AUV-BENCH: AESTHETIC UNDERSTANDING AND GENERATION EVALUATION FOR USER INTERFACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal foundation models are increasingly used for evaluating and generating user interfaces (UIs), often producing seemingly reasonable aesthetic judgments and visually plausible pages. However, under professional design scrutiny, their behavior can differ substantially from that of human designers. In professional design practice, designers rely on a systematic set of aesthetic principles that consistently guide judgment, diagnosis, repair, and creation. A coherent aesthetic capability should therefore connect aesthetic judgment with design actions. Existing evaluations, however, typically assess these abilities in isolation, making it difficult to determine whether task-level success reflects a shared aesthetic understanding or merely fragmented task-specific competence. To address this gap, we introduce AUV-BENCH, developed in collaboration with professional UI designers around 1,395 executable web interfaces and four tasks: aesthetic scoring, diagnosis, repair, and text-to-UI generation. The tasks share a pool of UIs and aesthetic principles, with diagnosis and repair further aligned on 660 controlled-degradation instances to enable instance-level analysis of judgment and action. Evaluation of 12 models reveals a capability imbalance: models show moderate agreement with professional designers in holistic aesthetic scoring, yet exact diagnosis-chain success peaks at only 24.7%. On the aligned diagnosis–repair cases, correct judgments and successful repairs do not consistently coincide, exposing a *Judgment–Action Gap* between identifying aesthetic problems and successfully acting on them. In open-ended generation, even leading models achieve only moderate aesthetic quality under human-calibrated evaluation. Overall, current models exhibit partial aesthetic competence, but still lack the fine-grained understanding and judgment–action coherence required for reliable UI design.

1 INTRODUCTION

Multimodal foundation models are increasingly taking on the dual roles of "critics" and "designers" of user interfaces (UIs): they can evaluate the visual quality of interfaces (Duan et al., 2024; Chen et al., 2024) and generate visually plausible webpages from visual or natural-language requirements (Si et al., 2025; Awal et al., 2025). Yet under professional design scrutiny, both their aesthetic judgments and generated interfaces still exhibit substantial gaps from those of experienced human designers (Huang et al., 2024b; An et al., 2026a). More importantly, evaluating these abilities in isolation cannot reveal whether they are supported by a shared and transferable understanding of aesthetic principles, or merely reflect task-specific competence. In professional design practice, experienced designers internalize accumulated aesthetic knowledge and design experience into stable judgment criteria, and consistently apply them across interface evaluation, problem diagnosis, design repair, and open-ended creation (Cross, 2004; Lawson, 2004). Therefore, a model with mature UI aesthetic capability should not merely perform well on individual tasks; its aesthetic knowledge should transfer across tasks and consistently connect design judgment with corresponding design actions. We define this consistency as *Aesthetic Judgment–Action Coherence*. This raises a central question: *Have current multimodal models developed a coherent capability that connects aesthetic judgment with design action?*

Existing evaluations are not yet well suited to systematically answer this question. As shown in Table 1, prior work has primarily followed two directions: one focuses on UI generation, editing, and front-end implementation, evaluating visual fidelity, functional correctness, and executability (Lin

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.

Table 1: Comparison of existing benchmarks along key dimensions for evaluating UI aesthetic judgment and design action. **Expert Aesthetic GT** indicates whether aesthetic ground truth is provided or validated by professional designers or relevant domain experts. **Principle-Grounded** denotes explicit evaluation against predefined aesthetic design principles, while **Controlled Violations** indicates controlled manipulation of specific aesthetic principles. **Judgment–Action Linkage** indicates whether aesthetic diagnosis and corrective action are directly paired on the same interface and controlled aesthetic violation, enabling instance-level analysis of judgment–action coherence. ✓ and ✗ denote support and non-support, respectively.

Benchmark	Benchmark Setup		Principle Grounding		Core Capabilities				Judgment–Action Linkage	Statistics	
	Executable UI	Expert Aesthetic GT	Principle-Grounded	Controlled Violations	Aesthetic Scoring	Aesthetic Diagnosis	Aesthetic Repair	Text-to-UI Generation		Domain	Scale
Benchmarks for UI Generation											
Design2Code (Si et al., 2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	Web UI	484 pages
WebUIBench (Lin et al., 2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	Web UI	21K QA / 0.7K+ sites
WebMMU (Awal et al., 2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	Web UI	8.1K tasks / 2,059 pages
WebGen-Bench (Lu et al., 2026)	✓	✗	✗	✗	✗	✗	✗	✓	✗	Web App	647 tests
WebCoderBench (Liu et al., 2026)	✓	✗	✗	✗	✗	✗	✗	✓	✗	Web App	1,572 requirements
UI-Bench (Jung et al., 2025)	✓	✓	✗	✗	✗	✗	✗	✓	✗	Web UI	300 sites / 4K+ judgments
Benchmarks for Aesthetic Evaluation											
UICrit (Duan et al., 2024)	✗	✓	✗	✗	✓	✓	✗	✗	✗	Mobile UI	983 UIs / 3,059 critiques
DesignProbe (Lin et al., 2024)	✗	✓	✗	✗	✗	✗	✗	✗	✗	Graphic Design	∼1.6K questions
PhotoBench (Qi et al., 2025)	✗	✗	✗	✗	✓	✗	✗	✗	✗	Photography	1.5K questions
AesEval-Bench (An et al., 2026a)	✗	✗	✓	✓	✓	✓	✗	✗	✗	Graphic Design	4.5K QA / ∼1.2K designs
AesGuide (Du et al., 2026)	✗	✓	✗	✗	✓	✓	✗	✗	✗	Photography	1K eval / 10.7K total
AUV-BENCH	✓	✓	✓	✓	✓	✓	✓	✓	✓	**Web UI**	**1,395 base UIs**

et al., 2025; Liu et al., 2026; Lu et al., 2026); the other focuses on aesthetic understanding, evaluating scoring, design feedback, and defect localization (Wang et al., 2026; Lin et al., 2024; Du et al., 2026). However, these evaluations typically treat performance on individual tasks as separate endpoints. Successfully generating a visually reasonable interface does not directly indicate that a model can reliably judge its design quality; likewise, accurately identifying an aesthetic issue does not guarantee that the corresponding judgment can guide an effective repair. Studying this connection requires evaluation to be organized around shared aesthetic principles, with judgment and corrective action aligned on the same interface and target defect. Existing evaluations provide limited support for such principle-level and instance-level linkage, making it difficult to determine whether success across different tasks reflects the consistent use of aesthetic design knowledge (Imteyaz et al., 2026; An et al., 2026a).

To address this gap, we introduce AUV-BENCH, an **A**esthetic **U**nderstanding and generation e**V**aluation for UIs. An overview of AUV-BENCH is shown in Figure 1. We work with experienced UI designers to organize systematic aesthetic principles, recruit 51 professional designers for human annotation, and construct a shared data foundation containing 1,395 executable Web UIs. Based on this foundation, *Aesthetic Scoring* and *Aesthetic Diagnosis* evaluate aesthetic understanding, while *Aesthetic Repair* and *Text-to-UI Generation* evaluate design action.

To further study *Aesthetic Judgment–Action Coherence*, we use controlled degradations to align diagnosis and repair on the same webpage, and introduce *Judgment–Action Association* (JAA) to measure the instance-level association between judgment correctness and repair success. Our systematic evaluation reveals clear imbalances across aesthetic capabilities: frontier models show moderate agreement with expert judgments in holistic aesthetic scoring, while precise defect attribution and localization remain more challenging; correct judgments and successful repairs are generally positively associated, yet they do not form a stable correspondence, revealing a clear *Judgment–Action Gap*. In open-ended generation, even relatively strong models achieve only moderate aesthetic quality under human-calibrated evaluation, further distinguishing relative model advantages from design quality measured on a professional rating scale. Our main contributions are as follows:

- **A unified benchmark for UI aesthetic understanding and design action.** Unlike existing benchmarks that evaluate aesthetic understanding or UI generation in isolation, AUV-BENCH evaluates scoring, diagnosis, repair, and text-to-UI generation over 1,395 executable web interfaces, enabling unified evaluation from judgment to action.
- **A principle-grounded protocol for cross-task aesthetic evaluation.** We operationalize professional aesthetic principles into task-specific evaluation criteria and 13 controlled source-level violations with executable verification. Diagnosis and repair are further paired on 660 controlled-degradation instances, enabling direct instance-level measurement of whether aesthetic judgments translate into corrective actions.
- **New empirical insights into the limits of current UI aesthetic capabilities.** Models show moderate holistic aesthetic judgment, yet exact fine-grained diagnosis peaks at only 24.7%. More

importantly, successful repair does not consistently coincide with correct explicit judgment, while strong relative generation performance still yields only moderate absolute aesthetic quality, revealing substantial gaps between apparent task competence and coherent aesthetic understanding.

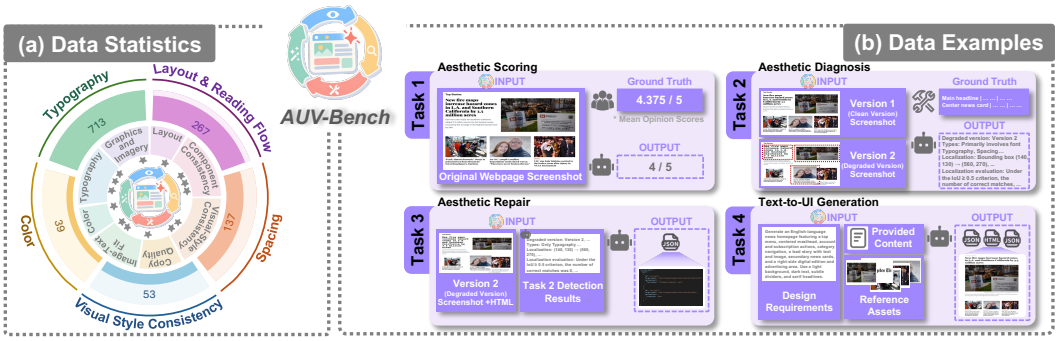

Figure 1: Overview of AUV-BENCH. (a) Data Statistics and (b) representative Data Examples of four tasks spanning aesthetic understanding (*Aesthetic Scoring* and *Aesthetic Diagnosis*) and design action (*Aesthetic Repair* and *Text-to-UI Generation*).

2 RELATED WORK

UI Generation and Front-End Development. Early works such as Rico, pix2code, and GUI skeleton generation established large-scale UI modeling and visual-to-code generation (Deka et al., 2017; Beltramelli, 2018; Chen et al., 2018). More recent systems, including Design2Code, Sketch2Code, and UI2Code, advance webpage reconstruction, code generation, and iterative refinement (Si et al., 2025; Li et al., 2025; Yang et al., 2025; Wan et al., 2025). WebUIBench, FullFront, WebMMU, and DesignBench broaden evaluation to UI understanding and general front-end capabilities (Lin et al., 2025; Sun et al., 2025; Awal et al., 2025; Xiao et al., 2025; He et al., 2026), while FrontendBench, WebGen-Bench, and WebCoderBench emphasize executable and functional correctness (Zhu et al., 2025; Lu et al., 2026; Liu et al., 2026). More recently, UI-Bench evaluates the design quality of generated interfaces, and Design Theater studies whether design reasoning supports downstream implementation (Jung et al., 2025; Imteyaz et al., 2026). These works increasingly extend UI evaluation from reconstruction and functionality toward design quality and action.

Aesthetic and Design Evaluation. Computational aesthetics has traditionally focused on holistic preference prediction, as exemplified by AVA and NIMA (Murray et al., 2012; Talebi & Milanfar, 2018). With multimodal models, AesBench evaluates aesthetic understanding, while DesignProbe and GPT-based evaluation examine reasoning over design properties and principles (Huang et al., 2024b; Lin et al., 2024; Haraguchi et al., 2024; Lin et al., 2026). UICrit introduces structured and localized UI critiques, while DesignPref studies individual visual-design preferences (Duan et al., 2024; Peng et al., 2025; Ko et al., 2026). Recent benchmarks move toward finer-grained analysis: AesEval-Bench covers aesthetic scoring and defect localization, UXBench evaluates detailed UI and UX reasoning, and UXBench-Actionability measures whether critiques support downstream repair (An et al., 2026a; Mao et al., 2026; Wang et al., 2026; Jeon et al., 2026). However, these works typically evaluate aesthetic scoring, diagnosis, repair, or generation separately, or connect only a subset of them. AUV-BENCH instead places all four capabilities under shared professional aesthetic principles, enabling systematic analysis of how aesthetic judgment translates into design action.

3 AUV-BENCH

This section presents the design of AUV-BENCH, including its task formulation, benchmark construction, and evaluation protocol.

3.1 OVERVIEW

AUV-BENCH evaluates UI aesthetics through four complementary capabilities built on a shared pool of executable reference UIs and professional aesthetic guidelines. Specifically, we introduce

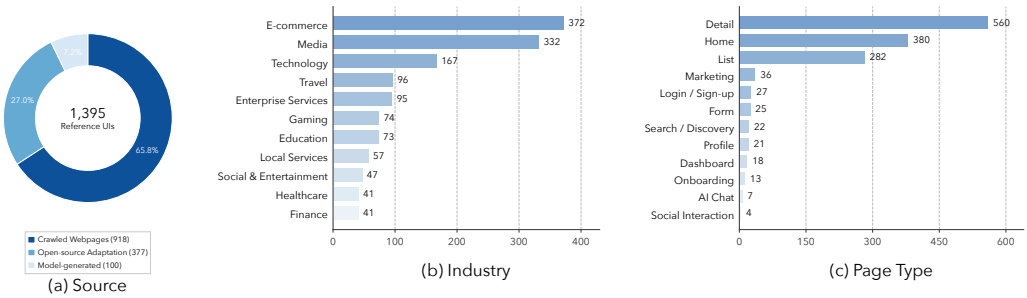

Figure 2: Distribution of the 1,395 reference UIs. (a) Data sources. (b) Industry categories. (c) Page types.

four tasks: (1) **Aesthetic Scoring** takes a UI image as input and outputs eight dimension-level aesthetic scores on a 1–5 scale, measuring holistic aesthetic judgment; (2) **Aesthetic Diagnosis** takes a pair of clean and degraded UI images and outputs the detected aesthetic violation(s) together with their affected regions, evaluating principle-level diagnosis and localization; (3) **Aesthetic Repair** takes a degraded UI image, its HTML source code, and the model's own diagnosis, and outputs a source-code patch that corrects the identified aesthetic issues; and (4) **Text-to-UI Generation** takes a natural-language webpage requirement together with textual content and image assets, and outputs an executable webpage without access to a reference screenshot. **Aesthetic Diagnosis** and **Aesthetic Repair** share the same controlled degradation instances, enabling direct instance-level analysis of judgment–action alignment.

