STATEEDIT-BENCH: FROM LOCAL EDITS TO STATEFUL PRESENTATION WORKFLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Presentation editing requires interpreting later requests in light of previously established objects, decisions, constraints, and edit boundaries. Yet existing evaluations primarily measure fulfillment of the current request or properties of the final output, leaving the preservation and selective revision of accepted state weakly observed. We introduce STATEEDIT-BENCH, a 126-unit benchmark suite with 218 goal instances on 58 distinct input decks: 70 single-turn local edits, 10 continuation or template extensions, and 46 three-goal stateful sessions (138 sequential goals). Each stateful goal advances the committed artifact and is evaluated with an edit contract covering current postconditions, retained constraints, artifact validity, and authorization-aware preservation. We also design SE-PPT-EDIT, a presentation-editing skill that combines explicit state tracking, scope-aware planning, and artifact verification. In paired no-skill/full-skill runs across three models, the equal-model descriptive macro changes from 69.8% to 75.8% ($+6.0$ pp), with the largest gain on continuation and template filling, followed by local editing. On stateful multi-turn sessions, however, the macro changes by only $+2.2$ pp, and the model-level changes are not statistically significant. These findings position StateEdit-Bench as a diagnostic benchmark: it exposes contract-state failures that current-request evaluation can miss, while showing that the evaluated procedural deployments do not yet reliably improve multi-turn state maintenance.

1 INTRODUCTION

Existing evaluations of presentation editing commonly assess whether the output satisfies the current instruction (Guo et al., 2024; Ofengenden et al., 2026; Gandhi et al., 2026). This criterion is incomplete for interactive editing. After a user identifies a chart, accepts a visual treatment, and later revises one property, a correct agent must recover the same chart, preserve accepted decisions, modify only the newly authorized property, and continue from the committed deck. A plausible current edit can therefore fail by reverting an earlier decision or changing protected content (Figure 1).

We study this problem as *stateful presentation editing*. Multi-turn interaction is not itself evidence of stateful evaluation: consecutive requests may remain independently solvable. A goal is stateful only when at least one condition required for correctness was established before the current request and is not restated.

This measurement problem imposes three design requirements. (R1) An explicit edit contract must track new, retained, revised, and revoked constraints and authorized scope. (R2) A cumulative native-artifact chain must pass each committed output to the next goal without replacing failed intermediate artifacts with oracle states. (R3) Conjunctive evaluation must require current fulfillment, artifact validity, retained constraints, and authorization-aware preservation jointly. These requirements distinguish stateful evaluation from turn count or final-output quality alone: the evaluator must check what should survive and when a constraint is lost.

We introduce STATEEDIT-BENCH, a 126-unit benchmark suite with 218 goal instances: 70 single-turn local-edit tasks, 10 continuation or template-extension tasks, and 46 three-goal stateful sessions, using 58 distinct input decks in total. Analysis of real-world customer usage data identified two needs that motivate the stateful and extension strata: executing edits that depend on prior interaction history and extending existing presentations. Five continuation tasks require agents to extend an existing deck while preserving its original slides and maintaining narrative and visual continuity.

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

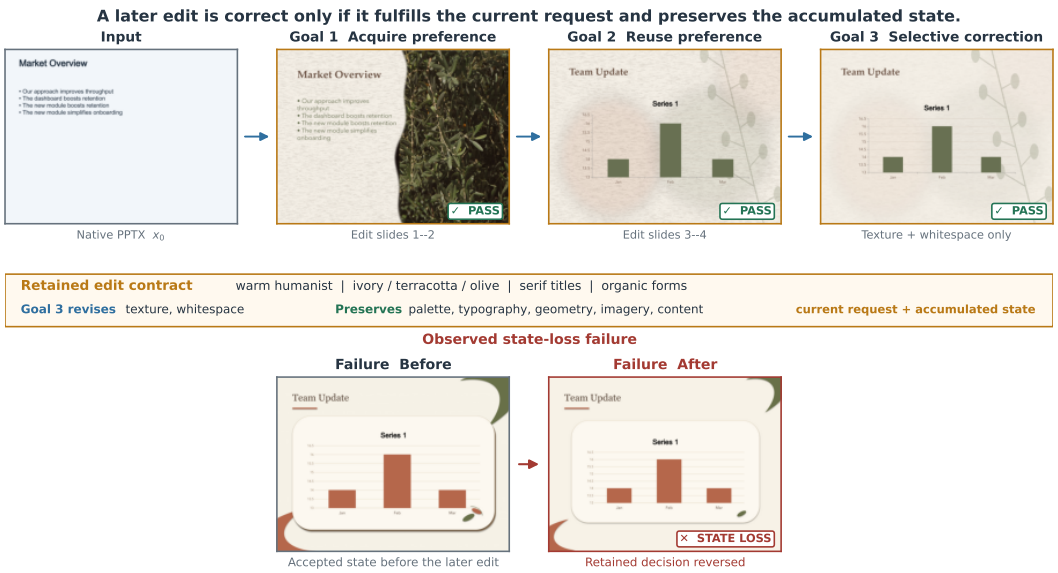

Figure 1: **A StateEdit-Bench session grounds later requests in an evolving native artifact and edit contract.** Goal 1 establishes a visual preference and limits the edit to slides 1–2; Goal 2 reuses the accepted preference on slides 3–4 while preserving prior edits; Goal 3 changes only texture and whitespace. The audited failure illustrates why per-turn fulfillment is insufficient: a plausible current edit can still lose a retained decision. Although Goal 3 requests changes only to texture and whitespace, the agent also removes the terracotta leaf and curved stem from the previously established botanical motif at the lower right. This changes retained decorative geometry rather than only the requested visual dimensions.

