VIEWCRAFT: EVOLVING WHAT MULTIMODAL AGENTS SEE AND HOW THEY READ IT

Anonymous authors
Paper under double-blind review

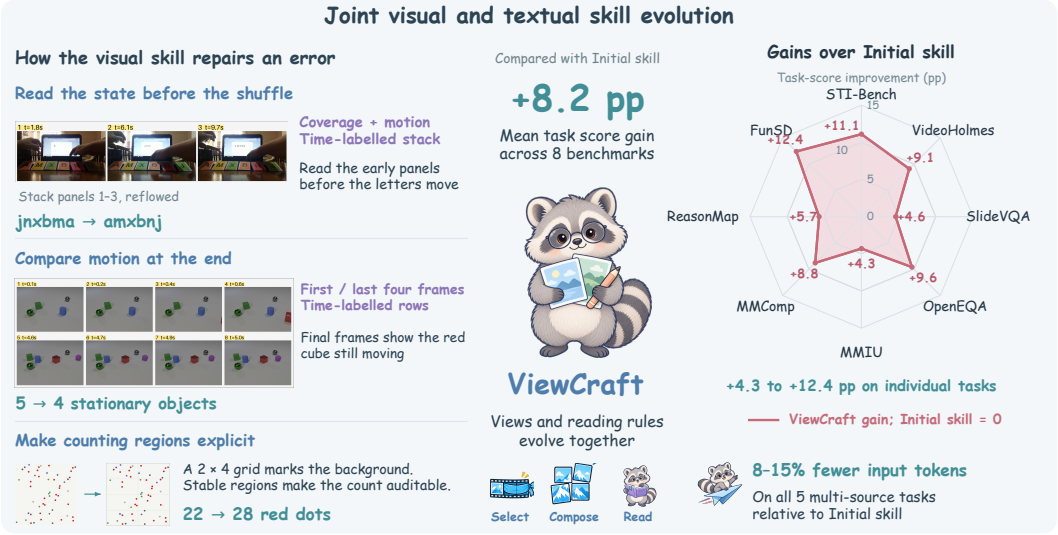

Figure 1: **Joint evolution of evidence interfaces and reading procedures.** Left: early frames recover pre-shuffle letter order, endpoint rows expose motion, and a background grid supports counting. Center/right: gains over Initial skill use a shared 0–15 point radial scale (mean 8.2; range 4.3–12.4 across eight tasks). On all five multi-source tasks, input tokens fall 8–15% from Initial skill.

ABSTRACT

Reusable skills improve multimodal agents without updating model weights, but their effectiveness depends on both the evidence presented and the procedure used to read it. We introduce ViewCraft, which jointly evolves an executable evidence interface and its reading procedure. The interface selects, transforms, and organizes observations; the reader interprets the resulting views. Failure traces guide revisions, replay and paired evaluation retain useful changes, and interface projection recovers visual edits from eligible rejected joint proposals. Across eight benchmarks, ViewCraft improves over direct execution by an average of 6.1 points in task score, with gains on every benchmark. The advantage persists at a matched input-token budget. On all five multi-source tasks, the learned interface uses 8–15% fewer input tokens than Initial skill. The learned skills also transfer across executors and outperform direct execution in every evaluated transfer, supporting evidence construction and reading as a coupled, reusable skill.

1 INTRODUCTION

Multimodal reasoning needs relevant, legible evidence and a procedure matched to its organization. A missed video segment or cropped-away field obstructs downstream reasoning; a reader written for the original input may also overlook timestamps or spatial relations introduced by a new view. We ask whether the interface exposes the needed evidence and whether its reader uses that organization.

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.

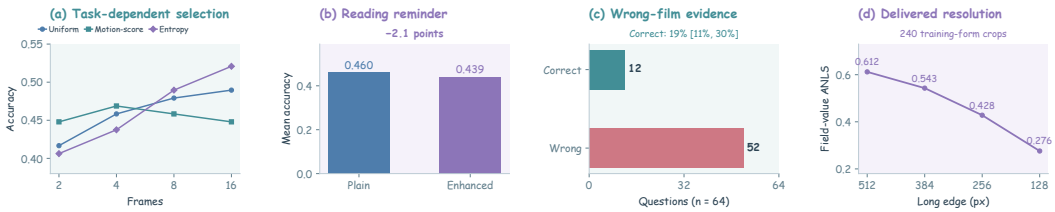

Figure 2: **Four probes of the evidence interface.** (a) Accuracy for three selectors at four frame budgets on 96 questions from 26 non-test films. (b) Mean accuracy across these 12 settings with plain and enhanced prompts. (c) With their frames replaced by another film, 12 of 64 previously correct questions are answered correctly (Wilson 95% CI $[11\%, 30\%]$). (d) Mean field-value ANLS on 240 training-form crops at four delivered resolutions. All probes are independent of Table 2's test sets.

An end-state counting question illustrates both requirements. Early and late frames establish which objects moved; timestamped rows expose that comparison. The reader must match objects across rows before counting those that remained stationary. Figure 1 shows temporal and counting repairs.

Skill evolution improves frozen executors through reusable procedures and workflows (Yang et al., 2024; Khattab et al., 2024; Fernando et al., 2024; Zhang et al., 2025; Hu et al., 2025; Yang et al., 2026). Multimodal systems add visual experience, prompts, and tools (Xu et al., 2026; Zhang et al., 2026; Jiang et al., 2026; Choi et al., 2026; Sun et al., 2026). We make the evidence interface a saved, executable mapping that selects, transforms, and composes observations for each question so that readers can replay and evaluate its revisions.

Evidence-interface probes. Neither a single selector nor more frames consistently wins: selector rankings reverse with frame budget, and Motion-score declines after its peak (Figure 2). The probes use 96 VideoHolmes questions from 26 non-test films and 240 field crops from FunSD training forms, with zero overlap with the main test sets.

A stronger reading reminder lowers mean accuracy on unchanged inputs. Reducing the resolution of oracle field crops also lowers field-value ANLS despite preserving the requested content. Selection and legibility therefore offer concrete targets for an editable evidence interface (Appendix A).

We introduce **ViewCraft**, which jointly evolves the evidence presented to a frozen executor and the procedure used to read it. The resulting skill preserves both how observations are selected and organized and how their relationships are interpreted.

Our contributions are threefold:

- We formulate multimodal skill evolution as jointly refining what agents see and how they read it. The reusable skill pairs an executable evidence interface with its reader, incorporating instance-conditioned evidence selection, transformation, and presentation.
- We introduce **ViewCraft**, which uses failure traces, visual replay, and paired development evaluation to evolve the interface and reader while freezing the executor. Interface projection retests visual edits from eligible rejected joint proposals with the incumbent reader to recover useful interface changes.
- Across eight benchmarks, ViewCraft improves over Direct by 6.1 points on average and remains 5.2 points ahead at a matched input-token budget. It also outperforms restricted and same-space search baselines, and the learned skills transfer across executors.

2 Related Work

2.1 Feedback-driven agent optimization

Automatic prompt optimizers generate, select, or evolve instructions from task feedback (Zhou et al., 2023; Yang et al., 2024; Fernando et al., 2024; Zeng et al., 2026a), while DSPy and GEPA optimize compound LM programs from execution trajectories (Khattab et al., 2024; Agrawal et al., 2026). Language agents also retain verbal feedback, distilled experience, or executable skills (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023), and systems such as ADAS, AFlow, and AgentSquare

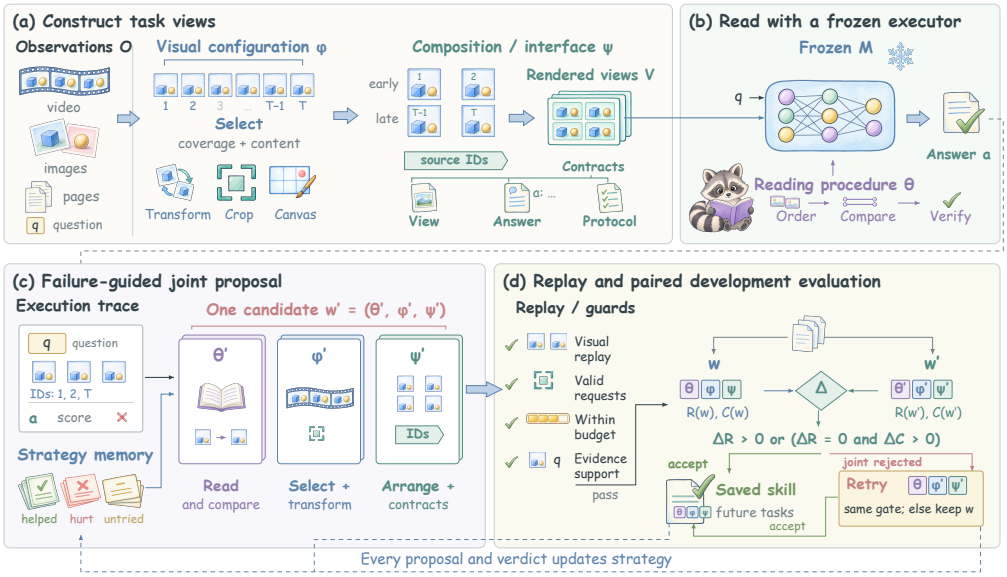

Figure 3: **Joint view construction and evolution.** (a) ϕ selects/transforms evidence; ψ arranges views and surface contracts. (b) Frozen M follows θ. (c) Traces guide proposals. (d) Replay and guards precede the paired gate. Projection retries the interface with incumbent θ when support is non-degraded and budget remains. Every verdict updates strategy.

Table 1: **What becomes part of an editable skill?** Columns track reuse, visual support, coupled editing, executable visual transformations, replay-screened paired evaluation, and interface-only retries. Checks denote explicit capabilities; a dash denotes a capability outside the described design. ViewCraft combines all six capabilities in its evolution loop.

Method family	Reusable state	Visual support	Reader–visual co-edit	Executable visual edits	Replay + paired gate	Interface retry
Textual skills	✓	–	–	–	–	–
Grounded multimodal skills	✓	✓	–	–	–	–
MPO (Choi et al., 2026)	✓	✓	✓	–	–	–
Dynamo (Sun et al., 2026)	✓	✓	✓	✓	–	–
VIEWCRAFT	✓	✓	✓	✓	✓	✓

search broader workflows or module configurations (Hu et al., 2025; Zhang et al., 2025; Shang et al., 2025; Yu et al., 2026). Most of this state is textual or structural. ViewCraft makes the evidence interface itself part of the persistent state and evolves it together with its reader.