3.2 Dataset Statistics

AUV-Bench comprises 1,395 executable reference UIs covering 11 industry categories and 12 page types (Figure 2). The crawled subset spans 386 web domains, and textual content covers 29 languages. Aesthetic Scoring and Text-to-UI Generation use all pages, while Aesthetic Diagnosis and Aesthetic Repair share 660 controlled-degradation cases with 1,209 annotated aesthetic violations across 13 rule types. Counting each task independently yields 4,110 evaluation instances. Aesthetic Scoring retains ratings from 44 professional UI designers after quality control. Further details are provided in Appendix C.

3.3 Task Formulation

Let h and x denote a UI's executable HTML source and rendered screenshot. Superscripts ref and deg denote reference and degraded versions; controlled degradation injects aesthetic violations while preserving content. Aesthetic Scoring predicts eight-dimensional scores $\hat{\mathbf{s}}$ from x^{ref}, using designer mean opinion scores (MOS) $\mathbf{s}$ as targets. Aesthetic Diagnosis and Aesthetic Repair share instances with violation annotations v specifying affected aesthetic dimensions and regions. Diagnosis identifies the degraded screenshot and predicts the corresponding annotations $\hat{v}$; repair uses $(h^{\mathrm{deg}}, x^{\mathrm{deg}}, \hat{v})$ to correct the violations. For Text-to-UI Generation, q comprises the webpage requirement, textual content, and image assets; the model generates source $\hat{h}$, rendered as $\hat{x}$, without access to x^{ref}.

3.4 Benchmark Construction

Figure 3 summarizes our construction pipeline. We build four tasks from 1,395 executable reference UIs through expert annotation, controlled aesthetic degradation, and requirement reconstruction. Professional UI designers co-develop the guidelines and review reference pages, degradation rules, and reconstructed requirements.

Professional Aesthetic Guidelines. Together with three senior UI designers, each with at least five years of industry experience at leading Internet companies, we develop aesthetic guidelines grounded in HCI and visual-design principles across seven dimensions: color, spacing, layout, typography, imagery, shadows, and global visual consistency. These guidelines define a shared design-principle space, which we further adapt to the evaluation objectives of different tasks. For

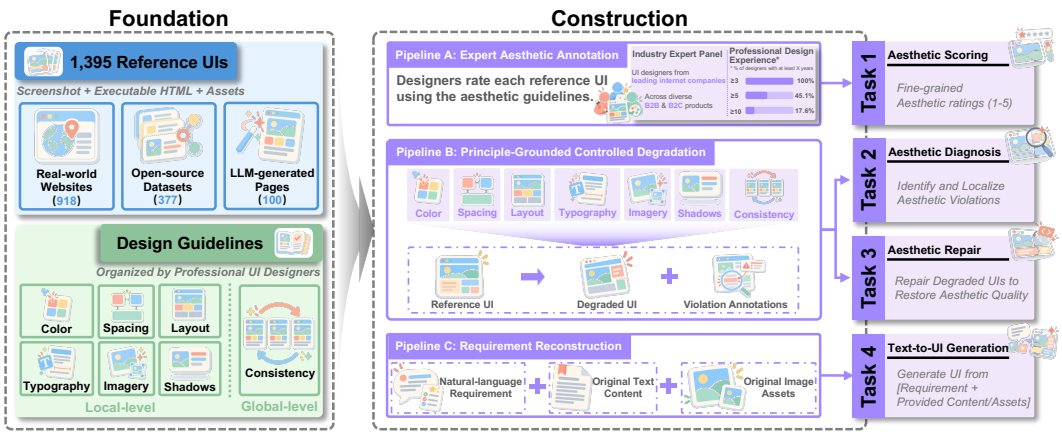

Figure 3: Overview of AUV-BENCH construction. Starting from 1,395 executable UIs and professional aesthetic guidelines, we build four complementary tasks through expert annotation, controlled degradation, and requirement reconstruction. The controlled-degradation pipeline draws inspiration from all seven guideline dimensions and retains five operational categories for source-level editing and verification.

Aesthetic Scoring, we reorganize the principles into eight perceptually coherent dimensions that can be reliably judged from rendered UIs: color, typography, graphics and imagery, layout, component consistency, visual-style consistency, copy quality, and image–text fit. Each dimension is rated on a shared 1–5 scale; detailed definitions are provided in Table 7. For *Aesthetic Diagnosis* and *Aesthetic Repair*, we retain principles that can yield perceptible, localizable, and attributable violations while supporting source-level editing and post-rendering verification. We organize them into five operational dimensions: typography, layout and reading flow, spacing, visual-style consistency, and color, and further instantiate 13 controlled aesthetic violation rules, detailed in Table 9.

For *Aesthetic Diagnosis* and *Aesthetic Repair*, the evaluation must instead correspond to specific aesthetic problems that can be explicitly identified and corrected. We therefore reorganize principles that admit perceptible, localizable, and attributable violations, while also supporting source-level modification and post-rendering verification, into five operational dimensions: typography, layout and reading flow, spacing, visual-style consistency, and color. Within these dimensions, we further instantiate the relevant design principles as 13 executable aesthetic violation rules, which are used to construct controlled degradations and define corresponding diagnosis and repair targets. The complete rule set is provided in Table 9.

Aesthetic Scoring Annotation. Each reference UI is initially rated independently by three professional designers using the shared eight-dimensional, 1–5 rubric. Annotators follow common rating criteria and review visual exemplars to calibrate their interpretation of the scale. After annotator-level quality control, retained ratings are aggregated into dimension-level MOS, which serve as evaluation targets.

Principle-Grounded Controlled Degradation. We first screen reference UIs for rule applicability and existing violations, applying a degradation only when the required structure is present and the target region satisfies the corresponding principle. Each intervention alters a small subset of eligible elements; designers iteratively review the rules and perturbation parameters. Automatic validation retains samples only when the intended defects are perceptible and localized, the detected violation set matches the injected recipe, and no unintended layout changes or unrelated aesthetic violations occur. We construct single-violation and compositional cases at mild and severe levels. Each retained case includes reference and degraded UIs with violation annotations, shared by Diagnosis and Repair.

Requirement Reconstruction. For *Text-to-UI Generation*, GPT-5.4 reconstructs an English requirement from each reference UI, describing its purpose, information structure, major functions, and overall visual direction while omitting precise visual parameters. Professional UI designers review each requirement and its objectively verifiable checklist, preserving functional intent while leaving visual implementation choices to the evaluated model.

Details of the guidelines, annotation, reference UI collection, controlled degradation, and requirement reconstruction are provided in Appendices A, B, C.1, C.2, and C.3, respectively.

3.5 Evaluation Protocol

We evaluate multimodal foundation models across four complementary tasks, covering aesthetic scoring, diagnosis, repair, and open-ended UI generation.

Task 1: Aesthetic Scoring. For instance i, the model predicts $\hat{\mathbf{s}}_i$ from x_i^{ref} on a 1–5 scale. We compare $\hat{\mathbf{s}}_i$ with $\mathbf{s}_i$ using *Spearman Rank Correlation (SRCC)* and *Mean Absolute Error (MAE)*, measuring ranking agreement and absolute error, respectively. Both metrics are computed across instances for each dimension and averaged equally across all eight dimensions.

Task 2: Aesthetic Diagnosis. We randomly order x_i^{ref} and x_i^{deg} as candidates V_1 and V_2. The model identifies the degraded candidate and predicts $\hat{v}_i$, evaluated against v_i. For degradation detection, we report *Balanced Accuracy (BAcc)*, defined as $\mathrm{BAcc} = (\mathrm{Recall}_{V_1} + \mathrm{Recall}_{V_2})/2$, where the two recalls correspond to instances with the first or second candidate degraded, respectively. For violation attribution, the labels are the five aesthetic dimensions: typography, layout and reading flow, spacing, visual style consistency, and color. We use *detection-gated Macro-J*, where $J_d = \mathrm{TPR}_d - \mathrm{FPR}_d$ and $\text{Macro-}J = \frac{1}{5}\sum_{d=1}^{5} J_d$. Here, d indexes the five attribution dimensions, and TPR_d and FPR_d denote their true-positive and false-positive rates, respectively. An incorrect degradation detection results in an empty attribution prediction. Localization is evaluated using *version-aware Micro-F1@0.5*, where regions in $\hat{v}_i$ and v_i are matched only on the correct version and with intersection over union (IoU) of at least 0.5.

We further report the *Exact Chain Success Rate (ECS)*, which requires all three stages to be correct:

$$\mathrm{ECS} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[D_i \wedge A_i \wedge L_i\right], \tag{1}$$

where D_i, A_i, and L_i are Boolean indicators for correct degradation detection, exact matching of the predicted and ground-truth dimension sets, and localization of all ground-truth regions without false positives, respectively. Here, N is the full number of evaluation instances in the current task, $\mathbf{1}[\cdot]$ equals one if its condition holds and zero otherwise, and $\wedge$ denotes logical conjunction.

Task 3: Aesthetic Repair. The model receives $(h_i^{\mathrm{deg}}, x_i^{\mathrm{deg}}, \hat{v}_i)$, using its own Task 2 diagnosis, and outputs a constrained source-code patch. The patched page is re-rendered for deterministic rule, DOM, and visual checks. We report the *Repair Pass Rate*. For instance i, the Boolean indicator R_i denotes repair success:

$$R_i = F_i \wedge C_i \wedge P_i \wedge S_i, \tag{2}$$

where F_i indicates that all target violations are repaired, C_i that no excessive collateral damage is introduced, P_i that the original content is preserved, and S_i that visual changes remain within the permitted repair scope. The final score is $\mathrm{RepairPass} = N^{-1}\sum_{i=1}^{N} \mathbf{1}[R_i]$. An incorrect $\hat{v}_i$ can still yield a successful repair if all four conditions hold.

Task 4: Text-to-UI Generation. Given q_i, the model generates $\hat{h}_i$ without access to x_i^{ref}. We evaluate the rendered $\hat{x}_i$ along two complementary axes.

First, GPT-5.4 scores $\hat{x}_i$ using the Task 1 rubric, yielding raw scores $a_{i,d}$ on a 1–5 scale, where d indexes its eight dimensions. To correct bias relative to designer ratings (Niculescu-Mizil & Caruana, 2005; Deng et al., 2025), we apply dimension-specific monotonic calibration functions $f_d(\cdot)$, learned exclusively from Task 1 MOS and frozen before Task 4 evaluation. For a valid generation, the calibrated score is $\mathrm{AesJudge}_i = \frac{1}{8}\sum_{d=1}^{8} f_d(a_{i,d}) \in [1,5]$. Invalid or failed generations receive zero. For model m, we report $\mathrm{AesJudge}_m = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AesJudge}_i$ over the fixed benchmark denominator N, yielding a model-level score in $[0,5]$. Judge validation and calibration details are provided in Appendix E.

Second, an independent multimodal judge performs blind pairwise comparisons of the rendered screenshots produced by two models for the same q_i. When both webpages render successfully, the judge determines a win, loss, or tie based on their overall aesthetic quality. If only one webpage

Table 2: Main results across the four evaluation tasks and cross-task analysis. ↑ (↓) indicates that higher (lower) is better. – indicates that JAA is undefined because no instance satisfies $J_i = 1$.

Model	Aesthetic Scoring		Aesthetic Diagnosis				Aesthetic Repair	Text-to-UI Generation		Judgment–Action Association (ϕ)↑
	SRCC↑	MAE↓	Det. BAcc↑	Attr. Macro-J↑	Loc. F1@0.5↑	ECS↑	Repair Pass↑	Cal. Aes. Judge↑	Pairwise WR↑	
Frontier Models										
Claude Opus 5	0.56	**0.81**	**91.10**	**58.90**	**36.70**	**24.70**	**75.60**	3.13	71.50	0.32
Doubao-Seed-2.1-Pro	0.50	1.10	79.60	36.20	7.60	4.10	45.00	2.78	62.90	0.31
GPT-5.6 Sol	0.55	0.90	87.60	55.40	28.20	19.40	65.50	3.17	78.00	0.38
Grok 4.6	0.54	0.95	81.20	42.70	6.50	2.90	70.00	3.08	62.50	0.25
Kimi-K3	0.57	1.10	83.80	43.30	26.70	15.20	57.30	**3.18**	**79.50**	**0.44**
Qwen3.7-Plus	**0.57**	1.25	60.80	22.00	1.60	0.90	34.20	3.09	57.80	0.07
Qwen3.8-Max	0.54	1.15	79.50	40.10	20.50	9.40	54.70	3.04	78.20	0.30
Open-Weight Models										
InternVL3.5-8B-Instruct	0.23	1.46	43.20	1.30	0.00	0.00	0.91	1.75	21.40	−0.01
LLaVA-OneVision2-8B-Instruct	0.45	0.94	40.90	-2.90	0.00	0.00	0.30	1.11	12.50	0.00
Qwen3-VL-8B-Instruct	0.45	1.56	58.70	8.70	1.20	0.80	0.61	1.84	28.10	−0.01
MiniCPM-V-4.6	0.05	1.65	48.40	3.10	0.00	0.00	0.15	0.53	5.30	–
Expert Models for Image Aesthetics Assessment										
AesExpert-7B	0.13	1.12	16.70	-0.40	0.00	0.00	0.00	0.16	1.40	–

renders successfully, it wins automatically; if both fail, neither receives credit. For model m, its *Overall Pairwise Win Rate* is

$$\text{PairwiseWR}_m = \frac{1}{P-1} \sum_{j \neq m} \frac{W_{m,j} + 0.5T_{m,j}}{N}, \tag{3}$$

where P is the number of participating models, j indexes the other models, and $W_{m,j}$ and $T_{m,j}$ count model m's wins and ties against model j, respectively. All queries remain in the fixed denominator N, including cases where both webpages fail to render. PairwiseWR lies in $[0, 1]$ and is reported as a percentage, with higher values indicating stronger relative preference across the benchmark. Losses and rendering failures receive zero credit.

