theta_0)$ and $f_c = f(\cdot;\theta_c)$. They share the task-harness policy, tools, and per-call limits; the task-conditioned generator determines the admitted graph for each execution. The candidate system may apply one adapter globally or route a terminal-stage adapter

Table 1: **System comparison after benchmark-informed adaptation.** Complete Life (405), HippoCamp (581), retail (115), and LongMemEval-S (500) sets; Qwen3.5-9B judges Life/HippoCamp. Values are percentages except Score (/5); Δ is PersonalArc minus Base in points. Life columns retain the original all-system score summaries (Orig. Yes/Cov.); strict task Yes Rate and rubric-micro Coverage for Base and PersonalArc appear in Table 10. Bold/underline mark the highest/second-highest values within each backbone where shown. † marks accuracies aligned to integer counts from reported percentages (Appendix B.1).

| | CL-bench Life | | HippoCamp | | $\tau$-bench retail | | LongMemEval-S | |
|---|---|---|---|---|---|---|---|---|
| System | Orig. Yes | Orig. Cov. | Acc. | Score | pass$^1$ | pass$^4$ | Acc. | F1 |
| **Qwen3.6-27B** | | | | | | | | |
| Base Search | 11.43 | 11.86 | 44.06 | 2.611 | 47.22 | 33.33 | 57.20 | 54.60 |
| Terminal Agent | 10.94 | 12.05 | 45.61$^\dagger$ | 2.667 | 50.00 | 36.11 | 57.80 | 55.60 |
| ReAct | 11.32 | 12.63 | 47.33$^\dagger$ | 2.722 | 52.78 | 37.35 | 59.20 | 57.00 |
| Self-Consistency@5 | 11.16 | 12.42 | 48.02$^\dagger$ | 2.778 | 52.84 | 41.67 | 60.00 | 58.20 |
| Mixture-of-Agents | 11.38 | 12.71 | 49.74$^\dagger$ | 2.833 | 53.62 | 42.65 | 60.60 | 58.80 |
| Matched Control SFT | 10.86 | 12.31 | 52.15$^\dagger$ | 3.000 | 50.00 | 38.89 | 58.00 | 56.10 |
| Mutation SFT | 11.79 | 12.88 | 51.81$^\dagger$ | 2.944 | 55.56 | 44.44 | 61.20 | 59.40 |
| `PersonalArc` | **16.22** | **14.13** | 55.94 | **3.442** | 58.33 | 47.22 | **63.40** | **61.20** |
| Δ vs. Base Search | +4.79 | +2.27 | +11.88 | +0.831 | +11.11 | +13.89 | +6.20 | +6.60 |
| **Gemma-3-12B** | | | | | | | | |
| Base Search | 8.72 | 9.34 | 30.29$^\dagger$ | 2.167 | 30.56 | 19.44 | 47.80 | 46.70 |
| Terminal Agent | 8.34 | 9.51 | 32.36$^\dagger$ | 2.222 | 33.33 | 22.22 | 49.60 | 47.80 |
| ReAct | 8.59 | 9.78 | 34.94$^\dagger$ | 2.278 | 36.11 | 25.00 | 51.00 | 49.20 |
| Self-Consistency@5 | 8.48 | 9.66 | 34.60$^\dagger$ | 2.333 | 35.45 | 27.78 | 51.80 | 50.40 |
| Mixture-of-Agents | 8.67 | 9.85 | 37.01$^\dagger$ | 2.389 | 38.89 | 27.91 | 52.40 | 51.00 |
| Matched Control SFT | 8.27 | 9.44 | 38.04$^\dagger$ | 2.500 | 37.13 | 26.23 | 49.80 | 48.30 |
| Mutation SFT | 9.06 | **10.02** | 40.10$^\dagger$ | 2.444 | 41.67 | 30.56 | 53.00 | 51.60 |
| `PersonalArc` | **10.94** | 9.97 | 44.41$^\dagger$ | **2.684** | 44.44 | 33.33 | **55.20** | **53.40** |
| Δ vs. Base Search | +2.22 | +0.63 | +14.11$^\dagger$ | +0.517 | +13.88 | +13.89 | +7.40 | +6.70 |
| **Mistral-Small-24B** | | | | | | | | |
| Base Search | 10.18 | 10.12 | 35.97$^\dagger$ | 2.389 | 41.61 | 27.38 | 53.20$^\dagger$ | 51.30 |
| Terminal Agent | 9.74 | 10.54 | 37.69$^\dagger$ | 2.444 | 44.45 | 30.56 | 54.20 | 52.10 |
| ReAct | 10.03 | 11.36 | 40.10$^\dagger$ | 2.500 | 47.22 | 33.33 | 55.60 | 53.50 |
| Self-Consistency@5 | 9.90 | 11.21 | 39.59$^\dagger$ | 2.556 | 47.32 | 36.51 | 56.40 | 54.70 |
| Mixture-of-Agents | 10.13 | 11.46 | 41.14$^\dagger$ | 2.611 | 50.06 | 36.84 | 57.00 | 55.30 |
| Matched Control SFT | 9.66 | 10.97 | 43.37$^\dagger$ | **2.722** | 44.49 | 33.43 | 54.40 | 52.60 |
| Mutation SFT | 10.51 | 11.79 | 48.36$^\dagger$ | 2.667 | 50.00 | 38.89 | 58.20 | 56.30 |
| `PersonalArc` | **11.58** | **13.03** | 50.26$^\dagger$ | 2.611 | 52.78 | 41.67 | **60.20** | **58.30** |
| Δ vs. Base Search | +1.40 | +2.91 | +14.29$^\dagger$ | +0.222 | +11.17 | +14.29 | +7.00$^\dagger$ | +7.00 |

only to verification, repair, and finalization. Writing $\bar{t} = (x_t, W_t)$, $a_b = f_b(\bar{t})$, and $a_c = f_c(\bar{t})$, the harness exposes the runtime task, authorized workspace, and paired answers to the frozen composer $\phi$. The hidden contract and evaluation rubric remain outside this interface:

$$a_f = \phi(\bar{t}; [a_b; a_c])\,. \tag{10}$$

The composer merges compatible details, resolves conflicts against the authorized workspace, and preserves concrete entities, dates, numbers, timelines, and attribution. The harness checks the synthesis output before comparison: an empty, failed, or unparsable call returns $a_b$ immediately; a valid $a_f$ proceeds to the decision stages. Appendix B describes the retrieval provenance associated with these answers.

