MuTUI-Bench: Benchmarking Multi-Turn Image Generation and Editing for Visual Interfaces

Anonymous authors
Paper under double-blind review

Abstract

A visual interface editor must change the intended element, preserve relevant context, and interpret user intent across turns. We introduce MuTUI-Bench, a compact image-output benchmark with three tracks: *Design* for stylistic and structural editing, *Assist* for sequential mouse guidance, and *Simulate* for rendering browsing-action consequences. The corpus contains 76 tasks and 274 turns: 255 design edits across seven domains and twelve edit types, and 19 interaction turns across six websites. An audit of two archived image-generation systems separates image quality from instruction correctness, removes repeated judgments, and reports matched comparisons with task-level uncertainty. On Design, mean image quality exceeds 8.3/10 while instruction correctness remains below 6.1/10; icon replacement is particularly weak despite polished outputs. The interaction tracks expose complementary demands on pointing and state rendering, although their small size and reference-conditioned evaluation limit generalization. MuTUI-Bench provides an inspectable testbed for studying what should change, what should persist, and what should happen next in visual UI editing.

1 Introduction

Consider three requests made while looking at a webpage: "make this button cyan without moving it," "show me where to click next to buy this product," and "show the page after opening the product menu." All require understanding a visible interface and generating a new image. Yet the intended transformations differ. The first changes a design attribute while preserving layout. The second adds a spatially grounded interaction cue. The third changes the interface state in a way that is consistent with the selected control. Across a sequence of requests, the editor must also account for earlier changes or for the newly observed page.

These settings differ in both their inputs and success criteria. A screenshot can look professional while failing its purpose: an icon may remain unchanged, a pointer may identify the wrong control, or a generated dialog may omit the requested fields. These cases motivate evaluating the *relationship between the request, the current screen, and the generated result*, rather than judging the result in isolation.

Existing benchmarks evaluate related but distinct settings. WebArena and VisualWebArena assess agents that operate on websites (Zhou et al., 2023; Koh et al., 2024); Design2Code evaluates screenshot-to-code reconstruction (Si et al., 2024); MagicBrush and ImgEdit include instruction-based and multi-turn image editing (Zhang et al., 2023; Ye et al., 2025). More recent work studies generative GUI environments and iterative interface maintenance (Li et al., 2026a;b; Peng et al., 2026; Fu et al., 2026). Our focus is the shared evaluation of three image-output tasks: local interface editing, goal-directed pointing, and rendering the screen state resulting from an action.

We develop MuTUI-Bench around three tracks: **Design**, **Assist**, and **Simulate**. Our contributions are an explicit task formulation linking these capabilities; a corpus and coverage analysis spanning 76 short trajectories; and a reproducible audit of archived outputs from two image-generation systems. The archived scores show that high image-quality scores do not always coincide with high scores for satisfying the requested change. They also show differences across edit categories and disagreement between absolute and pairwise judgments. We report these as diagnostic findings

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.

under the recorded protocol, with the limits of short trajectories, sparse interaction coverage, and automated judging made explicit.

2 Related Work

Interface understanding and reconstruction. Rico supports data-driven UI research with mobile screens and interaction traces (Deka et al., 2017). Design2Code evaluates the generation of code that reconstructs a reference webpage (Si et al., 2024). These resources establish the importance of layout and visual fidelity, but reconstructing an entire page does not by itself test whether a model can selectively modify an existing screenshot. MT-Web2Code directly addresses multi-turn regional reconstruction and localized modification in codebases (Li et al., 2026b). EvoGenUI-Bench tests maintenance of executable interfaces as user requests evolve, including interface behavior and externally grounded state (Peng et al., 2026). Our Design track shares their concern for preserving prior content, but takes raster images as both observations and outputs; no source code, DOM, or executable artifact is required.

Instruction-guided image editing. InstructPix2Pix learns image editing from paired instructions and images (Brooks et al., 2022). MagicBrush explicitly includes iterative edits, while AnyEdit broadens editing diversity and ImgEdit includes a dedicated multi-turn evaluation suite (Zhang et al., 2023; Yu et al., 2024; Ye et al., 2025). Thus, neither local editing nor multi-turn image generation is unique to MuTUI-Bench. The distinction lies in the interface-specific constraints: an icon's identity, the legibility and exact wording of a label, the alignment of controls, and the relationship between an action and a new screen. A visually similar result may fail one of these requirements even when it is acceptable as an unconstrained illustration.