Judgment–Action Association. To quantify judgment–action association, we pair diagnosis and repair outcomes for each instance i. Correct judgment is indicated by $J_i = D_i \wedge A_i$, requiring correct degradation detection and exact dimension attribution. Repair success is given by R_i in Eq. 2.

Over the $N = 660$ paired instances, the marginal judgment and repair success rates are $p_J = N^{-1} \sum_i \mathbf{1}[J_i]$ and $p_R = N^{-1} \sum_i \mathbf{1}[R_i]$, respectively, and the joint success rate is $p_{JR} = N^{-1} \sum_i \mathbf{1}[J_i \wedge R_i]$. We measure their association using the binary ϕ coefficient, termed *Judgment–Action Association* (JAA):

$$\text{JAA} = \phi_{J,R} = \frac{p_{JR} - p_J p_R}{\sqrt{p_J(1-p_J)\, p_R(1-p_R)}}. \tag{4}$$

JAA measures whether correct diagnosis and successful repair tend to occur on the same instances. Positive values indicate more frequent co-occurrence than expected under independence, values near zero indicate little instance-level association, and negative values indicate less frequent co-occurrence. All instances remain in the fixed benchmark denominator, including invalid or failed repair outputs; JAA is undefined when either J or R has zero variance. Together with the task-specific performance metrics, JAA provides an instance-level characterization of *Judgment–Action Coherence*.

4 Experiments

4.1 Experimental Setup

We evaluate a diverse set of multimodal models spanning frontier models, open-weight general-purpose models, and a specialized model for image aesthetics assessment. The frontier models include Claude Opus 5, Doubao-Seed-2.1-Pro, GPT-5.6 Sol, Grok 4.6, Kimi-K3, Qwen3.7-Plus, and Qwen3.8-Max. For open-weight models, we evaluate InternVL3.5-8B-Instruct (Wang et al., 2025), LLaVA-OneVision2-8B-Instruct (An et al., 2026b), Qwen3-VL-8B-Instruct (Bai et al., 2025), and MiniCPM-V-4.6 (Cui et al., 2026). We additionally include AesExpert-7B (Huang et al., 2024a), a specialized model for image aesthetics assessment. All models are evaluated under the same task-specific protocol. More details are provided in Appendix D.

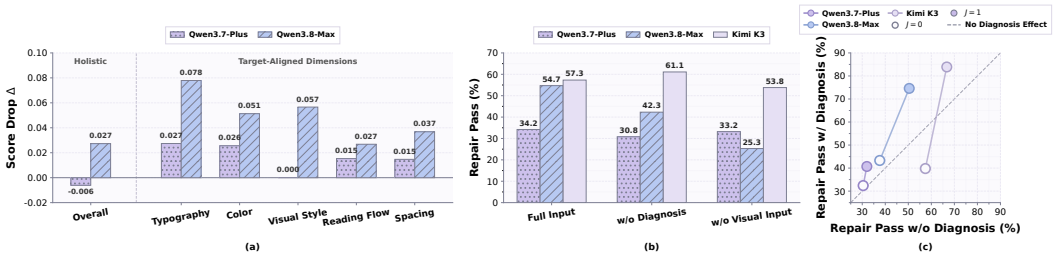

Figure 4: In-depth analyses of aesthetic sensitivity and judgment–action coherence. (a) Aesthetic score changes under controlled degradations. (b) Repair performance under input ablations. (c) Repair with versus without Task 2 diagnosis; points above (below) the diagonal indicate positive (negative) diagnosis effects.

4.2 Main Results

❶ Current models capture overall aesthetic quality to some extent, but fine-grained principle-level diagnosis remains a major bottleneck. In Aesthetic Scoring, frontier models show moderate agreement with professional designers, suggesting that they can partially capture the overall aesthetic quality of a UI. However, performance drops substantially when evaluation moves from holistic assessment to identifying specific aesthetic problems. Models can often distinguish the degraded interface, yet still struggle to determine *which* aesthetic principle is violated and *where* the problem occurs, with the best exact diagnosis-chain success reaching only 24.7%. These results indicate that progressing from coarse aesthetic perception to precise, principle-grounded visual diagnosis remains a key limitation of current models.

❷ Aesthetic judgment and repair are positively associated, yet a clear judgment–action gap remains. Most frontier models exhibit positive Judgment–Action Association, with JAA values ranging from 0.07 to 0.44, indicating that correct principle-level diagnosis tends to coincide with successful repair on the same instances. However, this alignment remains incomplete: successful repairs can occur without correct explicit judgment, while correct judgments do not always lead to successful repairs. These results reveal a persistent judgment–action gap, suggesting that current models have yet to develop a reliably coherent aesthetic capability that consistently connects principle-level understanding with corrective design action.

❸ Open-ended UI generation remains challenging, and relative superiority does not imply high absolute design quality. In Text-to-UI Generation, models must determine layout, visual hierarchy, typography, spacing, color, and component organization directly from natural-language requirements, textual content, and visual assets, without access to a reference screenshot or an existing implementation scaffold. Although frontier models show clear differences in relative performance, their human-calibrated absolute aesthetic scores remain only 2.78–3.18 out of 5, while some models achieve Pairwise Win Rates close to 80%. This discrepancy shows that consistently outperforming competing models does not necessarily correspond to strong absolute design quality. Overall, current models remain limited in their ability to autonomously coordinate multiple aesthetic principles and consistently synthesize high-quality interfaces in open-ended settings.

4.3 In-depth Analysis

❶ Current models' aesthetic scores often fail to distinguish controlled changes in specific aesthetic attributes. We use the controlled degradation samples from Aesthetic Diagnosis to evaluate Qwen3.7-Plus and Qwen3.8-Max on aesthetic scoring. As shown in Figure 4 (a), for 83.5% and 86.4% of samples, respectively, the overall score remains unchanged after degradation, with average changes of only -0.006 and 0.027. Even on dimensions directly corresponding to the injected defects, most scores remain unchanged. Qwen3.8-Max shows relatively higher local sensitivity: typography degradation reduces the corresponding score by 0.078 on average, compared with 0.027 for Qwen3.7-Plus, yet both changes remain small relative to the 1–5 scale. These results indicate that current models can respond to some local aesthetic changes, but their principle-level sensitivity remains weak and rarely propagates to overall aesthetic judgment.

❷ Current models rely on HTML-based shortcuts for UI repair instead of fully grounding their decisions in visual input. We isolate the contribution of visual input by removing screenshots while keeping the diagnosis, HTML, evaluation, and GT unchanged. As shown in Figure 4 (b), repair Pass drops only from 34.2% to 33.2% for Qwen3.7-Plus and from 57.3% to 53.8% for Kimi-

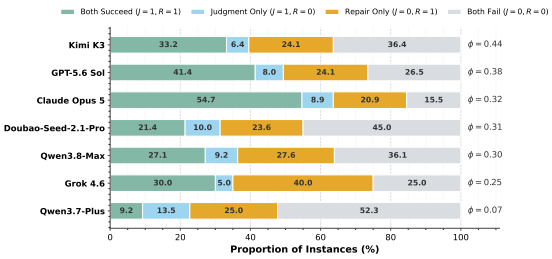

Figure 5: Decomposition of judgment–action outcomes over the 660 paired Diagnosis–Repair instances. Each bar separates $(J{=}1, R{=}1)$, $(J{=}1, R{=}0)$, $(J{=}0, R{=}1)$, and $(J{=}0, R{=}0)$ outcomes; JAA (ϕ) summarizes their instance-level association.

Table 3: Text-to-UI Generation with the Claude Code harness. ΔRank compares Pairwise WR rankings with the same seven models in the main evaluation.

Model	Cal. Aes. Judge↑	Pairwise WR↑	ΔRank vs. Main
Kimi-K3	**3.13**	**65.86**	–
GPT-5.6 Sol	3.09	55.91	↑ 1
Qwen3.7-Plus	3.10	55.38	↑ 4
Qwen3.8-Max	3.12	51.34	↓ 2
Claude Opus 5	3.09	48.66	↓ 1
Doubao-Seed-2.1-Pro	3.06	40.86	↓ 1
Grok 4.6	3.03	31.99	↓ 1

K3, suggesting that many repairs can be completed mainly from the structural, stylistic, and attribute information in the diagnosis and HTML. In contrast, Qwen3.8-Max drops sharply from 54.7% to 25.3%, indicating stronger reliance on explicit visual grounding. This model-level variation shows that Task 3 is not driven purely by visual understanding: some models can exploit executable HTML structure to bypass sufficient modeling of the page's visual state, forming an HTML-based shortcut.

❸ **Where does the Judgment–Action Gap come from?** To unpack the scalar JAA score, we decompose each paired Diagnosis–Repair instance into four outcomes according to judgment correctness J and repair success R. Figure 5 reveals substantial off-diagonal mass across models: correct judgments may fail to yield successful repairs, while many successful repairs occur without correct explicit judgment. This asymmetry is particularly pronounced for some models, showing that strong repair performance can arise without correspondingly reliable aesthetic understanding. The results make the Judgment–Action Gap explicit: judgment and action are associated, but do not form a stable instance-level correspondence.

❹ **Correct aesthetic diagnosis substantially improves downstream repair.** We group samples by judgment correctness, where $J = 1$ denotes correct degraded-UI identification and exact matching of the affected aesthetic dimensions, and compare Aesthetic Repair with and without the model's own diagnosis. As shown in Figure 4 (c), when $J = 1$, the diagnosis improves repair by $+8.7$, $+24.2$, and $+17.2$ percentage points for Qwen3.7-Plus, Qwen3.8-Max, and Kimi-K3, respectively. When $J = 0$, the benefit is much weaker and model-dependent; for Kimi-K3, repair performance drops by 17.5 points. These results indicate that accurate aesthetic judgments can provide useful design guidance, whereas unreliable judgments offer substantially less consistent downstream value.

❺ **Does the harness help? A coding harness does not consistently improve UI generation quality.** We further evaluate whether equipping frontier models with a Claude Code harness, which enables iterative code generation and editing, improves Text-to-UI Generation. As shown in Table 3, the harness does not consistently improve calibrated aesthetic scores, which remain tightly clustered around 3.0. The harness also changes the relative ordering of models: Qwen3.7-Plus rises from seventh to third among the seven evaluated models, whereas Qwen3.8-Max drops from second to fourth. This reshuffling suggests that models differ substantially in how effectively they exploit agentic coding scaffolds. More importantly, additional opportunities for code manipulation alone do not consistently translate into higher aesthetic quality, indicating that open-ended UI generation remains constrained by the model's underlying aesthetic decision-making capability rather than merely by its coding interface.

5 Conclusion

We introduce AUV-Bench, a benchmark for jointly evaluating UI aesthetic judgment and design action across scoring, diagnosis, repair, and generation. Experiments show that current models exhibit partial aesthetic competence but still struggle with fine-grained diagnosis, open-ended generation, and consistent judgment–action alignment. We hope AUV-Bench supports future progress toward more coherent and reliable aesthetic capabilities for UI design.

AI USE STATEMENT

In this work, we used generative AI tools for four purposes. First, large language models (LLMs) were used to assist with language polishing and improving the clarity of the manuscript. Second, LLMs were used to assist with code development and debugging; all AI-assisted code was reviewed, verified, and tested by multiple authors before use in our experiments. Third, generative AI models were used during benchmark construction, including generating a subset of the webpages and reconstructing natural-language requirements for Text-to-UI Generation, as described in the paper. Fourth, GPT-5.4 was used as an automatic aesthetic judge for Text-to-UI Generation; its scoring behavior was validated and calibrated against professional-designer annotations from Aesthetic Scoring before being applied to the generation results. All AI-assisted outputs used in the final work were reviewed by the authors, and human-reviewed or validated where specified in the corresponding benchmark construction and evaluation procedures. We take responsibility for the final content of this work, including text, code, data, experimental results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will release the source code and dataset upon acceptance of the paper to facilitate reproducibility and future research.

REFERENCES

Arctanx An, Shizhao Sun, Danqing Huang, Mingxi Cheng, Yan Gao, Ji Li, Yu Qiao, and Jiang Bian. Can vision language models assess graphic design aesthetics? a benchmark, evaluation, and dataset perspective. *arXiv preprint arXiv:2603.01083*, 2026a.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, et al. Llava-onevision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026b.

Rabiul Awal, Mahsa Massoud, Aarash Feizi, Zichao Li, Suyuchen Wang, Christopher Pal, Aishwarya Agrawal, David Vazquez, Siva Reddy, Juan A Rodriguez, et al. Webmmu: A benchmark for multimodal multilingual website understanding and code generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25129–25156, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Tony Beltramelli. pix2code: Generating code from a graphical user interface screenshot. In *Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems*, pp. 1–6, 2018.

Nathalie Bonnardel, Annie Piolat, and Ludovic Le Bigot. The impact of colour on website appeal and users' cognitive processes. *Displays*, 32(2):69–80, 2011.

Chunyang Chen, Ting Su, Guozhu Meng, Zhenchang Xing, and Yang Liu. From ui design image to gui skeleton: a neural machine translator to bootstrap mobile gui implementation. In *Proceedings of the 40th International Conference on Software Engineering*, pp. 665–676, 2018.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*, 2024.

Nelson Cowan. The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57, 2010.

James H Creager and Douglas J Gillan. Toward understanding the findability and discoverability of shading gradients in almost-flat design. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 60, pp. 339–343. SAGE Publications Sage CA: Los Angeles, CA, 2016.

Nigel Cross. Expertise in design: an overview. *Design studies*, 25(5):427–441, 2004.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pp. 845–854, 2017.

Zhijie Deng, Zhouan Shen, Ling Li, Yao Zhou, Zhaowei Zhu, Yanji He, Wei Wang, and Jiaheng Wei. Lm-mixup: Text data augmentation via language model based mixup. *arXiv preprint arXiv:2510.20449*, 2025.

Tianxiang Du, Hulingxiao He, and Yuxin Peng. Venus: Benchmarking and empowering multimodal large language models for aesthetic guidance and cropping. *arXiv preprint arXiv:2602.23980*, 2026.

Peitong Duan, Chin-Yi Cheng, Gang Li, Bjoern Hartmann, and Yang Li. Uicrit: Enhancing automated design evaluation with a ui critique dataset. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–17, 2024.

