PT Jurnal Cendekia Indonesia
Journal of Information Systems and Business Technology
Homepage: https://journal.jci.co.id/jisbt
Vol. 02 No. 03 (2026) Page: 132-143
E-ISSN: 3109-8886

From Enterprise UI Screenshots to Trustworthy Front-End Prototypes: Structural–Visual Consistency, Accessibility, Progressive Rendering, and an Evidence Layer for LLM-Assisted Design Critique

Wei Dong[1], Sierra Campbell[2], Lei Wu[3], Gabriel Ross[4]

[1]Department of Computer Science, University of Rochester, Rochester, NY, USA,
[2]Department of Computer Science, Pennsylvania State University, University Park, PA, USA
[3]Department of Computer Science, The Ohio State University, Columbus, OH, US

Emai: [1*]w_dong_rochester@protonmail.com

Abstract

Screenshot-to-code systems are often judged by visual resemblance, yet deployable prototypes also require coherent DOM structure, accessible semantics, and predictable rendering. We evaluated these dimensions on 484 Design2Code screenshot–HTML pairs using deterministic screenshot features, static markup audits, and CSS-hash-grouped five-fold cross-validation. Random Forest and Extra Trees predicted DOM size with R^2 values of 0.215 and 0.244, respectively, but accessibility-burden R^2 remained below zero, confirming that visual fidelity cannot substitute for markup inspection. Train-fold-only visual retrieval scored 78.26 on a composite consistency measure; structural reranking scored 79.59, and a deterministic evidence reranker combining predicted structure, accessibility, and static critical-render-path profiles scored 79.63. Its 1.37-point gain over visual retrieval was significant (95% bootstrap confidence interval 0.92–1.86; one-sided Wilcoxon $p < 0.001$) and cost 0.36 visual-similarity points. Conservative remediation removed 218 of 1,981 detected issues, raised zero-issue pages from 59 to 79, and preserved normalized body text and embedded CSS on all pages. On a fixed 25-page diagnostic subset, mean SSIM relative to each page's complete-CSS render increased from 0.693 for semantic HTML to 0.784 with typography; the complete renders independently reached median SSIM 0.996 against the stored screenshots. These findings establish an auditable evidence layer for subsequent LLM-assisted critique rather than an evaluated language-model critic.

Keywords: Design2Code, Screenshot-To-Code, Multimodal AI, Front-End Generation, Accessibility, Progressive Rendering, Prototype Retrieval, LLM Design Critique, Structural–Visual Consistency.

Abstrak

Sistem screenshot-to-code sering kali dinilai berdasarkan kemiripan visual, namun prototipe yang siap diterapkan juga memerlukan struktur DOM yang koheren, semantik yang aksesibel, dan rendering yang dapat diprediksi. Kami mengevaluasi dimensi-dimensi ini pada 484 pasangan screenshot-HTML dari dataset Design2Code menggunakan fitur screenshot deterministik, audit markup statis, dan validasi silang lima lipatan (five-fold cross-validation) yang dikelompokkan berdasarkan hash CSS. Model Random Forest dan Extra Trees memprediksi ukuran DOM dengan nilai R^2 masing-masing sebesar 0,215 dan 0,244, namun nilai R^2 untuk aspek aksesibilitas tetap di bawah nol; hal ini mengonfirmasi bahwa fidelitas visual tidak dapat menggantikan pemeriksaan markup. Metode pengambilan visual (visual retrieval) yang hanya menggunakan data latih memperoleh skor 78,26 pada ukuran konsistensi gabungan; metode reranking struktural memperoleh skor 79,59, sedangkan metode reranking berbasis bukti deterministik—yang menggabungkan prediksi struktur, aksesibilitas, dan profil statis critical-render-path—memperoleh skor 79,63. Peningkatan sebesar 1,37 poin dibandingkan metode pengambilan visual tersebut terbukti signifikan (interval kepercayaan bootstrap 95%: 0,92–1,86; uji Wilcoxon satu sisi: $p < 0,001$) dengan konsekuensi penurunan 0,36 poin pada metrik kemiripan visual. Perbaikan konservatif berhasil mengatasi 218 dari 1.981 masalah yang terdeteksi, meningkatkan jumlah halaman bebas masalah dari 59 menjadi 79, serta mempertahankan teks isi (body text) yang telah dinormalisasi dan CSS yang disematkan (embedded CSS) pada semua halaman. Pada subset diagnostik tetap yang terdiri dari 25 halaman, nilai rata-rata SSIM—relatif terhadap hasil rendering dengan CSS lengkap untuk setiap halaman—meningkat dari 0,693 (untuk HTML semantik) menjadi 0,784 (dengan penerapan tipografi); hasil rendering lengkap tersebut secara independen mencapai nilai median SSIM 0,996 jika dibandingkan dengan screenshot aslinya. Temuan ini menetapkan lapisan bukti yang dapat diaudit untuk proses kritik lanjutan yang dibantu oleh LLM, alih-alih sekadar mengandalkan model bahasa sebagai kritikus yang dievaluasi.

Diterima Redaksi: 01-08-2026 | Selesai Revisi: 06-08-2026 | Diterbitkan Online: 07-08-2026

***Kata Kunci**: Design2Code, Screenshot-To-Code, AI Multimodal, Pembuatan Front-End, Aksesibilitas, Progressive Rendering, Pengambilan Prototipe, Kritik Desain LLM, Konsistensi Struktural-Visual.*

A. INTRODUCTION

Design2Code established a real-world benchmark for converting webpage screenshots into executable front-end implementations and paired 484 curated webpages with reference images and HTML [1]. Its Hugging Face conversion preserves each screenshot–HTML pair in a compact Parquet representation [2]. The pages were selected from the C4 validation corpus introduced for large-scale text-to-text transfer learning [3]. This lineage provides authentic webpage structure while keeping the present evaluation small enough for exhaustive static analysis, grouped cross-validation, and per-page error inspection.

Interface implementation has progressed from screenshot-to-sequence translation in Pix2code [4] and hand-drawn mockups in Sketch2Code [5] to richer webpages and framework code. A controlled sketch-to-prototype evaluation likewise separates executable HTML/CSS structure from rendered visual consistency [6], reinforcing that code shape and resemblance must be measured independently. WebSight supplies large-scale synthetic screenshot–HTML pairs [7], Web2Code joins page understanding with generation [8], and divide-and-conquer generation decomposes complex pages to reduce omissions [9]. Visual-critic training further shows that code candidates can be improved through image-conditioned feedback without performing inference-time rendering for every update [10]. Mobile-screen analytics has also linked observed interface complexity to alternative presentation modes [11], illustrating that visual structure can guide downstream design decisions rather than serve only as a similarity target. These advances enable rapid prototyping but leave a gap between visual resemblance and implementation trustworthiness.

A screenshot hides the DOM tree, landmarks, accessible names, reading order, CSS organization, and resource dependencies. Pix2Struct learns simplified markup from pixels [12], ScreenAI models UI localization and semantics [13], and WebSRC combines text, metadata, HTML, and imagery [14]. MarkupLM represents markup directly [15], while Graph4GUI models semantic and spatial relations [16]. These studies establish layout, structure, and semantics as related but non-identical interface views.

Design critique is the final trust layer. UICrit structures expert feedback around concrete interface properties [17], and guideline-grounded LLM feedback has been tested on mockups [18]. A UICrit-based study further evaluates whether LLM-authored UI critiques align with human graphic-design judgment [19]. Text-grounded rationale interfaces bind review statements to explicit layout metadata [20], while visual-brief frameworks expose hierarchy, intent, and risk as inspectable decisions [21]; designer-editable brief cards extend the same idea to structured pre-generation artifacts [22]. Self-Refine iterates feedback [23], whereas CRITIC grounds correction in external tools [24]. LLM judges also exhibit position, verbosity, and self-enhancement biases [25], and unguided self-correction remains unreliable [26]. A front-end critic therefore needs pixel, DOM, accessibility, and rendering evidence before producing advice. Rubric-grounded formative-feedback work in education provides a transferable auditability principle—structure the evidence before language-model verbalization [27]—without supplying a UI-specific rubric.

