INVARIANTCANVAS: EVIDENCE-GROUNDED REVISION WITH REGRESSION-AWARE ACCEPTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense visual artifacts must satisfy exact text and numerical requirements while preserving layout, visual relations, and readability. However, iterative generation-and-critique systems may introduce new violations when repairing local defects. We introduce INVARIANTCANVAS, an evidence-grounded framework for reliable dense visual generation. The framework decomposes requests into verifiable requirements, generates editable Hypertext Markup Language (HTML) and Scalable Vector Graphics (SVG) artifacts, and combines browser-based checks, OCR, source inspection, and visual audits with explicit provenance. A central adjudicator decides whether the available evidence justifies a revision. A revised artifact is accepted only when the targeted requirement changes from FAIL to PASS and no previously passed mandatory requirement regresses; otherwise, the previous version is retained. An independent review then checks the exact image selected for delivery.

INVARIANTCANVAS achieves 95.86% overall and 94.83% Hard-split accuracy on BizGenEval, and 0.94 question-level and 0.62 infographic-level accuracy on IGen-Bench. The results demonstrate the importance of evidence-grounded adjudication and regression-aware revision for reliable dense visual generation.

1 INTRODUCTION

Dense visual artifacts, such as business graphics, charts, slides, posters, and scientific figures, must satisfy many fine-grained requirements at the same time. They must reproduce exact text and numerical values, place elements in the requested order and positions, and preserve relations among labels, axes, marks, legends, connectors, and containers. Correctness also depends on cross-element consistency: a value shown in a table may need to match a chart mark and a textual statement, while an arrow must connect the intended source and target. Recent generation systems have improved visual quality and text rendering (Peng et al., 2025; Zhang et al., 2026a), and code-based methods make artifacts editable through SVG, TikZ, or diagram programs (Belouadi et al., 2024; Rodriguez et al., 2025; Wei et al., 2025; Belouadi et al., 2025). However, editability alone does not make such artifacts safe to revise. Fixing an overlap may remove a label, moving a chart may change its relation to an axis or legend, and regenerating one component may change a value that is repeated elsewhere.

This problem is difficult because different verification tools observe different aspects of an artifact. Browser-based checks can measure positions, visibility, clipping, overflow, and selected geometric relations, but they provide limited evidence about semantic intent and cross-element meaning. OCR can detect text discrepancies, but may be unreliable for small or stylized text. Visual review can assess the complete canvas, hierarchy, legibility, and relations that require global context, but may miss a small numerical error or report a preference that was not required by the request. These observations should not be treated as equivalent or reduced to a single score. An edit that improves one region may still break a required value, relation, or visual binding elsewhere.

Iterative critique can improve generation (Singh & Zheng, 2023; Gou et al., 2024), but finding a defect is only part of the problem. A system must also decide whether the finding is supported by the request, whether it justifies a revision, and whether the revised artifact should replace the current version. A reported issue may result from incomplete evidence, may concern an unrequested design choice, or may be valid but introduce a new failure elsewhere. This raises a basic question: *when is feedback reliable enough to justify changing the artifact?*

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.

We introduce INVARIANTCANVAS, a framework that separates verification evidence from revision decisions. It decomposes a request into individually checkable requirements, preserving exact wording, expected content, quantity scope, and requested relations. These requirements are linked to the visual elements that should realize them in an editable HTML/SVG artifact. Browser-based checks, OCR, source inspection, and visual review provide version-specific evidence with explicit provenance. Each method may return PASS, FAIL, or UNKNOWN when the available evidence is missing or unclear. An UNKNOWN result is not treated as a pass and does not by itself justify an edit.

The central adjudicator uses the original request, the current artifact, and the available evidence to decide whether a finding is a request-grounded defect and whether it justifies a revision. After a candidate revision is rendered, the complete artifact is evaluated again, and the adjudicator decides whether the candidate can replace the current version. The candidate is accepted only when the targeted mandatory requirement no longer fails and no mandatory requirement that previously passed becomes a failure. Otherwise, the previous version is retained. Once a candidate is accepted, the revision loop ends and the selected image proceeds to a fresh delivery review without access to the revision history. A supported violation found during this review returns the artifact to the revision process.

We evaluate INVARIANTCANVAS on BizGenEval and IGenBench using both question-level and whole-artifact metrics. In our comparison pool, INVARIANTCANVAS obtains strong results on both types of evaluation. Qualitative cases illustrate how the framework handles fine-grained numerical consistency and visual relations. In the cases shown, direct image generation and image-based editing are less consistent when values, positions, labels, and graphical relations must remain coordinated. These cases illustrate the task addressed by INVARIANTCANVAS and are not intended to isolate the effect of any individual component.

Our contributions are:

- We develop a request-grounded representation that links exact content, quantities, cross-element relations, and visual properties to elements in an editable HTML/SVG artifact.
- We separate heterogeneous verification evidence from revision decisions. Browser-based checks, OCR, source inspection, and visual review preserve their provenance and may return UNKNOWN, while a central adjudicator decides whether a finding supports revision and whether a revised artifact may replace the current version.
- We define a conservative acceptance policy for revised artifacts and evaluate the complete framework on BizGenEval and IGenBench using official metrics and qualitative cases involving fine-grained numerical and relational constraints.