Mary C Dyson. How physical text layout affects reading from screen. *Behaviour & information technology*, 23(6):377–393, 2004.

Andrew J Elliot and Markus A Maier. Color psychology: Effects of perceiving color on psychological functioning in humans. *Annual review of psychology*, 65:95–120, 2014.

J Shawn Farris, Keith S Jones, and Brent A Anders. Acquisition speed with targets on the edge of the screen: An application of fitts' law to commonly used web browser controls. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 45, pp. 1205–1209. SAGE Publications Sage CA: Los Angeles, CA, 2001.

Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Bohua Chen, Yi Su, Dongping Chen, Siyuan Wu, Xing Zhou, et al. Webcode2m: A real-world dataset for code generation from webpage designs. In *Proceedings of the ACM on Web Conference 2025*, pp. 1834–1845, 2025.

Daichi Haraguchi, Naoto Inoue, Wataru Shimoda, Hayato Mitani, Seiichi Uchida, and Kota Yamaguchi. Can gpts evaluate graphic design based on design principles? In *SIGGRAPH Asia 2024 Technical Communications*, pp. 1–4. 2024.

Zehai He, Wenyi Hong, Zhen Yang, Ziyang Pan, Mingdao Liu, Xiaotao Gu, and Jie Tang. Vision2web: A hierarchical benchmark for visual website development with agent verification. *arXiv preprint arXiv:2603.26648*, 2026.

Yipo Huang, Xiangfei Sheng, Zhichao Yang, Quan Yuan, Zhichao Duan, Pengfei Chen, Leida Li, Weisi Lin, and Guangming Shi. Aesexpert: Towards multi-modality foundation model for image aesthetics perception. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5911–5920, 2024a.

Yipo Huang, Quan Yuan, Xiangfei Sheng, Zhichao Yang, Haoning Wu, Pengfei Chen, Yuzhe Yang, Leida Li, and Weisi Lin. Aesbench: An expert benchmark for multimodal large language models on image aesthetics perception. *arXiv preprint arXiv:2401.08276*, 2024b.

Kashif Imteyaz, Kaif Imteyaz, Nakul Rajpal, Kaif Shaikh, Michael Muller, and Saiph Savage. Design theater: A benchmark for generative ui. *arXiv preprint arXiv:2607.22928*, 2026.

Jaehyun Jeon, Min Soo Kim, Janghan Yoon, Sumin Shim, Yejin Choi, Hanbin Kim, Dae Hyun Kim, and Youngjae Yu. Do mllms capture how interfaces guide user behavior? a benchmark for multimodal ui/ux design understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44269–44294, 2026.

Sam Jung, Agustin Garcinuno, and Spencer Mateega. Ui-bench: A benchmark for evaluating design capabilities of ai text-to-app tools. *arXiv preprint arXiv:2508.20410*, 2025.

Wendy A Kellogg. Conceptual consistency in the user interface: Effects on user performance. In *Human–Computer Interaction–INTERACT'87*, pp. 389–394. Elsevier, 1987.

Jisu Ko, Jinyoung Choi, Cielo Morales, Dajung Kim, and Minsam Ko. Criticmate: Stagewise human–ai co-critique in single-screen ui evaluation. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–22, 2026.

Bryan Lawson. Schemata, gambits and precedent: some factors in design expertise. *Design studies*, 25(5):443–457, 2004.

Ryan Li, Yanzhe Zhang, and Diyi Yang. Sketch2code: Evaluating vision-language models for interactive web design prototyping. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3921–3955, 2025.

Jieru Lin, Danqing Huang, Tiejun Zhao, Dechen Zhan, and Chin-Yew Lin. Designprobe: A graphic design benchmark for multimodal large language models. *arXiv preprint arXiv:2404.14801*, 2024.

Jieru Lin, Danqing Huang, Tiejun Zhao, Dechen Zhan, and Chin-Yew Lin. Can multimodal large language models understand graphic design? a comparative study. *IEEE Transactions on Multimedia*, 2026.

Zhiyu Lin, Zhengda Zhou, Zhiyuan Zhao, Tianrui Wan, Yilun Ma, Junyu Gao, and Xuelong Li. Webuibench: a comprehensive benchmark for evaluating multimodal large language models in webui-to-code. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15780–15797, 2025.

Jonathan Ling and Paul van Schaik. The influence of line spacing and text alignment on visual search of web pages. *Displays*, 28(2):60–67, 2007.

Chenxu Liu, Yingjie Fu, Wei Yang, Ying Zhang, and Tao Xie. Webcoderbench: Benchmarking web application generation with comprehensive and interpretable evaluation metrics. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11632–11666, 2026.

Zimu Lu, Yunqiao Yang, Houxing Ren, Haotian Hou, Han Xiao, Ke Wang, Weikang Shi, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Webgen-bench: Evaluating llms on generating interactive and functional websites from scratch. *Advances in Neural Information Processing Systems*, 38, 2026.

Ruichao Mao, Zhou Fang, Teng Guo, Hao Yang, Yaping Li, Shaohua Peng, Maji Huang, Xiaoyu Lin, Shuoyang Liu, Xuepeng Li, et al. Reasoning for mobile user experience with multimodal llms: Task, benchmark, and approach. *arXiv preprint arXiv:2606.13192*, 2026.

George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 2408–2415. IEEE, 2012.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632, 2005.

Zirui Pang, Haosheng Tan, Yuhan Pu, Zhijie Deng, Zhouan Shen, Keyu Hu, and Jiaheng Wei. When vlms meet image classification: Test sets renovation via missing label identification. *arXiv preprint arXiv:2505.16149*, 2025.

Yi-Hao Peng, Jeffrey P Bigham, and Jason Wu. Designpref: Capturing personal preferences in visual design generation. *arXiv preprint arXiv:2511.20513*, 2025.

Daiqing Qi, Handong Zhao, Jing Shi, Simon Jenni, Yifei Fan, Franck Dernoncourt, Scott Cohen, and Sheng Li. The photographer's eye: Teaching multimodal large language models to see and critique like photographers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24807–24816. IEEE, 2025.

Mirjam Seckler, Klaus Opwis, and Alexandre N Tuch. Linking objective design factors with subjective aesthetics: An experimental study on how structure and color of websites affect the facets of users' visual aesthetic perception. *Computers in Human Behavior*, 49:375–389, 2015.

Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on image processing*, 15(2):430–444, 2006.

Hamid R Sheikh, Alan C Bovik, and Gustavo De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on image processing*, 14(12):2117–2128, 2005.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2code: Benchmarking multimodal code generation for automated front-end engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3956–3974, 2025.

Haoyu Sun, Huichen Will Wang, Jiawei Gu, Linjie Li, and Yu Cheng. Fullfront: Benchmarking mllms across the full front-end engineering workflow. *arXiv preprint arXiv:2505.17399*, 2025.

Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE transactions on image processing*, 27(8):3998–4011, 2018.

Yuxuan Wan, Chaozheng Wang, Yi Dong, Wenxuan Wang, Shuqing Li, Yintong Huo, and Michael Lyu. Divide-and-conquer: Generating ui code from screenshots. *Proceedings of the ACM on Software Engineering*, 2(FSE):2099–2122, 2025.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Wenjie Wang, Yue Huang, Zipeng Ling, Han Bao, Xiaonan Luo, Yu Jiang, Shiyi Du, Yuexing Hao, Xiaomin Li, Yuchen Ma, et al. Uxbench: Measuring the actionability of llm-generated ux critiques. *arXiv preprint arXiv:2606.16262*, 2026.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yuqian Wu, Zhijie Deng, Wei Chen, Junwei Li, Yutian Jiang, Junle Chen, Zhengjun Huang, Qingxiang Liu, Jing Tang, Jiaheng Wei, et al. Lifeside: Benchmarking agents as lifelong digital companions. *arXiv preprint arXiv:2606.04660*, 2026.

Jingyu Xiao, Ming Wang, Man Ho Lam, Yuxuan Wan, Junliang Liu, Yintong Huo, and Michael R Lyu. Designbench: A comprehensive benchmark for mllm-based front-end code generation. *arXiv preprint arXiv:2506.06251*, 2025.

Zhen Yang, Wenyi Hong, Mingde Xu, Xinyue Fan, Weihan Wang, Jiale Cheng, Xiaotao Gu, and Jie Tang. UI2Code^{N}: A Visual Language Model for Test-Time Scalable Interactive UI-to-Code Generation. 2025.

Hongda Zhu, Yiwen Zhang, Bing Zhao, Jingzhe Ding, Siyao Liu, Tong Liu, Dandan Wang, Yanan Liu, and Zhaojian Li. Frontendbench: A benchmark for evaluating llms on front-end development via automatic evaluation. *arXiv preprint arXiv:2506.13832*, 2025.

Appendix

TABLE OF CONTENTS

A Aesthetic Design Principles **15**

A.1 color . . . 15

A.2 Spacing . . . 17

A.3 Layout . . . 18

A.4 Typography and Text Layout . . . 20

A.5 Imagery . . . 22

A.6 Shadow . . . 23

A.7 Consistency . . . 25

B Human Annotation and Evaluation Details **26**

B.1 Annotation Procedure . . . 26

B.2 Evaluation Criteria and Ground-Truth Distribution . . . 28

B.3 Inter-Annotator Reliability . . . 28

B.4 Model Evaluation Prompt . . . 29

B.5 Annotation Examples . . . 29

C More Details of AUV-BENCH **30**

C.1 Reference UI Collection and Quality Assurance . . . 30

C.2 Controlled Aesthetic Violations and Task Construction . . . 33

C.3 Requirement Reconstruction . . . 37

D Experimental Details **37**

D.1 Model and Inference Settings . . . 37

D.2 Input Preprocessing and Asset Organization . . . 37

E More Experimental Analysis **38**

E.1 Validation and Calibration of GPT-5.4 as an Automatic Aesthetic Judge . . . 38

E.2 Additional Fine-Grained Analyses . . . 39

E.3 Performance across Industries . . . 41

A AESTHETIC DESIGN PRINCIPLES

Perceptual similarity to reference designs provides a useful measure of visual reconstruction quality, but does not directly characterize whether an interface follows the aesthetic principles used in professional design practice. To provide a principle-level basis for UI aesthetic evaluation, we collaborated with professional UI designers to co-develop a structured aesthetic design guideline. The guideline draws on established HCI and visual-design principles, industry design conventions, and explicit standards where applicable, with an emphasis on properties that can be described quantitatively from a rendered interface or its design source.

We organize the guideline into seven dimensions: color, spacing, layout, typography, imagery, shadow, and consistency. For each dimension, we describe the underlying design rationale and the corresponding design rules. When the co-developed guideline specifies an explicit quantitative or programmatic implementation, we retain it as part of the operational definition. Table 4 summarizes the rules for which such implementation procedures are specified. Representative violating and conforming examples are provided throughout the section.

Table 4: Summary of aesthetic rules with explicitly specified quantitative or programmatic checks in the co-developed professional UI design guideline.

Dimension	Rule	Implementation
color	Primary : secondary : accent $= 6:3:1$	Measure the pixel proportion of each color role and compute its KL divergence from the target ratio
	Normal text/background contrast ≥ 4.5:1; large text $\geq$ 3:1; graphics and UI components $\geq$ 3:1	Compute the WCAG contrast ratio from foreground and background relative luminances using Eq. (6)
Spacing	Element sizes and spacing use 4, 12, or integer multiples of 8 pixels	Compute the residual of element coordinates and dimensions modulo 8, while admitting 4 and 12 as additional values
Typography	At most one CJK and one Latin typeface per page	Extract typeface information from the design source and compare typeface usage
	$h_{\text{line}} = s + 8$ and at most five type sizes	Extract font-size and line-height parameters from the design source
	Font weights use $\{400, 500, 600\}$ and at most three weights per application	Extract `font-weight` values from the design source
	At most 85 characters per line	OCR followed by character counting
	Text in tables and lists is left-aligned, while Arabic numerals and other numerical content are right-aligned	Check the consistency of the left or right boundary coordinates within a column
Imagery	Image aspect ratios belong to $\{1, \frac{4}{3}, \frac{16}{9}, \frac{3}{4}, \frac{9}{16}\}$	Compute the deviation between the rendered aspect ratio and the nearest admissible ratio
	Images remain sufficiently clear to be recognizable	Apply image-quality assessment to identified image regions
Shadow	Component shadow depth follows semantic levels 0 to 3	Recognize UI components, extract their shadow parameters, and compare them with the corresponding semantic level
Consistency	Equivalent containers maintain consistent corner radius, shadow, and spacing	Identify container components and compare their corresponding parameters
	Components of the same type follow one shared specification	Identify component classes and compare parameters across instances

A.1 COLOR

color plays an important role in visual communication and in conveying multiple types of information within an interface (Bonnardel et al., 2011; Seckler et al., 2015). A well-designed enterprise

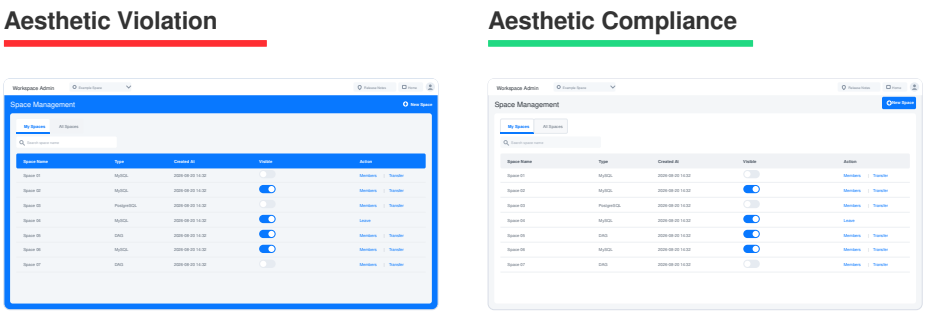

Figure 6: color proportion. The violating interface applies the accent color to a large portion of the page, while the conforming interface follows a more restrained color allocation.

application should use color clearly and consistently so that both functional information and product identity can be communicated effectively.