2.2 MULTIMODAL SKILL EVOLUTION

Multimodal skills incorporate spatial priors and interleaved support (Xu et al., 2026), state cards and keyframes (Zhang et al., 2026), and grounded experience for planning and tool use (Jiang et al., 2026). Visual prompting and adaptive selection expose relevant regions and moments (Yang et al., 2023; Hu et al., 2024; Wang et al., 2024; 2025; Yu et al., 2025; Zeng et al., 2026b). Skill optimizers refine reusable procedures through execution feedback (Yang et al., 2024; Khattab et al., 2024; Yang et al., 2026), while WikiSkill accumulates knowledge across updates (Tang et al., 2026). The Multimodal Prompt Optimizer (MPO) jointly optimizes textual and non-textual prompts (Choi et al., 2026), and Dynamo co-evolves reasoning skills and visual tools (Sun et al., 2026). Rejecting a joint proposal, however, can also discard a useful visual revision.

VIEWCRAFT saves an executable evidence interface together with its reader (Table 1). Replay reconstructs the views, and interface projection retests the visual component of eligible rejected joint proposals with the incumbent reader. We evaluate this design against MPO and Dynamo adaptations in the same search space (Table 4).

Table 2: **Held-out task scores and component contributions.** Scores pool three runs under Table 6's metrics. The Mean column averages tasks equally; the final row subtracts Direct. Bold marks column bests. Tints distinguish baselines, restored-coordinate states, and joint evolution; lavender and teal headers group single-source and multi-source tasks.

Method	MMComp	ReasonMap	FunSD	STI-Bench	VideoHolmes	SlideVQA	OpenEQA	MMIU	Mean
Direct	0.834	0.858	0.306	0.410	0.408	0.802	0.512	0.921	0.631
Initial skill	0.786	0.841	0.288	0.385	0.381	0.812	0.478	0.908	0.610
Text-only	0.848	0.876	0.372	0.430	0.424	0.837	0.528	0.926	0.655
Interface-only	0.779	0.856	0.341	0.466	0.452	0.834	0.545	0.935	0.651
θ-only	0.852	0.867	0.352	0.416	0.408	0.833	0.508	0.921	0.645
(ϕ, ψ)-only	0.784	0.850	0.325	0.448	0.430	0.828	0.526	0.926	0.640
ViewCraft	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**
ViewCraft − Direct	+0.040	+0.040	+0.106	+0.086	+0.064	+0.056	+0.062	+0.030	+0.061

3 VIEWCRAFT

ViewCraft searches for a reusable reader–interface state while keeping the executor frozen. Paired evaluation retains revisions that improve the responses to development tasks (Figure 3).

3.1 SKILL STATE AND OBJECTIVE

For q, observations $O = (o_1, \ldots, o_n)$, and target y, skill $w = (\theta, \phi, \psi)$ selects evidence via ϕ, arranges views and formats via ψ, and gives the frozen executor a reading procedure θ.

The renderer $\mathcal{W}_{\phi,\psi}$ constructs image attachments from observations and question context. For a frozen executor M, ViewCraft targets

$$\max_{\phi,\psi,\theta} \mathbb{E}_{(O,q,y)\sim\mathcal{D}} \left[R(M(\mathcal{W}_{\phi,\psi}(O,q),\theta), y) \right], \qquad M \text{ frozen.} \tag{1}$$

ViewCraft approaches Equation 1 through proposals, development evaluation, and rollback. Text-only edits θ, Interface-only edits (ϕ, ψ), and joint search permits both.

Coordinate partition. Reasoning and decision rules belong to θ, including polarity, verdict tables, zoom decisions, and mark reading. Rendering, view metadata, and answer/action formats remain on the interface side. Appendix D details the partition used in terminal crossings.

3.2 COMPLEMENTARY SUCCESS FACTORS

Let E mean that required evidence is selected and legible. Guessing or prior knowledge can also yield some correct answers without that evidence, so

$$\begin{aligned} P(\text{correct}) &= \underbrace{P(E)}_{\alpha}\underbrace{P(\text{correct} \mid E)}_{\beta} + \underbrace{P(\neg E)P(\text{correct} \mid \neg E)}_{\gamma}, \\ R^*(w) &= \alpha(\phi,\psi)\,\beta(\theta \mid \phi,\psi) \le P(\text{correct}). \end{aligned} \tag{2}$$

Equation 2 separates evidence availability, successful reading, and correctness without required evidence. The wrong-film probe estimates correctness after evidence replacement as 0.19 with interval $[0.11, 0.30]$. The evidence-absence probability weights this conditional rate in γ.

Two controls test explanations based on evidence delivery or input volume alone. Evolved views with the initial reader recover $+2.3$ of the $+8.2$-point joint gain under a fixed trajectory (Appendix Table 17); token-matched Direct recovers $+0.9$ points (Appendix Table 23). The remaining advantage connects evidence construction to its reading procedure.

3.3 FAILURE-GUIDED PROPOSALS

Traces record the question, sent views with geometry and provenance, response, and score. They guide revisions such as selecting a missed event or cropping an unreadable field. Interface contracts describe the views, answer format, and available requests; the reader specifies how to use them.

Table 3: Search strategies with matched submissions: 250 per arm across 24 task–seed runs; each run has four rounds and 96 training rollouts. Conditions share executor and interface. Scores pool three runs; filtering determines candidate evaluations.

Search strategy	MMComp	RMap	FunSD	STI	VHolm	Slide	OEQA	MMIU	Mean
Direct (no search)	0.834	0.858	0.306	0.410	0.408	0.802	0.512	0.921	0.631
MPO-adapted (prompt space)	0.852	0.869	0.368	0.437	0.428	0.834	0.524	0.922	0.654
Dynamo-style (fixed tools)	0.841	0.865	0.380	0.440	0.426	0.840	0.534	0.925	0.656
Independent, then combine	0.857	0.878	0.378	0.468	0.452	0.842	0.551	0.933	0.670
Staged (visual, then θ)	0.859	0.880	0.385	0.472	0.455	0.846	0.556	0.938	0.674
Alternating ($\theta \leftrightarrow \phi, \psi$)	0.861	0.885	0.391	0.478	0.461	0.849	0.562	0.940	0.678
Joint (ViewCraft)	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**

External guards check execution budgets and valid requests (C_{exec}), and support for question-derived evidence requirements (C_{ground}). A frozen verifier checks question-derived requirements against the sent views, without reference answers (Appendix H). Replay filters rendering edits in ϕ or ψ that leave the views unchanged. Reader-only and answer-text-only edits bypass this pixel check.

3.4 Paired evaluation

Let ΔR be the paired change in task score on the development set and ΔC the change in evidence support. After execution checks pass, the gate accepts when

$$\text{accept} \iff \Delta R > 0 \quad \text{or} \quad (\Delta R = 0 \ \wedge \ \Delta C > 0). \tag{3}$$

After a joint candidate changes visual fields but loses task score ($\Delta R < 0$), ViewCraft retests candidate (ϕ, ψ) with incumbent θ only if evidence support has not fallen ($\Delta C \geq 0$) and submission and gate allowances remain. This *interface projection* consumes another submission and paired evaluation (Algorithm 1). Acceptance advances w; rejection restores the incumbent.

At test time, the saved skill supplies views, actions, and reading rules without optimization. Strategy retains rejected proposals after rollback; only accepted expressions affect execution.

4 Experiments

4.1 Setting and comparisons

We evaluate eight visual benchmarks using frozen, source-disjoint splits: MMComposition (Hua et al., 2024), ReasonMap (Feng et al., 2026), FUNSD (Jaume et al., 2019), STI-Bench (Li et al., 2025), Video-Holmes (Cheng et al., 2025), SlideVQA (Tanaka et al., 2023), OpenEQA (Majumdar et al., 2024), and MMIU (Meng et al., 2025). The first three concentrate evidence in a single image or form; the remaining five require selecting and comparing observations distributed across sources.

Score accounting. Main comparisons pool three repetitions: each independently evolved terminal skill is tested once, while Direct and Initial skill are each executed three times. Binary scores use $S/(3n)$; continuous scores average the same repeated evaluations. FunSD uses field-value ANLS. Appendix A specifies construction, execution, and scoring.

Direct uses a bare call; Initial skill uses the empty-start interface. Text-only and Interface-only independently search the two sides under the clean partition. Terminal θ-only and (ϕ, ψ)-only restore one side of the joint result to its initial state, testing dependence within the learned pair.

Training traces guide proposals; gate items determine acceptance. Test questions remain outside both stages, and no form, film, or deck crosses the frozen splits.

All evolving methods share proposal and rollout budgets. Each arm runs 24 task–seed searches with four rounds, up to four submissions per round, and the training rollouts specified in Appendix A.

Gate sets range from 48 to 240 paired items (Table 6); candidate and incumbent use the same items. Halving or doubling these sets changes mean task score by at most 0.2 points (Appendix Table 7).

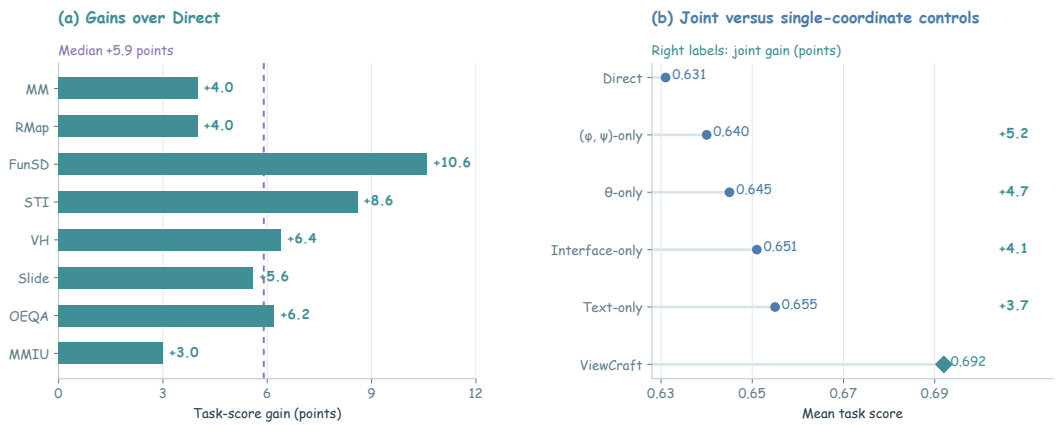

Figure 4: **Where the gain sits.** (a) Per-benchmark held-out gains over a bare Direct call (all eight positive; median $+5.9$ points). (b) Mean task score of the four single-coordinate controls against joint search (margins $+3.7$ to $+5.2$ points).