2 RELATED WORK

Structured visual generation. Early graphic-layout models learn geometric relations among elements or satisfy user-specified layout constraints (Li et al., 2019; Lee et al., 2020). Recent systems author richer editable representations. AutomaTikZ and TikZero synthesize scientific graphics programs from text (Belouadi et al., 2024; 2025); StarVector and Chat2SVG generate SVG code or templates (Rodriguez et al., 2025; Wu et al., 2025); and DiagramAgent combines planning, code generation, checking, and editing for structured diagrams (Wei et al., 2025). Layered design systems such as Accordion recover editable components from a global visual reference (Chen et al., 2025), while CreatiDesign conditions graphic synthesis on multiple user-provided controls (Zhang et al., 2026a). These methods motivate structured authoring, but our focus is the revision policy that governs mixed evidence and repeated repair across heterogeneous dense artifacts. Recent code-native systems also use HTML with Cascading Style Sheets (CSS) or SVG representations together with render-aware refinement, including DesignAsCode, Editable Visual Design, and FigTree (Liu et al., 2026b; Ye et al., 2026b; Liu et al., 2026a). Agentic systems likewise target scientific and academic illustrations through planning, rendering, and iterative validation, as in PaperBanana and AutoFigure (Zhu et al., 2026a;b).

GenClaw similarly separates cognitive structuring, executable-canvas construction, and visual review in a code-driven image agent (Ye et al., 2026a); INVARIANTCANVAS instead centers request-grounded requirements and conservative acceptance of revised artifacts across heterogeneous dense visual tasks.

Fine-grained verification and correction. Decomposing a request into assertions improves the sensitivity of visual alignment evaluation and enables iterative feedback (Singh & Zheng, 2023). CRITIC likewise demonstrates that external tools can support iterative self-correction (Gou et al., 2024). For visualization, Draco represents design knowledge as constraints, and VizLinter detects and repairs violations in Vega-Lite specifications (Moritz et al., 2019; Chen et al., 2022). Recent visualization and diagram systems make selected properties inspectable through heterogeneous evaluation, validator agents, learned critics, executable construction, or intermediate representations (Chen et al., 2024; Ouyang et al., 2025; Pan et al., 2025; Kim et al., 2026; Kumar et al., 2025; Li et al., 2026b). INVARIANTCANVAS builds on the principle of decomposed checking but preserves the source of each observation, treats missing evidence as UNKNOWN, and requires an explicit adjudication step before evidence can authorize an edit.

Persistent specifications and repair-aware control. SCOPE maintains an evolving structured specification of semantic commitments, while AutoDesign studies reusable harness optimization for long-horizon design (Ren et al., 2026; Luo et al., 2026). Recent multi-agent editing systems also maintain object state, contextual memory, or iterative validation during revision (Zhao et al., 2026; Ye et al., 2026c; Zhang et al., 2026b). Concurrent PosterVisor compiles paper-grounded semantic–geometric contracts for scientific posters and rechecks affected checkpoints after repair (Li et al., 2026a). These works make clear that persistent specifications and generation frameworks are not, by themselves, our novelty. Relative to dependency checks during revision, INVARIANTCANVAS contributes two distinct decisions: evidence adjudication determines whether an observation justifies an edit, and post-render comparison determines whether the revised artifact may replace the preceding version. Explicit UNKNOWN states and an image-based delivery review add two further observable behaviors. The resulting audit trail records dismissed false positives, rejected revisions, audit requirement coverage, and defects returned from delivery review rather than treating the delivered image as the only observable outcome.

3 PROBLEM SETTING

Let p be a user request, A an editable visual artifact, and γ a fixed rendering configuration. A deterministic browser renders the artifact as

$$I = R_\gamma(A).$$

The request is decomposed into an ordered collection of requirements $\mathcal{R} = (r_1, \ldots, r_n)$. Each requirement records its request span, type, expected content or property, scope, and relevant relations. Requirement types include exact text, cardinality, ordering, numerical consistency, spatial relations, and visual properties such as legibility. The design specification marks each requirement as mandatory or advisory and links it to the visual elements intended to realize it.

A verification method v evaluates requirement r_i using the request, artifact, rendered image, and method-specific evidence:

$$z_{v,i}(p, r_i, A, I) \in \{\text{PASS}, \text{FAIL}, \text{UNKNOWN}\}. \tag{1}$$

Each result includes a provenance record. UNKNOWN means that the available evidence is insufficient or ambiguous; it is neither a pass nor evidence of a defect. Different methods may disagree, so their results are retained separately. The adjudicator compares these records with the original request and the rendered image rather than reducing them to a single score.

A requirement may have multiple visual realizations. For example, a value may appear in text, a table, and a chart, while a relation may involve several elements such as endpoints, connectors, and labels. We therefore record each requirement's links to its associated visual elements, evidence, and revision history. These links support targeted revision and comparison across artifact versions.

4 METHOD

4.1 OVERVIEW

INVARIANTCANVAS treats dense visual authoring as a structured generation and verification problem rather than as an unconstrained prompt-to-image step. Figure 1 organizes the implementation into



Figure 1: Overview of INVARIANTCANVAS through one audited revision. The request is compiled into linked requirements and an editable browser-rendered artifact. Evidence confirms a layout defect in c_0, but the first revision c_1 introduces a rendering regression and is rejected. A second revision c_2 is generated from the retained c_0, passes complete re-evaluation and independent delivery review, and is finalized. Compact side branches summarize clean delivery, dismissed false alarms, full regeneration, and late-defect re-entry.

five functional stages. A request is first decomposed into categorized requirements grounded in the request, with external sources used only when the request calls for additional content. A structured design specification then links each requirement to the visual elements intended to realize it and to an annotated HTML/SVG source. Chromium produces the rendered image together with Document Object Model (DOM) and geometric measurements, after which deterministic and learned verification methods record evidence for each artifact version. This evidence either enters adjudication or, when no mandatory requirement is known to be violated, proceeds directly to an independent delivery review. For each request, the framework maintains a structured ordered requirement collection $\mathcal{R}$, a design specification D, an editable source artifact A, its rendered image $I = R_\gamma(A)$, and evidence tied to each artifact version. The output therefore includes both the rendered image and traceable links from requested content to its intended visual realization and supporting evidence. We denote artifact version t by $c_t = (A_t, I_t, \gamma)$.