Evaluation has the same split. CLIP supplies a high-level visual representation [28], SSIM measures structural image similarity [29], LPIPS uses deep perceptual features [30], and CIEDE2000 formalizes color difference [31]. Work on distilling VMAF into an edge-deployable predictor illustrates the appeal of compact, machine-readable quality tokens [32], but such learned proxies remain task-dependent and were not used here. Assignment methods support block matching [33], while tree-edit distance compares DOM structures [34]. Design2Code combines text, position, color, and text-masked CLIP similarity [1], but no visual metric determines whether controls have accessible names, headings are coherent, or rendering work is avoidable.

Accessibility is especially resistant to pixel-only assessment. WCAG 2.2 defines testable outcomes [35], WAI-ARIA specifies accessible widget semantics [36], and axe-core automates only a subset of checks [37]. Trust also extends beyond accessibility: text-based dark-pattern research shows that manipulative intent can be carried by button labels, banners, and other interface microcopy [38]. Layout analysis can identify clipping and overlap [39]. Thus, a visually plausible prototype can remain inaccessible or behaviorally misleading, and static findings represent machine-detectable risks rather than WCAG conformance.

Rendering adds another dimension: Largest Contentful Paint marks principal-content appearance [40], while Speed Index integrates visual completeness [41]. Progressive PDF rendering likewise distinguishes early functional visibility from full completion [42], although document slices and CSS layers are different technical objects. Design2Code contains no network traces, so the browser experiment here uses offline CSS-layer reconstruction only and does not claim network speedups. Separating semantic markup, layout, typography, and complete CSS reveals visual contributions without inventing latency measurements.

An evidence layer also needs a presentation contract. Scientific-search cards separate claim anchors, source passages, ranking rationales, and uncertainty [43]. Medical explanation and patient-safety cards similarly separate visual evidence, calibrated risk, stated boundaries, and next actions [44], [45]; a separate response-interface benchmark treats risk presentation itself as an evaluable object [46]. E-commerce recommendation cards expose reasons and

confidence rather than a bare ranking [47], while review-grounded recommendation work evaluates whether displayed evidence actually supports the underlying score [48]. These domain studies do not validate screenshot-to-code fidelity. Their relevance is the recurring principle that measured evidence should remain inspectable after natural-language formatting.

The same pattern appears in visual decision support. Financial dashboards emphasize hierarchy, marked evidence, and visible caveats [49]; capacity-dashboard briefs translate forecast risk into compact design decisions [50]; and counseling-support cards make evidence and human review visible [51]. A therapist-facing copilot likewise ranks retrieved rationales for human decision makers rather than replacing them [52]. Together with structured advertising briefs, these examples motivate a designer-facing schema that keeps measurements, uncertainty, and recommended action in separate regions. The present experiment evaluates the evidence that could populate such a schema, not the usability of the schema itself.

Source grounding must also survive interface formatting. Accounting-review cards separate claims, source excerpts, citation trails, and review actions [53], while a multilingual humanitarian RAG interface makes evidence, answerability, abstention, and escalation visible as distinct states [54]. These patterns motivate explicit states for accepted, uncertain, and rejected front-end critiques; they do not supply fidelity measurements for the present corpus.

This study contributes a grouped evaluation of screenshot-to-HTML signal prediction, train-fold-normalized prototype retrieval with structural and trust-aware reranking, conservative accessibility remediation on all 484 documents, and staged CSS reconstruction on a fixed browser diagnostic subset. The deterministic reranker ranks candidates through measurable discrepancies, supplying an auditable evidence layer for subsequent language-model feedback. This retrieve-then-explain separation parallels integrated code search and summarization [55] and evidence-grounded RAG for private technical documents [56]. Evidence-chain work further motivates word-level source attribution [57] and deterministic provenance explanations [58], while calibrated abstention addresses cases with insufficient evidence coverage [59]. Natural-language critique and model-generated front ends are outside the evaluated pipeline.

B. METODE

Bagian ini pada dasarnya menjelaskan pelaksanaan dan A. Dataset and integrity controls

The Design2Code Parquet conversion [2] contained 484 rows with PNG bytes in image.bytes, HTML in text, and null image.path values. The 77,578,732-byte file had SHA-256 digest d56c4675ddccad8bdd1485cd9e34b352d1d2552edc603e9f7340db7165663c70. Every pair was nonempty, with no duplicate screenshot, raw HTML, or whitespace-normalized HTML hashes. The pages originate from the C4 validation corpus [1]–[3]. We created independent cross-validation folds, summarized with the measurement pipeline in Figure 1.

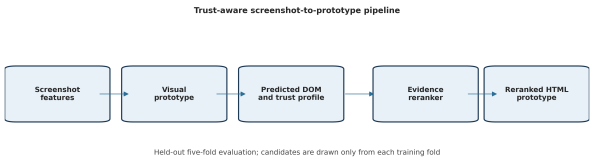

Figure 1 Evaluation pipeline. Screenshot descriptors support grouped prediction and train-fold-only prototype retrieval; static HTML audits and browser reconstruction supply independently measurable trust evidence.

Screenshots were decoded as RGB images. HTML was parsed with the recovery mode of lxml.html; all 484 documents produced a DOM. Embedded CSS was stripped of comments and whitespace, lowercased, hashed, and used as the grouping key. This procedure produced 471 style groups: 458 singletons and 13 two-page groups. Grouped five-fold cross-validation kept every identical normalized embedded-CSS hash within one fold, preventing exact style-hash matches from appearing simultaneously in training and testing.

Table 1 reports the complete corpus summary used by all static and cross-validated experiments.

Table 1 Descriptive Summary Of The 484 Design2code Pairs

Measure	Min.	Q1	Median	Mean	Q3	Max.
Screenshot width (px)	1,280.00	1,280.00	1,280.00	1,280.14	1,280.00	1,304.00
Screenshot height (px)	720.00	906.25	1,351.00	1,481.51	1,952.50	3,328.00
HTML size (KB)	1.70	22.09	61.57	71.79	109.11	235.47
DOM elements	12.00	81.75	133.00	158.24	217.00	528.00
DOM depth	3.00	8.00	10.00	11.48	14.00	31.00
Unique tag types	8.00	18.00	22.00	22.19	26.00	45.00
Text characters	32.00	774.50	1,357.50	1,414.01	1,983.50	3,339.00
Accessibility burden	0.00	4.73	10.79	15.04	19.67	100.00
CRP burden index	0.26	22.86	29.67	28.96	36.76	65.15

Most files referenced by relative image or CSS URLs were absent. Browser analysis therefore used a fixed diagnostic subset comprising row indices 3, 10, 18, 43, 58, 61, 85, 96, 108, 131, 142, 147, 169, 175, 186, 206, 214, 223, 236, 259, 301, 342, 392, 397, and 429. The identifiers and their order were frozen before staged comparisons and saved with the results; all 25 were retained, including pages with lower complete-render agreement. Network-blocked complete-

CSS Chromium renders achieved median SSIM 0.996 and pixel MAE 1.16/255 against stored screenshots. Browser findings describe this diagnostic subset, while static analyses used all 484 pages.

B. Screenshot, DOM, accessibility, and rendering features
Each screenshot was resized to 64 × 64 pixels. The 422-dimensional visual vector contained log geometry; RGB and luminance moments; luminance entropy; colorfulness; white/dark shares; gradients; three 12-bin color histograms; a 16 × 16 luminance thumbnail; 4 × 4 spatial RGB means; and horizontal/vertical luminance and edge projections. Ablations used geometry, color, layout, and combined features.

The HTML parser counted elements and unique tag types, measured maximum ancestor depth, and extracted normalized body text after excluding style, script, noscript, and template content. Counts were recorded for links, images, controls, headings, semantic landmarks, inline style attributes, and embedded style blocks. Table 2 summarizes the most frequent elements. These measurements supplied the structural target vector: DOM elements, maximum depth, unique tags, and normalized body-text characters per DOM element.