We distinguish three color roles. The *theme color* represents the product identity and is generally associated with the brand color. It is commonly used for primary buttons and text, important operation states, and highlighted information. The theme color is typically extended into a tonal scale within the same color family. *Functional colors* represent explicit information and states, including success, error, failure, warning, and links. Their use should follow common user expectations and remain consistent within the same product system. *Neutral colors* are widely used for text, backgrounds, borders, and dividers.

color proportion. To avoid excessive color usage, we adopt a 60:30:10 allocation. Approximately 60% of the interface is assigned to the primary color, which in enterprise applications is commonly the color of large surface or content regions. Approximately 30% is assigned to secondary colors, which are typically neutral colors, and the remaining 10% is reserved for accent colors drawn from the functional and theme palettes.

For quantitative assessment, we estimate the empirical distribution $\hat{p}$ of the three color roles over interface pixels and compare it with the target distribution

$$p^\star = (0.6, 0.3, 0.1).$$

The deviation is measured using

$$D_{\mathrm{KL}}\left(\hat{p} \,\|\, p^\star\right). \tag{5}$$

Figure 6 shows a representative violation in which the accent color occupies an excessively large region.

Contrast. Color contrast determines whether textual and graphical information can be reliably perceived, as illustrated by the confirmation messages in Figure 7. Following the WCAG AA requirements[1], we adopt a minimum contrast ratio of $4.5{:}1$ for normal text against its background. Large text requires a minimum contrast ratio of 3:1, where large text is defined as bold text of at least $18.66\,\mathrm{px}$ or regular text of at least $24\,\mathrm{px}$. Graphics and user-interface components require a minimum contrast ratio of 3:1.

WCAG is a set of accessibility guidelines developed by the W3C Web Accessibility Initiative to improve the accessibility of Web content for users with different perceptual and cognitive abilities. For implementation, we compute the WCAG contrast ratio as

[1] https://www.w3.org/TR/WCAG22/

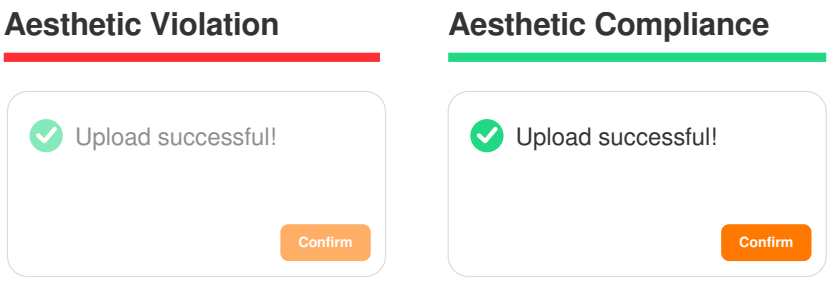

Figure 7: color contrast. A confirmation message with insufficient foreground-background contrast, compared with a version satisfying the prescribed contrast requirement.

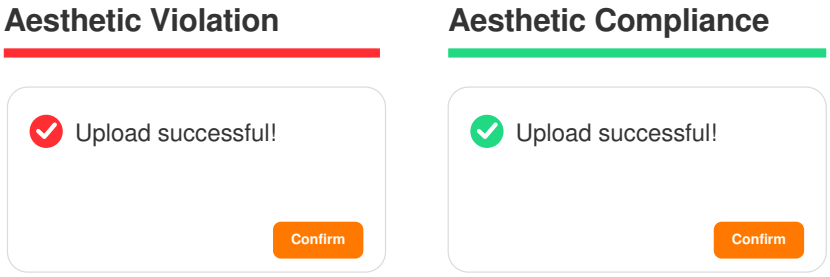

Figure 8: Functional color semantics. The violating example uses a color whose conventional semantic meaning conflicts with the represented state, while the conforming example follows the expected color semantics.

$$\text{contrast} = \frac{\max(Y_f, Y_b) + 0.05}{\min(Y_f, Y_b) + 0.05}, \tag{6}$$

where Y_f and Y_b denote the foreground and background relative luminances in $[0, 1]$, respectively, computed from linearized sRGB values following WCAG 2.2.

color semantics. Functional colors should conform to common semantic expectations, as colors can carry context-dependent semantic and affective associations (Elliot & Maier, 2014). Red commonly communicates urgency or intensity and is therefore suitable for promotional actions and error states. Blue commonly conveys technology and professionalism and is frequently used in technology-oriented products. Green is associated with nature and safety and is commonly used for environmental contexts or successful states. Maintaining these semantic associations reduces unnecessary interpretation effort and makes interface states easier to understand. Figure 8 contrasts red and green indicators for the same success state.

A.2 Spacing

Spacing defines the basic spatial relationships among interface elements. Enterprise applications commonly use a grid system to establish visual order, with 8 pixels serving as a widely adopted base unit. An even-valued base unit is compatible with common display environments, while a regular

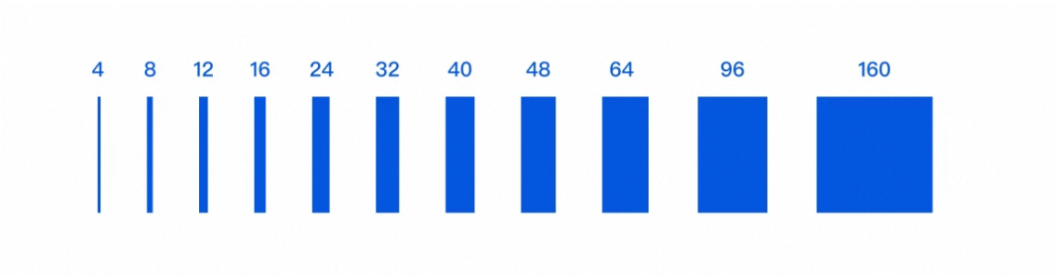

Figure 9: Spacing scale. Representative spacing values derived from the 8-pixel base grid, together with the additional 4- and 12-pixel increments.

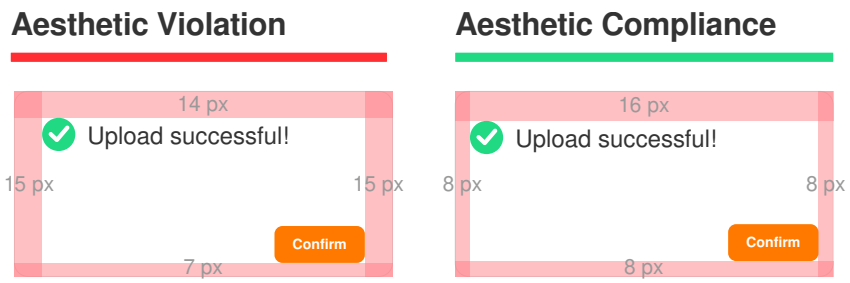

Figure 10: Spacing grid. The violating layout contains spacing values that do not follow the prescribed grid, while the conforming version uses admissible spacing increments.

grid also reduces arbitrary layout decisions and provides a shared spatial vocabulary between design and implementation.

We therefore require interface elements to be arranged using an 8-pixel base grid. To support denser layouts and smaller local gaps, 4 and 12 pixels are additionally admitted as valid spacing values. The resulting spacing scale (Figure 9) includes values such as 4, 8, 12, 16, 24, 32, 40, 48, 64, 96, and 160 pixels.

For a set of measured element coordinates and dimensions $\{x_i\}$, we evaluate deviation from the grid using

$$\mathcal{P} = \sum_i (x_i \bmod 8)\,, \tag{7}$$

while additionally admitting 4 and 12 pixels as valid values. Figure 10 illustrates irregular gaps of 14, 15, and 7 pixels and their corresponding conforming alternatives.

A.3 Layout

Layout is a fundamental part of enterprise interface design and provides the basis for consistent interaction and visual organization. We consider three established principles for organizing important information and actions: Fitts' Law, Miller's Law, and the Gutenberg Diagram.

Fitts' Law. Fitts' Law states that larger and closer targets can be reached more quickly and with fewer errors than smaller and more distant targets (Fitts, 1954). Screen edges have a particular advantage because pointer movement cannot continue beyond the screen boundary, making controls near the edge easier to acquire (Farris et al., 2001).

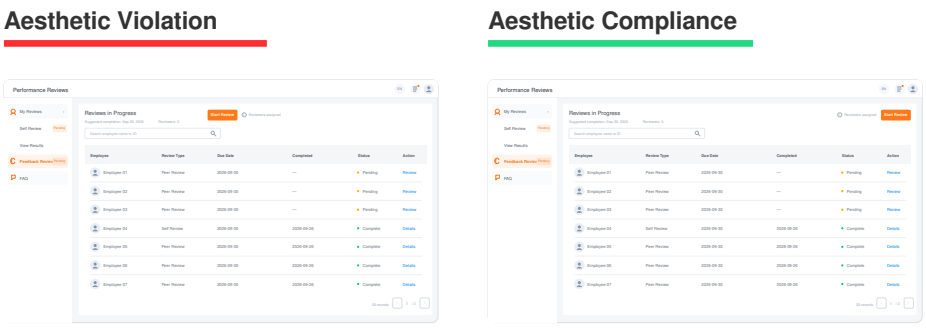

Figure 11: Fitts' Law. A primary action placed far from the screen edge, compared with a placement that follows the recommended edge distance.

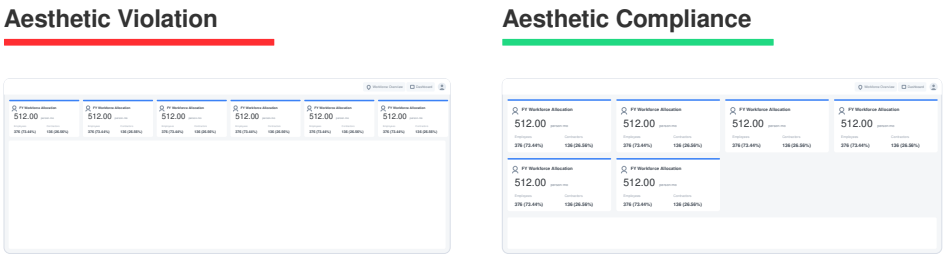

Figure 12: Miller's Law. Six similar content blocks are presented as one group in the violating example, while the conforming version reorganizes them into smaller groups.

At the same time, frequently used controls should not be positioned so close to the boundary that accidental activation becomes likely. Balancing these two considerations, we recommend that frequently used desktop controls, especially primary actions, be positioned approximately 5% to 12% of the viewport dimension from the nearest edge, as illustrated by the primary action in Figure 11. Corners should be used preferentially when appropriate.

Miller's Law. Human information processing capacity is limited. The classical formulation of Miller's Law suggests that short-term memory can simultaneously maintain approximately five to nine items (Miller, 1956). Later work suggests that effective working-memory capacity is often closer to three to five chunks (Cowan, 2010).

For enterprise applications, we adopt the more conservative interpretation and recommend that a group of similar content blocks contain no more than five items. When more content is required, the blocks can be reorganized into multiple groups or rows, as illustrated by the six-block layout in Figure 12.

Gutenberg Diagram. For interfaces following a left-to-right reading convention, visual attention typically begins in the upper-left region and progresses toward the lower-right region. The Gutenberg Diagram divides the page into four quadrants and emphasizes the upper-left and lower-right regions as the beginning and ending areas of the reading path.

Accordingly, we recommend placing the most important identifying or promotional information in the upper-left region, such as logos or other high-priority information. As illustrated by the confirmation dialog in Figure 13, the most important operation, such as a confirmation action, is preferentially placed in the lower-right region.

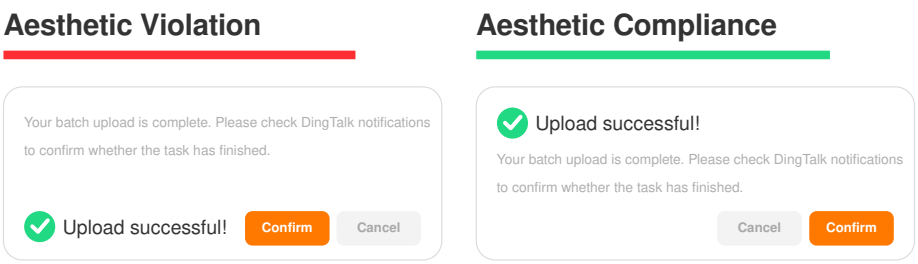

Figure 13: Gutenberg Diagram. The violating example places important information and actions against the expected reading flow, while the conforming layout follows the recommended upper-left to lower-right organization.

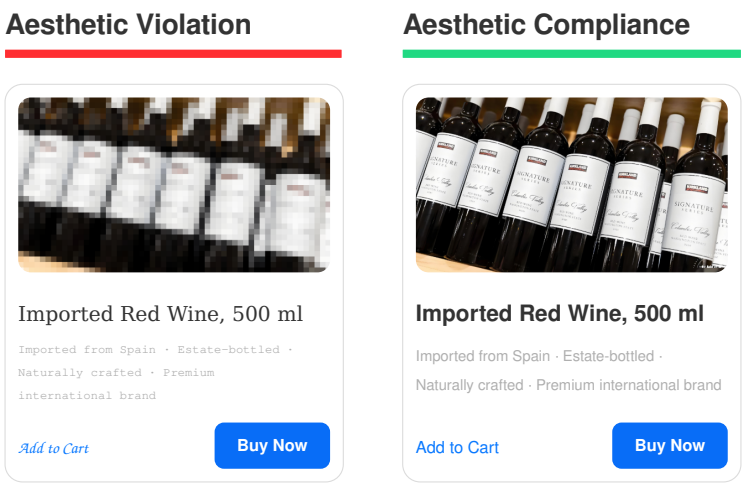

Figure 14: Typeface discipline. The violating example mixes several unrelated typefaces within one interface, while the conforming version uses a unified typeface system.

A.4 Typography and Text Layout

Typography is one of the fundamental components of systematic interface design. Enterprise users rely on text to understand information and complete their work, making a coherent type system important for both reading efficiency and task efficiency (Dyson, 2004).

Typeface. For efficiency and platform compatibility, we recommend using the system default typeface in enterprise applications. Within a single page, no more than one CJK typeface and one Latin typeface should be used. Figure 14 contrasts mixed typefaces with a unified typeface system.

This rule can be checked by extracting the typeface information from the design source and comparing the set of typefaces used on the page.