Table 4: Baselines and value decomposition at matched budget (per arm: 250 submitted proposals; per task–seed run: four rounds, 96 rollouts). Random proposals samples uniformly from the same expression grammar under the same paired gate; MPO-fullspace and Dynamo-gen optimize ViewCraft's executable interface; row 1 adapts prompt-space search; row 5 uses the full joint space and paired gate with replay, interface projection, and the evidence tiebreak disabled.

	MMComp	RMap	FunSD	STI	VHolm	Slide	OEQA	MMIU	Mean
MPO-adapted (prompt space)	0.852	0.869	0.368	0.437	0.428	0.834	0.524	0.922	0.654
Random proposals (same space, gate)	0.852	0.869	0.378	0.445	0.434	0.842	0.539	0.928	0.661
Dynamo-gen (tool generation)	0.857	0.876	0.381	0.459	0.444	0.835	0.545	0.903	0.662
MPO-fullspace (UCB selection)	0.861	0.878	0.385	0.463	0.450	0.838	0.549	0.907	0.666
Gate-only joint	0.861	0.887	0.393	0.478	0.457	0.844	0.560	0.932	0.676
ViewCraft	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**

Adapted baselines. MPO-adapted retains paired text/non-text updates; its image-side prompt controls frame selection, crops, and grid or stack attachments. Dynamo-style composes reusable select-frame, crop-box, and tile-grid calls, with rendering primitives held as library code. These implementations share the executor, splits, paired gate, and submission budget.

Across its runs, each arm submits 250 candidates and performs 200–208 paired evaluations. Retries consume submissions; replay filters unchanged renderings. Matching submissions retains the mechanisms' effect on gate access.

4.2 Task scores and coordinate effects

Joint evolution leads every benchmark (Table 2; Figure 4). Reading changes improve use of visible evidence, while interface edits expose relations lost through sampling or layout. Their relative value varies by task: SlideVQA benefits more from reading changes despite requiring multiple sources. Joint search accommodates both bottlenecks.

The terminal crossing writes $v = (\phi, \psi)$ and measures

$$I = R(\theta_1, v_1) - R(\theta_1, v_0) - R(\theta_0, v_1) + R(\theta_0, v_0). \tag{4}$$

The learned reader and interface help each other: all terminal interactions are positive, with mean $+1.8$ points (one-sided sign test $p = 0.004$). Seven bootstrap intervals exclude zero; SlideVQA's includes it (Figure 5c). Timestamp rows and canvas marks make this dependence concrete: the interface exposes a relation that its reader knows how to use.

Co-adaptation remains positive on average under all three coordinate partitions (Appendix Table 19). Moving render-referencing clauses changes the magnitude while preserving the direction under both clause-content and strict partitions. Each crossing measures dependence under its stated definition.

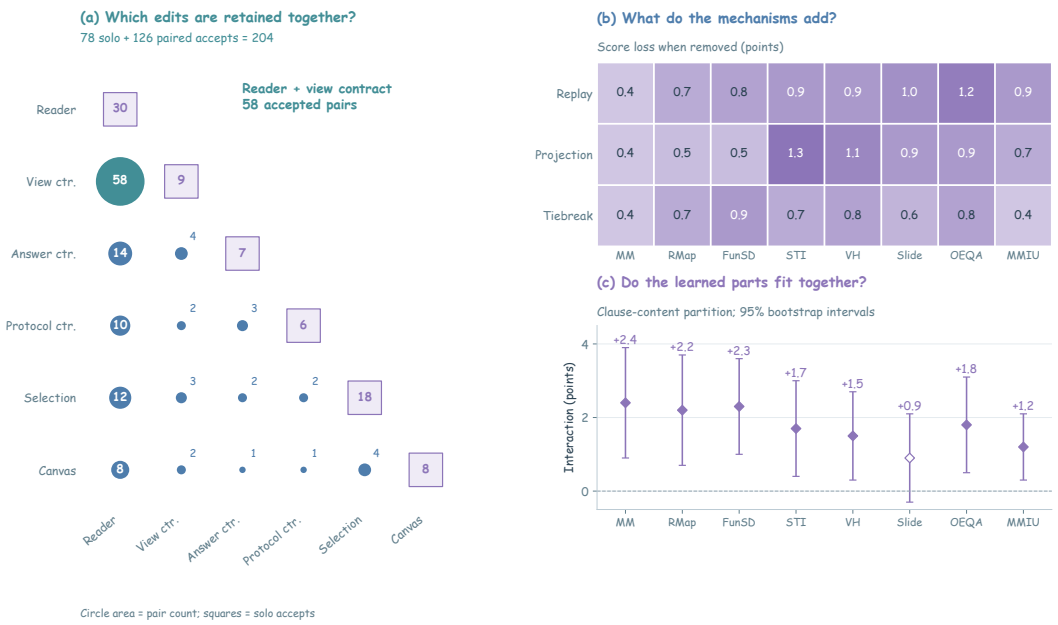

Figure 5: **Matched search and co-adaptation.** (a) Accepted co-edits across the three-arm log: circle area encodes pair counts and diagonal squares solo accepts; total accepts 204. (b) Full-model score minus each ablation, in points (Table 24). (c) Terminal interaction I under the clause-content partition; bars show 95% seed-stratified media-cluster bootstrap intervals, including zero for SlideVQA.

Every task-level estimate favors ViewCraft, with an exact sign-test $p = 0.008$. Holm correction across the comparisons retains FunSD and STI-Bench as individually significant (Appendix Table 13). Tests average runs per item and resample whole source media; the appendix reports run variation.

4.3 Search strategies at matched budget

Joint search leads alternating search by $+1.3$ points (cluster-bootstrap CI $[+0.6, +2.1]$). All eight task-level gaps are positive (exact sign-test $p = 0.008$), as are the three seed means (Appendix B).

The ladder in Table 4 separates access to the executable interface, candidate selection, failure guidance, and retention. Random same-space proposals improve over prompt-space search; UCB selection and failure-guided proposals add successive gains. The complete retention pipeline then reaches the highest score. ViewCraft leads the strongest same-space baseline by $+2.6$ points (CI $[+1.3, +3.9]$).

ViewCraft also leads Dynamo-gen and random proposals in the shared expression space. Appendix B reports the ladder increments and paired statistics.

The ordering in Table 3 reflects how components adapt to one another. Independent search combines separately learned partners; staged search fixes an interface before its reader adapts; alternating search revisits them in turn. Joint proposals can change the exposed relation and its reading rule together.

MPO-fullspace and Dynamo-gen are extensions that grant the adapted methods ViewCraft's executable search space. Trace-guided proposals lead random same-space proposals by $+1.5$ points; the retention mechanisms add $+1.6$. Their common expression grammar makes these gains attributable to the evaluated search configurations rather than access to different edits.

An evaluation-matched joint arm retains a mean score of 0.691 versus 0.692 for full ViewCraft, preserving the advantage under both submission and evaluation matching (Appendix A).

4.4 Mechanisms and evaluation cost

The reader and view contract form the most frequently accepted pair, with 58 co-edits (Figure 5a). Defining what panels contain frequently accompanies a change in how to read them. Solo accepts retain one-sided revisions; terminal crossings test clause-content co-adaptation.

Table 5: **Skill transfer across executors.** Bold marks the highest held-out score within each executor and task; tints group target executors. Self-transfer pools three runs; the Gemini diagonal reproduces Table 2. Cross-transfer uses the source's median-gate run. Direct uses the same pooling.

Executor	Skill source	MMComp	FunSD	STI-Bench	VideoHolmes
gemini-3.8-flash	Direct	0.834	0.306	0.410	0.408
	gemini-3.8-flash	**0.874**	**0.412**	0.496	**0.472**
	gpt-5.5	0.868	0.385	**0.508**	0.465
	claude-sonnet-4-6	0.852	0.379	0.484	0.458
gpt-5.5	Direct	0.850	0.271	0.567	0.550
	gemini-3.8-flash	0.871	**0.388**	0.623	0.598
	gpt-5.5	**0.881**	0.377	**0.641**	**0.612**
	claude-sonnet-4-6	0.865	0.354	0.627	0.593
claude-sonnet-4-6	Direct	0.623	0.221	0.520	0.444
	gemini-3.8-flash	0.693	0.308	0.556	0.495
	gpt-5.5	0.702	0.298	0.564	**0.518**
	claude-sonnet-4-6	**0.709**	**0.318**	**0.569**	0.507

Mechanism effects and their overlap. Replay filters inert renderings, projection recovers useful interfaces from rejected joint proposals, and the evidence tiebreak resolves equal task scores. Removing a single stage costs 0.7–0.9 points; removing all stages costs 1.6 (Appendix Table 24). Every paired interval excludes zero. The joint loss reveals overlap among stages.

Asymmetric projection has the highest mean score among the retry designs (Appendix Table 15). Keeping the verified interface while reverting the reader gains $+0.6$ points over the symmetric direction. Replay saves 30 joint-arm gate evaluations; all 21 projection retries enter the saved skill.

Removing the evidence tiebreak lowers held-out scores on every benchmark (Appendix Table 24). The verifier agrees with human consensus on 89% of audited judgments (Appendix H).

Most of the improvement survives with answer and protocol contracts fixed: the contract-frozen condition remains within 0.5 points of full ViewCraft on average. Format-normalized rescoring also preserves the advantage on FunSD (Appendix Table 16).

The gate links local edits to retained skill quality by scoring complete responses on paired items. Replay and guards save evaluations; projection preserves eligible visual improvements after joint rejection. Their ablations measure which improvements survive into deployment (Appendix H).

4.5 Transfer and input cost

The learned pairing remains useful after changing the executor. On held-out questions, every transferred skill exceeds its target executor's Direct baseline by at least 1.5 points (Table 5).

Three transfers exceed target-evolved skills: GPT-5.5 improves Gemini on STI-Bench and Sonnet on VideoHolmes; Gemini improves GPT-5.5 on FunSD. The remaining transfers fall within 3.3 points of target-evolved skills, preserving the pairing's value across executors.

The MMComp instance illustrates what transfers (Appendix I). Its reader grounds option nouns and builds a verdict table; the interface colors notes by option and explains their identities. These saved rules construct and interpret evidence for new questions without target-specific search.

Evidence construction trades input volume for relevant detail (Appendix Table 21). It adds tokens over Direct while learned selection reduces them from Initial skill on multi-source tasks.