**Order-swapped consensus and numeric retention.** The comparator $\psi$ scores the same pair under both presentation orders, and $\mathrm{Rec}_{\mathcal{N}}(a_f \mid a_b)$ gives the fraction of number and date tokens in Base that survive in Fusion, or one if Base contains none. The decision vector is

$$\mathbf{g}(a_b, a_f) = \big(\underset{c}{\mathbb{1}}[\psi(a_b, a_f) = a_f]\,\underset{c}{\mathbb{1}}[\psi(a_f, a_b) = a_f]\;,\; \underset{\rho}{\mathrm{Rec}_{\mathcal{N}}}(a_f \mid a_b)\big)^\top. \tag{11}$$

The first coordinate $c$ requires order-invariant preference for Fusion. The second, $\rho$, measures surface-form retention of Base numbers and dates. Together they control presentation-order sensitivity and preservation of concrete details; task scoring measures the factual quality of the selected answer. Appendix B lists the default threshold and model roles.

**Deterministic output control.** The harness routes the decision vector to its final output controller. Let $\Pi_0(x, y) = x$ and $\Pi_1(x, y) = y$, and interpret $\succeq$ componentwise. For a threshold $\theta \in [0, 1]$, the controller returns

$$a_t^\star = \Pi_{\mathbb{1}[\mathbf{g}(a_b, a_f) \succeq (1,\theta)^\top]}(a_b, a_f). \tag{12}$$

The controller returns $a_f$ when both checks pass and $a_b$ verbatim otherwise. Algorithm 1 specifies offline learning in Parts 1–2 and per-task harness execution in Part 3.

**Paired evaluation.** Paired evaluation compares each selected answer with its Base answer under the same judge. Task-level score differences distinguish improved, degraded, and tied outcomes; Appendix A defines these statistics.

## 4 Experiments

We evaluate overall quality, component behavior, and inference cost through three research questions:

- **RQ1: Overall effectiveness.** How does `PersonalArc` compare with scaffold, aggregation, and supervised baselines across benchmarks and backbones?
- **RQ2: Component behavior.** How do learning, fusion, and the retention threshold affect quality?
- **RQ3: Quality and cost.** How does performance change under common token ceilings, and how much of each ceiling does a method actually use?

### 4.1 Experimental Setup

**Benchmarks and metrics.** The evaluation covers CL-bench Life (405 tasks) (Dou et al., 2026a), HippoCamp (581 tasks across profiles; accuracy at score $\geq 3/5$ and mean Score) (Yang et al., 2026), $\tau$-bench retail (115 tasks; pass$^1$/pass$^4$) (Yao et al., 2025), and LongMemEval-S (500 questions; QA accuracy and auxiliary F1) (Wu et al., 2025). Life Yes Rate is the fraction of tasks passing every rubric item, and Coverage is the fraction of all rubric items passed. The Base–`PersonalArc` pairs use these definitions; the broader Life configuration comparisons retain their original score summaries (Appendix B.1).

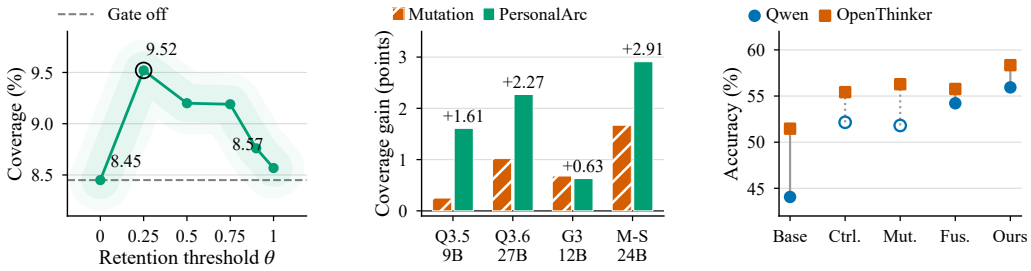
**Adaptation data and execution isolation.** We construct reference-informed synthetic supervision from each benchmark's full task set, including its questions, personal files or histories, reference answers, and rubrics, then evaluate the adapted system on the original tasks. The comparisons characterize the combined effects of this supervision and the inference strategy. Each question runs in an isolated `bwrap` environment, with HippoCamp additionally separating role-specific personal-file contexts. Base and student executions use isolated environments before the final fusion stage.

Table 2: **Life configuration comparison.** Qwen3.5-9B generation/judgment on 405 tasks. Orig. Yes/Cov. retain the original configuration score summaries; ΔCov. is the point difference from Base Search in that summary. Bold marks column maxima.

| Configuration | Orig. Yes ↑ | Orig. Cov. ↑ | ΔCov. |
|---|---|---|---|
| Base Search | 7.69 | 7.91 | 0.00 |
| Auto Harness | 7.58 | 7.37 | -0.54 |
| Original stage SFT | 7.63 | 7.69 | -0.22 |
| Mutation stage SFT | 8.11 | 8.16 | +0.25 |
| Terminal-repair SFT | 7.97 | 7.92 | +0.01 |
| Rubric-blind fusion | 8.41 | 8.52 | +0.61 |
| Two-way selector | 8.73 | 8.62 | +0.71 |
| `PersonalArc` | **9.29** | **9.52** | **+1.61** |

Table 3: **Default Life inference cost.** Qwen3.5-9B means over 405 tasks: input/output tokens and end-to-end latency. Bold marks the lowest costs.

| System | Tokens (k) ↓ | Time (s) ↓ |
|---|---|---|
| Base Search | **15.2** | **23** |
| Terminal Agent | 25.4 | 45 |
| ReAct | 33.3 | 52 |
| SC@5 | 21.7 | 35 |
| MoA | 27.1 | 57 |
| Control SFT | 23.8 | 41 |
| Mutation + repair | 29.5 | 49 |
| `PersonalArc` | 24.5 | 44 |




(a) Life threshold sweep. (b) Life rubric comparisons. (c) HippoCamp judge sensitivity.

Figure 2: **Threshold, backbone, and judge comparisons.** Life panels (a–b) use the original Coverage convention (Appendix B.1). (a) Qwen3.5-9B on full Life ($n = 405$); ring: $\theta = 0.25$; halo: visual guide; dashed "gate off": numeric gate disabled, consensus retained. (b) Life differences from Base (Table 14). (c) Two judges score the same Qwen3.6-27B answers on 581 HippoCamp tasks (Table 5); open circles mark count-aligned accuracies. Ctrl./Mut./Fus.: Control SFT, Mutation SFT, ungated fusion.

**Generating models and judges.** The main comparison uses Qwen3.6-27B (Qwen Team, 2026b), Gemma-3-12B (Gemma Team, 2025), and Mistral-Small-24B (Mistral AI, 2025). Qwen3.5-9B judges Life and HippoCamp and generates and judges the component and budget studies (Qwen Team, 2026a). OpenThinkerAgent-32B re-scores the same Qwen3.6-27B HippoCamp answers to assess judge sensitivity (OpenThoughts-Agent Team, 2026).