Type scale and line height. A type scale is a regular set of font sizes used to establish textual hierarchy. Line height defines the vertical space occupied by each line of text. We take

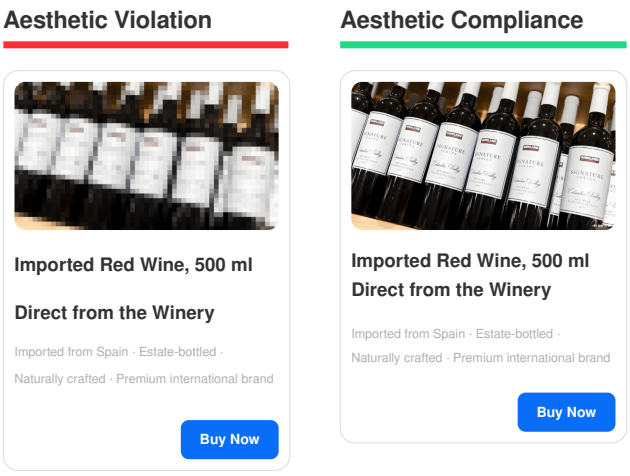

Figure 15: Type scale and line height. The violating example uses line spacing that weakens the continuity of the text block, while the conforming version follows the additive line-height rule in Eq. equation 8.

$$h_{\text{line}} = 1.5s$$

as a common accessibility-oriented reference, where s denotes the font size. However, a fixed multiplicative ratio causes the additional vertical space to increase with font size. For large display text, this may weaken the visual continuity between lines, particularly when several font sizes are mixed within the same interface.

We therefore adopt an additive rule to control the spacing between lines across font sizes, as illustrated in Figure 15:

$$h_{\text{line}} = s + 8 \quad \text{(pixels)}. \tag{8}$$

The constant 8 also gives line heights close to the $1.5\times$ reference for the commonly used body-text sizes of 14 px and 16 px. To maintain a clear reading hierarchy, no more than five distinct type sizes should be used within one application.

The rule is checked by extracting font-size and line-height parameters from the design source.

Font weight. Font weight is an important typographic variable for expressing hierarchy and distinguishing important content. We use $\{400, 500, 600\}$ as the preferred set of weights and allow no more than three font-weight values within one application. The link and button labels in Figure 16 illustrate consistent font weights across actions.

The implementation extracts the `font-weight` value of each text element and checks whether the observed values follow the prescribed set and cardinality.

Line length and alignment. Line length has been shown to affect on-screen reading and information retrieval (Dyson, 2004). To maintain reading comfort in Web applications, we recommend no more than 85 characters in a single line of text. This property can be checked through OCR followed by character counting.

Text alignment and spacing can affect visual-search performance on Web pages (Ling & van Schaik, 2007). For tables and lists, textual content should be left-aligned, while Arabic numerals and other numerical values should be right-aligned. This arrangement (Figure 17) allows textual entries to share a stable starting position and numerical values to share a stable ending position.

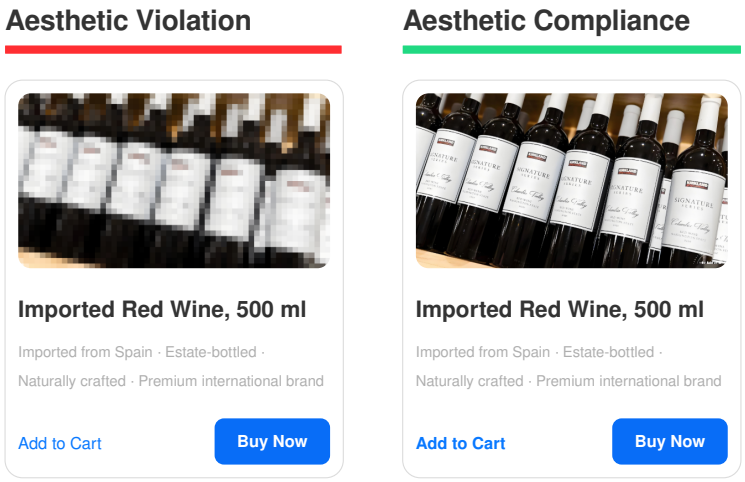

Figure 16: Font weight. The violating example uses an inconsistent weight assignment, while the conforming example follows the prescribed font-weight system.

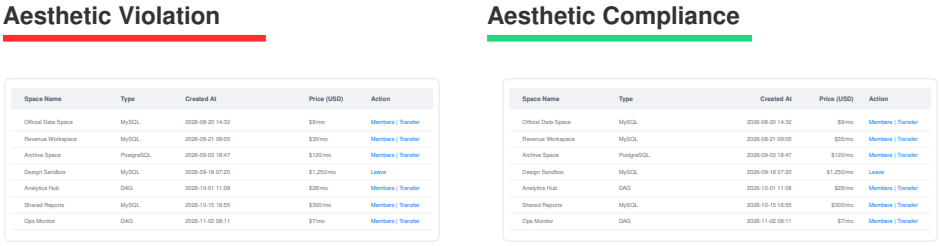

Figure 17: Text and numerical alignment. Numerical values are left-aligned in the violating table, while the conforming version right-aligns numerical content and left-aligns text.

For a left-aligned column, we describe the condition

$$x_{\text{left}} = C, \tag{9}$$

while a right-aligned column satisfies

$$x_{\text{right}} = W - C, \tag{10}$$

where W denotes the column width and C denotes the corresponding inset.

A.5 Imagery

Images increase the dimensionality of information conveyed by an interface and can provide visual explanations that complement textual content. Their presentation quality is primarily affected by geometric proportion and visual clarity.

Aspect ratio. Different image proportions support different presentation scenarios. Excluding full-width and full-screen banner images, we identify several commonly used aspect ratios.

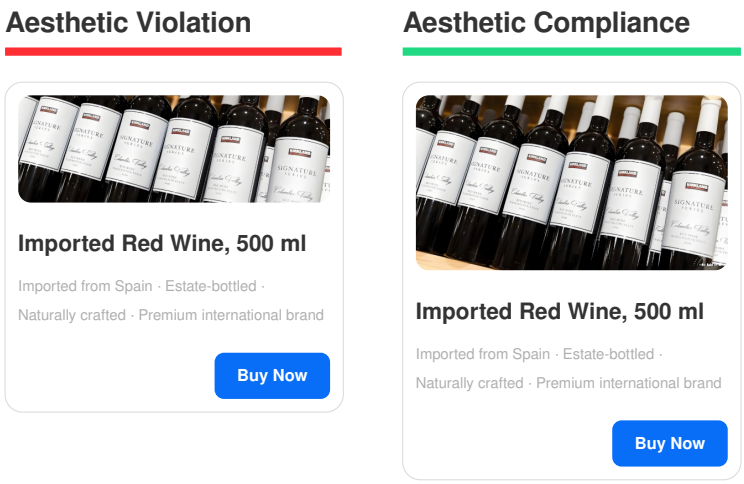

Figure 18: Image aspect ratio. The violating example distorts the image proportion, while the conforming version preserves an admissible aspect ratio.

A 1:1 image provides a simple composition with strong subject presence and is commonly used for products, avatars, and close-up content. A 4:3 image provides a compact frame that is relatively easy to compose. A 16:9 image provides a wider field of view and is widely used for video-oriented content.

The complete admissible set specified in our guideline is

$$\mathcal{R} = \left\{1, \frac{4}{3}, \frac{16}{9}, \frac{3}{4}, \frac{9}{16}\right\}. \tag{11}$$

For an image with width w and height h, its aspect ratio is compared with the nearest admissible value. We write the criterion as

$$\min_{r \in \mathcal{R}} \left| \frac{w}{h} - r \right| < 0.05. \tag{12}$$

Figure 18 illustrates the visual distortion caused by compressing an image vertically.

Image clarity. Recognizability is treated as a basic requirement for image presentation. Images should remain sufficiently clear for users to identify their visual content. The pixelated image in Figure 19 illustrates how reduced clarity obscures visual detail. We propose applying image-quality assessment algorithms to image regions identified in the interface.

Several commonly used image-quality metrics are listed as references for this purpose. Their definitions and properties are summarized in Table 5.

A.6 Shadow

Shadows originate from the physical relationship between objects and surfaces at different distances. User interfaces reproduce this visual cue to communicate the height and layer relationships among elements (Creager & Gillan, 2016). Components at different semantic layers therefore use different shadow properties.

We divide interface elements into four semantic levels. **Level 0** represents elements resting directly on the base surface. These include buttons, inputs, search fields, tags, tables, links, pagination con-

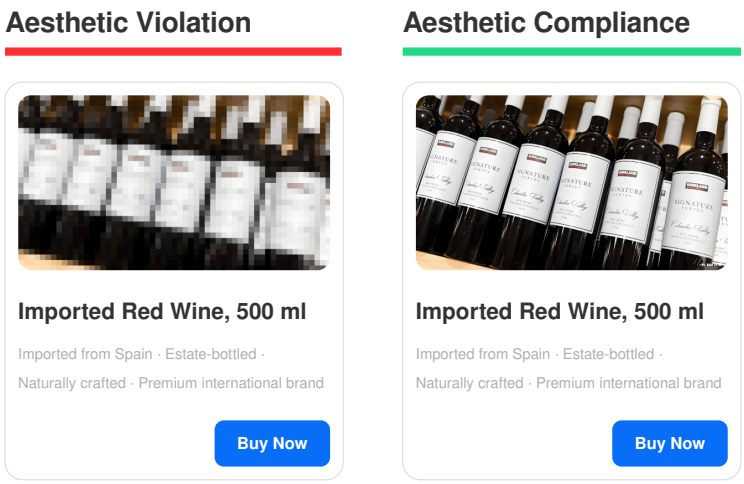

Figure 19: Image clarity. The image in the violating example is insufficiently clear, while the conforming version preserves recognizable visual detail.

Table 5: Image-quality metrics included in our professional UI design guideline as references for evaluating image clarity.

Metric	Definition and properties
PSNR	Peak signal-to-noise ratio is a full-reference image-quality metric that compares the maximum possible signal power with the power of corrupting noise. It is measured in decibels, and larger values indicate less distortion. Because it is based on pixel-level error, it does not explicitly model the visual characteristics of human perception.
SSIM	Structural similarity evaluates image similarity from luminance, contrast, and structure (Wang et al., 2004). Its value lies in $[-1, 1]$, with larger values indicating less distortion. In practical computation, an image can be divided into local windows. For N windows, the average structural similarity is $$\mathrm{MSSIM}(X, Y) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{SSIM}(x_i, y_i).$$
IFC	The information fidelity criterion evaluates image quality using natural scene statistics and characteristics of the human visual system. It measures the mutual information between a test image and a reference image (Sheikh et al., 2005).
VIF	Visual information fidelity extends the information-based formulation of IFC and focuses on the amount of visual information lost between a test image and its reference (Sheikh & Bovik, 2006).
MSE / RMSE	Mean squared error and root mean squared error measure pixel-level error between corresponding images. They are simple objective measures but do not explicitly account for characteristics of human visual perception.

trols, steps, breadcrumbs, switches, radio buttons, checkboxes, and progress bars. Their projection coincides with the component itself, so no visible shadow is used.

Level 1 represents low-level elevation and is used for navigation elements and card hover states. **Level 2** represents components that expand from elements on the base surface and remain associated with them, including drop-down containers and drawers. **Level 3** represents high-level components used for prominent prompts and operations, including dialogs, modal windows, and toast notifica-

Table 6: Shadow hierarchy specified in our professional design guideline, following the Fusion Design shadow scale.

Level	Typical components	Offset d	Blur	color
0	Buttons, inputs, search fields, tags, tables, links, pagination, steps, breadcrumbs, switches, radio buttons, checkboxes, and progress bars	0	0	`rgba(0,0,0,0)`
1	Navigation and card hover states	1 px	3 px	`rgba(0,0,0,0.12)`
2	Drop-down containers and drawers	2 px	4 px	`rgba(0,0,0,0.12)`
3	Dialogs, modal windows, and toasts	20 px	30 px	`rgba(0,0,0,0.15)`

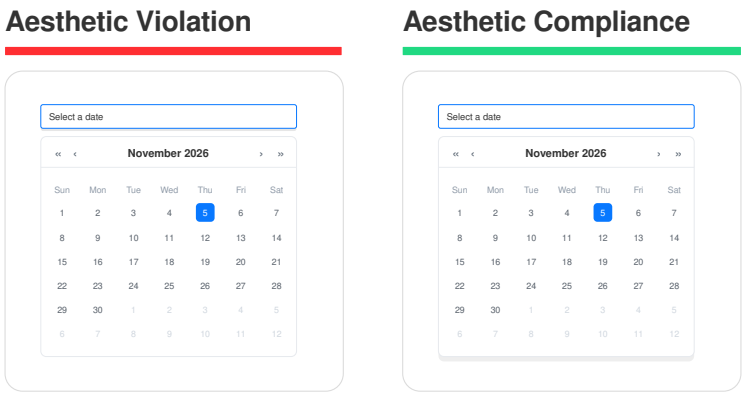

Figure 20: Shadow hierarchy. The violating example uses a shadow treatment inconsistent with the component's semantic level, while the conforming example follows the prescribed shadow hierarchy.

tions. For example, Figure 20 shows how a shadow distinguishes an expanded date picker from the base surface.

The shadow parameters specified in our guideline are summarized in Table 6. For Levels 1 to 3, the same offset magnitude is applied in five directions. These directions are (d, d), $(0, -d)$, $(d, 0)$, $(0, d)$, and $(-d, 0)$. The spread value is zero in all cases.

For implementation, UI components are first identified and their shadow parameters are extracted. The observed parameters are then compared with the semantic level associated with each component type.

A.7 Consistency

In addition to the visual quality of individual elements, overall interface quality depends strongly on whether visual rules are applied consistently across the page (Kellogg, 1987). We consider consistency at two levels: container consistency and component consistency.

Container consistency. Containers carry information and separate different content regions. Equivalent containers within the same page should use consistent corner radii, shadows, and spacing between containers. Figure 21 illustrates how consistent corner radii and spacing establish a regular arrangement of containers.

For implementation, container components are identified and their corner radius, shadow, and spacing parameters are compared across equivalent instances.

Component consistency. Components are among the most frequently repeated elements in an application. We therefore require one page to use components derived from a single specification.