Five template-grounded composition tasks require agents to organize supplied articles into complete presentations by selecting and adapting existing template layouts, preserving their design language, and removing placeholder content. These extensions test content synthesis under existing-artifact constraints rather than generation from a blank deck. The stateful sessions connect three goals: establishing an editing decision, reusing it in a history-dependent follow-up, and selectively revising one aspect while retaining the others. Agents must resolve references to earlier targets or preferences and continue editing the actual preceding output. Authored contracts and transition-level checks operationalize R1–R3: a correct final artifact cannot erase an earlier failed transition.

We design SE-PPT-EDIT, a presentation-editing skill that combines explicit state tracking, scope-aware planning, and artifact verification, and use it as a diagnostic deployment of procedural knowledge. Paired no-skill/full-skill runs share the model, task, source, simulated-user policy, harness, and evaluator. Aggregate local and continuation gains coexist with small, statistically non-significant multi-turn changes. A masked failure audit examines the errors behind these scores.

Our contributions are threefold. First, we formalize stateful artifact editing as history-conditioned correctness over an evolving edit contract and introduce a 126-unit benchmark suite spanning local edits, deck continuation, template-grounded composition, and a 46-session stateful diagnostic core. Second, we develop a multi-evidence evaluation framework that jointly measures current fulfillment, retained constraints, native-artifact validity, and authorization-aware preservation. Third, we design SE-PPT-Edit and use the skill as a diagnostic stress test of the benchmark across three models. The study characterizes model-specific responses across the three task strata. Upon acceptance, we will publicly release the benchmark, SE-PPT-Edit, evaluators, configurations, and legally redistributable traces.

2 RELATED WORK

Presentation generation, editing, and evaluation. Research on automatic presentations spans document summarization and paper-to-slide conversion (Sun et al., 2021; Sefid et al., 2021; Bandy-

opadhyay et al., 2024; Maheshwari et al., 2024; Mondal et al., 2024), as well as multimodal, code-generating, and agentic systems that produce editable artifacts (Ge et al., 2025; Zheng et al., 2025; Xu et al., 2025; Tang et al., 2025; Liang et al., 2025; Xie et al., 2025; Zeng et al., 2025; Ozden et al., 2026; Zheng et al., 2026). TSBench evaluates instruction-driven slide editing (Jung et al., 2026), while PPTC includes multi-turn task completion (Guo et al., 2024). Other benchmarks cover reference-image reconstruction, computer-use workflows, and dynamic template updates (Ofengenden et al., 2026; Gandhi et al., 2026; Zhou et al., 2026). MemSlides combines personalized generation with scope-aware multi-turn local revision (Jin et al., 2026). StateEdit-Bench complements this literature by making history-conditioned correctness the evaluated construct: later requests require recovering accepted decisions, resolving references against an evolving native artifact, and preserving properties outside the authorized scope.

Structured visual and layout generation. Presentation quality depends on the joint organization of text, images, and geometry. Recent layout methods model controllable generation with discrete or continuous diffusion, serialized constraints, language or code prompting, and explicit text-to-layout parsing (Inoue et al., 2023; Chai et al., 2023; Hui et al., 2023; Jiang et al., 2023; Lin et al., 2023b;a; Zhang et al., 2023; Tang et al., 2023). Content-aware poster and design template systems additionally evaluate visual-textual placement and theme consistency (Hsu et al., 2023; Weng et al., 2024; Shabani et al., 2024). These works motivate our open-ended visual criteria, but layout quality alone cannot determine whether an edit preserves earlier user decisions or leaves protected slide content unchanged.

Interactive agent benchmarks. General agent evaluations measure tool use, multi-step reasoning, and task outcomes (Liu et al., 2024; Mialon et al., 2024; Wang et al., 2024a). Web and GUI benchmarks extend this setting to realistic sites, mobile devices, and desktop applications (Deng et al., 2023; Zhou et al., 2024; Koh et al., 2024; Drouin et al., 2024; Xie et al., 2024; Lin et al., 2024; Cao et al., 2024; Rawles et al., 2025). These benchmarks test planning, action grounding, and execution in interactive environments. We focus on artifact transitions whose validity depends on interaction history. Turn count alone is insufficient: consecutive commands may remain independently solvable, whereas a stateful goal changes truth value when retained preferences, antecedents, or edit boundaries are removed from the history.

Procedural knowledge, memory, and judge-based evaluation. Reasoning-and-acting methods, self-refinement, verbal reinforcement, and learned tool use show how external procedures can improve execution without changing a task definition (Yao et al., 2023; Madaan et al., 2023; Shinn et al., 2023; Schick et al., 2023; Wang et al., 2023). Agent memory systems similarly retain experience or workflows across decisions (Park et al., 2023; Packer et al., 2023; Wang et al., 2024b). SE-PPT-Edit adopts this procedural perspective to study model-specific editing behavior (Ling et al., 2026; Hsiao et al., 2025). Model-based judges provide scalable semantic and visual assessment (Liu et al., 2023; Kim et al., 2024; Zhu et al., 2023; Chen et al., 2024), but also exhibit calibration and self-preference risks (Zheng et al., 2023; Chen et al., 2024; Panickssery et al., 2024). We therefore combine judge evidence with native postconditions, artifact validity, preservation checks, and a limited human-agreement audit rather than treating a single judge score as ground truth.