**Comparison systems.** Base Search and Terminal Agent adapt HippoCamp's retrieval and terminal interfaces (Yang et al., 2026). ReAct interleaves reasoning with tool actions (Yao et al., 2023). Aggregation baselines use Self-Consistency@5 (SC@5) (Wang et al., 2023) and Mixture-of-Agents (MoA) (Wang et al., 2025). Matched Control SFT and Mutation SFT compare original and mutated supervision, using tasks synthesized with DeepSeek-V4-Flash (DeepSeek-AI, 2026). `PersonalArc` combines the student with fusion and selection; rubric-blind Fusion is the candidate before gating. Appendix B.2 records corpus sizes and training settings; Eq. (9) defines the matching criterion.

**Evaluation protocol.** Comparisons use the same tasks and scorers. The main table compares complete systems; the threshold analysis reports sensitivity to numeric retention. Budget runs specify total online-token ceilings and realized consumption, with $\theta = 0.25$ as the default. Appendix B describes the evaluation protocol, Appendix C formalizes the control conditions, and Appendix E presents repeated evaluations and cross-backbone summaries.

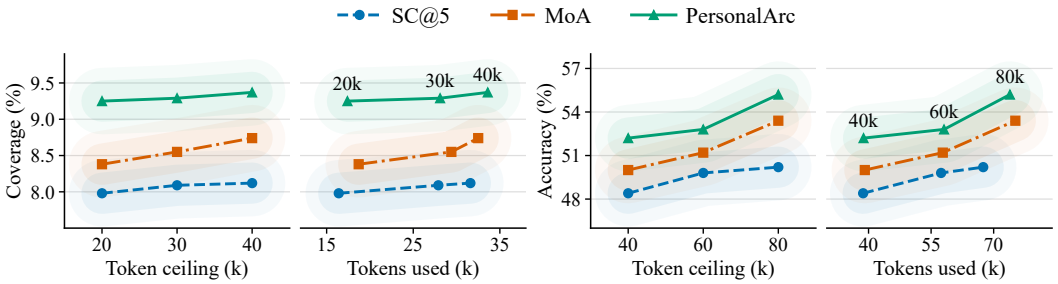
### 4.2 Main Results (RQ1)

**Gains across generating backbones.** `PersonalArc` leads LongMemEval-S accuracy and F1 across all three backbones (Table 1). It exceeds Mutation SFT, the strongest alternative, by 2.00–2.20 accuracy points and improves F1 over Base by 6.60–7.00 points. Qwen3.6-27B accuracy rises from 57.20% for Base to 61.20% with Mutation SFT and 63.40% with the full configuration. On Life, strict task success rises from 11.36% to 16.30% with Qwen3.6-27B, from 8.64% to 11.11% with Gemma-3-12B, and from 10.12% to 11.60% with Mistral-Small-24B (Table 10).

**Accuracy and graded quality on HippoCamp.** For Qwen3.6-27B answers, the recorded correct-task counts are 325/581 for `PersonalArc` versus Base's 256 under the primary judge, and 339 versus 299 under the second judge (Figure 2c). Its mean Scores are 3.442 and 3.690 (Table 5). The difference from Base persists across these evaluations; Appendix B.1 distinguishes reported counts from arithmetic reconstructions.

### 4.3 Component Analysis (RQ2)

**Training and fusion configurations.** Table 2 compares original-stage SFT, mutation-stage SFT, fusion, and selection configurations using the original Life score summaries. Figure 2b shows Mutation SFT and `PersonalArc` differences from Base across backbones. The strict-task and rubric-micro results in Table 10 complement these component diagnostics. Appendix C.1 explains selection for a fixed candidate pair.




(a) Life: ceiling. (b) Life: consumption. (c) LME-S: ceiling. (d) LME-S: consumption.

Figure 3: **Quality and token use under common budgets.** Qwen3.5-9B on Life ($n = 405$, a,b) and LongMemEval-S ($n = 500$, c,d). Life quality uses the originally reported Coverage score; LongMemEval-S quality is accuracy. Tokens are online-call totals in thousands/task; SC@5 uses five samples. Labels in (b,d) mark PersonalArc ceilings; halos are visual guides.

**Sensitivity to numeric retention.** Figure 2a shows the Life threshold sweep under the original configuration score. At $\theta = 0$, every valid candidate passes numeric retention while adoption still requires agreement in both comparison orders. Fully ungated fusion adopts every valid candidate. The sweep characterizes sensitivity to the retention rule at the reported operating point.

### 4.4 Inference Efficiency (RQ3)

**Quality and realized cost.** The default Life cost is 24.5k tokens and 44 seconds per task for `PersonalArc`, compared with Base's 15.2k and 23 seconds and MoA's 27.1k and 57 seconds (Table 3). On LongMemEval-S (Figures 3c and 3d), `PersonalArc` exceeds MoA by 1.6–2.2 accuracy points, using fewer tokens at the 40k and 80k ceilings and 0.3k more at 60k. Figures 3a and 3b show Life quality under the same budget analysis; Appendix D.1 records all operating points.

### 4.5 Case Study: Evidence-Aligned Fusion

**Aligning receipt evidence.** A HippoCamp task asks whether FairPrice raised prices (Yang et al., 2026). Three receipts show Coca-Cola Classic (1.5 L) at SGD 2.85 in March 2023 and March 2024, and SGD 3.10 in March 2025 (Figure 4). Matching item and package establishes a SGD 0.25 increase; basket totals also depend on quantities and other purchases.

**What the harness controls.** The schematic binds each amount to its item, size, date, and source, then illustrates comparison and numeric retention before routing to Fusion or Base. The case provenance in Appendix B lists the receipts and explains the evidence chain.

**Why the distinction matters.** Swapping dates changes the price trend while retaining every number. Evidence alignment establishes the factual relation; the harness uses comparison and retention signals to choose the final answer.

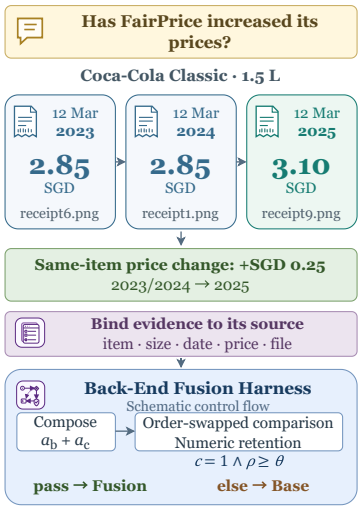



Figure 4: **A receipt-based HippoCamp case.** Receipts establish the item-level increase; the lower panel shows schematic fusion control.