Table 2 Most Frequent Html Elements

Element	Corpus count	Pages	Pages (%)	Median when present
	17,790	477	98.6%	25.0
<a>	13,309	478	98.8%	22.0
	6,883	391	80.8%	13.0
	6,813	394	81.4%	9.0
<p>	3,413	444	91.7%	6.0
	2,078	387	80.0%	4.0
<img>	1,870	368	76.0%	3.0
<style>	1,181	484	100.0%	1.0
<td>	997	85	17.6%	8.0
<input>	995	227	46.9%	3.0
<h3>	759	206	42.6%	2.5
<h2>	709	249	51.4%	2.0
<h1>	536	382	78.9%	1.0
<button>	525	189	39.0%	2.0
<tr>	436	85	17.6%	4.0

The accessibility profile comprised ten deterministic checks: absent document language, missing or empty title, missing viewport metadata, images without an alt attribute, controls without a visible or programmatic name, actionable links without a name, heading-rank skips, absent H1, tables without header cells, and duplicate IDs. Empty alt="" was not counted as a missing alternative because it can correctly mark a decorative image [35], [37]. Heading skips and duplicate IDs were treated as risk signals, not automatic declarations of WCAG failure. The accessibility burden index divided detected issues by four document-level opportunities plus the applicable image, control, link, heading, table, and ID opportunities, then scaled the result to 0–100.

Table 3 gives the corpus prevalence and multiplicity of every check.

Table 3 Machine-Detectable Accessibility-Risk Prevalence

Check	Issues	Pages	Pages (%)	Median among affected
Missing Document Language	192	192	39.7%	1.0
Controls Missing Name	380	164	33.9%	2.0
Heading Rank Skips	187	157	32.4%	1.0
Images Missing Alt	590	149	30.8%	2.0
Missing Viewport Metadata	106	106	21.9%	1.0
Missing H1	102	102	21.1%	1.0
Duplicate Ids	245	67	13.8%	2.0
Tables Missing Headers	163	66	13.6%	1.0
Links Missing Name	12	7	1.4%	2.0
Missing Or Empty Title	4	4	0.8%	1.0

The static critical-render-path profile contained synchronous external scripts in head, blocking stylesheets, embedded head-style kilobytes, inline head-script kilobytes, HTML kilobytes, non-lazy images after the first image, and inline-style attributes. Table IV reports the measured component distributions. The CRP burden index was fixed before evaluation:

$$\min(100,\ 12S + 5L + 2\min(10,C) + 2\min(10,J) + 0.12\min(250,H) + 0.8\min(15,I) + 0.15\min(40,A)),$$

where S is synchronous head scripts, L is blocking stylesheets, C is head CSS kilobytes, J is head inline-script kilobytes, H is HTML kilobytes, I is non-lazy images after the first, and A is inline-style attributes. The index is a static workload indicator rather than a latency estimate.

Table 4 Static Critical-Render-Path Signals

Component	Total	Pages	Pages (%)	Median	Q3
Html Kb	34,746.3	484	100.0%	61.57	109.11
Head Style Kb	26,345.1	470	97.1%	35.21	85.12
Inline Style Attributes	4,590.0	337	69.6%	3.00	10.00
Nonlazy Images After First	1,362.0	264	54.5%	1.00	3.00
Blocking Stylesheets	11.0	5	1.0%	0.00	0.00
Sync Head Scripts	0.0	0	0.0%	0.00	0.00
Head Inline Script Kb	0.0	0	0.0%	0.00	0.00

Cross-Modal Prediction

Six HTML-derived targets were predicted from screenshot features: DOM element count, DOM depth, unique tag count, normalized body-text density, accessibility burden, and CRP burden. DOM count and text density were transformed with log(1+x) during fitting and converted

back before scoring. Four fixed models were evaluated: a training-fold mean baseline; Ridge regression with $\alpha = 10$ and standardized predictors and targets; a Random Forest with 350 trees, minimum leaf size 4, and 70% feature sampling; and Extra Trees with 350 trees, minimum leaf size 3, and 75% feature sampling. Every preprocessing operation was fit inside the training fold. The reported metrics are out-of-fold MAE, MAE divided by the target interquartile range, R^2, and Spearman correlation.

Screenshot-Conditioned Prototype Retrieval And Evidence Reranking

Prototype retrieval treated the HTML paired with a training screenshot as a candidate template for the held-out design. It did not claim to reproduce the held-out page's text or content. For each held-out query, the candidate pool contained only pages in that query's training fold. The methods were a centroid representative, color nearest neighbor, layout nearest neighbor, complete visual-feature nearest neighbor, structural reranking, evidence reranking, and a Top-25 oracle. Visual distance and every similarity normalization scale were estimated within the training fold. Structural and evidence methods reranked the 25 nearest visual candidates.

An Extra Trees regressor with 400 trees predicted the query's structural, accessibility, and static CRP profiles from screenshot features. Structural reranking minimized $0.70\, d\_v + 0.30\, d\_s$. The evidence reranker minimized $0.55\, d\_v + 0.25\, d\_s + 0.10\, d\_a + 0.10\, d\_r$, where d_v, d_s, d_a, and d_r are normalized visual, structural, accessibility, and static CRP gaps. The Top-25 oracle selected the candidate with the highest measured composite score from the same 25 candidates, making it a per-query upper bound for these rerankers. It did not contribute to model fitting. The evidence reranker was deterministic; no language model contributed to candidate selection or any reported metric. Retrieval-quality prediction research treats relevance, redundancy, and diversity as distinct diagnostics [60]; in this study, the reported composite remains an explicit evaluation objective rather than a learned estimate of downstream generation quality.

Candidate quality was scored on six 0–100 scales. Visual similarity combined 32 × 32 luminance similarity (50%), RGB histogram intersection (30%), and edge-profile similarity (20%). DOM similarity was cosine similarity over 38 tag counts. Structural similarity normalized absolute differences in DOM count, depth, unique tags, and text density by the training fold's 95th-percentile pairwise difference. Accessibility and CRP-profile similarity used the same train-fold-only normalization on their respective profiles. Composite consistency weighted visual 35%, DOM 30%, structure 15%, accessibility 10%, and CRP profile 10%. These weights were fixed before retrieval and express the study's enterprise-trust objective rather than a universal preference function.

Accessibility Remediation

The remediation pass copied information already present in each document and did not generate new prose. It added standard viewport metadata when missing and promoted existing form-field prompt text to an accessible control name. It did not guess document language, image purpose, heading hierarchy, table semantics, or missing titles. The audit was rerun after transformation. Evaluation recorded issue count, zero-issue pages, normalized body-text identity, embedded-CSS identity, and serialized byte change.

Progressive CSS Reconstruction

The 25 diagnostic pages were reconstructed cumulatively: semantic HTML and browser defaults; layout declarations; typography; and complete CSS. Statement at-rules were removed from intermediate stages, font-face rules entered with typography, and conditional wrappers retained only around filtered inner declarations. Chromium 149 rendered 100 full-page screenshots at 1× device scale with network blocked. Intermediate renders were aligned to the complete-CSS canvas and measured by SSIM [29], PSNR, pixel MAE/RMSE, edge F1, RGB histogram intersection, and serialized bytes. Complete CSS was the within-page reference rather than an additional scored treatment. This CSS-layer experiment is separate from the static CRP profile used in retrieval and does not estimate network performance or Web Vitals.

Statistical Analysis

All randomized operations used seed 202407. Model comparisons used the same grouped folds. Evidence-reranker gains over visual retrieval were evaluated with 10,000 paired bootstrap resamples and a one-sided Wilcoxon signed-rank test. The 95% confidence interval was the 2.5th to 97.5th percentile of bootstrap mean differences. Complexity analysis divided pages into equal-frequency low, medium, and high groups by DOM element count. Per-page predictions, selections, scores, render metrics, audit results, and feature matrices were retained for every reported aggregate.