Token-matched Direct recovers $+0.9$ of the $+6.1$-point margin; three-sample self-consistency reaches a similar mean (Appendix Table 23). The remaining gain connects evidence to its reader.

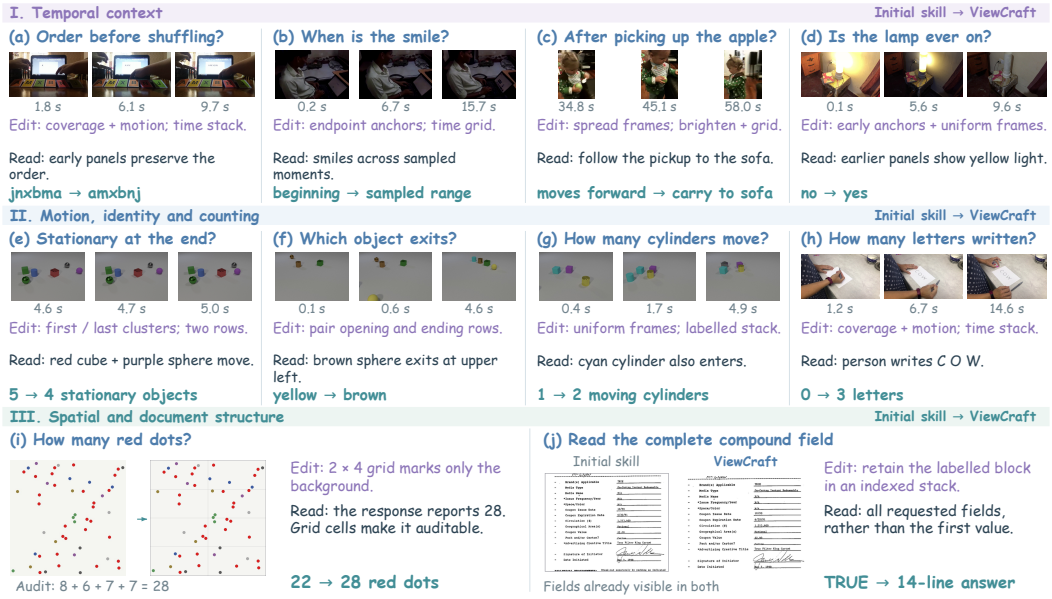

Figure 6: **Ten repairs linking views to readings.** Panels link questions, view edits, reading summaries, and answer changes.

Reuse amortizes search, while execution adds inspection. Latency rises by 7.6 seconds per item over Direct (Appendix Table 22). Task score and execution cost characterize reuse.

4.6 Case studies

Figure 6 shows how view edits expose usable relations. Endpoint frames reveal motion when the reader matches objects across rows; early letter panels recover the pre-shuffle state when read before later frames. Both repairs combine temporal coverage with a rule for interpreting it.

A background grid preserves dot colors while creating stable counting regions. The reader can localize partial counts and check them against the image, making existing evidence easier to use.

Both form inputs contain the requested fields, but the initial reader returns one value while the evolved reader preserves the compound query. The response must retain field–value associations as well as the required format. Appendix Figure 8 illustrates regressions when labels or option letters are lost.

5 Conclusion

ViewCraft makes evidence construction a persistent, editable part of a multimodal skill. The central design choice is to evolve what the executor sees together with how it reads. Frame selection determines which moments are available, layout exposes relationships across observations, and resolution controls whether fine details remain legible. A reading procedure turns these views into comparisons, counts, and decisions. The probes and recorded repairs connect these concrete operations to the two questions that motivate the work: whether the necessary evidence is visible, and whether the reader can use its spatial and temporal organization. Joint evolution improves both parts while keeping the executor frozen. It outperforms single-coordinate search and same-space adapted baselines, and its advantage over Direct persists at matched input budget. Replay and paired evaluation make candidate views inspectable and tie retention to task outcomes; interface projection recovers useful visual edits from rejected joint candidates. The resulting state can be applied to new questions and transferred across executors. On all five multi-source tasks, learned selection also sends 8–15% fewer input tokens than Initial skill. Together, these results identify an actionable unit of reuse: an executable evidence interface paired with the procedure that uses its rendered views to answer new questions.

REPRODUCIBILITY STATEMENT

Appendix A gives the evaluation protocol; Appendix B reports statistical tests; Appendix H defines the gate and round algorithm. Appendix I typesets a complete evolved skill.

AI USE STATEMENT

Generative Al was used to assist in outlining, drafting, and language refinement for this work. All final arguments, analysis, factual verification, and editorial decisions remain solely the responsibility of the author.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-holmes: Can mllm think like holmes for complex video reasoning? *arXiv preprint arXiv:2505.21374*, 2025.

Yumin Choi, Dongki Kim, Jinheon Baek, and Sung Ju Hwang. Multimodal Prompt Optimization: Why Not Leverage Multiple Modalities for MLLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6cb71a5c9585c9a9fdd246ae32b309d3-Abstract-Conference.html.

Sicheng Feng, Song Wang, Shuyi Ouyang, Lingdong Kong, Zikai Song, Jianke Zhu, Huan Wang, and Xinchao Wang. Reasonmap: Towards fine-grained visual reasoning from transit maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41077–41088, 2026.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=9ZxnPZGmPU.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=t9U3LW7JVX.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.09403.

Hang Hua, Yunlong Tang, Ziyun Zeng, Liangliang Cao, Zhengyuan Yang, Hangfeng He, Chenliang Xu, and Jiebo Luo. MMComposition: Revisiting the Compositionality of Pre-trained Vision-Language Models. *arXiv preprint arXiv:2410.09733*, 2024. URL https://arxiv.org/abs/2410.09733.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 international conference on document analysis and recognition workshops (ICDARW)*, volume 2, pp. 1–6. IEEE, 2019.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. XSkill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026. URL https://arxiv.org/abs/2603.12056.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od.

Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5622–5632. IEEE, 2025.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16488–16498. IEEE, 2024.

Fanqing Meng, Jin Wang, Chuanhao Li, Quanfeng Lu, Hao Tian, Tianshuo Yang, Jiaqi Liao, Xizhou Zhu, Jifeng Dai, Yu Qiao, et al. Mmiu: Multimodal multi-image understanding for evaluating large vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 38405–38453, 2025.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic llm agent search in modular design space. In *International Conference on Learning Representations*, volume 2025, pp. 3841–3865, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Yutao Sun, Yanting Miao, Hao-Xuan Ma, et al. Dynamo: Dynamic Skill-Tool Evolution for Vision-Language Agents. *arXiv preprint arXiv:2606.30185*, 2026. URL https://arxiv.org/abs/2606.30185.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 13636–13645, 2023.

Liyan Tang, Cyrus Rashtchian, Chun-Sung Ferng, Andrew Tomkins, Da-Cheng Juan, and Tu Vu. WikiSkill: Compiling agent experience into persistent knowledge for skill evolution. arXiv preprint arXiv:2608.27454, 2026. URL https://arxiv.org/abs/2608.27454.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *Computer Vision – ECCV 2024*, pp. 58–76, 2024. doi: 10.1007/978-3-031-72989-8_4. URL https://doi.org/10.1007/978-3-031-72989-8_4.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for llm reasoning on long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3272–3283, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VideoTree_Adaptive_Tree-based_Video_Representation_for_LLM_Reasoning_on_Long_CVPR_2025_paper.html.

Binxiao Xu, Ruichuan An, Bocheng Zou, and Hang Hua. Agent Skills Should Go Beyond Text: The Case for Visual Skills. *arXiv preprint arXiv:2606.01414*, 2026. URL https://arxiv.org/abs/2606.01414v1.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*, 2023. URL https://arxiv.org/abs/2310.11441.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026. URL https://arxiv.org/abs/2605.23904.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. Frame-Voyager: Learning to query frames for video large language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.03226.

Yongsheng Yu, Ziyun Zeng, Zhiyuan Xiao, Zhenghong Zhou, Hang Hua, Wei Xiong, and Jiebo Luo. Aurora: Unified video editing with a tool-using agent. *arXiv preprint arXiv:2605.18748*, 2026.

Ziyun Zeng, Hang Hua, and Jiebo Luo. Mira: Multimodal iterative reasoning agent for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9563–9573, 2026a.

Ziyun Zeng, Hang Hua, Bocheng Zou, Mu Cai, Rogerio Feris, and Jiebo Luo. Mementogui: Learning agentic multimodal memory control for long-horizon gui agents. *arXiv preprint arXiv:2605.18652*, 2026b.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=z5uVAKwmjf.

Kangning Zhang, Shuai Shao, Qingyao Li, Jianghao Lin, Lingyue Fu, Shijian Wang, Wenxiang Jiao, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. MMSkills: Towards multimodal skills for general visual agents. *arXiv preprint arXiv:2605.13527*, 2026. URL https://arxiv.org/abs/2605.13527.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=92gvk82DE-.

A Evaluation protocol

All conditions use frozen splits, the same executor, and paired item identities. Direct uses one bare call; Initial skill uses the empty-start execution interface. Table 6 lists source units and scoring rules. Sampling uses the full test set below 500 questions, half at 500–1000, and a quarter above 1000, preserving whole source media and category coverage. The selection is frozen before runs.

Score accounting. Every accuracy in the main comparisons is the pooled mean over three repetitions with denominator $3n$: evolving conditions are run as three independent searches (different seeds), each final skill evaluated once on the held-out set; Direct and Initial skill are executed three times each (repetition applies uniformly across conditions). For binary metrics this makes every entry an integer multiple of $1/(3n)$. FunSD, SlideVQA, and OpenEQA use continuous scores (field-value ANLS, ANLS, and LLM-Match respectively), so their means are unconstrained. This pooling yields MMIU's $685/720 \approx 0.951$ over three runs.

Evidence-interface probes. The probes use a frozen executor on 96 VideoHolmes questions from 26 non-test films and 240 field crops from FunSD training forms. These samples have zero overlap with the main test sets. Three selectors use the same questions at four frame budgets; plain and enhanced prompts are compared over these settings. Oracle crops hold requested content fixed while varying delivered resolution.

Wrong-film probe. Replacing the frames for 64 previously correct questions with another film yields 12 correct and 52 wrong answers, with no abstentions or mismatch flags (Figure 2c).

Table 6: Evaluation units, source-disjoint splits, and scoring. Cells report questions / unique source media. Train, gate, and test draws have pairwise-disjoint source media; no form, film, diagram, or deck appears in two splits. Gate questions are never shown to the proposal stage, which uses training traces only.