## 5 Conclusion

`PersonalArc` connects audited stage supervision and answer adoption through Build–Learn–Fuse. Restricted task harnesses define stage-visible evidence and artifacts; the fusion harness combines Base and student answers through comparison, numeric retention, and fallback. Across four benchmarks, experiments measure quality and cost after benchmark-informed adaptation. The full system improves LongMemEval-S accuracy and F1 across three backbones and leads aggregation baselines at every tested token ceiling, linking reusable supervision to controlled deployment.

### AI USE STATEMENT

For manuscript preparation, we used language models only to polish grammar and wording in author-written prose and to help locate a subset of the literature discussed in Section 2. AI coding assistants helped implement the bar plots in Figure 2 and the line plots in Figure 3, and convert author-provided Excel tables into the LaTeX format used in Tables 2, 3, and 5. These manuscript-preparation tools were not used at any stage to generate research ideas, define research objectives, run or interpret experiments, or produce any reported numerical results. Model calls for synthetic-task construction, auditing, teacher execution, fusion, and evaluation are described in Section 3 and Appendix B. The authors are responsible for the manuscript and its research claims.

### ETHICS STATEMENT

Task execution uses question-specific file environments. Offline synthesis sends benchmark questions, personal context, reference answers, and rubrics to a model API. The resulting supervision is used for benchmark-informed adaptation, as described in Section 4.1.

### REPRODUCIBILITY STATEMENT

To support reproduction, Section 3 and Algorithm 1 specify execution contracts, trace admission, distillation, and answer selection. Section 4.1 describes the benchmarks, models, comparison systems, and evaluation setup; Appendix B details the data-construction scope and runtime settings. Appendix B.1 defines score aggregation, and Appendix B.2 records corpus sizes and adapter-training settings. Appendix C formalizes the selection controls, while Appendices D and D.1 explain cost accounting, token ceilings, and realized consumption. Appendix E reports repeated evaluations and cross-backbone comparisons. Our implementation is available in the anonymous code repository.

## REFERENCES

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. Hugging Face model card for DeepSeek-V4-Flash, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash. Online documentation accessed on September 15, 2026.

Shihan Dou, Yujiong Shen, Chenhao Huang, Junjie Ye, Jiayi Chen, Junzhe Wang, Qianyu He, Shichun Liu, Changze Lv, Jiahang Lin, Jiazheng Zhang, Ming Zhang, Shaofan Liu, Tao Ji, Zhangyue Yin, Cheng Zhang, Huaibing Xie, Jianglu Hu, Jingcheng Deng, Lincheng Li, Minda Hu, Shaolei Wang, Syrus Zhao, Weichao Wang, Yan Lei, Yang Liu, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Ziliang Zhao, Pluto Zhou, Tao Gui, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, Di Wang, and Shunyu Yao. CL-bench Life: Can language models learn from real-life context? *arXiv preprint arXiv:2604.27043*, 2026a. Preprint available at https://arxiv.org/abs/2604.27043.

Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, Tao Gui, Zuxuan Wu, Xipeng Qiu, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, Di Wang, and Shunyu Yao. CL-bench: A benchmark for context learning. *arXiv preprint arXiv:2602.03587*, 2026b. Preprint available at https://arxiv.org/abs/2602.03587.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. Preprint available at https://arxiv.org/abs/2503.19786.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fef126561bbf9d4467dbb8d27334b8fe-Abstract-Conference.html.

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. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wei-Chieh Huang, Weizhi Zhang, Yuchen Wu, Yankai Chen, Eric Hanchen Jiang, Wooseong Yang, Yiwei Yang, Henry Peng Zou, Hanrong Zhang, Ying Nian Wu, Haolun Wu, Kai-Wei Chang, Philip S. Yu, Xue Liu, and Aylin Caliskan. Harness the memory: A holistic evaluation of memory substrates in memory agents. *arXiv preprint arXiv:2608.15008*, 2026a. Preprint available at https://arxiv.org/abs/2608.15008.

Yue Huang, Wenjie Wang, Han Bao, Yuchen Ma, Xiaonan Luo, Yi Nian, Haomin Zhuang, Zheyuan Liu, Yue Zhao, and Xiangliang Zhang. MemoHarness: Agent harnesses that learn from experience. *arXiv preprint arXiv:2607.14159*, 2026b. Preprint available at https://arxiv.org/abs/2607.14159.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.792. Proceedings available at https://aclanthology.org/2023.acl-long.792/.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2603.28052.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026. Preprint available at https://arxiv.org/abs/2604.25850.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*, 2025. Preprint available at https://arxiv.org/abs/2502.11528.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. AutoHarness: Improving LLM agents by automatically synthesizing a code harness. In *ICLR 2026 Workshop on AI with Recursive Self-Improvement*, 2026. doi: 10.48550/arXiv.2603.03329. URL https://openreview.net/forum?id=g9rEYVNn5T.

Jinghao Luo, Yuchen Tian, Chuxue Cao, Ziyang Luo, Hongzhan Lin, Kaixin Li, Chuyi Kong, Ruichao Yang, and Jing Ma. From storage to experience: A survey on the evolution of LLM agent memory mechanisms. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41622–41652. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2069. URL https://aclanthology.org/2026.findings-acl.2069/.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Mistral AI. Mistral Small 3. https://mistral.ai/news/mistral-small-3, 2025. Online documentation accessed on September 13, 2026.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. AgentInstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*, 2024. Preprint available at https://arxiv.org/abs/2407.03502.

OpenThoughts-Agent Team. OpenThinkerAgent-32B. Hugging Face model card, 2026. URL https://huggingface.co/open-thoughts/OpenThinkerAgent-32B. Online documentation accessed on September 15, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. Preprint available at https://arxiv.org/abs/2310.08560.

Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-9B. Online documentation accessed on September 15, 2026.

Qwen Team. Qwen3.6-27B. Hugging Face model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.6-27B. Online documentation accessed on September 15, 2026.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=h0ZfDIrj7T.

Kun Wang, Guibin Zhang, ManKit Ye, Xinyu Deng, Dongxia Wang, Xiaobin Hu, Jinyang Guo, Yang Liu, and Yufei Guo. MAS$^2$: Self-generative, self-configuring, self-rectifying multi-agent systems. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qumy27hMDY.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Yue Xu, Qian Chen, Zizhan Ma, Dongrui Liu, Wenxuan Wang, Xiting Wang, Li Xiong, and Wenjie Wang. Toward personalized LLM-powered agents: Foundations, evaluation, and future directions. *arXiv preprint arXiv:2602.22680*, 2026. Preprint available at https://arxiv.org/abs/2602.22680.