3 StateEdit-Bench

3.1 Task formulation and construction

An editing session begins with a native deck x_0. At goal t, the agent receives request q_t, the previously committed artifact x_{t-1}, and the observable history h_{t-1}, then produces a response and candidate artifact. A goal is stateful when at least one condition required for correctness was established before q_t and is not restated. Each evaluator-side edit contract tracks new, retained, revised, and revoked constraints together with authorized scope:

$$\mathcal{C}_t = (\mathcal{N}_t, \mathcal{K}_t, \mathcal{U}_t, \mathcal{D}_t, \Omega_t), \tag{1}$$

where the sets encode new, retained, revised, and revoked constraints and Ω_t is the mutable scope. Benchmark authors construct contracts from requests, resolved references, and previously accepted decisions, specifying Ω_t at slide, object, and property-family granularity. Properties outside this scope are protected by default. A transition succeeds only when the current request is fulfilled, active

Table 1: Full benchmark composition. Byte-identical input decks are counted once; cross-stratum reuse makes the total smaller than the sum. Stateful units have three goals; the other units have one.

Stratum	Units	Goal instances	Input decks	History role
Local editing	70	70	36	independent
Continuation/template extension	10	10	10	independent
Stateful T1–T4 and TC	46	138	41	dependent
Total	126	218	58	80 independent / 46 dependent

Table 2: Editing-task families across the complete benchmark. Local and extension counts are tasks; multi-turn counts are three-goal sessions, giving 126 evaluation units in total.

Family	Definition	Local	Extensions	Multi-turn
T1	Refinement of existing element properties	11	0	6
T2	Content replacement and structural modification	19	0	10
T3	Slide-level and cross-slide layout or style adjustment	16	0	10
T4	Open-ended visual editing and source-conditioned composition	14	10	15
TC	Multiple non-conflicting edits within one request	10	0	5

constraints remain true, the artifact is valid, and mutations remain within scope. The supplementary material gives the complete formalization. All source PPTX files were drawn from de-identified customer usage data. We screened more than 3,000 customer editing queries to curate the benchmark tasks.

The complete benchmark contains 126 evaluation units and 218 goal instances: 70 single-turn local-edit tasks, 10 continuation or template-extension tasks, and 46 stateful sessions with 138 sequential goals over 41 source decks. The stateful set is one stratum, not the full benchmark: all three strata together use 58 distinct input PPTX files after deduplicating byte-identical inputs. Each multi-turn session has three goals—establish or clarify, refer or reuse, and selectively revise—and preserves the actual artifact chain $x_0 \to x_1 \to x_2 \to x_3$. Failed intermediate artifacts are never replaced by oracle states. Native decks cover editable text, tables, charts, and mixed layouts.

The extension stratum comprises five deck-continuation tasks and five template-grounded composition tasks. Continuation preserves an existing three-slide prefix and appends two slides grounded in supplied materials, requiring narrative progression, factual fidelity, and visual continuity. Template-grounded composition converts a supplied article into a seven- or eight-slide presentation by selecting, duplicating, removing, and reordering candidate template slides. It requires preserving the template design, removing sample content, and avoiding unsupported facts or reconstruction from blank slides. Each extension is one goal with all requirements stated in the request and supplied artifacts; unlike the stateful sessions, it does not require recovering unstated constraints from earlier dialogue.

Tasks were available during benchmark, evaluator, or skill development; pretraining overlap is unverified. Public distribution requires source-level license review. The supplement details provenance and authored examples.

Table 2 defines the editing task families. Interaction uses a deterministic user policy: the first relevant clarification receives an authored slot answer, whereas repeated or irrelevant questions receive fixed responses without additional task information. Agents choose whether and when to clarify. Responses and artifact transitions are recorded throughout the session.

3.2 Capability coverage

We annotate 126 StateEdit evaluation units and all released units from TSBench (Jung et al., 2026), PPTC, and MemSlides under one requirement–support codebook. MemSlides releases runtime code, but the inspected experiment package provides only a one-slide generation smoke test and a three-round local-revision example, rather than the complete evaluation set reported in the paper. Its coverage counts therefore characterize these two public examples. A unit provides valid support

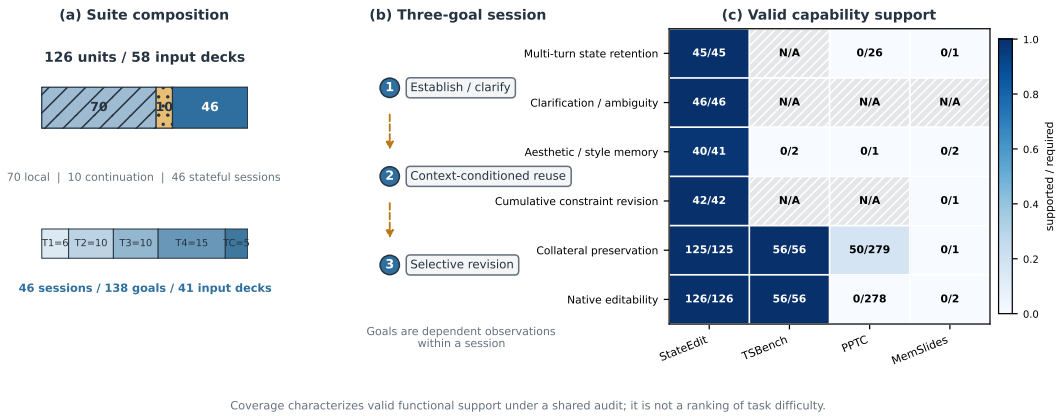

Figure 2: **StateEdit-Bench combines broad editing competence with a stateful diagnostic core.** The suite contains 70 single-turn local edits, 10 continuation or template extensions, and 46 three-goal stateful sessions (126 units and 218 goal instances in total) using 58 distinct input decks; 41 serve the stateful subset. Under a shared requirement–support audit, StateEdit directly supports clarification, retained state, aesthetic memory, cumulative revision, preservation, and native editability. High StateEdit coverage reflects the benchmark design objective.