C. RESULTS AND DISCUSSION

Corpus Structure And Trust Signals

The corpus was visually and structurally diverse. Screenshot width was almost constant: 479 of 484 pages were 1,280 pixels wide, while height ranged from 720 to 3,328 pixels. Median HTML size was 61.57 KB, median DOM size was 133 elements, and median maximum depth was 10. The screenshots therefore encoded substantial variation through full-page height and spatial density rather than viewport width. HTML size and element count were right-skewed, as shown in Figure 2.

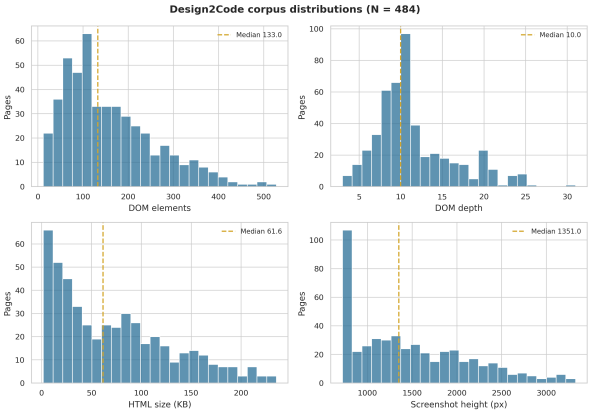

Figure 2 Distributions Of Screenshot Height, HTML Size, DOM Size, And DOM Depth Across 484 Screenshot–HTML Pairs.

The markup remained dominated by conventional containers and navigation elements. The corpus contained 17,790 div elements, 13,309 anchors, 6,883 list items, 6,813 spans, 3,413 paragraphs, and 1,870 images. Semantic HTML was present but less frequent: 131 pages contained main, 243 contained nav, and 221 contained footer. This imbalance matters for generative systems because visual grouping can be reproduced with generic containers even when the semantic hierarchy is weak.

Static accessibility risks were common and heterogeneous. Missing document language affected 192 pages (39.7%). Unnamed controls affected 164 pages (33.9%), heading-rank skips affected 157 (32.4%), and absent image alternatives affected 149 (30.8%). Missing viewport metadata affected 106 pages, while missing H1 affected 102. The image audit found 590 img elements without an alt attribute; explicit empty alternatives were not counted. Only seven pages contained an actionable, unnamed link because most anchor elements in the sanitized benchmark no longer carried an href attribute. Figure 3 makes the main pattern clear: risks requiring markup knowledge were not rare edge cases.

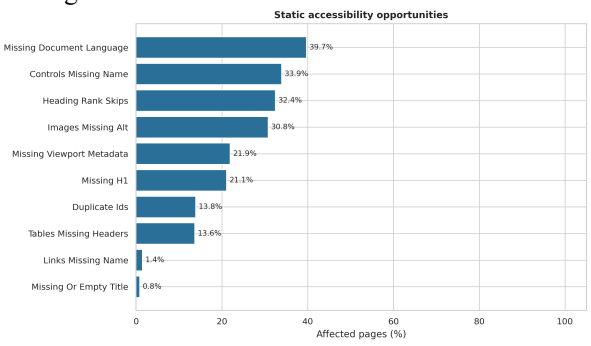

Figure 3 Prevalence Of Ten Machine-Detectable Accessibility-Risk Checks Across The Corpus.

The CRP profile reflected Design2Code's static construction. No document contained a script, and only five pages used a blocking external stylesheet. In contrast, 470 pages contained embedded head CSS, 337 used inline style attributes, and 264 contained at least two images without lazy-loading markup. Consequently, the CRP index primarily measured document and inline-style burden rather than JavaScript execution. Figure 4 shows that HTML size, embedded CSS, and non-lazy images dominated the measured profile. This distribution is appropriate for controlled HTML/CSS analysis but narrower than a production single-page application.

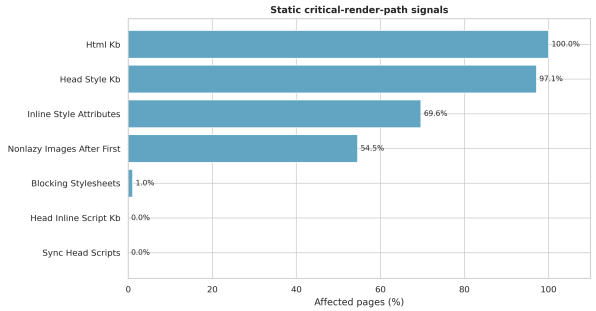

Figure 4 Static Critical-Render-Path Signals. Embedded Head CSS, Document Size, Inline Styles, And Non-Lazy Images Dominate This Script-Free Corpus.

Screenshot-To-HTML Signal Prediction

Table V reports grouped out-of-fold results. Random Forest achieved the best mean R^2 across the six targets (0.145), followed closely by Extra Trees (0.143). Extra Trees predicted DOM count with MAE 62.12 elements and R^2 0.244, depth with MAE 3.30 and R^2 0.153, and unique tag count with MAE 4.01 and R^2 0.258. The corresponding Spearman correlations were 0.615, 0.422, and 0.516, showing useful rank information despite substantial absolute uncertainty. CRP burden was moderately predictable (R^2 0.189), while normalized body-text density was weakly predictable (R^2 0.039).

Table 5 Grouped Five-Fold Screenshot-To-Html Prediction

Target	RF MAE	RF R^2	RF ρ	ET MAE	ET R^2	ET ρ
DOM elements	62.95	0.215	0.618	62.12	0.244	0.615
DOM depth	3.31	0.161	0.446	3.30	0.153	0.422
Unique tags	3.98	0.265	0.547	4.01	0.258	0.516
Text density	4.89	0.047	0.407	4.92	0.039	0.361
Accessibility burden	11.27	-0.034	0.041	11.21	-0.028	0.073
CRP burden index	8.10	0.214	0.470	8.27	0.189	0.440

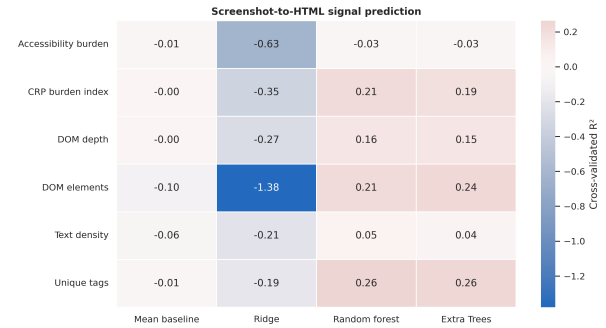

Figure 5 Visualizes The Target-Level R^2 Values For All Four Predictors.

Figure 5. Grouped five-fold prediction performance by model and HTML-derived target. Negative R^2 values indicate performance below the training-fold mean.

Accessibility burden was not predictable from pixels: Extra Trees produced R^2 −0.028 and Random Forest produced R^2 −0.034. This negative result is central rather than incidental. Language attributes, alternative text, control labels, header cells, and duplicate IDs have no necessary pixel signature. A model can infer that a page is complex, yet it cannot establish whether its code exposes accessible semantics. Screenshot-to-code evaluation therefore requires an HTML audit even when the visual score is high. The ablation in Table VI clarifies what the visual models learned. Layout features gave the strongest DOM-count R^2 (0.249), depth R^2 (0.165), and unique-tag R^2 (0.278). Geometry alone reached DOM-count R^2 0.204 because screenshot height tracked the amount of page content. Color features were weaker for structure but gave the best text-density R^2, 0.068. The combined vector did not consistently exceed layout alone, which indicates that additional color and texture dimensions added variance without supplying reliable semantic evidence. Figure 5 also exposes Ridge regression's instability on the high-dimensional features: its DOM R^2 was −1.379 even though its rank correlation was positive. Tree ensembles handled the nonlinear height, density, and edge relationships more effectively.