Zhe Yang, Shulin Tian, Kairui Hu, Shuai Liu, Hoang-Nhat Nguyen, Yichi Zhang, Zujin Guo, Mengying Yu, Zinan Zhang, Jingkang Yang, Chen Change Loy, and Ziwei Liu. HippoCamp: Benchmarking contextual agents on personal computers. In *Computer Vision – ECCV 2026*, volume 17071 of *Lecture Notes in Computer Science*, pp. 98–118. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37016-7_6. URL https://doi.org/10.1007/978-3-032-37016-7_6.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. $\tau$-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Lingzhi Yuan, Chenghao Deng, Fangxu Yu, Souradip Chakraborty, Mohammad Rostami, and Furong Huang. FlowBank: Query-adaptive agentic workflows optimization through precompute-and-reuse. *arXiv preprint arXiv:2606.11290*, 2026. Preprint available at https://arxiv.org/abs/2606.11290.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.181. URL https://aclanthology.org/2024.findings-acl.181/.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-0439. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/136a45cd9b841bf785625709a19c6508-Abstract-Conference.html.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 75834–75852. PMLR, 2025b. URL https://proceedings.mlr.press/v267/zhang25bi.html.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-Harness: Harnesses that improve themselves. *arXiv preprint arXiv:2606.09498*, 2026a. Preprint available at https://arxiv.org/abs/2606.09498.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. MemSkill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*, 2026b. Preprint available at https://arxiv.org/abs/2602.02474.

Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, Xiaoman Pan, Lian Xiong, Jingguo Liu, Philip S. Yu, and Xian Li. PersonaAgent: Bridging memory and action for personalized LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26421–26439. Association for Computational Linguistics, 2026c. doi: 10.18653/v1/2026.findings-acl.1315. URL https://aclanthology.org/2026.findings-acl.1315/.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen K. Ahmed, and Yu Wang. Personalization of large language models: A survey. *Transactions on Machine Learning Research*, 2025c. URL https://openreview.net/forum?id=tf6A9EYMo6.

## A Additional Result Comparisons

Tables 4, 10, and 5 report full-set comparisons. The second HippoCamp judge scores the same Qwen3.6-27B answers, fixing generation while changing the evaluator. Aggregate differences quantify net changes; Eq. (13) specifies the corresponding task-level analysis.

**Paired score differences and task outcomes.** For task metric $m$, define $\delta_t = m(a_t^\star) - m(a_{b,t})$. A pass over $\mathcal{Q}$ returns the mean paired difference and the sign counts indexed by $s \in \{+, -, 0\}$:

$$(\Delta,\, n_s)^\top = \sum_{t \in \mathcal{Q}} \big(|\mathcal{Q}|^{-1}\delta_t,\; \mathbb{1}[\operatorname{sgn}\delta_t = s]\big)^\top. \tag{13}$$

For Life Yes Rate, $m$ is the indicator that every rubric item passes; for HippoCamp, it is binary correctness. The counts $n_+$, $n_-$, and $n_0$ partition tasks into improved, degraded, and tied outcomes under the same Base answers and judge. Life Coverage compares the sums of passed rubric items over the same total rubric count in Eq. (14).

Table 4: **Life configuration differences from Base Search.** Qwen3.5-9B generation/judgment on 405 tasks. Orig. Yes/Cov. repeat the original configuration summaries in Table 2; paired strict Yes and micro Coverage appear in Table 10.

| System | Orig. Yes (%) | Orig. Cov. (%) | ΔYes | ΔCov. |
|---|---|---|---|---|
| Base Search | 7.69 | 7.91 | 0.00 | 0.00 |
| Mutation SFT | 8.11 | 8.16 | +0.42 | +0.25 |
| `PersonalArc` | 9.29 | 9.52 | +1.60 | +1.61 |

Table 5: **Judge sensitivity on HippoCamp.** Both judges score the same Qwen3.6-27B answers on all 581 tasks. Accuracy is computed over all tasks; † marks reconstructions from reported percentages. Score is the mean rating out of five. Bold marks the best unflagged accuracy and mean Score within each judge block.

| System | Acc. (%) | Score (/5) |
|---|---|---|
| **Qwen3.5-9B judge** | | |
| Base Search | 44.06 | 2.611 |
| Matched Control SFT | 52.15† | 3.000 |
| Mutation SFT | 51.81† | 2.944 |
| Rubric-blind fusion | 54.22 | 2.732 |
| `PersonalArc` | **55.94** | **3.442** |
| **OpenThinkerAgent-32B judge** | | |
| Base Search | 51.46 | 3.213 |
| Matched Control SFT | 55.42 | 3.195 |
| Mutation SFT | 56.28 | 3.321 |
| Rubric-blind fusion | 55.77 | 3.452 |
| `PersonalArc` | **58.35** | **3.690** |

**Accuracy under a change of judge.** The reported gain over Base is 69 correct tasks under Qwen3.5-9B (325 versus 256) and 40 under OpenThinkerAgent-32B (339 versus 299). The secondary-judge margin over Mutation SFT is twelve correct tasks. Primary-judge Control and Mutation accuracies use reconstructed counts of 303 and 301, marked with †. With generated answers held fixed, the recorded gain over Base varies in magnitude across the two judges.

**Graded-score rankings.** The full system leads mean Score under both judges. Under Qwen3.5-9B, it reaches 3.442, followed by Matched Control SFT at 3.000 and Mutation SFT at 2.944. Under OpenThinkerAgent-32B, it reaches 3.690, ahead of fusion at 3.452. The full system's mean-score gain over Base is 0.831 under the primary judge and 0.477 under the second. Reporting accuracy and mean Score captures both the frequency of correct answers and differences in their graded quality.

## B Protocol and Reproducibility Details

**System identities.** Base Search names the tabulated baseline, $a_b$ denotes the inference fallback answer, and $\mathcal{H}_0$ denotes the default execution graph. The Life cost row "Mutation + repair" corresponds to terminal-repair SFT in Table 2. Budget runs use separate operating points from the default costs in Table 3.

**Admission rules.** The syntax verdict admits programs that parse under the restricted grammar, rejecting unknown stage types, arbitrary Python or shell calls, and free-form control flow. The structural verdict checks unique stage identifiers, forward and bounded routing, stage count, and the final-answer declaration. The dependency verdict checks types, schemas, tool scopes, artifact producers, and explicit evidence inputs for answer, verify, repair, and final stages. The policy verdict rejects inappropriate tool mixtures, missing retrieval on large workspaces, resource violations, and high-risk custom-stage designs. All four checks must pass before execution. Static admission establishes conformance to these declarations; execution success also depends on the available tools and the runtime environment in which the graph runs.