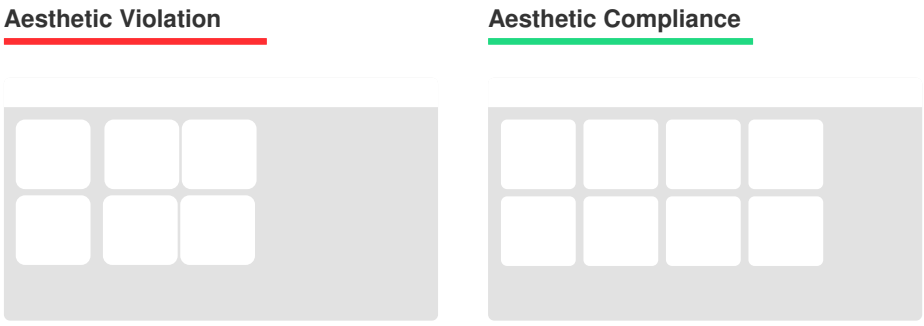

Figure 21: Container consistency. Equivalent containers use different corner radii, shadows, or spacing in the violating example, while the conforming version applies a consistent container specification.

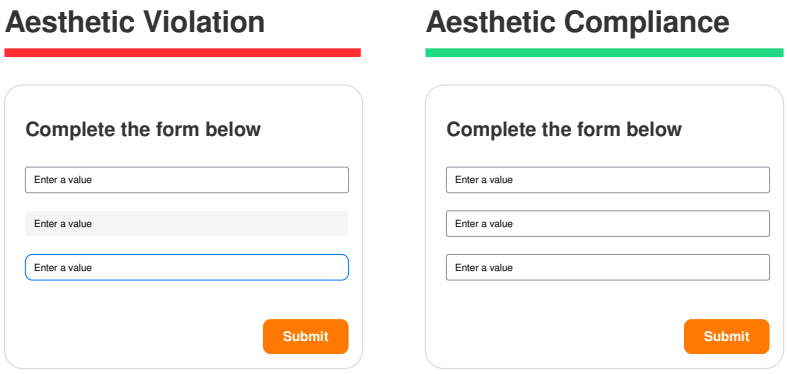

Figure 22: Component consistency. Instances of the same component type use different specifications in the violating example, while the conforming version follows a shared component specification.

Instances of the same component type should maintain consistent parameters, as illustrated by the input fields in Figure 22.

For implementation, component-recognition methods can be used to identify component instances on the page. Parameters are then compared within each component class to determine whether components of the same type remain consistent.

B Human Annotation and Evaluation Details

B.1 Annotation Procedure

We initially recruited 51 professional UI designers, all with at least 3 years of industry experience in interface and product design. Among them, 45.1% had at least 5 years of experience, including 17.8% with at least 10 years. All had worked at leading Internet and technology companies and had substantial experience with B2B or B2C products. All annotators held at least a bachelor's degree and had received formal training in design-related disciplines, including interaction design, visual communication, and related fields. All annotators followed the same annotation guidelines throughout the annotation process. Before formal annotation, a professional UI designer prepared a set of visual rubric exemplars to operationalize the 1–5 rating scale. For each of the eight aesthetic dimensions, the exemplar set contained one representative UI for each score level, resulting in 40

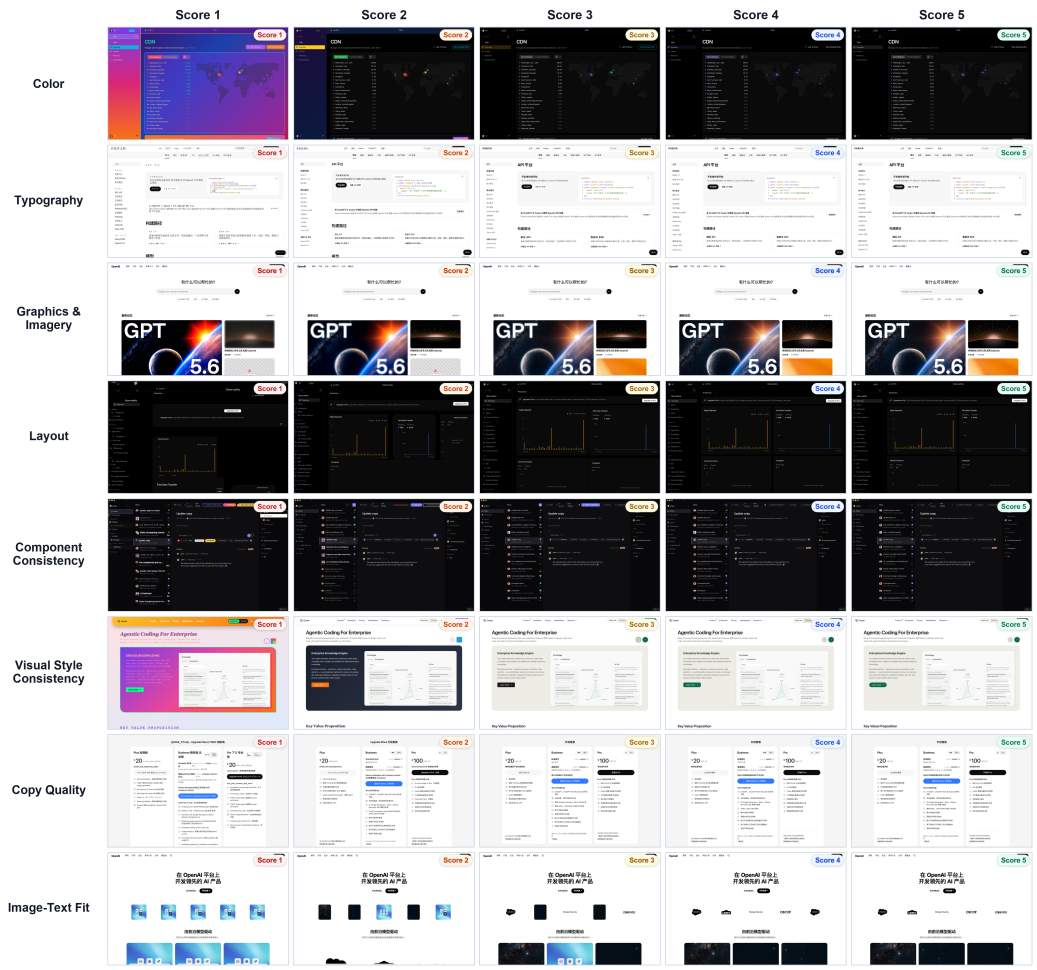

Figure 23: Visual rubric exemplars used to calibrate annotators before formal annotation. Rows correspond to the eight aesthetic dimensions, and columns correspond to rating levels from 1 to 5.

examples in total. All annotators reviewed the same exemplars together with the scoring guidelines before beginning formal annotation. This calibration step was intended to align their interpretation of the five score levels and reduce differences in individual rating scales. Figure 23 presents the complete exemplar set.

Each of the 1,395 reference UIs was independently rated by three designers along eight aesthetic dimensions, yielding 33,480 dimension-level ratings before quality control.

Given the importance of annotation quality for reliable model evaluation (Pang et al., 2025), we manually inspected the rating distribution of each designer as an annotator-level quality-control step. We excluded annotators exhibiting degenerate response patterns, defined as assigning the same extreme score (either 1 or 5) to all ratings they submitted. Seven annotators met this criterion, and all annotations from these annotators were removed. The final annotation set therefore contains ratings from 44 professional designers.

Our aesthetic guidelines contain a broader set of principle-level criteria, including rules for color, spacing, layout, typography, imagery, shadow, and consistency. To derive the dimensions used for human evaluation, we worked with the designers to consolidate these fine-grained principles into perceptually coherent categories that can be reliably judged from rendered UI screenshots. Closely related principles were grouped into broader evaluation dimensions. For example, spacing and spatial organization are jointly considered under layout, while component-level regularity and visual treatments across the page are evaluated separately as component consistency and visual

Table 7: Fine-grained criteria used for human UI evaluation.

Criterion	Definition
Color	Whether the page palette is visually harmonious, colors are used consistently, and sufficient contrast is maintained between text and background.
Typography	Whether font choices, sizes, weights, line heights, and textual hierarchy are clear, consistent, and readable.
Graphics & Imagery	Whether images, icons, illustrations, and other visual assets are clear, intact, appropriately proportioned, and visually well presented.
Layout	Whether page structure, information hierarchy, alignment, spacing, grouping, and space utilization are visually appropriate.
Component Consistency	Whether components with equivalent functions or hierarchy follow consistent rules in their size, structure, appearance, and state representation.
Visual Style Consistency	Whether corner radii, shadows, borders, line weights, icons, and other visual treatments form a coherent visual language across the page.
Copy Quality	Whether headings, body text, buttons, hints, and other textual content are clear, natural, concise, and appropriate for the page context.
Image–Text Fit	Whether images, icons, and other visual elements semantically match their associated text, function, and surrounding context.

style consistency. Shadow-related principles are assessed under visual style consistency, together with borders, corner radii, and other visual treatments. We retain color and typography as separate dimensions and assess graphical content under graphics and imagery. We additionally include copy quality to assess the clarity and appropriateness of textual content, and image–text fit to assess the semantic agreement between visual elements and their associated text or context. This adaptation yields the shared eight-dimensional rubric used for Aesthetic Scoring and Text-to-UI Generation (Tasks 1 and 4).

For each UI, annotators provided ratings from 1 to 5 along eight aesthetic dimensions: color, typography, graphics and imagery, layout, component consistency, visual style consistency, copy quality, and Image–Text Fit. For each evaluation, annotators were also asked to provide a brief comment summarizing their assessment. In addition, they recorded a concise description of the most salient design issue together with its corresponding region or component when applicable. The resulting annotations therefore contain both quantitative ratings and qualitative design feedback for each UI.

Ground-truth aggregation. After annotator-level quality control, we compute the mean opinion score (MOS) over all retained expert ratings for each UI and aesthetic dimension. No additional rating-level outlier filtering is applied. The resulting MOS is rounded to one decimal place and is used as the ground-truth score for subsequent evaluation.

B.2 Evaluation Criteria and Ground-Truth Distribution

Fine-grained criteria. Annotators evaluate each UI along eight fine-grained aesthetic criteria. Each criterion is independently rated from 1 to 5 according to its corresponding scoring rubric. Table 7 summarizes the definition of each criterion.

Ground-truth distribution. The benchmark ground truth consists of eight dimension-level MOS scores for each UI. Figure 24 compares the score distributions of the three individual annotators with the final aggregated MOS across all eight aesthetic dimensions. For visualization only, the MOS values are mapped to the nearest 1–5 rubric level, while all quantitative evaluations use the one-decimal MOS scores.

B.3 Inter-Annotator Reliability

We further assess the reliability of the professional aesthetic annotations. Given the ordinal 1–5 rating scale and the rotating-rater setting, we use ordinal Krippendorff's α to quantify inter-annotator agreement for each of the eight aesthetic dimensions. As shown in Table 8, α ranges from 0.328 to 0.431, with a macro-average of 0.368. Layout shows the highest agreement ($\alpha = 0.431$), while Color shows the lowest ($\alpha = 0.328$). Visual-style Consistency, Copy Quality, and Image–Text Fit obtain α values of 0.366, 0.334, and 0.402, respectively, and do not exhibit substantially larger

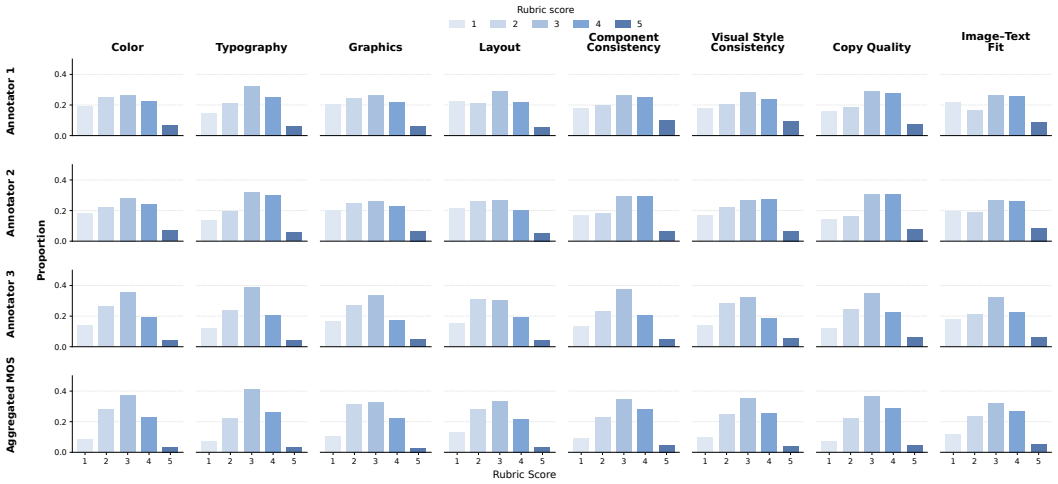

Figure 24: Score distributions from three individual annotators and the aggregated human MOS across eight aesthetic dimensions.

Table 8: Inter-annotator reliability across aesthetic dimensions. We report ordinal Krippendorff's α for inter-annotator agreement and ICC(1,3) for the reliability of the average of three ratings.

Dimension	**Krippendorff's α**	**ICC(1,3)**
Color	0.328	0.596
Typography	0.334	0.609
Graphics & Imagery	0.346	0.614
Layout	0.431	0.694
Component Consistency	0.399	0.672
Visual-style Consistency	0.366	0.635
Copy Quality	0.334	0.610
Image–Text Fit	0.402	0.667
Macro Avg.	**0.368**	**0.637**

disagreement than the other dimensions. Overall, the results indicate that professional designers share some common judgment on fine-grained UI aesthetics, while noticeable individual variation remains.

Since the benchmark uses the mean of the retained expert ratings as its ground-truth score, we additionally examine the reliability of rating aggregation. On the UIs retaining all three valid ratings, we use a one-way random-effects intraclass correlation coefficient (ICC), which is appropriate for the rotating-rater setting. The single-rating reliability, ICC(1,1), ranges from 0.329 to 0.431 across dimensions, whereas the reliability of the average of three independent ratings, ICC(1,3), ranges from 0.596 to 0.694, with a macro-average of 0.637. These results show that aggregating multiple professional ratings improves score reliability relative to relying on a single designer, while fine-grained aesthetic assessment still retains non-negligible inter-expert variation.

B.4 Model Evaluation Prompt

For model evaluation, we convert the human annotation criteria into a unified evaluation prompt. All evaluated models receive the same task instruction, scoring criteria, and output format. The complete prompt used for aesthetic scoring is shown in Figure 25, Figure 26, and Figure 27.