The revision path in Figure 1 begins only for an actionable defect: current-version evidence must support a violation of a mandatory requirement and identify what should change. Semantic defects concern requested content or relations, whereas rendering defects concern visible presentation, such as clipping or overlap. The central adjudicator may confirm or dismiss a finding, but it does not edit the source directly. A confirmed defect yields a targeted revision plan. All applicable verification methods are rerun on the new source and rendering. The revision replaces the preceding version only if the original criterion no longer fails and no previously passed mandatory requirement now fails; otherwise, the preceding version is retained. Both versions remain in the audit trail. A defect found by the independent delivery review re-enters the same process, whereas unavailable evidence supports neither acceptance nor an edit.

4.2 REQUEST-GROUNDED REQUIREMENT DECOMPOSITION

For a request p, the decomposition produces the ordered collection $\mathcal{R}$ of tracked requirements. Each requirement preserves the relevant request wording, expected visual property, applicable object and scope, whether it must be visible, and a suitable verification method. A stable identifier derived from this content links subsequent evidence to the source wording. We group requirements into content, quantity, appearance, layout, relation, and prohibition categories. Each item is evaluated separately, although the decomposition need not capture every implication of the request. A rule-based clause-representation check tests whether each request clause is represented. Passing this check does not establish semantic completeness or final-image correctness.

Two validation checks address especially consequential decomposition errors. Exact-text requirements retain quoted strings and punctuation rather than silently substituting a paraphrase. Distributive expressions retain both their count and scope: for example, "three labels for each series" is not weakened into three labels globally. A malformed decomposition is rejected. If another decomposition attempt is made, it receives the unchanged request together with the validation error, and its outcome is recorded separately from clause representation.

The requirement collection is then expanded into a structured design specification D. When possible, D maps each requirement r_i to one or more intended visual elements—for example, a title, table cell, chart mark, connector, or container—and to applicable verification methods. It also records the reading order, hierarchy, regions, layout, and stylistic constraints needed to coordinate the artifact. A shared numeric fact may link a caption, a table cell, and a chart mark; a relation may link a source element, target element, connector, and direction. These traceability links support construction and later inspection. They do *not* create additional user requirements: a deviation from a design choice proposed during planning is not, by itself, a violation of the request. Appendix A lists the fields retained for requirements, evidence, revision decisions, and run-level provenance.

4.3 STRUCTURED AUTHORING WITH MACHINE-READABLE ANNOTATIONS

An authoring model receives the request and structured design specification and emits one self-contained HTML document, using inline SVG where needed. This representation combines exact text, CSS layout, vector marks, tables, and connectors in a browser-renderable artifact. Machine-readable metadata link requirements to planned visual regions, entities, relations, and endpoints. Additional annotations identify chart marks and labels, table cells, relation endpoints, containers, and geometric anchors. These annotations encode the intended correspondence between requirements and elements; they are not, by themselves, evidence that the corresponding content is visible or correct in the rendered image.

The annotated source is rendered in Chromium to obtain the image and DOM-based geometric measurements. Element-to-requirement links make it possible to ask which element is intended to realize a requirement; rendered measurements make it possible to ask whether that element is visible, clipped, overlapping, or geometrically related as intended. Supported relations include containment, disjointness, left/right and above/below placement, explicit ordering, connector contact and direction, and requested relative size or proportion. The audit records the participating elements, their rendered boundaries, and the applicable absolute, relative, or ratio tolerance. When the request gives no tolerance, the defaults are one CSS pixel and 0.01 for normalized or ratio comparisons. Unsupported shapes or relations return UNKNOWN. Non-visible layout containers and guides are excluded from visible-shape measurements. The annotations alone do not constitute evidence: missing or ambiguous annotations produce an explicit evidence gap rather than an unsupported judgment of visual correctness.

An accepted HTML/SVG artifact may also be exported as a structured conditioning package for an external raster editor. The package contains the browser rendering, text and non-text geometry layers, and masks that distinguish protected text from editable regions. The external editor lies outside the INVARIANTCANVAS generation and acceptance loop. Its output creates new pixels, so evidence for the accepted rendering does not transfer; a separate pixel-domain review is required.

4.4 VERSION-SPECIFIC HETEROGENEOUS EVIDENCE

For artifact version $c_t = (A_t, I_t, \gamma)$ defined above, A_t is its editable HTML/SVG source, γ is the recorded rendering configuration, and $I_t = R_\gamma(A_t)$ is its rendered image. Each evidence record $e_{v,i}(c_t)$ is associated with that version and identifies the relevant requirement, linked visual element when available, evidence source, location, and provenance; its judgment is $z_{v,i}(A_t, I_t)$ from Eq. 1. Evidence from an earlier version is not assigned to a revision without rendering and evaluating that revision anew. We use four complementary families of verification methods:

1. **Browser-based deterministic checks** inspect clipping, overlap, visibility, exact strings and counts, consistency between tables and charts, connector endpoints and direction, whether connectors visibly touch their intended elements, and selected request-derived geometric relations.
2. **OCR** produces recognized text and bounding boxes and may flag potential omissions, mismatches, or readability problems. An OCR discrepancy remains evidence rather than a conclusive judgment about visible text.
3. **Source audit** inspects the HTML/SVG document for traceability links and structural facts that cannot be recovered reliably from pixels alone.
4. **Visual audit** examines the full current canvas and relevant crops of linked visual regions for hierarchy, legibility, omissions, and relations requiring perceptual context.

For a requirement r_i and verification method v, UNKNOWN means that the required observation is unavailable or ambiguous; it is neither a pass nor evidence of a visible defect. An UNKNOWN result neither authorizes a change nor silently supports satisfaction. Evidence sources remain distinct rather than being collapsed into one scalar score: a source-level link does not establish rendered legibility, and a visual assessment does not deterministically establish every small textual or numerical fact. We retain evidence-availability summaries for each verification method rather than an unqualified correctness score, so disagreement remains visible to adjudication.