GUI agents and grounding. WebArena evaluates functional task completion in reproducible web environments, and VisualWebArena extends evaluation to tasks that require visual understanding (Zhou et al., 2023; Koh et al., 2024). OSWorld studies computer use through execution-based evaluation (Xie et al., 2024). SeeClick and its ScreenSpot benchmark isolate the problem of locating interface elements from visual input (Cheng et al., 2024). Assist is related to grounding, but its output is a screenshot with a visual mouse pointer that a user can interpret. It therefore couples localization with visual preservation. It does not execute the indicated action or inherit the success criteria of an autonomous web agent.

Generative interface environments. GEBench is a close predecessor: it evaluates image generators as GUI environments, including dynamic interactions, trajectories, and grounding (Li et al., 2026a). Code2World predicts subsequent GUI states through renderable code (Zheng et al., 2026). GUI-CC further evaluates contextual consistency under reference-action trajectories and online agent loops (Fu et al., 2026). These works already cover important parts of Simulate. Our contribution is a complementary organization that places local design operations alongside assistance and state synthesis, with a domain-by-edit breakdown of the design component. We do not claim the first GUI simulation benchmark or a replacement for these larger environment evaluations.

3 Task Definition and Track Design

3.1 A Common Image-Output Interface

A task is a sequence $\tau_i = (x_{i,0}, g_i, \{u_{i,t}, r_{i,t}, y^*_{i,t}\}_{t=1}^{T_i})$, where $x_{i,0}$ is the initial screenshot, g_i is an optional overall goal, and $u_{i,t}$ is the current instruction. The recorded before-state $r_{i,t}$ and illustrative target $y^*_{i,t}$ are optional. A system produces

$$\hat{y}_{i,t} = F_\theta(b_{i,t}, u_{i,t}, g_i, h_{i,t}), \qquad b_{i,t} = \begin{cases} x_{i,0}, & t = 1, \\ \hat{y}_{i,t-1}, & \text{chained editing}, \\ r_{i,t}, & \text{reference-conditioned editing}, \end{cases} \tag{1}$$

where $h_{i,t}$ denotes the complete conversation history available at turn t. These input policies answer different questions. Chained editing tests a trajectory influenced by earlier generated outputs. Ref-

erence conditioning tests a requested transformation from a supplied screen, even if previous model outputs were incorrect. We retain the distinction throughout evaluation.

3.2 Design: Modify an Interface while Preserving Its Constraints

Design covers both appearance and structure. Its twelve edit types comprise color replacement, typography adjustment, decoration effects, icon replacement, text modification, element addition/removal, order adjustment, layout restructuring, responsive adaptation, size adjustment, style change, and position adjustment. A generation prompt establishes the initial visual style; subsequent turns request concrete edits. The evaluated turns concern editing these seeded interfaces, rather than scoring the initial text-to-image generation.

The intended invariant depends on the request. Changing an icon should preserve its surrounding label, line weight, and position. Responsive adaptation may legitimately relocate several elements while preserving their semantic roles. Consequently, "local" refers to the *scope of the requested change*, not necessarily a small fixed pixel region. The instructions specify that scope in language; the dataset does not provide segmentation masks or element boxes. This makes region selection part of the model's task, while also limiting deterministic measurements of unintended changes.

3.3 Assist: Ground the Next Step in a User Goal

Assist provides an overall goal and a sequence of observed screens. Each turn asks the model to indicate where the user should click by placing a visual mouse pointer over the target area. The three goals concern finding nearby accommodation for specified dates, purchasing a particular phone, and finding technical assistance on an automotive website. The repeated turn wording is intentional: the correct target depends on the goal and visible state rather than lexical diversity in the local instruction.

All ten Assist turns have recorded reference inputs. The task is sequential *guidance over supplied states*; the model does not execute clicks or obtain observations. A correct answer makes the intended control apparent while preserving recognizable interface context. Target images illustrate the pointing behavior but are not supplied to the absolute judge.