Table 3: Presentation-editing benchmark design. Design comparison only; released-package verifiability is reported separately. Native denotes a cumulative native artifact chain, MT denotes multi-turn interaction, History denotes history-conditioned correctness, and Scope denotes evaluation of mutations outside the authorized region. Y indicates explicit support, P partial or indirect support, and – no explicit evaluation target.

Benchmark	Native	MT	History	Scope	Evidence
StateEdit-Bench	Y	Y	Y	Y	structure, artifact, visual
TSBench (Jung et al., 2026)	P	P	P	P	structure and visual
PPTC (Guo et al., 2024)	P	Y	P	P	attribute matching
MemSlides (Jin et al., 2026)	P	Y	Y	P	task and visual

when the task requires a capability and the released package contains the source, history, artifact, and evaluation conditions needed to exercise it. StateEdit uses sessions as multi-turn units; the comparison benchmarks use their released task or session units. Figure 2 summarizes functional coverage, not benchmark difficulty; exact counts appear in the supplementary capability-coverage table.

StateEdit coverage is high by design. This descriptive released-package audit uses one retained annotation per unit and is not used as a primary quality or difficulty ranking. Two annotators divide the units under a shared codebook; the labels are descriptive rather than adjudicated gold. Separately, two masked annotators label the same failed trajectories; Table 7 reports pre-adjudication agreement. For PPTC, 26 released task definitions require retained state and 278 released task definitions require native editable outputs, but the referenced per-turn base and label files are not distributed. They therefore yield 0/26 and 0/278 valid support, respectively, under the released-artifact protocol; this is a statement about verifiability from the release, not the intended scope of PPTC. Unit definitions, the codebook, and item-level annotations are documented in the supplement.

Table 3 compares explicit evaluation targets; Figure 2 audits whether released materials support capability verification. These distinguish design scope from release verifiability. StateEdit-Bench makes the joint enforcement of R1–R3 explicit through authored contracts and transition-level checks.

MemSlides explicitly constrains edit scope and qualitatively compares changes to non-target regions. We record partial scope support to distinguish this evidence from a conjunctive benchmark-level preservation endpoint; the released-package audit remains a separate assessment.

The benchmark spans 80 history-independent units and 46 history-dependent units, including five contextual-continuation and 41 cumulative-revision units. These strata support capability and failure-mode analysis but do not compare semantically identical requests or estimate a numerical state-recovery cost.

3.3 Evaluation

The central design choice is conjunctive success (R3): fulfilling the latest request cannot compensate for losing an accepted decision or exceeding the authorized scope. With current fulfillment F_t, validity V_t, locality L_t, and retained constraints $\mathcal{K}_t$, we compute

$$S_t = F_t V_t L_t \prod_{c \in \mathcal{K}_t} \mathbb{1}[c(x_t)], \qquad S_{\text{session}} = \prod_{t=1}^{T} S_t. \tag{2}$$

The contract (R1) specifies what must survive, and the artifact chain (R2) exposes when it is lost. Together, these checks make supported failures of history-conditioned correctness observable at each transition, even if a later edit repairs the final file. Refusals, timeouts, missing outputs, corrupt files, and agent-side protocol failures remain adverse outcomes. For example, when a request authorizes a color change but not a font size change, an output that performs both mutations can satisfy the current request but fails authorization-aware preservation. Deterministic checks cover supported text, data, geometry, style, chart, table, and object-scope properties. Visual rubrics cover 20 local tasks, all 10 extensions, and 15 stateful sessions: 45/126 units and 75/218 goals. Aesthetic preference scores are auxiliary visual evidence, not terms in strict success. However, VLM-based validity gates enter extension success, and validity, persona adherence, and selective-update judgments enter success for five aesthetic-memory sessions. These judge-dependent outcomes cannot override failed structural checks. For other visual tasks, strict success does not establish aesthetic quality. The supplement separates rubric eligibility, score aggregation, missing-evidence handling, validator probes, and the limited human–VLM agreement audit.

4 Experimental Setup

We use paired deployments to compare local, extension, and multi-turn performance, then analyze failure types, endpoint changes, and external TSBench results. Human audits assess failure attribution and visual-judge agreement.

4.1 Procedural Diagnostic Deployment: SE-PPT-Edit

StateEdit-Bench defines stateful editing independently of agent architecture. We design SE-PPT-Edit, a presentation-editing skill, and use it as a diagnostic deployment for testing whether externalized procedural knowledge changes editing behavior without changing model parameters. The package combines model-visible instructions, structured records, deterministic file helpers, verification rules, and visual-editing guidance; it contains no held-out target artifacts or benchmark solutions.