4.5 EVIDENCE ADJUDICATION, REVISION, AND DELIVERY REVIEW

The evidence record supports a request-grounded audit of the current artifact version. The adjudicator receives the original request, the structured requirements, the full canvas, relevant region crops, and available evidence from the different verification methods. It may confirm a request-grounded defect, dismiss an unsupported finding or a design preference that was introduced during planning but not required by the request, request a targeted local revision, request full-document regeneration when the problem cannot be localized, accept the artifact, or terminate without accepting a result. The adjudicator does not edit the source directly. Any requested change is represented by a revision plan that specifies the affected elements, its basis in the request, supporting evidence, content that should be preserved, and the expected outcome.

For analysis, we retain the revision plan, the preceding and revised artifact versions, and the evidence for both. Complete re-evaluation reruns every applicable verification method on the revised source and rendering. A revision is accepted only if the targeted criterion no longer fails and no previously satisfied mandatory requirement becomes a failure. Otherwise, it remains in the audit trail without replacing the preceding version. This acceptance rule operates over explicitly recorded requirements and evidence, rather than over a universal image-quality score. It is therefore an operational policy over observed properties, not a proof that unobserved properties are preserved. We call this policy *transactional* in the limited sense that a proposed revision is accepted as a complete new version only after full re-evaluation; if it is rejected, the preceding accepted version remains unchanged.

Finally, an independent delivery review receives the original request, the structured requirements, and the exact selected image without the preceding revision history. It can accept the artifact, request another targeted revision, reject the artifact, or abstain when the evidence is insufficient. An UNKNOWN result neither justifies an edit nor silently supports acceptance. Another revision is requested only for a request-grounded defect or visible rendering defect supported by evidence from the current image. An accepted artifact is finalized together with its source, rendered image, and audit records for external evaluation.

5 EXPERIMENTS

5.1 BENCHMARKS AND METRICS

We use each benchmark's official protocol and scorer. BizGenEval contains 400 prompts spanning slides, charts, webpages, posters, and scientific figures, with Hard/Easy checks for layout, attribute binding, text, and knowledge (Li et al., 2026c). We also report evaluator-wide Question Accuracy, which is computed over the complete case set rather than by averaging the two split-level category means. IGenBench contains 600 cases across 30 infographic types and reports question-level accuracy (Q-ACC) and all-questions-correct infographic accuracy (I-ACC) (Tang et al., 2026). Their gap measures whether otherwise strong artifacts still contain an isolated error. Official scorers run only after finalization and do not participate in generation, revision, or candidate selection.

5.2 COMPARISON POOL AND IMPLEMENTATION

We compare the complete INVARIANTCANVAS configuration with each benchmark's closed- and open-source comparison pool. We report the benchmarks separately because their comparison sets and metrics differ; no cross-benchmark aggregate is formed. Fixed GPT-5.6 Sol snapshots perform decomposition, planning, initial authoring, adjudication, delivery review, and full regeneration, while GPT-5.6 Terra performs local revision and model-based audits (OpenAI, 2026). The delivery review is a fresh Sol call without revision history, so its independence is contextual rather than model-based. Chromium supplies rendered pixels and DOM geometry. Optional retrieval is used only when a request requires outside information, and the retrieved material is retained with the run. Each sample permits three revision attempts, up to two adjudicator-requested attempts, and one full regeneration. Model, inference, browser, font, canvas, dataset, and random-seed versions are recorded.

Evaluation protocol. Each official prompt is processed once under a fixed configuration, and only the finalized raster is submitted to the benchmark scorer. Official scorer outputs are unavailable during authoring, verification, and revision. Every candidate is rendered at the benchmark canvas size, and all applicable checks are rerun on the complete current version before it can replace the preceding accepted artifact. The delivery reviewer receives the original request, structured requirements, and selected image, but not prior critiques or revision history. This separates generation from external evaluation and makes the final review independent of the repair path. Attempt budgets, model snapshots, inference settings, browser and font environments, and case order are fixed across the corresponding benchmark; failed cases are retained rather than selectively rerun.

5.3 MAIN RESULTS

Table 1: BizGenEval results, reported as Hard/Easy (%); higher is better. The evaluator-wide Question Accuracy of INVARIANTCANVAS is 95.86% overall and 94.83% on Hard. PE denotes prompt enhancement.

Method	# Params	Layout	Attribute	Text	Knowledge	Macro avg.
Closed-source models						
GPT-Image-2	–	88.5/95.3	81.9/91.0	83.9/92.6	74.2/90.9	82.1/92.5
Nano-Banana-Pro	–	72.2/91.2	65.6/**92.2**	86.4/95.0	82.6/**96.2**	76.7/93.7
Qwen-Image-3.0	–	79.3/89.5	74.0/87.6	72.0/88.3	76.8/90.5	75.5/89.0
Nano-Banana-2.0	–	68.4/91.0	57.4/91.6	83.4/94.6	64.6/93.0	68.5/92.5
Seedream-5.0-Pro	–	75.5/87.6	74.1/87.3	63.2/81.1	49.8/75.3	65.7/82.8
Open-source models						
SenseNova-U1.5 + PE	8B	82.5/92.3	66.5/86.9	67.4/86.5	72.0/87.9	72.2/88.4
SenseNova-U1.5	8B	78.1/91.2	55.9/80.0	66.6/86.1	4.8/23.4	51.4/70.2
SenseNova-U1	8B	61.6/81.6	47.5/72.8	46.3/74.6	3.5/17.9	39.7/61.7
Emu3.5	32B	30.4/63.4	14.2/52.6	7.0/33.6	1.2/11.0	13.2/40.2
HunyuanImage-3.0	80B (13B active)	27.8/65.0	13.8/53.6	10.2/39.6	0.0/2.0	13.0/40.1
Z-Image	6B	26.8/69.2	2.6/47.6	2.8/45.0	0.6/13.2	8.2/43.8
Qwen-Image-2512	20B	22.2/70.6	1.2/47.8	1.8/39.2	0.0/6.4	6.3/41.0
FLUX.2-dev	32B	17.2/67.8	1.2/49.2	1.0/43.0	0.0/8.2	4.9/42.0
Qwen-Image	20B	10.4/51.2	0.2/22.2	0.6/17.6	0.0/4.4	2.8/23.8
Our complete system						
INVARIANTCANVAS	–	**93.4/95.4**	**89.4**/91.6	**91.0/96.0**	**85.2**/92.2	**89.8/93.8**