3.4 Simulate: Render the Consequence of a Browsing Action

Simulate asks for the image that should follow an action: navigating through a control, returning to a landing page, opening a menu or form, or scrolling to reveal further content. Its three website traces contain nine turns. Unlike Assist, the desired output is a changed page state, rather than an additional pointing cue. Some transitions are tightly constrained, such as opening a named menu; others, such as producing a pricing page, permit multiple visually plausible realizations. The benchmark tests visual action consequences, not the correctness of a hidden website backend.

4 Corpus Construction and Coverage

4.1 Design Seeds and Instruction Sequences

The Design corpus contains 70 synthetic initial interfaces, with ten tasks in each of seven domains: e-commerce, social media, dashboards, forms and authentication, gaming interfaces, finance, and chat/messaging. The stored seed prompts specify both content and style, including examples such as a neon product page, a minimalist financial dashboard, and a bold promotional landing page. All 70 initial images have resolution 1360×768. The supplied seed-generation script routes prompts to an image generator; the archived metadata retain the prompts and image paths, but not complete per-image provider provenance.

The instruction-generation scripts select an edit category, condition on the initial image, and include earlier instructions as textual context. Attribute-focused and structure-focused category pools contribute 126 and 129 turns, respectively, within the single Design track. Category selection discourages repetition within a trajectory. However, the generator observes the initial image rather than newly edited intermediate states. Textual continuity is encouraged by prompting, not guaranteed by

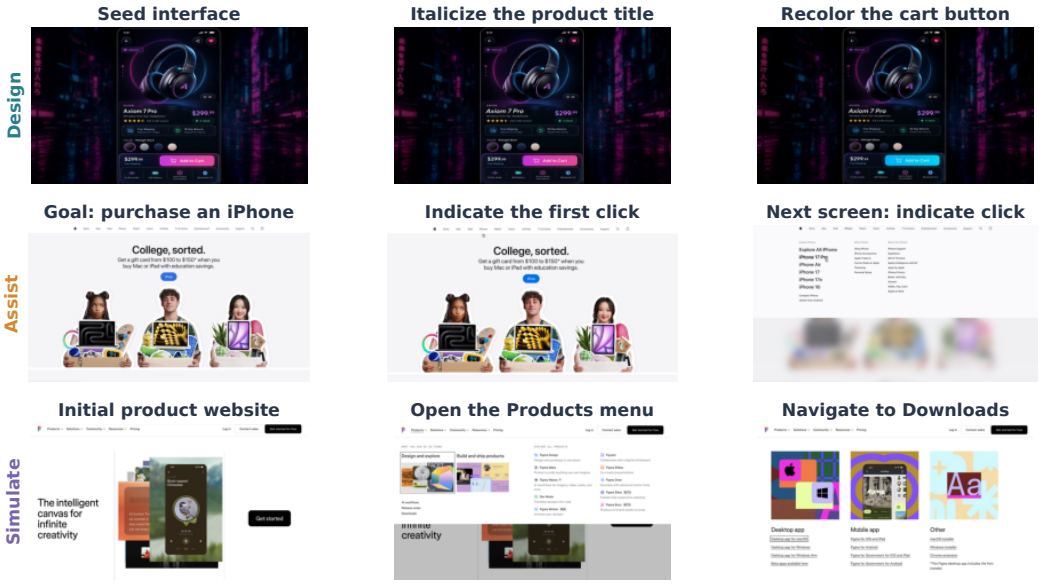

Figure 1: Actual corpus examples for the three tracks. Design shows the initial image followed by system G outputs for two edits. Assist and Simulate show recorded initial and target images; subsequent interaction turns can receive a new recorded before-state. Headings are concise English paraphrases, and the Assist goal is abbreviated. Images are resized for display.

a semantic validation stage. This matters when an early structural edit moves a control that later instructions still identify by its original position.

Design contains initial images and instructions without target after-images. Model outputs are evaluation artifacts rather than independent ground truth, and the instructions are model-generated. Evaluation therefore uses a judge conditioned on the interface and requested edit.

4.2 Website-Based Interaction Traces

The interaction component contains six website traces: Airbnb, Apple, and BMW for Assist; ClickHouse, ElevenLabs, and Figma for Simulate. These labels identify the represented websites and are not affiliations or endorsements. The traces provide before/after image paths and short Chinese instructions, sometimes mixed with English interface labels. Assist includes a nonempty goal for every task; the three Simulate tasks have no separate overall-goal text because their local instructions specify the requested transition.