The shared procedural core has three components. *State externalization* records the committed artifact, bindings, decisions, preferences, and scope in a compact ledger. An intent router selects editing or clarification workflows using this record.

Scope-constrained planning limits mutable slides, objects, and property families before execution. Open-ended requests become visual specifications; selective correction marks only the requested dimension as mutable.

Verify-and-commit execution keeps the source immutable while producing a candidate and checking postconditions, scope, preservation, and validity. Failed candidates are rolled back; open-ended edits permit one focused render-review correction. The supplementary material gives the full protocol.

All deployments share state externalization, artifact-chain reuse, scope control, and candidate verification. Model-facing guidance is adapted to tool-use behavior; package hashes identify the exact deployment used in each reported arm.

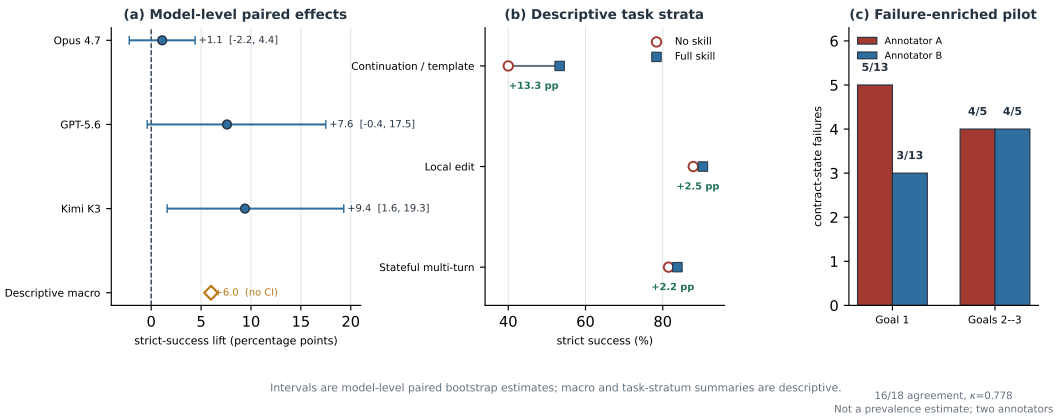

Figure 3: (a) Model-level changes in strict success, measured as full-skill minus no-skill in percentage points. The three blue circles show the point estimates for Opus, GPT, and Kimi, respectively, with equal weight assigned to local editing, continuation/template filling, and multi-turn tasks within each model. Horizontal error bars indicate 95% paired-bootstrap confidence intervals, and the vertical dashed line marks zero change. The hollow orange diamond shows the descriptive equal-model average across the three deployments. (b) Equal-model success rates by task type. Continuation and template filling show the largest gain (+13.3 percentage points), followed by local editing (+2.5); the multi-turn change is smaller (+2.2), with no statistically significant model-level improvement. (c) Two annotators independently classify 18 selected failed trajectories by the goal at which failure first appears. Contract-state failures include lost bindings or references, reversed retained decisions, and scope or preservation violations. Of 13 Goal-1 failures, annotators A and B assign 5 and 3 to this category; each assigns 4 of the 5 later-goal failures to it. Bar heights are counts, and labels show counts over the audited failures in each group, not benchmark-wide failure rates.

We evaluate Claude Opus 4.7, GPT-5.6-sol, and Kimi K3 through the same Hermes harness. Within each model pair, we fix the task, source deck, simulated user, terminal/file/code-execution tools, tool-turn and timeout limits, retry policy, renderer, and evaluator. The full-skill arm additionally loads the frozen instructions, workflows, and deterministic local helper; the helper performs no model calls or external retrieval. Refusals, timeouts, missing files, invalid artifacts, and post-initiation protocol failures remain adverse outcomes.

The reported conditions use frozen, versioned, model-adaptive deployments of a shared procedural core rather than byte-identical packages. We interpret within-model differences as paired deployment contrasts and use them as the primary evidence. The equal-model macro summarizes the aggregate pattern; neither it nor the bundled comparison identifies a uniform or component-level causal effect. Package hashes and run provenance identify the evaluated deployments.

The assigned population consists of 70 local edits, 10 continuation or template-filling tasks, and 46 three-goal sessions. Headline values are equal-model macros over within-model rates. Multi-turn inference uses paired session outcomes and exact McNemar tests. Multi-turn effect intervals use 100,000 basic paired-bootstrap resamples of sessions with fixed seed 2027; other model-level intervals use the frozen source-cluster bootstrap. At the maximum multi-turn sample of 46, the prospective 80%-power minimum detectable effect is 18.3–29.8 points across the reported discordance assumptions; the smaller GPT and Kimi samples are no more sensitive. Complete harness fields, estimands, and power calculations appear in the supplementary material.

5 Results and Analysis

5.1 Paired deployment results

Figure 3 reports model-level contrasts and the task-stratum macro. Overall strict success changes from 69.8% to 75.8% (+6.0 pp). Effects are heterogeneous: Kimi changes by +9.4 pp, GPT by +7.6 pp, and Opus by +1.1 pp. With the skill, GPT leads on local editing and multi-turn success, whereas

Kimi leads on extensions. This variation reveals distinct editing capabilities that a single aggregate score cannot fully characterize. The strongest aggregate change occurs in continuation and template filling (+13.3 pp), followed by local edits (+2.5 pp); the multi-turn macro changes by +2.2 pp.