Table 6 Extra Trees Feature Ablation (R^2)

Target	Geometry	Color	Layout	Combined
DOM elements	0.204	0.075	0.249	0.244
DOM depth	0.065	0.054	0.165	0.152
Unique tags	0.197	0.100	0.278	0.259
Text density	-0.074	0.068	0.026	0.041
Accessibility burden	-0.056	-0.021	-0.045	-0.029
CRP burden index	0.066	0.092	0.187	0.190

Prototype Retrieval And Evidence-Based Reranking

Table 7 gives the complete retrieval comparison. Visual fusion selected a highly similar training-fold screenshot and achieved visual similarity 86.81, but its DOM, structural, accessibility, and static CRP-profile scores were 72.30, 66.88, 78.64, and 82.88. The resulting composite was 78.26. Structural reranking raised DOM similarity to 74.91 and structural similarity to 71.40, producing a composite of 79.59. The evidence reranker produced a similar visual score of 86.45, DOM 74.48, structure 71.38, accessibility 79.61, and CRP-profile similarity 83.60. Its composite score, 79.63, was the best learned result. The strict Top-25 oracle reached 86.50 and never fell below the learned reranker on an individual query, leaving 6.88 points of mean selection headroom within the same candidate set.

Table 7 Train-Fold-Only Prototype Retrieval

Method	Visual	DOM	Structure	A11y	CRP	Composite
Centroid representative	73.90	73.32	66.73	80.52	82.50	74.17
Color nearest	86.09	73.52	66.81	77.93	82.07	78.21
Layout nearest	85.06	71.79	66.43	78.65	83.01	77.44
Visual fusion	86.81	72.30	66.88	78.64	82.88	78.26
Structural rerank	86.51	74.91	71.40	78.84	82.40	79.59
Evidence reranker	86.45	74.48	71.38	79.61	83.60	79.63
Top-25 oracle	84.84	90.38	82.32	85.02	88.48	86.50

Figure 6 compares every retrieval method across the displayed visual, DOM, structural, and weighted-composite scores; the complete six-score values are listed in Table VII.

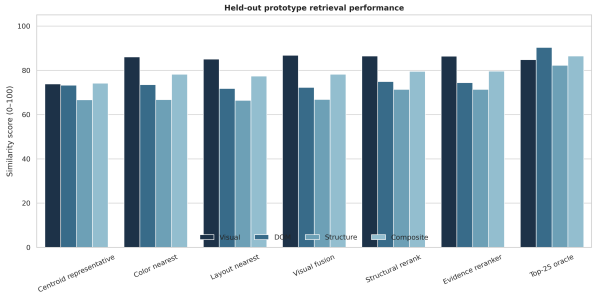

Figure 6 Prototype-Retrieval Consistency Across The Displayed Visual, DOM, Structural, And Weighted-Composite Dimensions.

Relative to visual fusion, the evidence reranker improved DOM similarity by 2.19 points (95% CI 1.07–3.33), structure by 4.51 (3.18–5.88), accessibility by 0.96 (0.08–1.84), static CRP-profile consistency by 0.72 (0.10–1.35), and composite consistency by 1.37 (0.92–1.86). The composite Wilcoxon test yielded $p = 1.37 \times 10^{-8}$; the accessibility and CRP-profile signed-rank p-values were 0.031 and 0.014. Table VIII therefore supports a strong structural gain and smaller but positive accessibility and static CRP-profile gains.

Table 8 Evidence Reranker Minus Visual Fusion

Metric	Mean Δ	CI low	CI high	Win rate	Wilcoxon p
DOM	2.187	1.068	3.325	19.2%	2.37×10^-4
Structure	4.505	3.175	5.877	23.3%	8.28×10^-11
Accessibility	0.963	0.080	1.844	15.1%	0.031
CRP profile	0.718	0.096	1.349	18.6%	0.014
Composite	1.374	0.922	1.862	22.7%	1.37×10^-8

The evidence reranker changed the nearest visual candidate for 152 of 484 queries (31.4%). Among changed selections, composite consistency improved in 72.4% and structural similarity improved in 74.3%. The mean visual score fell by 0.36 points because the reranker deliberately exchanged a small amount of pixel resemblance for code-level consistency. Figure 7 shows the tradeoff over the visual, structural, and full-trust stages. The largest gain occurred when structure first entered the objective; accessibility and static CRP evidence then produced a smaller additional composite improvement.

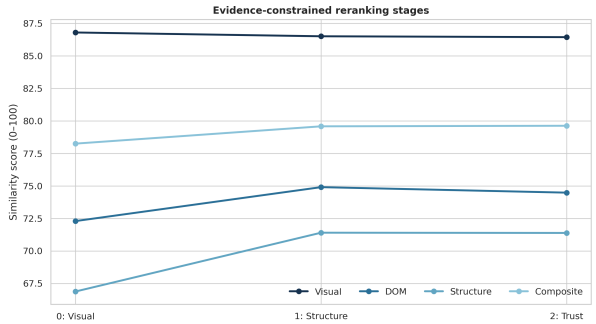

Figure 7 Three-Stage Candidate Selection: Visual Fusion, Structural Reranking, And The Complete Evidence Reranker

Complexity-stratified results were stable. The evidence reranker improved composite consistency over visual fusion from 78.16 to 79.57 for low-complexity pages, from 78.87 to 79.98 for medium pages, and from 77.74 to 79.34 for high-complexity pages. High-complexity structure remained hardest: its evidence-reranker structural score was 66.99, compared with 73.66 and 73.53 in the low and medium groups. Table IX indicates that the method did not obtain its aggregate gain from a single easy stratum.

Table 9 Composite Retrieval By Dom-Complexity Stratum

Complexity	Visual composite	Structural composite	Evidence composite	Evidence structure
Low	78.16	79.55	79.57	73.66
Medium	78.87	79.84	79.98	73.53
High	77.74	79.37	79.34	66.99

Figure 8 shows top-viewport retrievals for three queries with large composite gains. The examples demonstrate why evidence-based candidate selection is useful before natural-language feedback is generated: a visually close page can use a substantially different information hierarchy, while a slightly less similar screenshot can supply a more appropriate structural prototype.

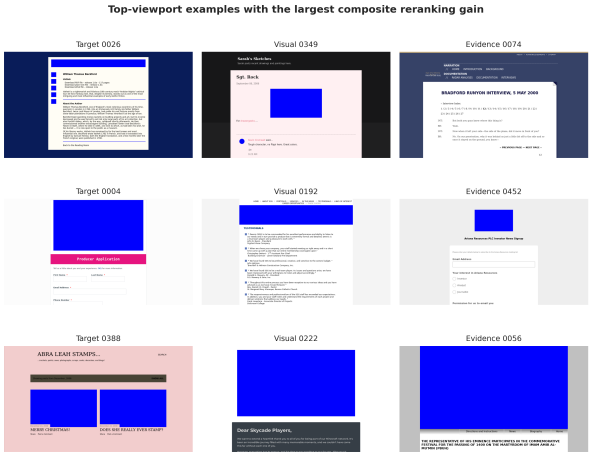

Figure 8 Top-Viewport Retrieval Cases With Large Composite Gains. The Evidence Reranker Accepts A Small Visual Tradeoff When The Alternative Prototype Better Matches The Query's Measured Structure And Trust Profile.

Accessibility Remediation

The initial audit counted 1,981 machine-detectable issues, or 4.09 per page. The conservative transformation removed 218 issues (11.0%), reducing the mean to 3.64 while the median remained 2. Zero-issue pages increased from 59 to 79. The transformation inserted 106 viewport declarations and promoted 112 existing form-field prompts to control names. It deliberately left all 192 missing language attributes and 590 missing image alternatives unchanged because neither page language nor image purpose could be established safely from the available evidence. Mean serialized markup increased by only 21.55 bytes. Table X gives every preservation and issue-count measure, while Figure 9 contrasts the before-and-after issue distribution and zero-issue rate.