Table 6: Original evaluation sets also used as sources for synthetic-data construction. Tables 8 and 9 record retained training sources and synthesized tasks.

| Benchmark | Evaluation tasks | Scope |
|---|---|---|
| CL-bench Life | 405 | Complete release |
| HippoCamp | 581 | All profiles |
| $\tau$-bench retail | 115 | Complete retail evaluation set |
| LongMemEval-S | 500 | Complete S evaluation set |

**Construction and evaluation scope.** The API constructs synthetic training tasks using questions, personal files or histories, standard answers, and rubrics covering the full original evaluation sets in Table 6. The student therefore receives supervision derived from those evaluation references, and the results measure adaptation on the same sources. Per-question `bwrap` environments govern runtime file visibility; the source-to-synthetic-task mapping governs training exposure. Table 8 records retained sources and stage rows, while Table 9 counts synthetic tasks by operator.

**Threshold calibration and output accounting.** The default is $\theta = 0.25$. Figure 2a shows the measured quality across threshold settings on the Life evaluation set.

Table 7: Model roles and recorded runtime settings.

| Role | Recorded model / setting |
|---|---|
| Main generation | Qwen3.6-27B; Gemma-3-12B; Mistral-Small-24B |
| Life analysis | Qwen3.5-9B |
| Mutation / teacher | DeepSeek-V4-Flash |
| Primary judge | Qwen3.5-9B |
| Secondary judge | OpenThinkerAgent-32B |
| Fusion / comparator | Qwen3.5-9B (Life); Qwen3.6-27B (HippoCamp) |
| Numeric gate | $\theta$=0.25; empty Base number set: $\rho$=1 |

Table 7 identifies the model roles and default numeric gate used in the reported comparisons.

**Fusion evidence and retrieval provenance.** In the HippoCamp implementation, Fusion inherits Base retrieval metadata; its retrieval F1 therefore describes Base retrieval. This retrieval measure is separate from LongMemEval-S answer F1.

**Case-study provenance.** Figure 4 uses task 232 from HippoCamp's released Bei full set, whose question is "Has FairPrice increased its prices?" The source files are `Receipts/receipt6.png`, `receipt1.png`, and `receipt9.png`. Their dates are 12 March 2023, 2024, and 2025, respectively; the Coca-Cola Classic 1.5 L line item costs SGD 2.85, 2.85, and 3.10. We checked these fields in the three raw receipt images against the official annotation. The figure records the item-level difference, SGD $3.10 - 2.85 = 0.25$, and reconstructs the evidence chain for qualitative analysis. The

accompanying figure data file records the immutable dataset revision, source URLs, and checksums. These source records support the comparison; runtime composition still accesses only the authorized workspace and paired answers declared in Section 3.4.

### B.1 Score aggregation and accounting

**Life rubric aggregation.** For task $t$, let $m_t$ be its number of rubric items and $y_{tj} \in \{0, 1\}$ indicate whether item $j$ passes. We define the metrics over the complete set $\mathcal{Q}$ as

$$\text{Yes} = \frac{100}{|\mathcal{Q}|} \sum_{t \in \mathcal{Q}} \prod_{j=1}^{m_t} y_{tj}, \qquad \text{Cov.} = 100 \frac{\sum_{t \in \mathcal{Q}} \sum_{j=1}^{m_t} y_{tj}}{\sum_{t \in \mathcal{Q}} m_t}. \tag{14}$$

Yes Rate is strict task accuracy: every rubric item must pass, consistent with the benchmark's task-solving criterion (Dou et al., 2026a). Coverage is micro-averaged rubric satisfaction. For example, passing 2/2 items on one task and 4/8 on another gives 50% Yes Rate and 60% Coverage. Their difference reflects both partial success and variation in rubric counts per task.

**Life reporting streams.** The all-system tables and diagnostic figures retain the originally reported Life Yes/Coverage summaries. In that reporting convention, Yes summarized passed rubric items and Coverage averaged the within-task rubric fractions. Table 10 reports the Base–`PersonalArc` pairs from the server-side aggregation using strict Yes Rate and rubric-micro Coverage in Eq. (14). We display the two reporting conventions separately.

The abstract's 27.0% and 18.2% Life summaries follow the originally reported all-system Yes/Coverage columns in Table 1; these labels refer to that original reporting convention. Each summary averages the relative improvement $100(p_{\text{full}} - p_{\text{Base}})/p_{\text{Base}}$ across the three main backbones using the corresponding displayed percentages $p$. The 6.77-point F1 gain averages absolute score differences in the LongMemEval-S columns. Table 10 gives the corresponding backbone-level comparisons under the explicit strict Yes and rubric-micro Coverage definitions.

**Binary accuracy and trial accounting.** Accuracy uses one binary judgment per task: $100k/n$ for integer correct count $k$. Twenty-three main-table cells use nearest-count reconstructions, $\hat{k} = \text{round}(np/100)$ with half-integers rounded upward. The † marker identifies these arithmetic reconstructions and their derived differences. HippoCamp primary-judge Base, fusion, and full-system counts are 256, 315, and 325; reconstructed Control and Mutation counts are 303 and 301. Secondary-judge counts are 299, 322, 327, 324, and 339 in Table 5 order. Mistral Base on LongMemEval-S uses 266/500 (53.20%). Rankings use unflagged accuracy values. Retail pass$^k$ aggregates cover the 115-task retail set; Base and student run in isolated environments before fusion.

### B.2 Training and corpus accounting

Table 8: Training-corpus records. Sources count training-data origins, and rows count supervision instances. Life V4 and V7 use different sources and corpus sizes. Life splits count constraint-changing/paraphrase rows; HippoCamp records original traces versus two mutation operators. Missing loss-token counts are shown as dashes. Table 9 separately counts synthetic tasks by mutation operator and benchmark.

| Corpus | Training sources | Rows | Supervision / split | Loss tokens |
|---|---|---|---|---|
| Life V4 | 229 | 1,000 | 891/109 | – |
| Life V7 terminal repair | 189 | 450 | 368/82 | – |
| HippoCamp original Control | 6 | 29 | Original traces | 4,513 |
| HippoCamp mutation V4 | 6 | 29 | Two mutation operators | 4,469 |

The recorded HippoCamp loss-token difference is $44/4513 \approx 0.975\%$, and both arms list four optimizer updates. Life repair uses rank 8, alpha 16, dropout 0.05, q/v/o targets, 40 updates, learning rate $3 \times 10^{-6}$, and seed 42. HippoCamp lists rank 16, alpha 32, dropout 0.05, q/k/v/o and gate/up/down targets, learning rate $10^{-4}$, one epoch, and seed 42. Its configured schedule has ten warmup steps and four executed updates. Training uses synthetic data derived from the original evaluation sources (Section 3.3); source identifiers connect each retained stage row to its construction task.