Of the 19 interaction turns, 15 explicitly provide a reference-before frame and four use the previous recorded target. Both supply recorded inputs rather than model-generated states. All 19 turns have target images for inspection; their collection history does not establish independent human validation of every target, so we call them *recorded targets* rather than verified functional outcomes.

4.3 Coverage Audit and Data Integrity

The audit checks task identifiers, sequence lengths, turn numbering, and referenced images directly from metadata. It verifies all 548 expected model outputs and 652 distinct image paths overall. Content hashes identify the analyzed snapshot; they cannot retroactively authenticate image bytes supplied to historical judge calls.

Figure 2 distinguishes broad coverage from strict balance. Each domain has ten tasks, but type counts range from 14 to 28, and two of 84 domain–type cells are empty. Subgroup results are therefore descriptive rather than estimates from a fully factorial design. Fine-grained labels are preserved within the Design track.

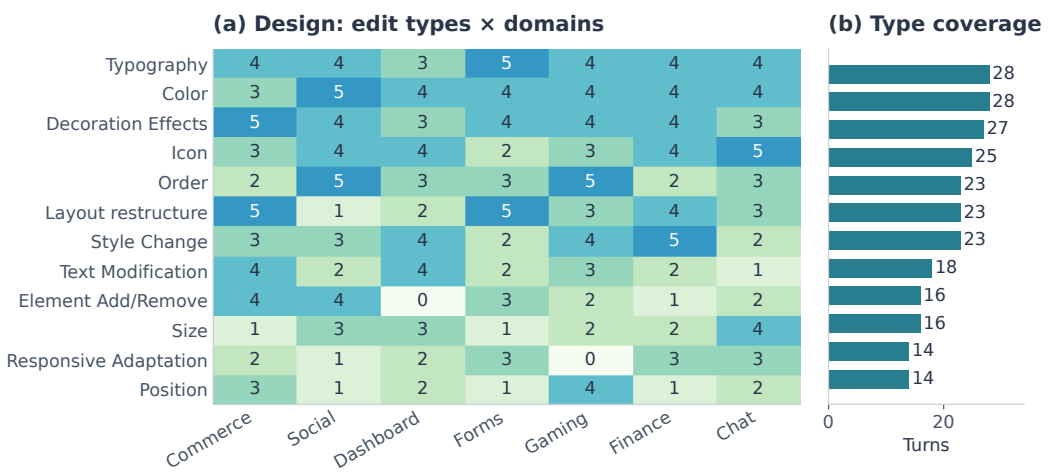

Figure 2: Design coverage computed from all 255 turns. Each cell reports the number of edits in a domain–type combination; the right panel gives type totals. Domain balance at the task level does not imply complete cross-product coverage: dashboard element addition/removal and gaming responsive adaptation are absent.

Table 1: Corpus statistics before filtering evaluation records. Instruction length is measured as whitespace-delimited words for English Design instructions and Unicode characters for the Chinese-dominant interaction instructions; these are not comparable tokenizer counts.

Track	Tasks	Turns	3-turn	4-turn	Instruction length
Design	70	255	25	45	71.8 words
Assist	3	10	2	1	21.0 characters
Simulate	3	9	3	0	24.1 characters

4.4 Sequence Length, Language, and Interaction Diversity

Table 1 summarizes the resulting corpus. All tasks contain three or four turns: 30 tasks have three turns and 46 have four. Design accounts for $255/274 = 93.1\%$ of all turns. Consequently, a pooled turn average would largely measure Design and obscure the two interaction tracks. We report track-specific results and avoid treating a single overall score as a representative measure of all three capabilities.

The 255 Design instructions are distinct and average 71.8 whitespace-delimited words: 61.2 for attribute operations and 82.3 for structural operations. Length reflects positional descriptions and preservation clauses, not an independent difficulty measurement. Assist has four distinct turn-number templates averaging 21.0 characters; its goal and current screen disambiguate the target. Simulate has nine distinct instructions averaging 24.1 characters.

The interaction coverage is limited: Simulate has only one scrolling turn, and each interaction website has one task. Website effects cannot be separated from intent, content, or resolution. Assist and Simulate are therefore diagnostic slices rather than representative samples of web usage.