Table 10 Conservative Accessibility Remediation

Auditable issues	1,981.00	1,763.00	-218.00
Mean issues per page	4.09	3.64	-0.45
Median issues per page	2.00	2.00	0.00
Zero-issue pages	59.00	79.00	20.00
Mean markup bytes	73,466.89	73,488.44	21.55
Normalized-body-text-preserved pages	484.00	484.00	0.00
CSS-preserved pages	484.00	484.00	0.00

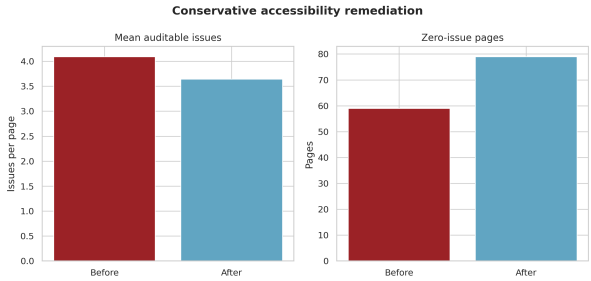

Figure 9 Accessibility-Risk Profile Before And After Conservative Semantic Remediation.

Normalized body text was identical, and embedded CSS was unchanged on every page. These preservation checks distinguish targeted semantic remediation from a broad rewrite that might change appearance or content. The remaining 1,763 issues were also informative: the procedure intentionally left document language, image alternatives, heading hierarchy, absent H1 elements, table semantics, duplicate IDs, and missing titles unresolved when the source did not contain enough evidence for a safe correction. A later LLM stage could propose richer repairs for those cases, provided that its output is validated against the same explicit audit profile and inspected for meaning. At the interface level, risk-card designs for LLM-agent oversight show how source trust, data sensitivity, and action consequences can be surfaced before a consequential step [61]. Before any language-model proposal is applied, evidence-based response self-verification offers a separate output gate [62]; that gate was not part of the current deterministic remediation.

Progressive Reconstruction

The 25 complete-CSS local renders closely reproduced the stored reference screenshots at the median: SSIM was 0.996, MAE was 1.16/255, and RGB histogram intersection was 0.993. Per-page SSIM ranged from 0.711 to 0.999, showing that browser or resource differences affected a minority of pages. Every staged comparison therefore used that page's own complete-CSS local render

as its reference, isolating the CSS-layer experiment from any stored-screenshot mismatch. Results appear in Table XI.

Table 11 Progressive Css Reconstruction On 25 Pages

Stage	Bytes	Share	SSIM	PSNR	MAE	Edge F1	Color
Semantic DOM	9,882.2	49.2 %	0.693	20.22	43.34	0.108	0.679
Layout	13,388.3	66.6 %	0.713	20.39	43.67	0.190	0.678
Typography	14,688.6	73.1 %	0.784	22.83	40.19	0.570	0.679
Complete CSS (reference)	20,097.8	100.0 %	—	—	—	—	—

Semantic HTML alone reached mean SSIM 0.693 and color intersection 0.679, but edge F1 was only 0.108. Layout declarations raised SSIM to 0.713 and edge F1 to 0.190. Adding typography produced the largest intermediate improvement: SSIM reached 0.784, PSNR 22.83 dB, and edge F1 0.570. Color intersection remained near 0.679 through the typography stage, confirming that backgrounds, borders, and explicit colors were concentrated in declarations reserved for the complete-CSS reference. Figure 10 displays the progressive reconstruction pattern without treating that reference as an additional finding.

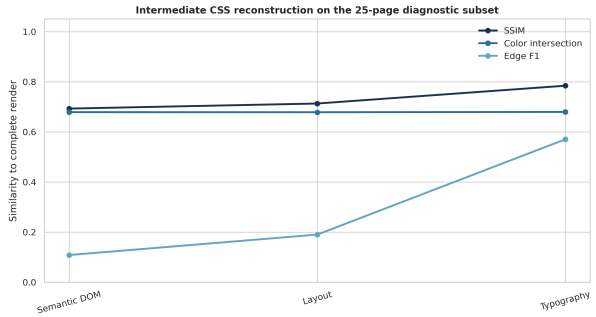

Figure 10 Three intermediate CSS reconstruction stages on the fixed 25-page diagnostic subset, measured against each page's complete-CSS reference.

All 100 browser renders completed; execution time varied with hardware load and was excluded from interpretation. The experiment shows which style layers recover appearance relative to the complete-CSS canvas. Typography contributed more than layout declarations alone on this fixed diagnostic subset, while complete decoration remained necessary for final colors and edges. A progressive generation system can use this ordering to expose a structurally valid page early, stabilize typography next, and defer fine visual decoration without claiming that the intermediate page is visually complete.

Implications And Limitations

The results support a separation of responsibilities. Visual encoders are useful for selecting a neighborhood of plausible layouts. A learned cross-modal model then estimates hidden structural properties. Static analyzers supply accessibility and CRP facts that pixels cannot reveal, while the browser experiment separately measures CSS-layer recovery. In the proposed integration, an LLM would receive those facts as evidence for critique, explanation, and code refinement. This sequence aligns with tool-grounded critique [24] and with evidence-constrained incident cards that keep raw signals visible beside recommended action [63]. Claim-verification results also show that auxiliary evidence-ranking signals can improve candidate ordering without replacing a strong first-stage retriever [64]. The architecture therefore avoids treating an LLM judge as an independent source of truth [25], [26], and gives engineering teams a traceable reason for rejecting a visually attractive prototype.

Three limitations bound the conclusions. First, Design2Code contains reference pages but no generated model outputs, execution traces, natural-language critiques, or human ratings for the candidates selected here. Retrieval measures prototype selection, not end-to-end code generation or an evaluated LLM critic. Second, most relative assets are absent from the Parquet conversion, so staged browser analysis used a fixed 25-page diagnostic subset that is not claimed to represent the full corpus. Third, the accessibility audit covers machine-detectable markup patterns and cannot establish WCAG conformance. Human review, keyboard testing, screen-reader evaluation, responsive breakpoints, and live resource loading remain necessary before deployment. Accordingly, the reported scores do not establish end-to-end deployment readiness. A future critic would also read page text and DOM content as untrusted input, so guardrails require continual red-teaming under changing attacks [65]. Visible privacy/data-integrity oversight [61] and behavior-level refusal controls [66] add distinct safeguards. A deterministic DevOps-copilot study likewise illustrates that generated-looking advice should remain subordinate to explicit policy checks [67]; no such agent was evaluated in this study.

D. CONCLUSIONS

Trustworthy screenshot-to-code requires more than a high visual match. On 484 Design2Code pairs, screenshot features predicted broad DOM complexity but did not predict accessibility burden. Train-fold-only retrieval produced strong visual prototypes, and deterministic evidence reranking improved structural and composite consistency with a small visual tradeoff. Conservative markup remediation removed 11.0% of detected risks without changing normalized body text or CSS. On the fixed 25-page diagnostic subset, staged reconstruction showed that semantic HTML and layout provide an early scaffold and typography raised mean SSIM to 0.784 relative to the complete-CSS reference.

The evaluated evidence layer is designed to sit between visual generation and LLM feedback. Its decisions are grounded in measured screenshot, DOM, accessibility, and static CRP profiles; the strict Top-25 oracle gap shows the remaining selection headroom; and its transformations are checked for preservation. Browser reconstruction supplies a separate view of CSS-layer recovery. This architecture

can turn design critique from an unconstrained opinion into an auditable step in front-end prototyping. Future evaluation should pair the evidence layer with named multimodal code generators, retain their raw HTML and critique traces, collect expert preferences, and execute complete pages under controlled network and assistive-technology conditions. If repair is made agentic, tool-level trajectory reliability should be evaluated before accepting the final artifact [68] rather than inferred from fluent output alone.

E. DAFTAR PUSTAKA

[1] C. Si, Y. Zhang, R. Li, Z. Yang, R. Liu, and D. Yang, "Design2Code: Benchmarking multimodal code generation for automated front-end engineering," in Proc. NAACL, 2025, pp. 3956–3974, doi: 10.18653/v1/2025.naacl-long.199.