Table 9: Synthesized training tasks recorded for each benchmark. Operator 1 preserves the answer through paraphrase; Operator 2 changes constraints and recomputes answers and evidence. Each total counts the training variants produced by the two operators for that benchmark.

| Benchmark | Operator 1 | Operator 2 | Total |
|---|---|---|---|
| CL-bench Life | 357 | 316 | 673 |
| HippoCamp | 647 | 502 | 1,149 |
| $\tau$-bench retail | 89 | 103 | 192 |
| LongMemEval-S | 513 | 332 | 845 |
| Total | 1,606 | 1,253 | 2,859 |

### B.3 Strict Life task accuracy and rubric coverage

Table 10 compares Base Search and `PersonalArc` under the strict task and rubric-micro definitions in Eq. (14). Each pair covers the complete 405-task Life set with Qwen3.5-9B judgment. The backbone-level gains in the table use absolute percentage-point differences.

Table 10: **Paired Life results by generating backbone.** Strict Yes Rate requires every task rubric to pass; Coverage counts passed rubric items over all items. $\Delta$ is `PersonalArc` minus Base Search in percentage points.

| Backbone | Base Yes | Ours Yes | $\Delta$Yes | Base Cov. | Ours Cov. | $\Delta$Cov. |
|---|---|---|---|---|---|---|
| Qwen3.6-27B | 11.36 | 16.30 | +4.94 | 11.86 | 14.13 | +2.27 |
| Gemma-3-12B | 8.64 | 11.11 | +2.47 | 9.34 | 10.37 | +1.03 |
| Mistral-Small-24B | 10.12 | 11.60 | +1.48 | 10.03 | 13.03 | +3.00 |
| Qwen3.5-9B | 7.65 | 9.38 | +1.73 | 7.91 | 9.52 | +1.61 |

## C Supplemental Controls

The configuration comparisons vary learning and answer-generation components under the execution policy in Section 4.1. Original-stage, mutation-stage, and terminal-repair SFT vary supervision choices; rubric-blind fusion, two-way selection, and the full system compare answer-generation and adoption choices in Table 2. A matched comparison fixes the training and inference budgets, model, judge, and source provenance as formalized in Eq. (9).

### C.1 Fixed-candidate gate analysis

For fixed valid candidates, $\theta = 0$ always passes numeric retention because $\rho \in [0, 1]$; final adoption still requires consensus $c = 1$. Increasing the numeric threshold accepts a subset of those candidates and returns Base for the remainder. These properties follow from Eq. (12); Figure 2a reports the Life response under the original Coverage score.

## D Cost and Admission Diagnostics

Table 11: Compile-time admission and fallback summaries on Life ($n = 405$), using Qwen3.5-9B.

| Configuration | First admission (%) | Default graph (%) |
|---|---|---|
| Matched Control SFT | 61.7 | 12.5 |
| Mutation + repair SFT | 68.0 | 9.4 |
| `PersonalArc` | 72.7 | 6.2 |
| Fusion harness | 84.4 | 3.1 |

Table 11 summarizes admission and fallback rates for generated harnesses. Before execution, the compiler applies the syntax, structural, dependency, and policy checks described in the admission rules in Appendix B.

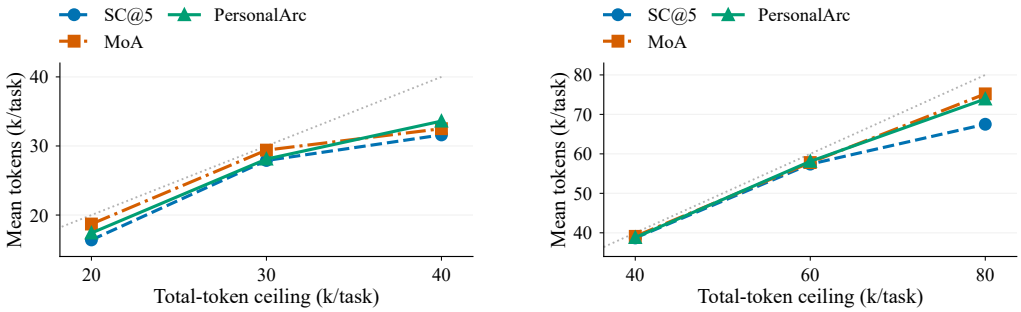
Offline synthesis, auditing, teacher execution, and adapter training are separate costs. Amortized cost uses an explicit number of served tasks. Online totals count candidate and comparator calls for accepted answers and every fallback decision made during execution.

### D.1 BUDGET AND REALIZED CONSUMPTION

Table 12 gives the complete measurements behind Figure 3. Life ceilings are 20k, 30k, and 40k tokens per task; LongMemEval-S ceilings are 40k, 60k, and 80k. Each setting covers the full benchmark with the same Qwen3.5-9B generation and judge. SC@5 retains five samples; the budget includes candidate generation, aggregation, selection, and recovery. The default operating points in Table 3 are reported separately from these budget-controlled runs.

Table 12: **Quality and tokens at each ceiling.** Qwen3.5-9B generation/judgment on Life (405 tasks) and LongMemEval-S (500 tasks). Life quality uses the originally reported Coverage score; LongMemEval-S reports accuracy. Cap and mean tokens are thousands per task. Bold marks the highest quality at each ceiling.

| | SC@5 | | MoA | | `PersonalArc` | |
|---|---|---|---|---|---|---|
| Cap (k) | Quality ↑ | Tokens (k) | Quality ↑ | Tokens (k) | Quality ↑ | Tokens (k) |
| **Life ($n = 405$): original reported Coverage** | | | | | | |
| 20 | 7.98 | 16.4 | 8.38 | 18.7 | **9.25** | 17.4 |
| 30 | 8.09 | 27.9 | 8.55 | 29.4 | **9.29** | 28.1 |
| 40 | 8.12 | 31.6 | 8.74 | 32.5 | **9.37** | 33.6 |
| **LongMemEval-S ($n = 500$): Accuracy** | | | | | | |
| 40 | 48.40 | 38.7 | 50.00 | 39.1 | **52.20** | 38.8 |
| 60 | 49.80 | 57.4 | 51.20 | 57.8 | **52.80** | 58.1 |
| 80 | 50.20 | 67.5 | 53.40 | 75.2 | **55.20** | 73.9 |




(a) Life, $n = 405$. (b) LongMemEval-S, $n = 500$.

Figure 5: **Realized consumption under common ceilings.** Full-set Qwen3.5-9B runs match Table 12: Life has 405 tasks and LongMemEval-S has 500. Both axes measure thousands of tokens per task, with gray diagonals indicating full use of the ceiling. Lines connect the three measured settings within each panel.