Opus changes by −1.4 pp on local editing and by +4.3 pp on multi-turn sessions. Excluding Opus, the descriptive overall lift is +8.6 pp, whereas the multi-turn lift remains only +1.2 pp. Thus, the aggregate improvement is not driven by the Opus deployment, and the separation between aggregate and multi-turn gains persists in this sensitivity analysis.

Table 4: Strict success (%) without/with the skill. Local edit denotes single-turn edits; Extensions denotes deck continuation and template-based content filling; Multi-turn denotes sessions in which all three goals succeed. Equal-stratum macro is the unweighted mean of these three success rates within each model.

Model	Local edit	Extensions	Multi-turn	Equal-stratum macro
Opus 4.7	87.1/85.7	40.0/40.0	78.3/82.6	68.3/69.4
GPT-5.6	89.9/92.8	30.0/50.0	87.8/87.8	69.2/76.9
Kimi K3	86.8/92.6	50.0/70.0	78.6/81.0	71.8/81.2
Equal-model macro	87.9/90.4	40.0/53.3	81.5/83.8	69.8/75.8
Sensitivity excl. Opus	88.4/92.7	40.0/60.0	83.2/84.4	70.5/79.1

Table 5: Endpoint decomposition as descriptive equal-model macro averages.

Endpoint	No skill	Full skill	Effect
Artifact delivery	98.8%	98.9%	+0.1 pp
Postcondition attainment	89.3%	92.4%	+3.1 pp
Authorization-aware preservation	95.6%	98.2%	+2.6 pp

5.2 Stateful Multi-turn Results

Table 6: Model-level paired inference on multi-turn sessions. Success rates are percentages; lift and its 95% paired-bootstrap CI are in percentage points. W/L counts sessions that change from failure to success or success to failure after adding the skill. p is from the exact McNemar test.

Model	No skill	Full skill	Lift	95% CI	W/L	Exact p
Opus 4.7	78.3	82.6	+4.3	[−4.3, +13.0] pp	3/1	0.625
GPT-5.6	87.8	87.8	0.0	[−7.3, +7.3] pp	1/1	1.000
Kimi K3	78.6	81.0	+2.4	[−2.4, +4.8] pp	1/0	1.000

Multi-turn strict success changes by +4.3 pp for Opus (3 wins, 1 loss), 0.0 pp for GPT (1 win, 1 loss), and +2.4 pp for Kimi (1 win, 0 losses); exact paired p-values are 0.625, 1.000, and 1.000. The tests contain only 4, 2, and 1 discordant pairs, respectively. All 95% intervals include zero, and the prospective MDE is much larger than the observed effects. The tests therefore neither establish improvement nor equivalence; the design is powered only for large paired effects under the stated discordance assumptions. These results do not establish that the observed local and continuation gains extend to multi-turn state maintenance. The separate trajectory audit below examines the contract-state failures underlying unsuccessful sessions.

5.3 Failure-Mode Analysis

To distinguish current-edit execution errors from violations of bindings, retained decisions, or protected scope, two masked annotators examine 18 failed trajectories from 12 sessions. We group binding/reference loss, retained-decision reversal, and scope/preservation violation into a contract-state category. Both annotators assign four of five later-goal failures to this category, illustrating failures that current-request fulfillment alone cannot characterize. Agreement on category membership is 16/18 ($\kappa = 0.778$; 95% bootstrap CI [0.471, 1.000]; Table 7). Scope violations can also occur at the

first goal, so the category is broader than history-induced failure. This selected sample characterizes observed failure modes rather than their prevalence; the supplementary material reports the complete category cross-tabulation and direct state-relatedness judgments.

Table 7: Agreement between two independent annotators on 18 failed trajectories. Contract-state failures group reference/binding loss, retained-decision reversal, and scope/preservation violations. The last row reports how many of the five later-goal failures each annotator assigns to this group, not inter-annotator agreement.

Field	Usable n	Exact agreement	Cohen κ
Primary failure category	18	12/18	0.509
First observable turn	18	16/18	0.745
Taxonomy-derived contract-state	18	16/18	0.778
Later-goal contract-state	5	A: 4/5; B: 4/5	descriptive

5.4 Endpoint trade-offs and evaluator evidence

Postcondition attainment changes from 89.3% to 92.4%, authorization-aware preservation from 95.6% to 98.2%, and artifact delivery from 98.8% to 98.9%. Procedural augmentation therefore changes several endpoints rather than one quality dimension.

To examine deployment behavior beyond StateEdit-Bench, we evaluate Kimi-K3 on 56 paired TSBench units. Recorded artifact delivery, postcondition completion, and preservation each increase from 54/56 without the skill to 56/56 with it. Complete external results, calibration evidence, and validator probes appear in the supplement.

5.5 Limitations

StateEdit-Bench is designed for controlled diagnosis of editing capabilities. Its authored tasks and source decks do not constitute a representative sample of presentation workflows. The current strata do not estimate state-recovery cost for semantically identical requests. The failure audit is small and visual calibration and validator coverage remain limited. Component ablations and broader model coverage are needed to explain the deployment differences. Source-license review and development exposure are documented in the supplementary data statement.

6 Conclusion

StateEdit-Bench brings local editing, deck continuation, template-grounded composition, and stateful interaction into a 126-unit suite with 218 goal instances. Explicit edit contracts, cumulative native-artifact chains, and multi-evidence evaluation make history-conditioned constraint maintenance an observable requirement alongside current-request fulfillment. Grounded in needs identified from real-world customer usage data, history-dependent editing and presentation extension broaden the evaluation beyond isolated edits and expose task-specific differences among agents. We also design SE-PPT-Edit, a skill for state tracking, scope-aware planning, and artifact verification. Paired deployments across three models improve local editing and continuation/template-filling scores in aggregate. Together, the benchmark, evaluation framework, and skill provide a reproducible basis for studying both content transformation and the preservation of accepted state throughout presentation editing.