Table 2: IGenBench results; higher is better. Q-ACC measures question-level correctness, whereas I-ACC requires every question for an infographic to be correct. Comp., Enc., Anno., Leg., and Deco. abbreviate completeness, encoding, annotations/callouts, legend/category mapping, and decorative elements; PE denotes prompt enhancement.

Content and local structure

Method	# Params	Comp.	Enc.	Order	Marks	Anno.
Closed-source models						
Nano-Banana-Pro	–	.84	.86	.90	.87	.93
GPT-Image-2	–	.89	.61	.93	.87	.94
Qwen-Image-3.0	–	.65	.43	.77	.77	.84
Seedream-5.0-Pro	–	.64	.47	.71	.73	.77
Open-source models						
SenseNova-U1.5 + PE	8B	.65	.47	.76	.77	.82
SenseNova-U1.5	8B	.41	.33	.58	.56	.66
SenseNova-U1	8B	.27	.23	.49	.45	.54
Qwen-Image	20B	.10	.13	.19	.29	.43
Z-Image-Turbo	6B	.10	.16	.16	.25	.38
HiDream-I1	17B	.01	.03	.03	.10	.07
FLUX.1-dev	12B	.00	.03	.01	.08	.06
Our complete system						
INVARIANTCANVAS	–	**.93**	**.91**	**.93**	**.91**	**.96**

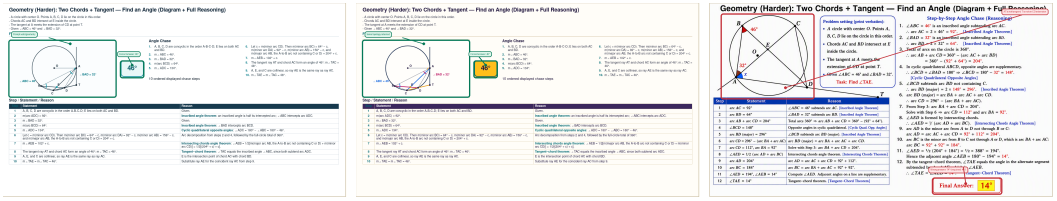
Global structure and overall accuracy

Method	# Params	Axes	Leg.	Chart	Title	Deco.	Q-ACC	I-ACC
Closed-source models								
Nano-Banana-Pro	–	.93	**.96**	.92	.98	.94	.90	.49
GPT-Image-2	–	.98	.95	.93	**.99**	**.95**	.88	.38
Qwen-Image-3.0	–	.84	.87	.85	.80	.88	.74	.13
Seedream-5.0-Pro	–	.85	.83	.83	.86	.88	.73	.17
Open-source models								
SenseNova-U1.5 + PE	8B	.82	.89	.84	.87	.92	.76	.17
SenseNova-U1.5	8B	.67	.83	.72	.83	.88	.62	.08
SenseNova-U1	8B	.61	.70	.65	.74	.82	.51	.04
Qwen-Image	20B	.37	.51	.48	.56	.78	.36	.01
Z-Image-Turbo	6B	.31	.58	.42	.61	.73	.35	.00
HiDream-I1	17B	.14	.10	.26	.19	.20	.11	.00
FLUX.1-dev	12B	.06	.01	.24	.09	.39	.10	.00
Our complete system								
INVARIANTCANVAS	–	**.99**	**.96**	**.95**	**.99**	.92	**.94**	**.62**

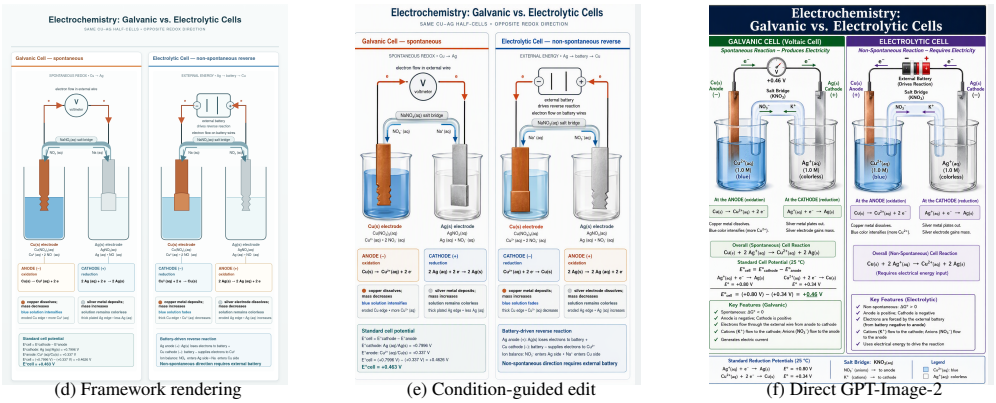
Overall performance. Tables 1 and 2 show that INVARIANTCANVAS achieves the best aggregate result among the reported open- and closed-source generators on both benchmarks. On the BizGenEval Hard split, its macro average is 89.8%, 7.7 percentage points above the strongest closed-source baseline, GPT-Image-2. The system leads every Hard category, with gains of 4.9 points in layout, 7.5 in attribute binding, 4.6 in text, and 2.6 in knowledge over the best competing score in each column. Its Easy macro average of 93.8% narrowly exceeds Nano-Banana-Pro at 93.7%, although the latter remains stronger on Easy attribute binding and knowledge by 0.6 and 4.0 points, respectively. The larger Hard-split margin indicates that the advantage is concentrated where several exact constraints must hold on the same artifact.