5 Evaluation Protocol

5.1 Archived Systems and Input Conditions

We analyze the stored outputs under two gateway identifiers: `gpt-image-2` (G) and `gemini-3.1-flash-lite-image` (M). Both systems have an output image for every corpus turn. No additional model calls or training are performed for this analysis. Both systems use multi-turn conversational inference and retain the complete conversation history for each request.

5.2 Two Complementary Judgment Criteria

Every absolute evaluation record identifies the judge as `gpt-5-nano-2025-08-07`. Its two criteria use a 0–10 scale, with higher values better. **Image quality (IQ)** examines only the generated image, considering structural coherence, visual fidelity, artifacts, overall naturalness, and text rendering. IQ is not an instruction-following metric. **Key-point correctness (KP)** receives the generated image, the evaluator's before-image, and the current instruction, with the overall goal prepended when present. Its rubric weights requirement fulfillment at 70% and visual quality at 30%. Requirement fulfillment includes accuracy, completeness, and preservation of important reference elements. These weights are instructions to the judge, not separately measured component scores.

KP is therefore a composite instruction-correctness judgment, not a geometric keypoint metric or an isolated preservation score. It does not ingest the recorded target image. Moreover, the Design evaluator uses the initial image as its reference at every turn, whereas inference chains earlier outputs. A later-turn score consequently assesses the current request against the original interface, not against the actual immediate input. We retain this behavior when reporting historical scores and explicitly avoid interpreting KP as a calibrated measure of cumulative preservation. This reference mismatch is a limitation of the archived protocol, not an unreported correction applied to the data.

5.3 Record Selection and Matched Comparisons

The absolute logs contain 1,240 rows for 548 distinct model–task–turn keys. Repeated rows include retries and incomplete responses. Some records have a valid IQ value but a null KP value even when the parser reports no error. Averaging every numeric entry would therefore weight turns unequally and can mix successful judgments with failed scoring attempts.

We use a score-independent selection rule. For each key, a record is complete only if both IQ and KP are finite numeric values in $[0, 10]$ and neither has a parse-error flag. We retain the latest complete record by timestamp, with physical line order breaking ties. A key without a complete record is excluded. For the headline comparison, a turn must have complete records for both systems. This leaves 247 Design, nine Assist, and nine Simulate turns: 265 matched turns, or 96.7% of the corpus. Nine model–turn keys lack complete scores; they correspond to nine distinct turns. Missing scores are not imputed or treated as generation failures.

For track c and metric q, we report the turn mean and paired difference on the same set $\mathcal{M}_c$:

$$\bar{q}_{m,c} = \frac{1}{|\mathcal{M}_c|} \sum_{(i,t)\in\mathcal{M}_c} q_{m,i,t}, \qquad \Delta_{q,c} = \frac{1}{|\mathcal{M}_c|} \sum_{(i,t)\in\mathcal{M}_c} \left(q_{G,i,t} - q_{M,i,t}\right). \tag{2}$$

We estimate percentile 95% intervals with 10,000 task-bootstrap replicates and seed 2027, retaining all matched turns within each sampled task. This respects within-task dependence. Intervals from three interaction tasks are coarse and do not measure judge reliability or provider nondeterminism.

5.4 Pairwise and Longitudinal Diagnostics

Pairwise logs ask which output has better IQ or KP. We select the latest valid winner separately for each task, turn, and criterion; valid responses are G, M, or tie after mapping the stored A/B ordering to system identities. Invalid winners are excluded per criterion. These comparisons have their own valid counts, rather than inheriting the matched absolute-score subset. The stored presentation order is fixed, so position bias remains possible, as emphasized in prior work on model judging (Zheng et al., 2023).

Turn-position plots describe which instructions occur at each index and are not controlled tests of error accumulation. We additionally compare first and last scores within the same 62 Design tasks for which both systems have complete scores at every turn. This paired endpoint diagnostic reduces changes in task composition, while still inheriting the reference mismatch and differing edit categories at the first and last turn. No judge-to-human calibration experiment is available in the archive.

Table 2: Matched absolute judgments for the archived configurations. G and M denote the two system identifiers in Section 5. IQ measures image quality; KP measures instruction correctness with a visual-quality component. Counts are matched turns per system, not accumulated log rows.