Benchmark	Medium	Train	Gate	Test	Metric
MMComp	image	48 / 48	96 / 96	151 / 151	option-letter accuracy
ReasonMap	transit diagram	24 / 2	48 / 2	153 / 4	route accuracy
FunSD	form	64 / 58	96 / 72	248 / 47	field-value ANLS
STI-Bench	video	36 / 12	72 / 18	221 / 36	option-letter accuracy
VideoHolmes	video	24 / 10	72 / 16	183 / 28	option-letter accuracy
SlideVQA	slide deck	40 / 10	60 / 14	155 / 30	ANLS
OpenEQA	video	12 / 4	48 / 8	95 / 10	LLM-Match (mean)
MMIU	image	240 / 240	240 / 240	240 / 240	option-letter accuracy

Table 7: Gate-size sensitivity: mean task score when every benchmark's gate set is scaled as shown, holding proposals, rollouts, and test sets fixed.

Gate scale (items/task)	8	25%	50%	100% (default)	200%
Mean task score	0.679	0.686	0.690	0.692	0.693

FunSD construction. FunSD is converted to field-value QA. Question-class entities become field labels; the reference answer is the vertically-nearest answer-class entity. Table 6 gives the constructed splits. The split unit is the form: no form appears in two splits, so every field question derived from one form stays with it. The metric is field-value ANLS (normalized edit similarity on the lowercased transcription), continuous in $[0, 1]$; Table 2 reports the pooled mean. Compound fields are constructed when a single question entity spans multiple lines; the reference retains the full multi-line value.

FunSD construction audit. Of 200 constructed pairs, 192 link the correct field (96.0%); the eight errors come from multi-column forms (4), multi-line answers (3), and nearest-entity ambiguity (1). With gold labels, Direct shifts 0.306 to 0.312 and ViewCraft 0.412 to 0.415.

Table 8: Per-arm development cost over 24 task–seed runs. Grouped methods each incur the listed cost. Tokens sum logged input tokens at measured per-call rates: executor 774, verifier 151, optimizer 3.4k. Shared initialization is executed once and reused by every arm.

Method	Proposals	Gate evals	Exec. calls	Verif. calls	Optim. calls	Tokens
Shared initialization	0	8	0.7k	0	0	0.5M
Text-only / Interface-only	250	200	45.2k	38.8k	384	42.2M
ViewCraft	250	208	47.0k	40.4k	384	43.8M
Alternating / Staged / Indep.	250	200	45.2k	38.8k	384	42.2M
MPO-adapted / MPO-fullspace	250	200	45.2k	38.8k	384	42.2M
Dynamo-style / Dynamo-gen	250	200	45.4k	38.8k	384	42.4M

Cost accounting. For the ViewCraft arm, the logged call counts give

$$47.0\text{k} \times 774 + 40.4\text{k} \times 151 + 384 \times 3.4\text{k} \simeq 36.4\text{M} + 6.1\text{M} + 1.3\text{M} = 43.8\text{M}.$$

This is 1.8M input tokens per task–seed run. Arm costs differ by under 4%; additional image-bearing gate evaluations account for the joint arm's higher cost.

Search-log scope and accounting. Each arm runs $8 \times 3 = 24$ task–seed searches. Every search has four rounds, up to four submissions per round, and 96 training rollouts. Submissions total 250 per arm, with 200–208 paired gate evaluations. The three main arms submit 750 candidates out of 1,152 available slots (65%) and perform 608 paired evaluations. Table 10 resolves these totals by arm.

Table 9: Development-evaluation accounting.

Submitted (incl. 21 projections)	750
Pre-gate rejects (64 inert / 78 guards)	142
Paired calls (204 accepts / 404 rejects)	608
Candidate evaluations	**608**

The gate accepts 204 candidates and rejects 404; accepted candidates include all 21 interface projections. Eight shared round-zero evaluations are outside these candidate counts. Strategy and ablation variants have their own ledgers.

Table 10: Submission and filtering ledger for the three main arms. Ordinary and projection columns sum to each arm's submissions or evaluations. This aggregate excludes strategy and ablation variants.

	Submitted		Pre-filtered		Evaluated	
	Ordinary	Projection	Inert	Guards	Ordinary	Projection
Joint	229	21	30	12	187	21
Text-only	250	0	0	50	200	0
Interface-only	250	0	34	16	200	0
Total	729	21	64	78	587	21

Projection candidates reuse validated visual edits and bypass redundant rendering replay; they still consume submission slots and a new paired evaluation. Evaluation counts therefore reflect filtering under matched submissions. All evolving methods share four rounds and 96 training rollouts. Capping joint evaluations at 200 yields mean score 0.691 versus 0.692 for full ViewCraft.

Reproducibility. ReasonMap has 1,008 route questions over 30 official city-transit diagrams, covering fewest-transfer, no-transfer, and station-order prompts. Appendix I typesets the exported skill. Search reproduction also requires starting skills, optimizer and verifier prompts, the expression interpreter, split manifests, and per-item predictions.

Implementation. Executor: gemini-3.8-flash, greedy decoding, one call per observation step; budgets per domain: 12–24 steps, 4–9 images per request. Optimizer: the same model, temperature 0.7, up to 4 attempts per round, the complete executed skill in Appendix I. Verifier for C_{ground}: the same model in verification mode (binary yes/no per requirement, options withheld), two calls per gate item. Stopping: fixed 4 rounds. All conditions share these settings. Cross-executor transfers and the case-study instance use the run with the median gate score among three seeds; the main table and self-transfer diagonal pool all three runs.

States and comparisons. Let $w_0 = (\theta_0, \phi_0, \psi_0)$ be Initial skill and $w_1 = (\theta_1, \phi_1, \psi_1)$ the terminal joint state. Text-only independently optimizes a reader θ_T, evaluating $(\theta_T, \phi_0, \psi_0)$. Interface-only independently optimizes (ϕ_V, ψ_V), evaluating $(\theta_0, \phi_V, \psi_V)$. The terminal θ-only row evaluates $(\theta_1, \phi_0, \psi_0)$; (ϕ, ψ)-only evaluates $(\theta_0, \phi_1, \psi_1)$. Both restore unused coordinates to the initial state. Under the clean partition, Interface-only excludes reasoning and decision rules; it includes rendering, view metadata, and surface formats.

For binary gate scores, one changed outcome moves ΔR by 0.4–2.1 percentage points at the default gate sizes. Candidate and incumbent use the same questions for both score and support differences.

Scoring and cost. Table 6 distinguishes option-letter accuracy, route accuracy, field-value ANLS, ANLS, and mean LLM-Match. Four benchmarks use option letters; ReasonMap uses route matching. Input tokens include all requests for an item.

Semantic content and answer format. The contract-frozen control and format-normalized rescore in Table 16 separate evidence use from answer presentation. Case records still expose local failures: omitting a field label changes the requested content, while a numerical value in place of an option letter can fail the answer contract despite a correct calculation.

B UNCERTAINTY AND CONTRACT CONTROLS

Table 11: Joint versus alternating search. Gaps and 95% seed-stratified media-cluster bootstrap CIs are in points. Permutation tests exchange whole source media (10,000 resamples), using per-item means over three runs.

Benchmark	Gap	95% CI	p	Benchmark	Gap	95% CI	p
MMComp	1.3	[0.1, 2.6]	0.06	VideoHolmes	1.1	[0.0, 2.4]	0.07
ReasonMap	1.3	[0.1, 2.6]	0.19	SlideVQA	0.9	[−0.2, 2.1]	0.12
FunSD	2.1	[0.8, 3.5]	0.01	OpenEQA	1.2	[0.2, 2.5]	0.09
STI-Bench	1.8	[0.5, 3.3]	0.04	MMIU	1.1	[0.0, 2.3]	0.08
Mean gap: 1.3 points 95% CI: [0.6, 2.1]							

Baseline and projection comparisons. Joint search leads alternating search in every seed. The corresponding gaps are $+1.5$, $+1.2$, and $+1.3$ points.

ViewCraft's gain over MPO-fullspace has CI $[+1.3, +3.9]$; all tasks favor ViewCraft (sign-test $p = 0.008$). The same-space margins over Dynamo-gen and random proposals appear in Table 12.

Table 12: Mean task-score increments along the matched-budget ladder, and final margins. All differences are in points.

Ladder step	Gain	ViewCraft comparison	Margin
Prompt space to random same-space	+0.7	MPO-fullspace	+2.6
Random to UCB selection	+0.5	Dynamo-gen	+3.0
UCB to failure-guided joint	+1.0	Random proposals	+3.1
Gate-only to full retention	+1.6	Random to failure-guided joint	+1.5

Table 13: Uncertainty versus Direct. Std is over three search runs. Gain CIs are 95% seed-stratified media-cluster bootstraps; permutation tests use each item's mean score over the three runs. Gains and intervals are in points.

Benchmark	ViewCraft (std)	Gain (pts)	Cluster CI (pts)	vs. Direct p
MMComp	0.874±0.007	4.0	[0.3, 7.8]	0.03
ReasonMap	0.898±0.008	4.0	[0.2, 7.9]	0.05
FunSD	0.412±0.015	10.6	[5.8, 15.2]	<0.001
STI-Bench	0.496±0.011	8.6	[2.9, 14.2]	0.006
VideoHolmes	0.472±0.010	6.4	[1.2, 11.6]	0.01
SlideVQA	0.858±0.007	5.6	[0.1, 11.1]	0.05
OpenEQA	0.574±0.014	6.2	[1.6, 10.9]	0.02
MMIU	0.951±0.005	3.0	[0.0, 6.0]	0.08

For MMIU, Direct's score of 0.921 leaves 7.9 points of headroom; ViewCraft recovers 3.0 points.

Significance testing. All tests operate on per-item means over the three runs. Paired permutation tests exchange whole source media over 10,000 resamples of these per-item means. Confidence intervals use seed-stratified media-cluster bootstrap resampling: whole films, forms, or decks are resampled within seed strata and then averaged. Search-run standard deviations are reported separately. Binary metrics use the same per-item protocol.

All task-level estimates favor ViewCraft (exact sign test $p = 0.008$). Across-task mean and median gains are $+6.1$ and $+5.9$ points. After Holm correction, FunSD and STI-Bench remain individually significant.

The contract-frozen condition preserves most of ViewCraft's improvement. Format-normalized rescoring applies a common scoring rule to the existing responses, holding visual inputs and generated answers fixed. On FunSD, it increases Direct by 1.2 points and ViewCraft by 0.9 points, preserving

Table 14: Across-task gain summaries and multiplicity correction versus Direct.