On IGenBench, INVARIANTCANVAS reaches 0.94 Q-ACC and 0.62 I-ACC, improving over Nano-Banana-Pro by 0.04 and 0.13, respectively. It leads or ties the best baseline on nine of ten diagnostic dimensions; decoration is the only dimension without a leading score. The substantially larger improvement in I-ACC is especially relevant because one failed question invalidates the entire infographic. This result pattern is consistent with the intended role of the verification harness: prompt requirements remain explicit across authoring, heterogeneous inspection, revision, and final acceptance, reducing the chance that individually plausible elements form an inconsistent whole. These full-system comparisons establish end-to-end effectiveness, while not isolating the contribution of any single component.

Representation-level comparison. Figure 2 compares framework renderings, external edits conditioned on those renderings, and independent direct generations. In the geometry example, the first two preserve the requested topology and answer. In the electrochemistry example, the conditioned edit preserves the reversed cell roles and outcomes more faithfully than direct generation while adding richer material appearance. Both the source and edit omit the positive charge on silver in two half-reactions.

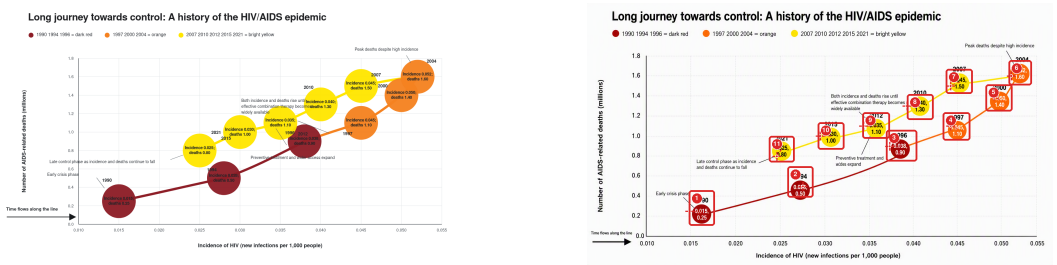


(a) Framework rendering (b) Condition-guided edit (c) Direct GPT-Image-2



(d) Framework rendering (e) Condition-guided edit (f) Direct GPT-Image-2

Figure 2: Representation-level comparisons on geometry (top) and electrochemistry (bottom). Each edited output is conditioned on the framework rendering to its left; the rightmost GPT-Image-2 output is generated independently.



(a) Source rendering (b) Raster edit

Figure 3: A post-edit data–position binding failure. Red overlays are analysis annotations. Although numeric labels remain visible, all 11 bubble centers in (b) are displaced from their axis-encoded coordinates in (a), showing why source evidence must be revalidated for edited pixels.

Failure analysis. Figure 3 isolates a visually plausible post-edit failure. The source places each of 11 bubbles at coordinates encoding date and life expectancy. The raster edit retains the labels and broad upward trend, yet shifts every center and breaks the data–position binding. A correct source therefore does not transfer correctness to edited pixels: geometric evidence must be recomputed and encoded relations checked before delivery. These edits are qualitative examples and are not included in the benchmark scores.

6 Conclusion

We presented InvariantCanvas, a verification harness linking request-grounded requirements to visual realizations and version-specific evidence. It admits revisions only after full re-evaluation and independent review of the delivered image. On BizGenEval and IGenBench, it outperforms the reported comparison pool, especially on the Hard and whole-infographic measures. The qualitative cases show that structured rendering can aid refinement, whereas edited pixels require fresh verification.

AI USE STATEMENT

During manuscript preparation, we used large language model (LLM) tools to refine the writing, assist with LaTeX formatting, and improve the clarity of technical descriptions. These tools also supported discussions of potential experimental controls and helped us review whether our claims were appropriately supported by the reported evidence. The authors made all decisions regarding experimental design, interpretation, and reporting, and take full responsibility for the paper's content.

ETHICS STATEMENT

By reducing the effort needed to create persuasive visual artifacts, the system may also facilitate the production of misleading charts, posters, or documents. Its verification procedures check adherence to the input request and consistency within the artifact; they do not establish the factual correctness, provenance, or suitability of the content for downstream use. Outputs intended for consequential applications should therefore be independently reviewed by domain experts.

REPRODUCIBILITY STATEMENT

Section 4 specifies the generation pipeline and the criteria for accepting revisions. Appendix A details the requirement, evidence, revision, and run-record schemas. For each run, we record the code and dataset revisions, the ordered list of evaluation cases, exact model identifiers, execution budgets, random seeds, and browser and font configurations. Run records also retain model calls, revision decisions, and delivery-review outcomes.

REFERENCES

Jonas Belouadi, Anne Lauscher, and Steffen Eger. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v3K5TVP8kZ.

Jonas Belouadi, Eddy Ilg, Margret Keuper, Hideki Tanaka, Masao Utiyama, Raj Dabre, Steffen Eger, and Simone Ponzetto. TikZero: Zero-shot text-guided graphics program synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17793–17806, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Belouadi_TikZero_Zero-Shot_Text-Guided_Graphics_Program_Synthesis_ICCV_2025_paper.html.

Jingye Chen, Zhaowen Wang, Nanxuan Zhao, Li Zhang, Difan Liu, Jimei Yang, and Qifeng Chen. Rethinking layered graphic design generation with a top-down approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16861–16870, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Rethinking_Layered_Graphic_Design_Generation_with_a_Top-Down_Approach_ICCV_2025_paper.html.