Track	System	Matched / total	IQ ↑	KP ↑
Design	G	247 / 255	8.72	6.09
Design	M	247 / 255	8.38	5.83
Assist	G	9 / 10	7.67	8.33
Assist	M	9 / 10	7.78	7.22
Simulate	G	9 / 9	8.78	7.89
Simulate	M	9 / 9	8.67	8.11

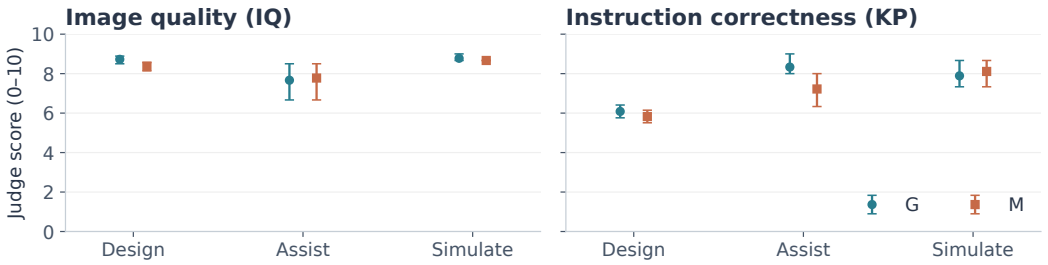

Figure 3: Matched track means and percentile 95% task-bootstrap intervals. Interaction intervals involve only three tasks and should not be read as population guarantees. M's Simulate IQ interval collapses because all three task means happen to equal 8.67; this does not imply a perfectly reliable judge.

6 Results and Analysis

6.1 Visual Quality and Instruction Correctness Separate

Table 2 reports the main results. On Design, G obtains IQ 8.72 and KP 6.09, while M obtains IQ 8.38 and KP 5.83. Both systems receive high IQ scores on average, while their KP scores are lower. This difference indicates that visual quality alone does not guarantee that the requested change was judged to be satisfied.

G's paired Design IQ advantage is $+0.35$, with a task-bootstrap interval of $[+0.25, +0.46]$. Its KP advantage is smaller, $+0.26$, with interval $[-0.15, +0.65]$. Thus, the matched scores support a clearer image-quality difference than a broad instruction-correctness ranking. This distinction is obscured if IQ is mislabeled as instruction quality or if all historical rows are averaged without deduplication.

On Assist, G has higher mean KP (8.33 versus 7.22) but slightly lower IQ (7.67 versus 7.78). The paired KP interval spans $[0.00, 2.00]$. These nine matched turns suggest a difference in pointing adequacy within the available cases, while remaining too sparse for a strong claim across unseen websites. Simulate reverses the absolute KP ordering: M obtains 8.11 and G 7.89, with a paired difference interval of $[-1.00, +1.33]$. Its IQ means are 8.67 and 8.78. The two interaction behaviors should therefore be reported separately even in a small corpus.

The higher KP values on the interaction tracks do not show that browsing is intrinsically easier than local editing. The tracks differ in instruction detail, language, reference conditioning, visual content, and number of constraints. In particular, provided interaction states remove some consequences of earlier model failures. These confounds rule out a causal claim about task difficulty from the track means alone.

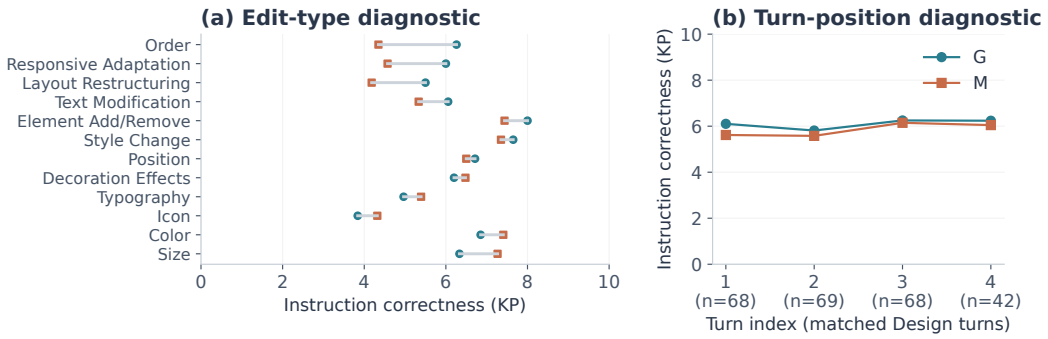

Figure 4: Design diagnostics using matched turns. (a) Per-category KP shows heterogeneous system scores, with icon replacement receiving the lowest means. (b) Turn-position means do not exhibit a consistent monotonic decline; later positions contain different instructions and fewer tasks. G uses circles and M squares throughout.