[2] SALT-NLP, "Design2Code-hf," Hugging Face Datasets, 2024. [Online]. Available: https://huggingface.co/datasets/SALT-NLP/Design2Code-hf

[3] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.

[4] T. Beltramelli, "pix2code: Generating code from a graphical user interface screenshot," in Proc. EICS, 2018, Art. no. 3, doi: 10.1145/3220134.3220135.

[5] A. Robinson, "Sketch2Code: Generating a website from a paper mockup," arXiv:1905.13750, 2019.

[6] Y. Chen and M. Li, "From Hand-Drawn Sketches to Interactive Web Prototypes: A Reproducible Vision-Language Approach with Structural and Visual Consistency Evaluation," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 364–384, Aug. 2025, doi: 10.51903/jtie.v4i2.490.

[7] H. Laurençon, L. Tronchon, and V. Sanh, "Unlocking the conversion of web screenshots into HTML code with the WebSight dataset," arXiv:2403.09029, 2024.

[8] S. Yun et al., "Web2Code: A large-scale webpage-to-code dataset and evaluation framework for multimodal LLMs," in Advances in Neural Information Processing Systems, 2024.

[9] Y. Wan et al., "Divide-and-conquer: Generating UI code from screenshots," Proc. ACM Softw. Eng., vol. 2, FSE, Art. FSE094, pp. 2099–2122, 2025, doi: 10.1145/3729364.

[10] D. Soselia, K. Saifullah, and T. Zhou, "Learning UI-to-code reverse generator using visual critic without rendering," arXiv:2305.14637, 2023.

[11] J. Zhang, "Adaptive User Interface Design for Volleyball Learning Apps: Empirical Evidence from Google Play Reviews and Mobile Screen Analysis," Int. J. Graph. Des., vol. 3, no. 1, pp. 175–195, May 2025, doi: 10.51903/ijgd.v3i1.3618.

[12] K. Lee et al., "Pix2Struct: Screenshot parsing as pretraining for visual language understanding," in Proc. ICML, 2023, pp. 18893–18912.

[13] G. Baechler et al., "ScreenAI: A vision-language model for UI and infographics understanding," in Proc. IJCAI, 2024, doi: 10.24963/ijcai.2024/339.

[14] X. Chen et al., "WebSRC: A dataset for web-based structural reading comprehension," in Proc. EMNLP, 2021, pp. 4173–4185, doi: 10.18653/v1/2021.emnlp-main.343.

[15] J. Li, Y. Xu, L. Cui, and F. Wei, "MarkupLM: Pre-training of text and markup language for visually rich document understanding," in Proc. ACL, 2022, doi: 10.18653/v1/2022.acl-long.420.

[16] Y. Jiang, C. Zhou, V. Garg, and A. Oulasvirta, "Graph4GUI: Graph neural networks for representing graphical user interfaces," in Proc. CHI, 2024, doi: 10.1145/3613904.3642822.

[17] P. Duan, C.-Y. Chen, G. Li, B. Hartmann, and Y. Li, "UICrit: Enhancing automated design evaluation with a UI critique dataset," in Proc. UIST, 2024, doi: 10.1145/3654777.3676381.

[18] P. Duan, J. Warner, Y. Li, and B. Hartmann, "Generating automatic feedback on UI mockups with large language models," in Proc. CHI, 2024, doi: 10.1145/3613904.3642782.

[19] Y. Li, S. Lu, and L. Zhao, "LLM-as-Design-Critic: Aligning AI-Generated UI Feedback with Human Graphic Design Judgment," Int. J. Graph. Des., vol. 3, no. 1, pp. 196–215, May 2025, doi: 10.51903/ijgd.v3i1.3661.

[20] Q. Wu, S. Meng, and J. Zhao, "Text-Grounded LLM-Assisted Design Rationale Interfaces: Turning Advertising Layout Metadata into Explainable UI/UX Decision Cards," Int. J. Graph. Des., vol. 3, no. 1, pp. 216–240, May 2025, doi: 10.51903/ijgd.v3i1.3713.

[21] H. Tu, S. Zhao, and A. Zhou, "Visual Brief Cards for Advertising Design: A Structured UI/UX Framework for Turning Creative Intentions into Graphic Design Decisions," Int. J. Graph. Des., vol. 3, no. 1, pp. 210–226, May 2025, doi: 10.51903/ijgd.v3i1.3714.

[22] J. Mu, Y. Lu, and E. Hwang, "Structured Visual Brief Interfaces for Advertising Design: A UI/UX Framework for Turning Creative Intentions into Designer-Editable Graphic Design Cards," Int. J. Graph. Des., vol. 4, no. 1, pp. 192–208, Apr. 2026, doi: 10.51903/ijgd.v4i1.3702.

[23] A. Madaan et al., "Self-Refine: Iterative refinement with self-feedback," in Advances in Neural Information Processing Systems, 2023.

[24] Z. Gou et al., "CRITIC: Large language models can self-correct with tool-interactive critiquing," in Proc. ICLR, 2024.

[25] L. Zheng et al., "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena," in Advances in Neural Information Processing Systems, 2023, pp. 46595–46623.

[26] J. Huang et al., "Large language models cannot self-correct reasoning yet," in Proc. ICLR, 2024.

[27] Q. Xin, "Auditable Automated Essay Scoring and Formative Feedback: A Rubric-Grounded Pipeline

for Secondary and Higher Education," J. Appl. Artif. Intell. Educ., vol. 2, no. 1, pp. 1–19, Jul. 2026, doi: 10.66053/jaaie.v2i1.348.

[28] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. ICML, 2021, pp. 8748–8763.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[30] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in Proc. CVPR, 2018, pp. 586–595, doi: 10.1109/CVPR.2018.00068.

[31] M. R. Luo, G. Cui, and B. Rigg, "The development of the CIE 2000 colour-difference formula: CIEDE2000," Color Res. Appl., vol. 26, no. 5, pp. 340–350, 2001, doi: 10.1002/col.1049.

[32] B. Zhou, H. Wang, and X. Chang, "Distilling VMAF into an Edge-Deployable Quality Predictor: A Pilot Shot-Level Proxy with LLM-Ready Quality Tokens," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 447–463, Aug. 2025, doi: 10.51903/jtie.v4i2.522.

[33] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 4, pp. 1679–1696, 2016, doi: 10.1109/TAES.2016.140952.

[34] M. Pawlik and N. Augsten, "Tree edit distance: Robust and memory-efficient," Inf. Syst., vol. 56, pp. 157–173, 2016, doi: 10.1016/j.is.2015.08.004.

[35] W3C, "Web Content Accessibility Guidelines (WCAG) 2.2," Oct. 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/

[36] W3C, "Accessible Rich Internet Applications (WAI-ARIA) 1.2," Jun. 2023. [Online]. Available: https://www.w3.org/TR/wai-aria-1.2/

[37] Deque Systems, "axe-core: Accessibility engine for automated Web UI testing," 2024. [Online]. Available: https://github.com/dequelabs/axe-core

[38] H. Xu, Y. Chen, and A. Med, "Automatic Detection and Explanation of Dark Patterns from Interface Microcopy: Empirical Comparison of BERT-Style Encoders, RoBERTa-Style Encoders, and LLM-Style Decoders on the ec-darkpattern Dataset," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 590–612, Dec. 2025, doi: 10.51903/jtie.v4i3.491.

[39] P. Panchekha, A. T. Geller, M. D. Ernst, Z. Tatlock, and S. Kamil, "Verifying that web pages have accessible layout," in Proc. PLDI, 2018, pp. 1–14, doi: 10.1145/3192366.3192407.