**Operating-point comparisons.** Figures 3b and 3d place quality against realized token consumption. On LongMemEval-S, the highest-ceiling PersonalArc point gains 1.8 accuracy points over MoA with 1.3k fewer tokens. Relative to SC@5 at that ceiling, it gains 5.0 points with 6.4k additional tokens. The cost of improved quality depends on the comparison method.

**Budget use varies by method and workload.** All reported mean token counts lie below their respective ceilings (Table 12). On LongMemEval-S, `PersonalArc` uses 38.8k, 58.1k, and 73.9k tokens as the ceiling increases (Figure 5). At the largest ceiling, SC@5 uses 67.5k and MoA uses 75.2k. A common ceiling thus bounds each method while allowing different realized token consumption.

**Returns from increasing the budget.** Doubling the LongMemEval-S ceiling from 40k to 80k adds 3.0 accuracy points for `PersonalArc` and 3.4 for MoA. Their memory-task accuracy improves with the larger allowance. Across the Life ceilings, `PersonalArc`'s original reported Coverage score rises from 9.25 to 9.37 points (Table 12).

## E Robustness and Cross-Benchmark Analysis

**Variation across repeated evaluations.** Table 13 summarizes three Life evaluations using the original reported Coverage score. The first full-system result also appears in Table 2 and Figure 2a; Base evaluations and budget sweeps use separate operating points.

Table 13: **Three Life evaluations.** Each repetition covers 405 tasks with Qwen3.5-9B generation/judgment. Values use the original reported Coverage score; the final column gives mean ± sample standard deviation across evaluations.

| System | Run 1 | Run 2 | Run 3 | Mean ± SD |
|---|---|---|---|---|
| Base Search | 7.70 | 7.64 | 7.73 | 7.690 ± 0.046 |
| Mutation SFT | 8.15 | 8.32 | 8.08 | 8.183 ± 0.123 |
| `PersonalArc` | 9.52 | 9.41 | 9.37 | 9.433 ± 0.078 |

Table 14: Life configurations across four generating backbones on 405 tasks with Qwen3.5-9B judgment. Values use the original reported Coverage scores from the configuration and main comparisons.

| System | Qwen3.5-9B | Qwen3.6-27B | Gemma-3-12B | Mistral-Small-24B |
|---|---|---|---|---|
| Base Search | 7.91 | 11.86 | 9.34 | 10.12 |
| Mutation SFT | 8.16 | 12.88 | 10.02 | 11.79 |
| `PersonalArc` | 9.52 | 14.13 | 9.97 | 13.03 |

**Recorded Life configurations across backbones.** Table 14 holds the complete Life set and judge fixed while varying the generating backbone. These original reported Coverage values underlie Figure 2b; Table 10 separately reports strict task accuracy and rubric-micro coverage for the Base–`PersonalArc` pairs.

**Benchmark compatibility.** LongMemEval-S reports QA accuracy and auxiliary answer F1. Retail results cover 115 tasks with pass$^1$ and pass$^4$; Base and student execute in isolated environments before final fusion. Table 15 summarizes the reported metrics across benchmarks.

Table 15: Benchmark sizes and reported metric definitions.

| Benchmark | Tasks | Metrics |
|---|---|---|
| CL-bench Life | 405 | Paired Yes: all rubrics pass per task; paired Coverage: micro rubric satisfaction; all-system: original reported Yes/Coverage summaries |
| HippoCamp | 581 | Accuracy at score $\geq 3/5$; mean Score |
| $\tau$-bench retail | 115 | pass$^1$ and pass$^4$ |
| LongMemEval-S | 500 | QA accuracy; auxiliary answer F1 |

# F PIPELINE PSEUDOCODE

**Algorithm 1:** Contract-Controlled Learning and Fusion Pipeline of `PersonalArc`

**Input:** *Offline*: benchmark source tasks $\mathcal{Q}_{\text{src}}$ with references, operators $\mathcal{O}$, pack $\mathcal{K}$, teacher $\theta_{\text{T}}$, base $\theta_0$, budget $R$, gate $\eta$. *Inference*: task $\bar{t}$, Base $f_b$, fusion $\phi$, comparator $\psi$, numeric-retention threshold $\theta$

**Output:** Learned agent $f_c$; answer $a_{\bar{t}}^{\star}$, equal to $a_b$ if the gate rejects

$\mathcal{Z} \leftarrow \varnothing$ // Initialize the audited trace buffer

/* Part 1: Audited Trace Construction */

**for** $(t, o) \in \mathcal{Q}_{\text{src}} \times \mathcal{O}$ **do**

  $t' \leftarrow \mu_o(t)$ // Ground request and contract in the workspace

  **if** $\prod_{j=1}^{5} z_j(t') = 0$ **then continue**

  $\mathcal{H}^{\star} \leftarrow \mathcal{H}_{i_{t'}^{\star}}$ by Eqs. (2)–(3) // Admit the first valid harness

  $\zeta^{g} \leftarrow \mathsf{Exec}_{\theta_{\text{T}}}(\mathcal{H}^{\star} \mid x_{t'}, W_{t'})$ // Teacher executes the admitted harness

  $\zeta^{\star} \leftarrow \mathcal{R}_{\mathcal{H}_0}^{\varpi, Q_{\text{post}}}(\zeta^{g})$ // Bounded recovery repairs the trace

  **if** $V_{\text{keep}}(t', \zeta^{\star}) = 1$ **then** $\mathsf{Append}_{\mathcal{Z}}(t', \zeta^{\star})$

/* Part 2: Stage-Level Distillation */

$\mathcal{D} \leftarrow \mathsf{StagePairs}(\mathcal{Z})$ by Eq. (7) // Project traces to stage pairs

$\Delta\theta^{\star} \leftarrow \mathsf{SFT}(\theta_0, \mathcal{D})$ by Eq. (8) // Low-rank update on stage pairs

$\theta_c \leftarrow \theta_0 + \Delta\theta^{\star}$, $f_c \leftarrow f(\cdot\,; \theta_c)$ // Learned agent for inference

/* Part 3: Back-End Fusion Harness */

$a_b \leftarrow f_b(\bar{t})$, $a_c \leftarrow f_c(\bar{t})$ // Paired execution on one task

$a_f \leftarrow \phi(\bar{t}; [a_b; a_c])$ // Compose from declared runtime inputs

**if** $a_f$ *is invalid* **then return** $(f_c, a_b)$

$\mathbf{g} \leftarrow \mathbf{g}(a_b, a_f)$ by Eq. (11) // Consensus and retention signals

$a_{\bar{t}}^{\star} \leftarrow \Pi_{\mathcal{K}[\mathbf{g} \succeq (1,\theta)^{\top}]}(a_b, a_f)$ // Admit Fusion or fall back exactly

**return** $(f_c, a_{\bar{t}}^{\star})$