6.2 Pairwise Preferences Tell a Different Story

Pairwise judgments strongly favor G: it wins 200 of 254 Design KP pairs, all ten Assist KP pairs, and eight of nine Simulate KP pairs. Yet the absolute Simulate KP mean slightly favors M. Absolute scores and direct preferences are therefore not interchangeable. Fixed presentation order, comparative salience, and evaluator framing are possible explanations, but the archive cannot distinguish them. Reversed-order judging and independent human calibration remain necessary follow-up experiments.

6.3 Edit-Type and Domain Diagnostics

Figure 4 breaks down Design KP on matched turns. Icon replacement receives low scores: 3.84 for G and 4.32 for M over 25 turns, despite IQ means of 8.96 and 8.52. Typography KP is also low, at 4.96 and 5.39. A targeted inspection shows both an unchanged person icon and a slider substitution contradicted by the archived judge rationale. Low scores therefore identify a fine-grained evaluation problem, not a validated failure rate: model errors and judge errors can coexist.

By contrast, element addition/removal reaches KP 8.00 for G and 7.44 for M; style change reaches 7.65 and 7.35. The relative ordering therefore does not follow a rule that smaller edits are easier. G has a larger advantage on order adjustment (6.26 versus 4.35), layout restructuring (5.50 versus 4.18), and responsive adaptation (6.00 versus 4.57). M scores higher on color replacement (7.41 versus 6.85) and size adjustment (7.27 versus 6.33). These are exploratory category means with different sample counts, not separately powered comparisons or proof of underlying mechanisms.

Domain averages likewise conceal type differences. G's KP ranges from 5.40 on gaming interfaces to 6.39 on chat/messaging; M ranges from 5.37 on gaming to 6.19 on social media. Because the domain–type matrix is not fully balanced, a domain difference may reflect which edits were requested rather than interface content alone.

6.4 Short Trajectories Do Not Establish Accumulating Failure

Across the 62 fully scored Design tasks, G's mean last-minus-first KP difference is -0.02 with interval $[-1.02, 0.97]$; M's is $+0.53$ with interval $[-0.55, 1.58]$. The corresponding IQ differences are $+0.05$ and -0.03. These observations do not support a general claim of monotonic quality decay across the available three- and four-turn tasks. They also do not show that cumulative preservation has been solved: endpoint instructions differ, and the scoring prompt does not provide the complete editing history to the judge. A dedicated persistence metric would need to track earlier requirements explicitly.

As a record-selection sensitivity check, replacing the latest complete absolute record with the earliest complete record leaves all matched means unchanged. In this archive, complete absolute judgments

are unique for the retained keys. Repeated log rows therefore do not supply independent generation trials or a reliable estimate of judge variance. The audit's main effect is to prevent incomplete attempts and repeated partial scores from distorting the aggregate.

7 Discussion and Limitations

Preservation requires semantic scope. A pixel-difference metric cannot alone distinguish legitimate layout changes from accidental drift, while global quality can overlook an unchanged icon. Language-based scope enables evaluation without source code but places responsibility on the judge. A stronger successor should attach element identities and preservation constraints—text, geometry, and semantic roles—to each turn. These constraints should persist until explicitly superseded. The present results motivate this requirement-tracking evaluation but do not implement it.

Limits of the collection. The corpus has short trajectories and sparse interaction coverage. Synthetic seeds can contain preexisting artifacts and may favor a system related to their generator; complete provenance is unavailable. Language differs across tracks, and multilingual robustness is untested. File checks establish integrity, not semantic validity or inter-annotator agreement.

Limits of archived evaluation. IQ and KP are not validated usability measurements. IQ's naturalness criteria may imperfectly match stylized interfaces, and KP mixes fulfillment with appearance. Fixed pairwise ordering, differing input/reference policies, and the Design reference mismatch constrain system comparisons. Missing generation-time hashes and request settings permit reproduction of this snapshot analysis, but not exact replay of provider responses.