AI use statement

Generative AI tools assisted with language polishing and translation. The authors reviewed the AI-assisted materials and take responsibility for the final manuscript, claims, and artifacts.

Reproducibility statement

Sections 3–5 define contracts, interaction protocols, metrics, and session-level analysis. The supplementary material details task construction, harness settings, power analysis, and validator checks. We

will publicly release StateEdit-Bench and the SE-PPT-Edit skill upon acceptance. The release will separately version tasks, evaluators, and skills, with run manifests, and license-cleared artifacts and traces. Failed and invalid trials remain in the declared denominator.

REFERENCES

Sambaran Bandyopadhyay, Himanshu Maheshwari, Anandhavelu Natarajan, and Apoorv Saxena. Enhancing presentation slide generation by LLMs with a multi-staged end-to-end approach. In *Proceedings of the 17th International Natural Language Generation Conference*, pp. 222–229, 2024. doi: 10.18653/v1/2024.inlg-main.18. URL https://aclanthology.org/2024.inlg-main.18/.

Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Yuchen Mao, Wenjing Hu, Tianbao Xie, Hongshen Xu, Danyang Zhang, Sida Wang, Ruoxi Sun, Pengcheng Yin, Caiming Xiong, Ansong Ni, Qian Liu, Victor Zhong, Lu Chen, Kai Yu, and Tao Yu. Spider2-V: How far are multimodal agents from automating data science and engineering workflows? In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shang Chai, Liansheng Zhuang, and Fengying Yan. LayoutDM: Transformer-based diffusion model for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18349–18358, 2023.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6562–6595, 2024. URL https://proceedings.mlr.press/v235/chen24h.html.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11642–11662, 2024. URL https://proceedings.mlr.press/v235/drouin24a.html.

Apurva Gandhi, Vishwas Suryanarayanan, Raja Hasnain Anwar, Firoz Shaik, Shubhang Desai, Thong Q. Nguyen, Muhammad Taqi Raza, Vishal Chowdhary, and Graham Neubig. PPT-Eval: A benchmark for computer-use agents on PowerPoint tasks. In *International Conference on Machine Learning*, 2026. URL https://microsoft.github.io/ppteval/.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. AutoPresent: Designing structured visuals from scratch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2902–2911, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ge_AutoPresent_Designing_Structured_Visuals_from_Scratch_CVPR_2025_paper.html.

Yiduo Guo, Zekai Zhang, Yaobo Liang, Dongyan Zhao, and Nan Duan. PPTC benchmark: Evaluating large language models for PowerPoint task completion. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8682–8701, 2024. doi: 10.18653/v1/2024.findings-acl.514. URL https://aclanthology.org/2024.findings-acl.514/.

Vincent Hsiao, Mark Roberts, and Leslie Smith. Procedural knowledge improves agentic LLM workflows. *arXiv preprint arXiv:2511.07568*, 2025. doi: 10.48550/arXiv.2511.07568. URL https://arxiv.org/abs/2511.07568.

Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. PosterLayout: A new benchmark and approach for content-aware visual-textual presentation layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6018–6026, 2023.

Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1942–1951, 2023.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10167–10176, 2023.

Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutFormer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18403–18412, 2023.

Ye Jin, Yangyang Xu, Jun Zhu, and Yibo Yang. MemSlides: A hierarchical memory driven agent framework for personalized slide generation with multi-turn local revision. *arXiv preprint arXiv:2606.17162*, 2026. doi: 10.48550/arXiv.2606.17162. URL https://arxiv.org/abs/2606.17162.

Kyudan Jung, Hojun Cho, Jooyeol Yun, Soyoung Yang, Jaehyeok Jang, and Jaegul Choo. Talk to your slides: High-efficiency slide editing via language-driven structured data manipulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3370–3399, 2026. doi: 10.18653/v1/2026.findings-acl.166. URL https://aclanthology.org/2026.findings-acl.166/.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

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

Xin Liang, Xiang Zhang, Yiwei Xu, Siqi Sun, and Chenyu You. SlideGen: Collaborative multimodal agents for scientific slide generation. *arXiv preprint arXiv:2512.04529*, 2025. URL https://arxiv.org/abs/2512.04529v1.

Jiawei Lin, Jiaqi Guo, Shizhao Sun, Weijiang Xu, Ting Liu, Jian-Guang Lou, and Dongmei Zhang. A parse-then-place approach for generating graphic layouts from textual descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23622–23631, 2023a.

Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutPrompter: Awaken the design ability of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/88a129e44f25a571ae8b838057c46855-Abstract-Conference.html.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Qinchen Wu, Mingyi Yan, Zhengyuan Yang, Lijuan Wang, and Mike Zheng Shou. VideoGUI: A benchmark for GUI automation from instructional videos. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

George Ling, Shanshan Zhong, and Richard Huang. Agent skills: A data-driven analysis of Claude skills for extending large language model functionality. *arXiv preprint arXiv:2602.08004*, 2026. doi: 10.48550/arXiv.2602.08004. URL https://arxiv.org/abs/2602.08004.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023. doi: 10.18653/v1/2023.emnlp-main.153.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023.