Summary	Value	Holm-adjusted comparison	p
Mean gain (points)	+6.1	FunSD	0.004
Median gain (points)	+5.9	STI-Bench	0.042
Relative gain range (%)	3.3–34.6	Other tasks	> 0.05
Median relative gain (%)	9.5		

Table 15: Projection design at matched budget (per arm: 250 submitted proposals). Gap is full ViewCraft minus the variant in points; CIs are 95% seed-stratified media-cluster bootstraps of the paired per-item difference; p is a paired permutation test on per-item means (10,000 resamples).

Variant	MM	RMap	FunSD	STI	VH	Slide	OEQA	MMIU	Mean	Gap [CI]	p
No retry (drop on joint rejection)	0.870	0.893	0.407	0.483	0.461	0.849	0.565	0.944	0.684	0.8 [0.3, 1.3]	0.004
Plain retry (fresh proposals)	0.872	0.893	0.408	0.484	0.463	0.850	0.566	0.944	0.685	0.7 [0.2, 1.2]	0.008
Symmetric projection (keep θ')	0.872	0.895	0.409	0.486	0.463	0.851	0.567	0.946	0.686	0.6 [0.1, 1.1]	0.020
No ΔC trigger (retry all eligible)	0.872	0.895	0.408	0.487	0.464	0.852	0.568	0.947	0.687	0.5 [0.1, 1.0]	0.030
Asymmetric projection (ours)	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**	—	—

Table 16: Isolating evidence from contract effects. "Evidence-only" freezes the answer and protocol contracts at their initial values and evolves only evidence selection, transformation, layout, and the paired reading rules. "Format-normalized" rescores all responses case-/unit-insensitively with letter extraction.

Condition	MMComp	RMap	FunSD	STI	VHolm	Slide	OEQA	MMIU
ViewCraft	0.874	0.898	0.412	0.496	0.472	0.858	0.574	0.951
format-normalized	0.876	0.900	0.421	0.498	0.474	0.861	0.581	0.953
Evidence-only (contracts frozen)	0.872	0.895	0.401	0.492	0.468	0.854	0.566	0.947
format-normalized	0.872	0.895	0.409	0.495	0.470	0.857	0.574	0.947

ViewCraft's advantage. Figure 8 distinguishes preservation of requested content from compliance with the answer format.

Table 17: Fixed-trajectory decomposition. Each row replays the terminal skill's recorded observation sequence and substitutes one coordinate. The evidence transplant delivers the evolved views to the initial reader; the reading transplant gives the evolved reader the initial views.

Replayed state	MMComp	RMap	FunSD	STI	VHolm	Slide	OEQA	MMIU	Mean
Initial skill (θ_0, v_0)	0.786	0.841	0.288	0.385	0.381	0.812	0.478	0.908	0.610
Evidence transplant (θ_0, v_1)	0.775	0.845	0.318	0.440	0.424	0.823	0.519	0.922	0.633
Reading transplant (θ_1, v_0)	0.848	0.861	0.345	0.410	0.403	0.829	0.503	0.918	0.640
ViewCraft (θ_1, v_1)	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**

With the observation trajectory fixed, the evidence and reading transplants each recover part of the joint gain. Table 18 reports their additive contrast; the terminal crossing in Figure 5c allows each reader to issue its own requests. Re-drawing the gate items shifts every arm by under half a point

Table 18: Fixed-trajectory contrasts relative to Initial skill, in points. The final row is the joint gain minus the two separate transplant gains.

Evidence transplant	+2.3
Reading transplant	+3.0
Joint pairing	+8.2
Joint-minus-transplants contrast	+2.9

and preserves the joint-over-alternating gap computed from the displayed means (Table 20).

Table 19: Terminal interaction I (points) under three coordinate partitions. Storage keeps exported files intact; clause-content assigns inference and decision rules to θ; strict additionally moves render-referencing clauses (note placement, rendered-color references) to the interface.

Partition	MM	RMap	FunSD	STI	VH	Slide	OEQA	MMIU	Mean
Storage fields	1.4	1.0	1.6	0.8	0.6	−0.3	0.5	0.8	+0.8
Strict (render clauses out of θ)	1.8	1.5	1.8	1.2	1.0	0.5	1.2	0.9	+1.2
Clause-content (used throughout)	2.4	2.2	2.3	1.7	1.5	0.9	1.8	1.2	+1.8

Table 20: Gate-split sensitivity. Differences use the displayed mean task scores; the final row reports joint minus alternating in points.

Strategy	Split A	Split B	Difference (points)
Alternating	0.678	0.674	−0.4
Gate-only joint	0.676	0.672	−0.4
ViewCraft	0.692	0.688	−0.4
Joint − Alternating (points)	+1.4	+1.4	0.0

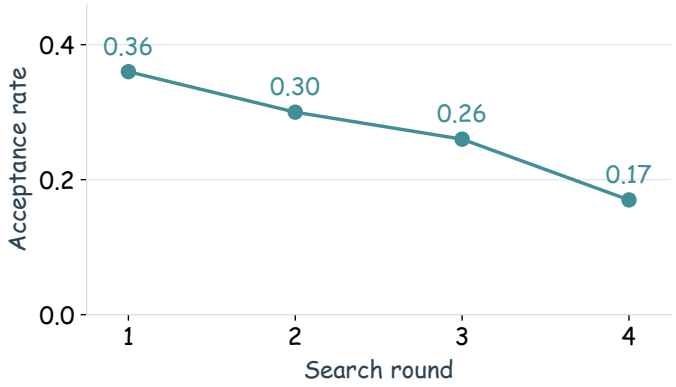

Figure 7: **Search selectivity.** Rates pool the three evolving arms and use submitted proposals, including projection retries and pre-gate rejects. Rejected replays and guard failures consume no development calls.

C Transfer and Search Diagnostics

Every evaluated transferred skill improves over its target executor's Direct baseline by at least 1.5 points. Three transfers also exceed the target-evolved skill; the remaining transfers fall within 3.3 points of it. The tightest margin over Direct is GPT-5.5 executing MMComp with the Sonnet skill.

Acceptance declines across search rounds, while score-drop rejections identify candidates that underperform the incumbent (Figure 7). These diagnostics characterize selection during search; the held-out comparisons establish the value of the retained skills. Reader–view-contract pairs are the most frequently accepted co-edit (Figure 5a).

Within the recorded accepted cohort (including projection), 153 of 204 edits both repair and regress items. The averages are 5.4 repairs and 2.9 regressions. The paired gate selects a net improvement even when some previously solved items regress.

The rejected FunSD proposal returned "None" on 89% of paired items, reducing the score from 0.273 to 0.075. Its demand for a literal query-label match excluded valid composite fields. The next proposal relaxed this rule through fuzzy matching.

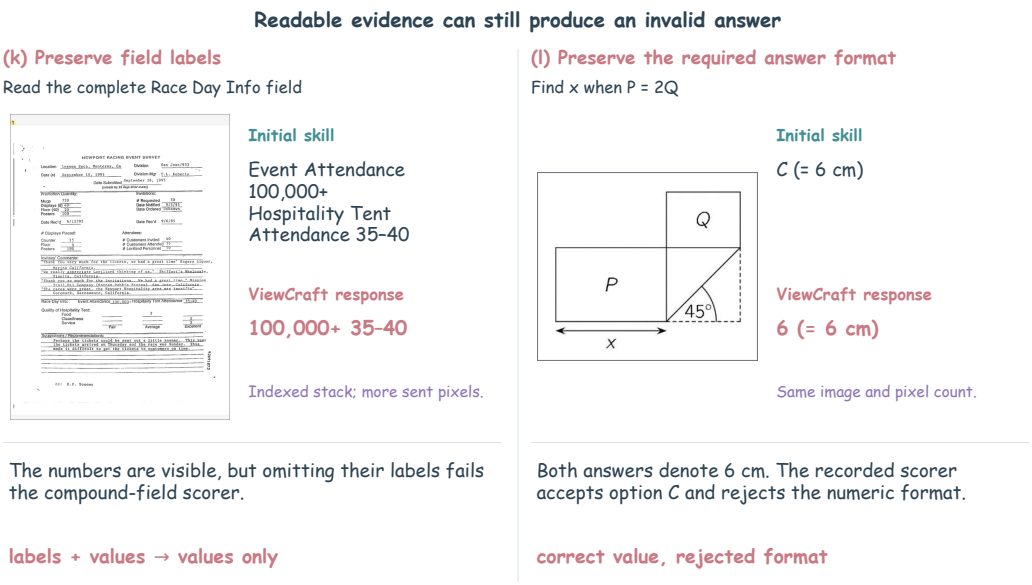

Figure 8: **Two answer-contract regressions.** In (k), the response retains the numbers but drops the labels required by the compound-field query. In (l), both responses denote 6 cm, yet only option letter C passes the recorded scorer.

D Discrete proposal space

The clean partition assigns all inference and decision rules to θ: polarity handling, verdict-table logic, when-to-zoom decisions, and mark-reading procedures. The interface contains executable rendering operations in ϕ (selection, transformation, crop admission, canvas), layout and view metadata in ψ, and answer/action surface formats. Crop admission controls whether a requested crop is rendered; deciding whether to request it belongs to the reader. The terminal crossing restores all reasoning clauses with θ, including those stored in interface files.

Clause-content assigns inference and decision rules to θ; mark placement and names of rendered colors belong to the interface vocabulary. Storage keeps exported files intact. The strict partition additionally moves every render-referencing clause to the interface before computing the crossing (Table 19).

In the exported instance, `answer_contract` contains both the single-letter format and decision rules for choosing that letter; `protocol_contract` contains both JSON action syntax and domain guidance. The former's polarity/verdict clauses and the latter's reading/zoom policy belong to θ under the re-partition. View descriptions, color identities, and action schemas remain on the interface side.

E Reading the recorded cases

Figure 6 presents ten repairs and Figure 8 two regressions. The figures pair questions and responses with the view changes and reading rules behind them.

Long video stacks are reflowed or excerpted for legibility, with their source timestamps retained. Each displayed view comes from evidence sent to the executor.

F Preserving query semantics and answer format

In (k), the evolved response keeps both numbers but drops field labels required by the compound-field scorer. In (l), both conditions read 6 cm, yet only the option letter passes. These cases distinguish preserving the requested field associations from producing an admissible answer format, motivating evaluation of the complete response.