[40] Y. Weiss and N. Peña Moreno, "Largest Contentful Paint," W3C Working Draft, Sep. 2023. [Online]. Available: https://www.w3.org/TR/largest-contentful-paint/

[41] W. Liu, X. Yang, H. Lin, Z. Li, and F. Qian, "Fusing Speed Index during web page loading," Proc. ACM Meas. Anal. Comput. Syst., vol. 6, no. 1, Art. no. 23, 2022, doi: 10.1145/3511214.

[42] H. Wang, Y. Ren, and X. Chang, "Layout-Aware Progressive PDF Rendering: AI Prioritization of PDF Slices to Reduce Time-to-Functional-First-Frame on FUNSD," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 425–446, Aug. 2025, doi: 10.51903/jtie.v4i2.523.

[43] J. Jin, "LLM-Style Evidence Cards for Scientific Search Interfaces: A UI/UX Design Framework for Retrieval Transparency, Ranking Trust, and Visual Evidence Hierarchy," Int. J. Graph. Des., vol. 3, no. 2, pp. 397–414, Oct. 2025, doi: 10.51903/ijgd.v3i2.3698.

[44] Z. S. Zhong, Q. Wu, and G. Mi, "Uncertainty-Aware Medical Image Explanation Cards: LLM-Generated Visual Explanations for AI-Assisted Radiology Interfaces," Int. J. Graph. Des., vol. 3, no. 2, pp. 415–436, Oct. 2025, doi: 10.51903/ijgd.v3i2.3616.

[45] B. Zhou, C. Li, and L. Liu, "Risk-Calibrated Patient-Facing AI Safety Cards: A UI/UX Design Framework for Rubric-Based Medical Risk Communication," Int. J. Graph. Des., vol. 3, no. 2, pp. 365–380, Oct. 2025, doi: 10.51903/ijgd.v3i2.3696.

[46] C. Li, B. Zhou, and K. Gao, "Risk-Calibrated Patient-Facing AI Safety Cards: A UI/UX Benchmark for Explainable Medical AI Response Interfaces," Int. J. Graph. Des., vol. 3, no. 2, pp. 381–394, Oct. 2025, doi: 10.51903/ijgd.v3i2.3709.

[47] B. Zhang, Y. Ren, and J. Zou, "LLM-Style Explainable E-Commerce Recommendation Cards: A UI/UX Design Framework for Trust-Calibrated Product Recommendation," Int. J. Graph. Des., vol. 3, no. 2, pp. 381–396, Oct. 2025, doi: 10.51903/ijgd.v3i2.3697.

[48] X. Chang, Y. Lu, and Z. S. Zhong, "Review-Grounded Explainable Recommendation with Faithfulness Evaluation on Amazon Reviews," J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 9–22, Jun. 2026, doi: 10.54732/jeecs.v11i1.2.

[49] Z. Li, S. Zhou, and Z. Zhou, "Financial Risk Dashboard Design for Institutional RWA Investors: Visual Hierarchy, Chart Comprehension, and Explainability in FinChart-Bench," Int. J. Graph. Des., vol. 3, no. 1, pp. 196–210, May 2025, doi: 10.51903/ijgd.v3i1.3715.

[50] G. Liu, S. He, and H. Wong, "LLM-Compatible Visual Brief Cards for AI Infrastructure Capacity Dashboards: A UI/UX Framework for Turning Forecast Risk into Graphic Design Decisions," Int. J. Graph. Des., vol. 3, no. 1, pp. 196–213, May 2025, doi: 10.51903/ijgd.v3i1.3723.

[51] Y. Zhang and H. Zhang, "Visualizing the Right Counseling Support: Evidence-Linked Recommendation Cards for Explainable Mental Health Intake Interfaces," Int. J. Graph. Des., vol. 3, no. 1, pp. 214–229, May 2025, doi: 10.51903/ijgd.v3i1.3722.

[52] Y. Zhang and H. Zhang, "A Therapist-Facing Session Copilot for Live Counseling Support: Reasoning-Guided Retrieval and Ranking from Multi-Turn Counseling Dialogues," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 464–486, Aug. 2025, doi: 10.51903/jtie.v4i2.547.

[53] K. Zhang, Y. Chen, and A. Qian, "Evidence-Grounded Accounting Disclosure Review Cards: A Visual Communication Framework for LLM-Style Explanations over SEC Financial Statements and Notes," Int. J. Graph. Des., vol. 3, no. 2, p. 395, Oct. 2025, doi: 10.51903/ijgd.v3i2.3710.

[54] Y. Chen and H. Xu, "Trust-Calibrated Multilingual RAG for Humanitarian Information Platforms: Empirical Evaluation on OMoS-QA for Migration Information Access," Int. J. Graph. Des., vol. 4, no. 1, pp. 141–164, Apr. 2026, doi: 10.51903/ijgd.v4i1.3552.

[55] Y. Li, "Findable then Explainable: Retrieval–Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset," J. Adv. Comput. Syst., vol. 4, no. 7, pp. 65–82, Jul. 2024, doi: 10.69987/JACS.2024.40706.

[56] B. Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," J. Adv. Comput. Syst., vol. 4, no. 3, pp. 109–125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[57] C. Li, J. Bai, and S. Wang, "Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications," J. Adv. Comput. Syst., vol. 4, no. 2, pp. 76–92, Feb. 2024, doi: 10.69987/JACS.2024.40207.

[58] J. Jin, "Evidence-Chain Reliable RAG: Hallucination Detection, Source Attribution, and Deterministic Provenance Explanations," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 520–533, Aug. 2025, doi: 10.51903/jtie.v4i2.535.

[59] Z. S. Zhong, J. Chen, E. Zhong, and X. Sun, "Evidence-Calibrated RAG for Unanswerable Question Answering: Retrieval Coverage, Abstention Calibration, and Hallucination-Proxy Analysis on SQuAD 2.0," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 502–520, Aug. 2025, doi: 10.51903/jtie.v4i2.536.

[60] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality Analysis and Evaluation Prediction of RAG Retrieval Based on Machine Learning Algorithms," arXiv preprint arXiv:2511.19481, 2025, doi: 10.48550/arXiv.2511.19481.

[61] W. Su, H. Rao, and E. Ma, "Privacy and Data-Integrity Risk Cards for LLM Agents: A UI/UX Design Framework for Secure Human Oversight under Prompt-Injection Attacks," Int. J. Graph. Des., vol. 4, no. 1, pp. 186–191, Apr. 2026, doi: 10.51903/ijgd.v4i1.3699.

[62] D. Zheng, B. Zhang, and J. Geibel, "VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 67–82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[63] J. Nie, G. Liu, C. Li, and T. Zou, "Evidence-Constrained Incident Visualization Cards for Distributed Cloud Logs: A UI/UX Framework for Turning Hadoop, OpenStack, and ZooKeeper Logs into Actionable SRE Design Interfaces," Int. J. Graph. Des., vol. 4, no. 1, pp. 179–185, Apr. 2026, doi: 10.51903/ijgd.v4i1.3703.

[64] W. Su, S. Chen, and E. Qian, "Narrative-Aware Scientific Claim Verification Agent with Evidence Ranking for ClimateCheck," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 327–340, Apr. 2026, doi: 10.51903/jtie.v5i1.549.

[65] D. Zheng, C. Li, and H. Davidson, "Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation," J. Adv. Comput. Syst., vol. 3, no. 2, pp. 35–49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[66] D. Zheng and C. Li, "Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 83–99, Jan. 2024, doi: 10.69987/JACS.2024.40107.

[67] B. Zhang, X. Sun, G. Liu, and B. Zhou, "LLM-Style DevOps Copilot for Cloud-Native Troubleshooting: Retrieval-Augmented Runbook Generation and Command-Safety Evaluation," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 104–118, Aug. 2026, doi: 10.51903/jtie.v5i2.534.

[68] Z. S. Zhong, C. Li, and H. Rao, "Trajectory Reliability Prediction for Generalist AI Agents: Tool-Use Failure Analysis and Success Forecasting on ZClawBench," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 341–360, Apr. 2026, doi: 10.51903/jtie.v5i1.539.