Nan Chen, Yuge Zhang, Jiahang Xu, Kan Ren, and Yuqing Yang. VisEval: A benchmark for data visualization in the era of large language models, 2024. URL https://arxiv.org/abs/2407.00981. arXiv preprint arXiv:2407.00981.

Qing Chen, Fuling Sun, Xinyue Xu, Zui Chen, Jiazhe Wang, and Nan Cao. VizLinter: A linter and fixer framework for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):206–216, 2022. doi: 10.1109/TVCG.2021.3114804.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Jinwoong Kim, Rui Yang, and Huishuai Zhang. GeoBuildBench: A benchmark for interactive and executable geometry construction from natural language, 2026. URL https://arxiv.org/abs/2605.13167. arXiv preprint arXiv:2605.13167.

Vishal Kumar, Shubhra Mishra, Rebecca Hao, Rizwaan Malik, David Broman, and Dorottya Demszky. DiagramIR: An automatic pipeline for educational math diagram evaluation, 2025. URL https://arxiv.org/abs/2511.08283. arXiv preprint arXiv:2511.08283; published at the Math-AI Workshop at NeurIPS 2025.

Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B. Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints. In *European Conference on Computer Vision*, pp. 491–506, 2020. doi: 10.1007/978-3-030-58580-8_29.

Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. LayoutGAN: Generating graphic layouts with wireframe discriminators. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJxB5sRcFQ.

Runze Li, Yukun Zhao, Can Xu, Yucheng Shen, Shuaiqiang Wang, Jianmin Wu, Lingyong Yan, and Dawei Yin. From transient prompts to persistent control: Scientific poster generation via recursive semantic-geometric contracts, 2026a. URL https://arxiv.org/abs/2609.17326.

Sifan Li, Yujun Cai, Hongkai Chen, and Yiwei Wang. GeoSVG-RL: Geometry-aware reinforcement learning for layout-constrained text-to-svg diagram generation, 2026b. URL https://arxiv.org/abs/2605.25447. arXiv preprint arXiv:2605.25447.

Yan Li, Zezi Zeng, Ziwei Zhou, Xin Gao, Muzhao Tian, Yifan Yang, Mingxi Cheng, Qi Dai, Yuqing Yang, Lili Qiu, Zhendong Wang, Zhengyuan Yang, Xue Yang, Lijuan Wang, Ji Li, and Chong Luo. BizGenEval: A systematic benchmark for commercial visual content generation, 2026c. URL https://arxiv.org/abs/2603.25732.

Yepeng Liu, Dasen Dai, Chengzhi Liu, Yiren Song, Hai Ci, Yu Zhang, Qi Zhang, Mike Zheng Shou, Xin Eric Wang, and Yuheng Bu. Figures as programs: Recursive generation of editable scientific figures, 2026a. URL https://arxiv.org/abs/2609.01006. arXiv preprint arXiv:2609.01006.

Ziyuan Liu, Shizhao Sun, Danqing Huang, Yingdong Shi, Meisheng Zhang, Ji Li, Jingsong Yu, and Jiang Bian. DesignAsCode: Bridging structural editability and visual fidelity in graphic design generation, 2026b. URL https://arxiv.org/abs/2602.17690. arXiv preprint arXiv:2602.17690.

Yaxin Luo, Haobin Jiang, Jialv Zou, Xu Huang, Wenhao Yan, Haodong Li, Zhengrong Yue, Jing Li, Xiaofu Chen, Xiaohan Zhao, Jiacheng Liu, Jiacheng Cui, Zhiqiang Shen, and Xiaotong Li. AutoDesign: Meta-harness optimization for long-horizon agentic design, 2026. URL https://arxiv.org/abs/2608.13560.

Dominik Moritz, Chenglong Wang, Greg L. Nelson, Halden Lin, Adam M. Smith, Bill Howe, and Jeffrey Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, 2019. doi: 10.1109/TVCG.2018.2865240.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026. Accessed 2026-09-26.

Geliang Ouyang, Jingyao Chen, Zhihe Nie, Yi Gui, Yao Wan, Hongyu Zhang, and Dongping Chen. nvAgent: Automated data visualization from natural language via collaborative agent workflow, 2025. URL https://arxiv.org/abs/2502.05036. arXiv preprint arXiv:2502.05036.

Bo Pan, Yixiao Fu, Ke Wang, Junyu Lu, Lunke Pan, Ziyang Qian, Yuhan Chen, Guoliang Wang, Yitao Zhou, Li Zheng, Yinghao Tang, Zhen Wen, Yuchen Wu, Junhua Lu, Biao Zhu, Minfeng Zhu, Bo Zhang, and Wei Chen. VIS-Shepherd: Constructing critic for LLM-based data visualization generation, 2025. URL https://arxiv.org/abs/2506.13326. arXiv preprint arXiv:2506.13326.

Yuyang Peng, Shishi Xiao, Keming Wu, Qisheng Liao, Bohan Chen, Kevin Lin, Danqing Huang, Ji Li, and Yuhui Yuan. BizGen: Advancing article-level visual text rendering for infographics generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23615–23624, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/

html/Peng_BizGen_Advancing_Article-level_Visual_Text_Rendering_for_Infographics_Generation_CVPR_2025_paper.html.

Tianfei Ren, Zhipeng Yan, Yiming Zhao, Zhen Fang, Yu Zeng, Guohui Zhang, Hang Xu, Xiaoxiao Ma, Shiting Huang, Ke Xu, Wenxuan Huang, Lionel Z. Wang, Lin Chen, Zehui Chen, Jie Huang, and Feng Zhao. SCOPE: Structured decomposition and conditional skill orchestration for complex image generation, 2026. URL https://arxiv.org/abs/2605.08043.

Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. StarVector: Generating scalable vector graphics code from images and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16175–16186, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Rodriguez_StarVector_Generating_Scalable_Vector_Graphics_Code_from_Images_and_Text_CVPR_2025_paper.html.

Jaskirat Singh and Liang Zheng. Divide, evaluate, and refine: Evaluating and improving text-to-image alignment with iterative VQA feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dfd0bd56e8a6f82d1619f5d093d5f9ca-Abstract-Conference.html.

Yinghao Tang, Xueding Liu, Boyuan Zhang, Tingfeng Lan, Yupeng Xie, Jiale Lao, Yiyao Wang, Haoxuan Li, Tingting Gao, Bo Pan, Luoxuan Weng, Xiuqi Huang, Minfeng Zhu, Yingchaojie Feng, Yuyu Luo, and Wei Chen. IGenBench: Benchmarking the reliability of text-to-infographic generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 36919–36954, 2026. doi: 10.18653/v1/2026.acl-long.1713. URL https://aclanthology.org/2026.acl-long.1713/.

Jingxuan Wei, Cheng Tan, Qi Chen, Gaowei Wu, Siyuan Li, Zhangyang Gao, Linzhuang Sun, Bihui Yu, and Ruifeng Guo. From words to structured visuals: A benchmark and framework for text-to-diagram generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13315–13325, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wei_From_Words_to_Structured_Visuals_A_Benchmark_and_Framework_for_CVPR_2025_paper.html.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2SVG: Vector graphics generation with large language models and image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23690–23700, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_Chat2SVG_Vector_Graphics_Generation_with_Large_Language_Models_and_Image_CVPR_2025_paper.html.

Junyan Ye, Jun He, Zilong Huang, Dongzhi Jiang, Xuan Yang, Rui Chen, and Weijia Li. GenClaw: Code-driven agentic image generation, 2026a. URL https://arxiv.org/abs/2605.30248.

Junyan Ye, Wei Liu, Dongzhi Jiang, Zichen Wen, Haodong Li, Zhutao Lv, Jiaxin Lin, Jinhua Yu, Jun He, Zilong Huang, Rui Chen, and Weijia Li. Editable visual design, 2026b. URL https://arxiv.org/abs/2609.04034. arXiv preprint arXiv:2609.04034.

Ruijie Ye, Jiayi Zhang, Zhuoxin Liu, Zihao Zhu, Siyuan Yang, Li Li, Tianfu Fu, Franck Dernoncourt, Yue Zhao, Jiacheng Zhu, Ryan Rossi, Wenhao Chai, and Zhengzhong Tu. Agent banana: High-fidelity image editing with agentic thinking and tooling, 2026c. URL https://arxiv.org/abs/2602.09084. arXiv preprint arXiv:2602.09084.

Hui Zhang, Dexiang Hong, Maoke Yang, Yutao Cheng, Zhao Zhang, Weidong Chen, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. CreatiDesign: A unified multi-conditional diffusion transformer for creative graphic design. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=Wtda8HpVp2.

Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Xiao Xu, Yan Shu, Yanran Zhang, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Huishuai Zhang, Dongyan Zhao, and Chenfei Wu. Qwen-Image-Agent: Bridging the context gap in real-world image generation, 2026b. URL https://arxiv.org/abs/2606.26907. arXiv preprint arXiv:2606.26907.

Haozhe Zhao, Shuzheng Si, Zhenhailong Wang, Zheng Wang, Liang Chen, Xiaotong Li, Zhixiang Liang, Maosong Sun, and Minjia Zhang. CRAFTER: A multi-agent harness for editable scientific figure generation from diverse inputs, 2026. URL https://arxiv.org/abs/2605.30611. arXiv preprint arXiv:2605.30611.

Dawei Zhu, Rui Meng, Yale Song, Xiyu Wei, Sujian Li, Tomas Pfister, and Jinsung Yoon. PaperBanana: Automating academic illustration for AI scientists, 2026a. URL https://arxiv.org/abs/2601.23265. arXiv preprint arXiv:2601.23265.

Minjun Zhu, Zhen Lin, Yixuan Weng, Panzhong Lu, Qiujie Xie, Yifan Wei, Sifan Liu, Qiyao Sun, and Yue Zhang. AutoFigure: Generating and refining publication-ready scientific illustrations. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=5N3z9JQJKq.

A REQUIREMENT, EVIDENCE, AND AUDIT RECORDS

Tracked requirement. Each requirement has a unique identifier and records a concise statement of the requested property, the corresponding request span and optional context, the entity to which it applies, its expected value and scope, any limiting conditions, whether it is mandatory or advisory, whether it must be visually present, and a suitable verification method. Exact text retains terminal punctuation; ordered values retain order and multiplicity. Distributive quantities record their scope explicitly.

Evidence record. An evidence record identifies the relevant requirements and visual elements, the verification method, status in $\{\text{PASS}, \text{FAIL}, \text{UNKNOWN}\}$, image checksum, measurements or localized observations, and any proposed revision. Browser-based, OCR, source-audit, and visual-audit evidence remain separate. Missing or ambiguous evidence cannot be interpreted as a pass.

Revision record. For each proposed revision, the audit records the targeted requirements, the evidence before and after the change, newly introduced or worsened semantic and rendering defects, the acceptance decision, and all reasons for rejection. Both the preceding and revised images remain available for analysis.

Run record. Each run record contains an experiment identifier, code revision, dataset revision, checksum of the ordered evaluation cases, random seed, exact model identifiers, revision budgets, browser version, canvas dimensions, and a checksum of the font package. For each artifact version, we retain the source document, rendered image, verification evidence, evidence-availability summary, adjudication outcome, revision and acceptance records, delivery-review record, and records of model calls. Missing evidence is written explicitly as UNKNOWN so it cannot be interpreted as a pass.