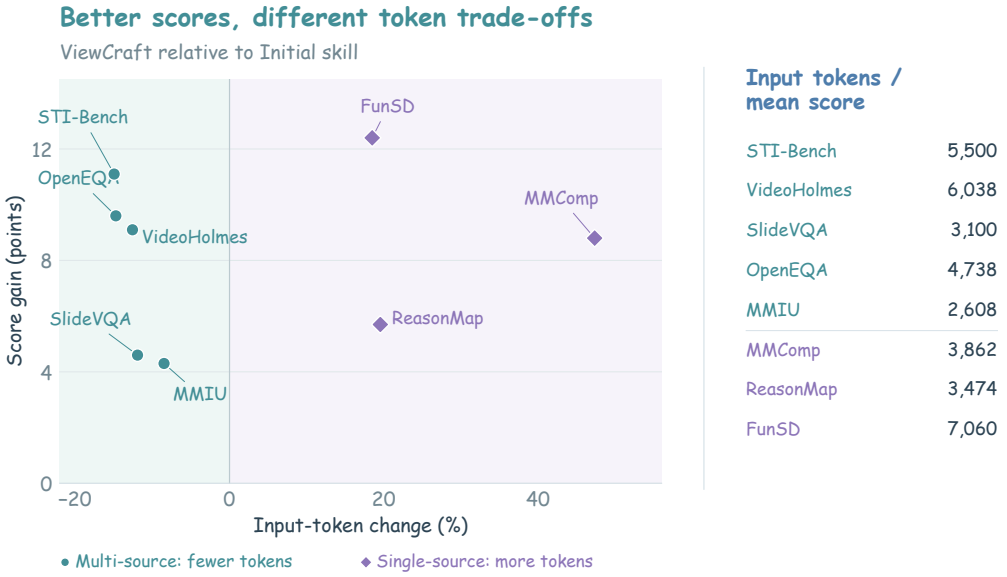

Figure 9: **Task scores and input-token cost.** Left: score gains and token changes relative to Initial skill; negative horizontal values mean savings. Circles denote multi-source tasks, diamonds single-source tasks. Right: ViewCraft input tokens divided by mean score (a score-normalized cost for continuous metrics). Counts include all requests; Appendix Table 21 lists absolute counts.

G ABSOLUTE INPUT-TOKEN COSTS

Table 21: Mean input tokens per item across all requests. Original counts determine Figure 9 and the appended ViewCraft-minus-Direct column. Relative to Direct, increases are 15–23% for multi-source tasks and 67–108% for single-source tasks; multi-source counts fall approximately 8–15% from Initial skill.

Benchmark	Direct	Initial skill	ViewCraft	VC − Direct
STI-Bench	2,380	3,210	2,730	+350
VideoHolmes	2,410	3,260	2,850	+440
SlideVQA	2,240	3,020	2,660	+420
OpenEQA	2,360	3,190	2,720	+360
MMIU	2,010	2,710	2,480	+470
MMComp	1,620	2,291	3,375	+1,755
ReasonMap	1,850	2,610	3,120	+1,270
FunSD	1,740	2,456	2,910	+1,170

Table 22: Deployment cost averaged equally over the eight tasks. Input means are the arithmetic means of Table 21; search cost is amortized over reuse.

	Input tok/item	Output tok/item	Calls/item	Latency (s/item)
Direct	2.08k	38	1.0	4.2
Initial skill	2.84k	61	2.6	9.4
ViewCraft	2.86k	69	3.4	11.8

Reuse amortizes the one-time search cost in Table 8. On multi-source tasks, input selection reduces input tokens relative to Initial skill. ViewCraft adds 7.6 seconds per item relative to Direct and generates more output tokens than Initial skill. Table 22 reports the arithmetic mean of the eight per-task input counts in Table 21, alongside output tokens, calls, and end-to-end latency.

Table 23: Task scores at matched and expanded input budgets. "Token-matched" gives Direct ViewCraft's measured per-item input budget through content-independent means (more frames or pages on multi-source tasks; upsampling plus a regular crop grid on single-source tasks). "×3" is self-consistency over three samples. Tokens/item is the task-equal-weight macro-average.

	MMComp	RMap	FunSD	STI	VHolm	Slide	OEQA	MMIU	Mean	Tok/item
Direct	0.834	0.858	0.306	0.410	0.408	0.802	0.512	0.921	0.631	2.08k
Direct (token-matched)	0.841	0.863	0.318	0.421	0.417	0.809	0.521	0.926	0.640	2.86k
Direct ×3 (self-consistency)	0.848	0.867	0.311	0.418	0.413	0.808	0.519	0.929	0.639	6.24k
Initial skill	0.786	0.841	0.288	0.385	0.381	0.812	0.478	0.908	0.610	2.84k
ViewCraft	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**	2.86k

At identical input-token budget, ViewCraft leads Direct by $+5.2$ points, with CI $[+3.6, +6.9]$ and positive gains on every task. Self-consistency with three Direct samples reaches a similar mean score (Table 23).

H GATE DEFINITIONS AND ROUND PROCEDURE

Table 24: Mechanism ablations across eight tasks and three seeds at 250 submissions per arm. Paired intervals and tests follow Table 15. Gate evaluations: full 208 (21 projections; 30 inert and 12 guard failures filtered), no replay 238, no projection 204, no tiebreak 208, and gate-only 200.

Variant	MM	RMap	FunSD	STI	VH	Slide	OEQA	MMIU	Mean	Gap [CI]	p
w/o replay filter	0.870	0.891	0.404	0.487	0.463	0.848	0.562	0.942	0.683	0.9 [0.4, 1.5]	0.002
w/o interface projection	0.870	0.893	0.407	0.483	0.461	0.849	0.565	0.944	0.684	0.8 [0.3, 1.3]	0.004
w/o evidence tiebreak	0.870	0.891	0.403	0.489	0.464	0.852	0.566	0.947	0.685	0.7 [0.2, 1.2]	0.008
Gate-only joint (all three off)	0.861	0.887	0.393	0.478	0.457	0.844	0.560	0.932	0.676	1.6 [0.8, 2.4]	<0.001
ViewCraft (full)	**0.874**	**0.898**	**0.412**	**0.496**	**0.472**	**0.858**	**0.574**	**0.951**	**0.692**	—	—

The optimizer extracts at most three requirements from the question alone. For item i, let H_i contain these atomic requirements and let $s_{ij} \in \{0, 1\}$ be the verifier's support judgment for requirement j given the actual sent views. For a nonempty requirement set,

$$C_i(w) = \frac{1}{|H_i|} \sum_{j \in H_i} s_{ij}(w), \qquad \Delta C = \frac{1}{|G|} \sum_{i \in G} [C_i(c) - C_i(w)]. \tag{5}$$

The same requirements and gate items are used for both states; reference answers are withheld from extraction and verification. Two annotators independently audited 100 stratified judgments, with pre-consensus Cohen's $\kappa = 0.76$, then resolved disagreements by discussion. The frozen verifier agreed with the resulting consensus on 89% of judgments. The tiebreak's 44 accepts concentrate on visual edits that pass replay. Removing it lowers task scores (Table 24); this ablation measures the rule's effect, while the annotation audit measures verifier agreement.

Replay checks rendered changes for proposals that edit the visual side. Reader-only and answer-text-only proposals bypass the pixel filter. This scope permits the solo reader accepts shown in Figure 5a. Execution checks still apply to every candidate before scoring.

Projection restores the incumbent reader while retaining a rejected candidate's visual edits. It is eligible after a task-score loss with $\Delta C \geq 0$ and requires both submission and gate allowances. The 21 accepted retries are included in the 204 accepted edits and in the 750 submitted candidates. Algorithm 1 takes the remaining gate and submission allowances as inputs.

Algorithm 1 One ViewCraft round with a shared gate allowance. Every candidate, including projection, consumes one submission. Each paired evaluation consumes one gate unit; pre-evaluation rejections consume no gate units (Table 9).

Require: Incumbent w with cached gate scores, training items T, gate items G, remaining gate allowance B, remaining submissions P, strategy S

If $P = 0$, return incumbent; otherwise propose c from training traces and S; $P \leftarrow P - 1$
Initialize verdict as rejected and next state as w
if c changes rendering fields and replay is pixel-identical **then**
    Record inert rejection in S; **return** (w, B, P, S)
end if
if c fails C_{exec} or $B = 0$ **then**
    Record failed check or exhausted budget; **return** (w, B, P, S)
end if
Evaluate c against w on G; obtain $\Delta R, \Delta C$; $B \leftarrow B - 1$
if $\Delta R > 0$ or ($\Delta R = 0$ and $\Delta C > 0$) **then**
    Accept c; next state $\leftarrow c$
else if c has joint reading/visual edits, $\Delta R < 0$, $\Delta C \geq 0$, and $B > 0$, $P > 0$ **then**
    $c_v \leftarrow (\theta_w, \phi_c, \psi_c)$; $P \leftarrow P - 1$; reuse validated rendering
    Evaluate c_v against w on G; $B \leftarrow B - 1$
    If the same score/support gate passes, accept c_v and set next state $\leftarrow c_v$
end if
Record every attempted state and verdict in S; retain w if no state passes
return (next state, B, P, S)

I A COMPLETE EVOLVED SKILL

The MMComp instance consists of `SKILL.md`, `VIEW.yaml`, and `LAYOUT.yaml`. These are storage containers, not the logical coordinate split: reasoning clauses in any file belong to θ, as specified in Appendix D. The reader first grounds option nouns and builds a verdict table. The crop rule is `zoom_gain > 1.05 or region_frac < 0.12`; canvas expressions color notes by option ordinal. The view contract explains those colors, and the answer contract requires the final option letter. These files make the relation between the reader, visual configuration, and interface concrete. The complete contents are typeset below, including four runtime slots: `{pool_size}`, `{max_crops}`, `{ans_header}`, and `{ans_example}`. The domain configuration binds these at load time. For MMComp, `pool_size`=8 source units per item and `max_crops`=2, matching `max_crops_per_step`; the protocol template supplies the answer header and example. The frozen artifact includes the template and these bindings. Markdown and YAML are typeset for reading with their content preserved.

I.1 READING PROCEDURE: SKILL.MD

mmcomposition Reading Procedure

What this domain contains

Items are single composite images — multi-panel layouts, presentation slides, side-by-side photo pairs, captioned images, or scene photos. The question probes spatial relationships, outlier detection, caption matching, or scene description. Answer options (A/B/C/D or Left/Right style) are always embedded in the question text you receive — never refuse on the grounds that options are missing.