Missing-score sensitivity. Eight Design turns are excluded from the paired analysis. Allowing their G-minus-M differences to range over $[-10, 10]$ bounds the full-corpus IQ difference to $[0.02, 0.65]$ and KP to $[-0.06, 0.56]$. These conservative arithmetic bounds are not confidence intervals and do not impute scores. Missingness could reverse the small KP advantage; the IQ difference retains its sign within these bounds.

Visual simulation versus executable behavior. Screenshots contain no DOM, action handler, or backend state: a plausible dialog cannot establish authentication. Reference-conditioned traces isolate local transformations without testing autonomous continuation. Executable probes and longer closed-loop trajectories are needed to assess whether generated states support subsequent actions.

Benchmark use and extensions. We treat the corpus as an evaluation set, without claiming a training split or held-out performance. Tuning on its prompts, screenshots, or judgments should be disclosed. Extensions should use task- or website-level splits, independently validate targets, align scoring references with actual inputs, expand interaction coverage, and balance context conditions across systems.

8 Conclusion

MuTUI-Bench organizes visual UI editing into Design, Assist, and Simulate: changing an interface, showing where to act, and rendering action consequences. Its 76 tasks and 274 turns support an analysis of domain coverage, edit types, and short interaction traces. Archived judgments separate visual quality from instruction fulfillment, while local inspection shows that model errors and judge errors can coexist. The results point to a practical evaluation requirement: an editor should satisfy the current request while preserving interface information needed in later turns.

REFERENCES

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. URL https://arxiv.org/abs/2211.09800.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024. URL https://arxiv.org/abs/2401.10935.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017. URL https://www.interactionmining.org/archive/rico.

Lin Fu, Zheyuan Yang, Tianhui Zhang, Jinbiao Wei, Guo Gan, Boxu Liu, Yilun Zhao, and Yu Rong. GUI-CC: Benchmarking contextual consistency of gui world models as agent environments. *arXiv preprint arXiv:2609.00048*, 2026. URL https://arxiv.org/abs/2609.00048.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024. URL https://arxiv.org/abs/2401.13649.

Haodong Li, Jingwei Wu, Quan Sun, Guopeng Li, Juanxi Tian, Huanyu Zhang, Yanlin Lai, Ruichuan An, Hongbo Peng, Yuhong Dai, Chenxi Li, Chunmei Qing, Jia Wang, Ziyang Meng, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. GEBench: Benchmarking image generation models as gui environments. *arXiv preprint arXiv:2602.09007*, 2026a. URL https://arxiv.org/abs/2602.09007.

Qiming Li, Shujie Hu, Haohan Liu, Xiaocheng Feng, Songxiang Liu, and Guanglu Wan. MT-Web2Code: Benchmarking coding agents on multi-turn regional reconstruction and localized modification. *arXiv preprint arXiv:2608.03474*, 2026b. URL https://arxiv.org/abs/2608.03474.

Yue Peng, Lanke Xia, Zihan Wang, Jiahao Ye, Ke Ning, and Hongyi Wen. EvoGenUI-Bench: Evaluating llms as multi-turn generative ui assistants. *arXiv preprint arXiv:2608.29387*, 2026. URL https://arxiv.org/abs/2608.29387.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2code: Benchmarking multimodal code generation for automated front-end engineering. *arXiv preprint arXiv:2403.03163*, 2024. URL https://arxiv.org/abs/2403.03163.

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. *arXiv preprint arXiv:2404.07972*, 2024. URL https://arxiv.org/abs/2404.07972.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. ImgEdit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025. URL https://arxiv.org/abs/2505.20275.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. AnyEdit: Mastering unified high-quality image editing for any idea. *arXiv preprint arXiv:2411.15738*, 2024. URL https://arxiv.org/abs/2411.15738.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. *arXiv preprint arXiv:2306.10012*, 2023. URL https://arxiv.org/abs/2306.10012.

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. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.

Yuhao Zheng, Li'an Zhong, Yi Wang, Rui Dai, Kaikui Liu, Xiangxiang Chu, Linyuan Lv, Philip Torr, and Kevin Qinghong Lin. Code2World: A gui world model via renderable code generation. *arXiv preprint arXiv:2602.09856*, 2026. URL https://arxiv.org/abs/2602.09856.

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. *arXiv preprint arXiv:2307.13854*, 2023. URL https://arxiv.org/abs/2307.13854.