Himanshu Maheshwari, Sambaran Bandyopadhyay, Aparna Garimella, and Anandhavelu Natarajan. Presentations are not always linear! GNN meets LLM for text document-to-presentation transformation with attribution. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15948–15962, 2024. doi: 10.18653/v1/2024.findings-emnlp.936. URL https://aclanthology.org/2024.findings-emnlp.936/.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

Ishani Mondal, Shwetha S, Anandhavelu Natarajan, Aparna Garimella, Sambaran Bandyopadhyay, and Jordan Boyd-Graber. Presentations by the humans and for the humans: Harnessing LLMs for generating persona-aware slides from documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2664–2684, 2024. doi: 10.18653/v1/2024.eacl-long.163. URL https://aclanthology.org/2024.eacl-long.163/.

Michael Ofengenden, Yunze Man, Ziqi Pang, Liang-Yan Gui, and Yu-Xiong Wang. PPTArena: A benchmark for PowerPoint editing. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2512.03042.

Tarik Can Ozden, Sachidanand VS, Furkan Horoz, Ozgur Kara, Junho Kim, and James M. Rehg. Narrative-driven paper-to-slide generation via ArcDeck. *arXiv preprint arXiv:2604.11969*, 2026. URL https://arxiv.org/abs/2604.11969.

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.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2197. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7f1f0218e45f5414c79c0679633e47bc-Abstract-Conference.html.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Athar Sefid, Prasenjit Mitra, Jian Wu, and C. Lee Giles. Extractive research slide generation using windowed labeling ranking. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pp. 91–96, 2021. doi: 10.18653/v1/2021.sdp-1.11. URL https://aclanthology.org/2021.sdp-1.11/.

Mohammad Amin Shabani, Zhaowen Wang, Difan Liu, Nanxuan Zhao, Jimei Yang, and Yasutaka Furukawa. Visual layout composer: Image-vector dual diffusion model for design layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9222–9231, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. D2S: Document-to-slide generation via query-based text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1405–1418, 2021. doi: 10.18653/v1/2021.naacl-main.111. URL https://aclanthology.org/2021.naacl-main.111/.

Wenxin Tang, Jingyu Xiao, Wenxuan Jiang, Xi Xiao, Yuhang Wang, Xuxin Tang, Qing Li, Yuehe Ma, Junliang Liu, Shisong Tang, and Michael R. Lyu. SlideCoder: Layout-aware RAG-enhanced hierarchical slide generation from design. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9015–9039, 2025. doi: 10.18653/v1/2025.emnlp-main.458. URL https://aclanthology.org/2025.emnlp-main.458/.

Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. LayoutNUWA: Revealing the hidden layout expertise of large language models. *arXiv preprint arXiv:2309.09506*, 2023.

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. doi: 10.48550/arXiv.2305.16291. URL https://arxiv.org/abs/2305.16291.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. GTA: A benchmark for general tool agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 75749–75790, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8a75ee6d4b2eb0b777f549a32a5a5c28-Abstract-Datasets_and_Benchmarks_Track.html.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024b. doi: 10.48550/arXiv.2409.07429. URL https://arxiv.org/abs/2409.07429.

Haohan Weng, Danqing Huang, Yu Qiao, Zheng Hu, Chin-Yew Lin, Tong Zhang, and C. L. Philip Chen. Desigen: A pipeline for controllable design template generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12721–12732, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Weng_Desigen_A_Pipeline_for_Controllable_Design_Template_Generation_CVPR_2024_paper.html.

Eric Xie, Danielle Waterfield, Michael Kennedy, and Aidong Zhang. SlideBot: A multi-agent framework for generating informative, reliable, multi-modal presentations. *arXiv preprint arXiv:2511.09804*, 2025. URL https://arxiv.org/abs/2511.09804.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Xiaojie Xu, Xinli Xu, Sirui Chen, Haoyu Chen, Fan Zhang, and Ying-Cong Chen. PreGenie: An agentic framework for high-quality visual presentation generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 3045–3063, 2025. doi: 10.18653/v1/2025.findings-emnlp.165. URL https://aclanthology.org/2025.findings-emnlp.165/.

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

Wenzheng Zeng, Mingyu Ouyang, Langyuan Cui, and Hwee Tou Ng. SlideTailor: Personalized presentation slide generation for scientific papers. *arXiv preprint arXiv:2512.20292*, 2025. URL https://arxiv.org/abs/2512.20292.

Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. LayoutDiffusion: Improving graphic layout generation by discrete diffusion probabilistic models. *arXiv preprint arXiv:2303.11589*, 2023.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14402–14418, 2025. doi: 10.18653/v1/2025.emnlp-main.728. URL https://aclanthology.org/2025.emnlp-main.728/.

Hao Zheng, Guozhao Mo, Xinru Yan, Qianhao Yuan, Wenkai Zhang, Xuanang Chen, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. DeepPresenter: Environment-grounded reflection for agentic presentation generation. *arXiv preprint arXiv:2602.22839*, 2026. URL https://arxiv.org/abs/2602.22839.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Kun Zhou, Jiakai He, Wenmian Yang, Zhensheng Wang, Yiquan Zhang, and Weijia Jia. Automatic slide updating with user-defined dynamic templates and natural language instructions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18263–18297, 2026. doi: 10.18653/v1/2026.findings-acl.909. URL https://aclanthology.org/2026.findings-acl.909/.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*, 2023.