Step 0 — Noun-grounding pre-pass (ALWAYS, before any other reasoning)

Before interpreting the image content against the question, do this explicitly:

1. Parse each answer option and extract the key concrete nouns (objects, people, animals, scene elements).
2. For each option, state whether its key noun(s) are **visible** or **absent** in the image.
3. Count how many options describe objects that are entirely absent from the image.
4. Note the question's polarity:
   - 'which does NOT apply' / 'pick the incorrect' / 'does not apply' / 'wrong description' → NEGATION-FRAMED: proceed to Step 0-V (Verdict Table) immediately.
   - 'which aligns / which is correct / right description / accurately describes' → AFFIRMATIVE: proceed to Step 0-V as well, marking each option APPLIES or ABSENT.

- 'yes/no' → after confirming subject locations with notes, fill in: 'The [subject] IS / IS NOT [relation] the [object]' before mapping to A/B.

5. Write this grounding check in your note field: 'Objects in options: [list]. Visible: [list]. Absent: [list]. Question asks for: [correct/incorrect/yes/no].'

This pre-pass prevents over-elimination (answering all letters) and polarity flips.

Step 0-V — MANDATORY VERDICT TABLE (for all multi-option questions)

Before committing to any letter, write a verdict table in your note field. Format exactly:

```
VERDICT TABLE
A: [one-line visual evidence or NOT FOUND IN IMAGE] → APPLIES / ABSENT
B: [one-line visual evidence or NOT FOUND IN IMAGE] → APPLIES / ABSENT
C: [one-line visual evidence or NOT FOUND IN IMAGE] → APPLIES / ABSENT
D: [one-line visual evidence or NOT FOUND IN IMAGE] → APPLIES / ABSENT
QUESTION POLARITY: [asks for INCORRECT / asks for CORRECT]
FINAL LETTER: [the letter whose verdict matches the polarity]
```

Rules:

- For negation-framed questions ('does NOT apply', 'pick the incorrect', 'wrong description'): the FINAL LETTER is the one marked ABSENT (or the most specifically wrong if multiple are absent).
- For affirmative questions ('which is correct', 'accurately describes', 'right description'): the FINAL LETTER is the one marked APPLIES.
- If ALL options are absent (full domain mismatch): rank by specificity of false claim — the option making the most concrete/specific false assertion is 'most incorrect'; that letter is FINAL LETTER.
- The answer field must contain exactly the FINAL LETTER from this table. No other letter may appear in the answer field.
- Drop one note on the image for each option (at the region where the object was sought — even if absent, mark the empty region). Note ordinal 1=A, 2=B, 3=C, 4=D. The canvas will color each note ring distinctly so you can visually confirm letter-to-region mapping.

Step 1 — Initial full read

Attach and read image 0. Identify:

- How many panels/sub-images are present
- Question type: spatial-relation | outlier | caption-match | scene-description | left-right | mismatch-detection
- Whether text in the image is legible at full scale
- Whether any embedded screen/monitor/framed photo contains the relevant content (screen-within-screen pattern)

Step 2 — Decide if one pass is enough

Answer immediately (after the verdict table) when:

- The scene content is unambiguous at full scale AND the noun-grounding pre-pass resolves the question
- The answer options are purely verbal and the image content is already understood

Do NOT answer immediately when:

- Slide or panel text is small and needs reading
- Two panels look similar and fine detail distinguishes them
- A spatial relationship depends on object overlap or occlusion
- A TV/monitor/framed image is embedded and may contain the described content

Step 3 — Zoom protocol (when needed)

For **outlier / slide questions**: zoom each panel individually to read titles, headers, colour schemes.

For **caption-match**: zoom embedded text and the dominant subject.

For **left-right / spatial**: zoom the region of interest. Place a note at the key anchor before answering.

For **screen-within-screen**: detect bright rectangular inset (TV, monitor, framed photo). Zoom to that inset at the tightest crop that captures the full screen content. Treat the inset as a separate visual context and re-run the noun-grounding pre-pass on it.

For **mismatch / all-absent scenarios**: zoom the most ambiguous region if any option's objects might be present but small. If after zooming objects are still absent, proceed to Step 0-V rule for all-absent.

Step 4 — Note before answering

Drop one note per option on the image at the region where that option's key object was found or sought. Note ordinal corresponds to option letter (1st note = A, 2nd note = B, 3rd = C, 4th = D). For absent objects, drop the note at the region where the object would be if it existed. The canvas colors these rings distinctly so you can visually confirm your letter mapping.

Step 5 — Answer

Read your VERDICT TABLE. Copy the FINAL LETTER to the answer field. That letter and only that letter goes in the answer field.

State the single letter only (or the exact number of letters the question format requires — usually one). Never answer 'Left' or 'Right' as a standalone string. Never list all four option letters unless the question explicitly asks for all that apply.

Anti-patterns to avoid

- Do NOT answer ABCD when only one answer is required, even if all options describe absent objects — select the single most specifically incorrect option.
- Do NOT skip the noun-grounding pre-pass (Step 0) or the verdict table (Step 0-V).
- Do NOT claim options are missing — they are always in the question text.
- Do NOT answer 'Left' or 'Right' if the options are A/B/C/D.
- Do NOT skip zooming on slide/panel grids or embedded screens.
- Do NOT issue more zoom steps than needed once the answer is clear.
- Do NOT let a letter appear in the answer field before the verdict table is complete.
- Do NOT select a letter based on what 'sounds plausible in the real world' when the image shows a completely different scene — always rank by visual contradiction to the actual scene.

I.2 VISUAL CONFIGURATION: VIEW.YAML

Rendering parameters

```
allow_upscale: false
crop_long_edge_px: 1024
crop_margin: 0.02
grid: null
max_crops_per_step: 2
min_crop_frac: 0.0
overview_long_edge_px: 1024
pixel_expr: ''
select_expr: ''
union_crop_for_multi_target: true
view_ops: []
zoom_gate_expr: zoom_gain > 1.05 or region_frac < 0.12
```

Canvas expression: `canvas_expr`

```
opt_a_r = where(nearest_idx == 1, clip(red * 0.4 + 220, 0, 255), red)
opt_a_g = where(nearest_idx == 1, clip(green * 0.4 + 40, 0, 255), green)
opt_a_b = where(nearest_idx == 1, clip(blue * 0.4 + 40, 0, 255), blue)
opt_b_r = where(nearest_idx == 2, clip(opt_a_r * 0.4 + 20, 0, 255), opt_a_r)
opt_b_g = where(nearest_idx == 2, clip(opt_a_g * 0.4 + 180, 0, 255), opt_a_g)
opt_b_b = where(nearest_idx == 2, clip(opt_a_b * 0.4 + 180, 0, 255), opt_a_b)
opt_c_r = where(nearest_idx == 3, clip(opt_b_r * 0.4 + 40, 0, 255), opt_b_r)
opt_c_g = where(nearest_idx == 3, clip(opt_b_g * 0.4 + 80, 0, 255), opt_b_g)
opt_c_b = where(nearest_idx == 3, clip(opt_b_b * 0.4 + 220, 0, 255), opt_b_b)
opt_d_r = where(nearest_idx == 4, clip(opt_c_r * 0.4 + 220, 0, 255), opt_c_r)
opt_d_g = where(nearest_idx == 4, clip(opt_c_g * 0.4 + 200, 0, 255), opt_c_g)
opt_d_b = where(nearest_idx == 4, clip(opt_c_b * 0.4 + 20, 0, 255), opt_c_b)
ring_mask = where(dist < 0.055, where(dist > 0.030, 1, 0), 0)
base_r = where(ring_mask > 0, opt_d_r, where(dist < 0.030, clip(red * 0.5, 0, 255),
red))
base_g = where(ring_mask > 0, opt_d_g, where(dist < 0.030, clip(green * 0.5, 0,
255), green))
```

```
base_b = where(ring_mask > 0, opt_d_b, where(dist < 0.030, clip(blue * 0.5, 0, 255),
blue))
rgb(base_r, base_g, base_b)
```

I.3 INTERFACE: LAYOUT.YAML

Request and history parameters

```
annotate_overview_boxes: false
context_ops: []
evict_policy: fifo
history_keep_last_n: 3
keep_overview: true
layout_expr: ''
max_images_per_request: 6
order: overview_first
```

View contract: `view_contract`

Image 0 is the single composite source image containing all panels, sub-images, slides, or scene content referenced by the question. Notes burned into the image appear as colored rings: note 1 (option A) = red/orange ring, note 2 (option B) = teal/cyan ring, note 3 (option C) = blue ring, note 4 (option D) = yellow ring. Each ring marks the region where that option's key object was sought — either found there, or the empty region where it would be if it existed. The ring color encodes option letter identity so the verdict table letter-to-region mapping can be visually confirmed.

Answer contract: `answer_contract`

A single uppercase option letter (e.g. A, B, C, or D). The answer field must contain ONLY this one letter — no other letters, no punctuation, no words. The letter must be copied from the FINAL LETTER line of the VERDICT TABLE written in the note field. For negation-framed questions the FINAL LETTER is the option marked ABSENT/INCORRECT. For affirmative questions the FINAL LETTER is the option marked APPLIES. Place all reasoning, the full verdict table, and evidence summaries in the note field only.

Protocol contract: `protocol_contract`

You are a visual reasoning agent. Each step you output exactly ONE JSON action block.

Actions

- `zoom`: {"action":"zoom","regions":[{"x0":...,"y0":...,"x1":...,"y1":...}],"image":N,"note":"why"} — coordinates are plain decimals in [0,1] in image N's frame, x0<x1, y0<y1. Up to 2 regions per step. Zooming re-reads original file so real detail is recovered. Zooming inside a zoom is allowed.
- `note`: {"action":"note","notes":[{"at":[x,y],"text":t,"image":N}]} — persistent marker visible in later views and zooms.
- `view`: {"action":"view","units":[i,...]} — attach additional source units (pool size: {pool_size}).
- `compose`: {"action":"compose","units":[...],"cols":c} — tile units into a labelled comparison sheet.
- `answer`: {"action":"answer","answer":"LETTER","note":"what you saw"}

Domain guidance

1. Read the full image first. Identify panel count and question type before doing anything else.
2. Answer in one step if the content is unambiguous. Zoom when slide text, panel labels, or fine scene detail is needed.
3. The answer options are in the question text — never say they are missing.
4. Always map the answer to the option LETTER (A/B/C/D). Never answer 'Left' or 'Right' as a standalone string.
5. Drop a `note` on the deciding evidence region before issuing `answer`.
6. Zoom budget: use only as many crops as needed. {max_crops} crops maximum.

{ans_header} Example: {ans_example}