B.5 Annotation Examples

We provide three representative annotation examples in Figures 28, 29 and 30 to illustrate the human evaluation and ground-truth aggregation process. Each example presents the original UI screenshot, the independent ratings and qualitative comments from three professional designers.

Aesthetic Scoring Prompt, Part I

Task
You are a professional UI design reviewer. Rate the given UI page screenshot on each of the following UI aesthetic dimensions according to the corresponding 1–5 scoring rubric.

1. Color
Evaluate whether the palette harmonizes with the page as a whole, colors are used consistently, and text background contrast is clear.

- **5:** Mature and harmonious palette with clear roles for primary, secondary, background, and accent colors. Color usage follows a consistent logic. Text background contrast is clear, with no noticeable color clashes or excessive saturation.
- **4:** Overall harmonious palette with a clear color system. Only a few minor issues are present, such as an occasionally distracting color, slightly weak local contrast, or slight overuse of accent colors. Overall appearance and readability remain strong.
- **3:** Generally acceptable, but several noticeable problems are present, such as locally uncoordinated colors, unclear color hierarchy, or insufficient text background contrast. The page remains readable and usable.
- **2:** Serious palette problems occur across multiple regions, such as conflicting colors, inconsistent color usage, excessive saturation, or poor contrast that noticeably affects readability.
- **1:** Large-scale color conflicts or abnormal contrast make substantial portions of the content difficult to recognize and substantially impair the page's presentation and comprehension.

2. Typography
Evaluate whether fonts, sizes, weights, line heights, and text hierarchy are clear, reasonable, and readable.

- **5:** Complete and consistent type system with clear hierarchy among headings, body text, captions, and other textual elements. Font sizes, weights, and line heights are well coordinated and comfortable to read.
- **4:** Overall clear type system with only a few minor issues in font size, weight, or line height. Reading and information hierarchy remain clear.
- **3:** The page remains readable, but several noticeable typography issues are present, such as insufficient distinction between headings and body text, unstable size hierarchy, inconsistent weight usage, or locally cramped or loose text.
- **2:** The typography system is noticeably disorganized. Multiple textual levels are difficult to distinguish, and inappropriate sizes, weights, or line heights significantly affect readability.
- **1:** Large amounts of text are difficult to read because of severely inappropriate font sizes, styles, or arrangements, and the textual hierarchy is largely ineffective.

3. Graphics & Imagery
Evaluate whether images, icons, illustrations, and other visual assets are clear, intact, appropriately proportioned, and visually well presented.

- **5:** Images are sharp, intact, correctly proportioned, and naturally cropped. Icons and other graphics are refined and recognizable, with no noticeable stretching, blurring, damage, or missing assets.
- **4:** Overall asset quality is good, with only a few minor issues such as slightly imperfect cropping or less refined icons. The visual experience remains strong.
- **3:** Assets remain usable, but several perceptible problems are present, such as slightly blurry images, suboptimal cropping, mediocre icon quality, or graphics with limited recognizability.
- **2:** Multiple important visual assets are blurry, stretched, abnormally cropped, low quality, or missing, noticeably degrading the page's visual quality.
- **1:** Large numbers of core images or graphics are broken, missing, unrecognizable, or severely distorted, substantially disrupting the presentation of the page.

Figure 25: Model evaluation prompt for aesthetic scoring, Part I.

C More Details of AUV-Bench

C.1 Reference UI Collection and Quality Assurance

Reference UI Collection Details. We apply quality filtering, rendering validation, and duplicate removal to obtain the final reference UI collection. The adapted subset comprises 300 pages from

Aesthetic Scoring Prompt, Part II

4. Layout
Evaluate whether page structure, information hierarchy, alignment, spacing, grouping, and space utilization are reasonable.

- **5:** The page has a clear overall structure and well-defined visual priorities. Elements are accurately aligned, spacing is stable and regular, related content is properly grouped, and sections are clearly separated. No noticeable crowding, abnormal whitespace, misalignment, occlusion, or overflow is present.
- **4:** Overall layout is mature and reasonable, with only a few minor spacing, alignment, or proportion issues. Reading flow and information structure remain clear.
- **3:** The main structure is generally sound, but several noticeable problems are present, such as inconsistent alignment, unstable spacing, locally crowded or loose areas, or insufficiently clear information hierarchy.
- **2:** Serious layout problems appear across multiple areas, including obvious misalignment, crowding, abnormal whitespace, confused hierarchy, or unbalanced space allocation. Information access is noticeably affected.
- **1:** The layout is severely disrupted, with widespread overlapping, overflowing, misplaced, or structurally disorganized elements that make the page difficult to comprehend.

5. Component Consistency
Evaluate whether components with the same function or hierarchy follow consistent design rules.

- **5:** Buttons, cards, inputs, tags, navigation elements, and other component types follow clear and unified rules. Components with the same function use stable dimensions, structures, and state representations.
- **4:** The component system is generally consistent, with only occasional minor deviations in size, structure, or style.
- **3:** A basic component system is recognizable, but several noticeable inconsistencies are present, such as same-level buttons with different sizes, same-type cards with different structures, or identical states represented in different ways.
- **2:** Many similar components lack unified rules, and the same functions or hierarchy levels are frequently represented using different designs, resulting in weak overall system consistency.
- **1:** Similar components differ substantially in structure or appearance, resulting in a strongly fragmented and visibly inconsistent component system.

6. Visual Style Consistency
Evaluate whether the visual language, including corner radii, shadows, borders, line weights, icon styles, and decorative treatments, is consistent.

- **5:** The page has a stable and unified visual language. Corner radii, shadows, borders, line weights, icons, and decorative treatments follow clear and consistent rules across sections.
- **4:** The overall visual style is consistent, with only minor local deviations such as occasional differences in corner radius, shadow, or border style.
- **3:** A generally coherent visual direction is present, but several noticeable inconsistencies appear across sections in corner radii, shadows, borders, icons, or decorative treatments.
- **2:** Visual languages differ substantially across sections. Corner radii, shadows, borders, strokes, and icon styles lack consistent rules, producing a fragmented visual appearance.
- **1:** Sections follow substantially different visual systems, resulting in a severely inconsistent and disorganized overall style.

Figure 26: Model evaluation prompt for aesthetic scoring, Part II.

WebCode2M (Gui et al., 2025)[2] and 77 pages from Design2Code (Si et al., 2025)[3]. The collection contains 1,292 desktop pages, 92 mobile pages, and 11 tablet pages.

Source Adaptation and Asset Recovery. The reference collection combines real-world webpages, adapted public UI datasets, and model-generated pages. Each page is packaged with its source code, localized assets, and a full-page screenshot. For adapted pages with missing or placeholder images, we recover semantically appropriate visual assets using the surrounding context and alternative text. This process replaces 2,046 visual elements across 310 pages.

Rendering Fidelity. Real-world webpages often depend on remote images, fonts, stylesheets, and other resources whose availability can change over time. We localize these dependencies to obtain

[2] xcodemind/webcode2m_purified
[3] SALT-NLP/Design2Code-HARD

Aesthetic Scoring Prompt, Part III

7. Copy Quality
Evaluate whether the page text is clear, natural, easy to understand, and appropriate for the page context.

- **5:** Headings, body text, buttons, hints, and other copy are accurate, natural, concise, and context appropriate. The information is clearly expressed.
- **4:** Copy is generally clear and appropriate, with only minor unnatural phrasing, redundancy, or imprecise wording.
- **3:** Core content remains understandable, but several noticeable problems are present, such as stiff phrasing, vague information, repeated content, mediocre organization, or occasional context inappropriate expressions.
- **2:** Multiple important pieces of text are unclear, context inappropriate, semantically incorrect, or incomplete, noticeably affecting comprehension.
- **1:** Large amounts of core text are garbled, meaningless, substantially incorrect, or incomprehensible, preventing the page from effectively conveying its main information.

8. Image–Text Fit
Evaluate whether images, icons, and other visual elements semantically match their associated text, function, and surrounding context.

- **5:** Images, icons, and text are highly compatible. Visual elements accurately express the corresponding content or function and effectively support information comprehension.
- **4:** Image text relationships are generally appropriate, with only a few semantically imprecise visual elements that have little impact on comprehension.
- **3:** Most image text relationships are reasonable, but several issues are present, such as weakly relevant images, icons with unclear semantics, or visual elements that require additional context to understand.
- **2:** Multiple important images or icons clearly mismatch their associated text or function, creating substantial potential for misunderstanding or incorrect expectations.
- **1:** Image text relationships are widely incorrect, irrelevant, or misleading, and key visual elements substantially interfere with correct interpretation.

Output Requirements
Output only the following JSON object with no additional text:

```
{
"color": { "score": <1--5>, "reason": "<one-sentence
justification>" },
"typography": { "score": <1--5>, "reason":
"<one-sentence justification>" },
"graphics": { "score": <1--5>, "reason":
"<one-sentence justification>" },
"layout": { "score": <1--5>, "reason": "<one-sentence
justification>" },
"component_consistency": { "score": <1--5>, "reason":
"<one-sentence justification>" },
"visual_style_consistency": { "score": <1--5>, "reason":
"<one-sentence justification>" },
"copy_quality": { "score": <1--5>, "reason":
"<one-sentence justification>" },
"image_text_fit": { "score": <1--5>, "reason":
"<one-sentence justification>" }
}
```

Evaluation Rules

1. Score each aesthetic dimension independently according to its corresponding rubric.
2. Keep each reason to a single sentence in English.
3. Output only the specified JSON object.

Figure 27: Model evaluation prompt for aesthetic scoring, Part III.

self-contained pages that can be executed offline. For pages with original reference screenshots, we re-render each localized page under the corresponding viewport configuration and compare the result with the original screenshot.

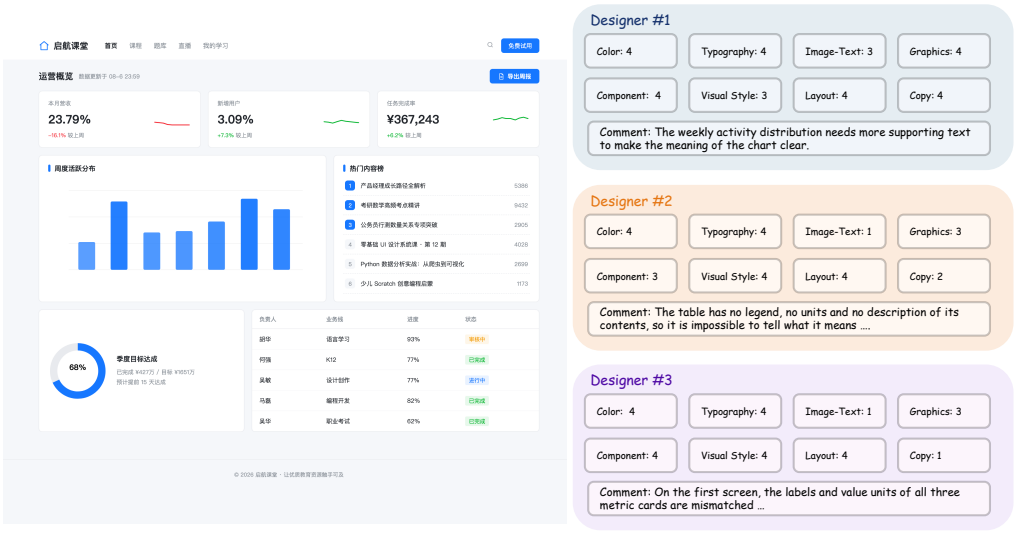

Figure 28: Representative human annotation example.

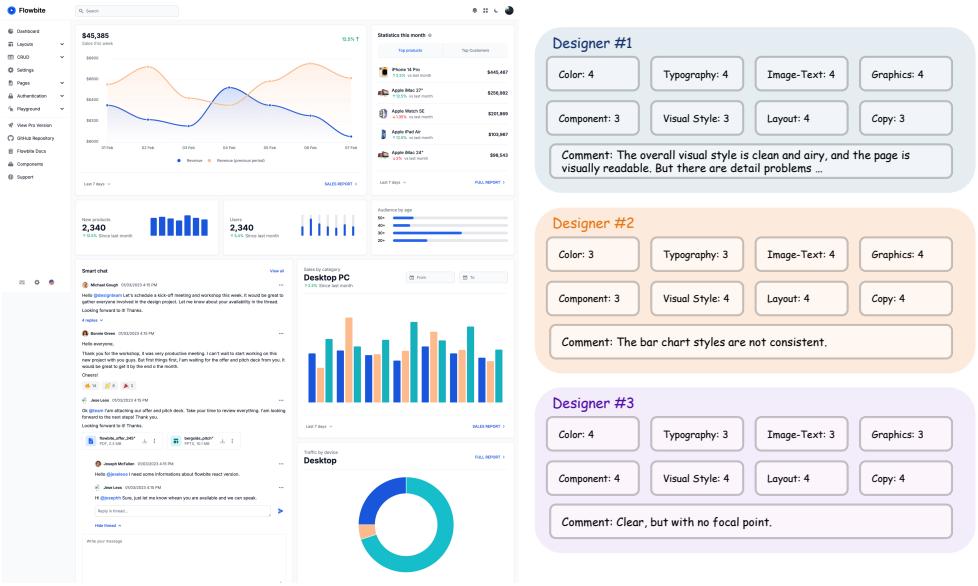

Figure 29: Representative human annotation example.

We measure rendering fidelity using pixel-level mean absolute error (MAE) and the structural similarity index (SSIM) (Wang et al., 2004). MAE measures pixel discrepancies, while SSIM measures structural visual similarity. We require $\mathrm{MAE} < 10$ and rank surplus candidates by SSIM; retained pages in this subset have a median SSIM of 0.994. Pages with substantial visual discrepancies are filtered or manually inspected for missing resources, layout drift, and abnormal rendering. Before finalizing the reference pool, professional UI designers inspect rendering quality, asset completeness, and layout integrity. These checks establish a stable reference for controlled degradation, reducing incidental rendering variation that could otherwise confound attribution of the injected defects.

C.2 Controlled Aesthetic Violations and Task Construction

Controlled Violation Taxonomy. Tasks 2 and 3 use controlled aesthetic violation rules that are localizable, editable in source code, and verifiable after rendering. We identify principles from the

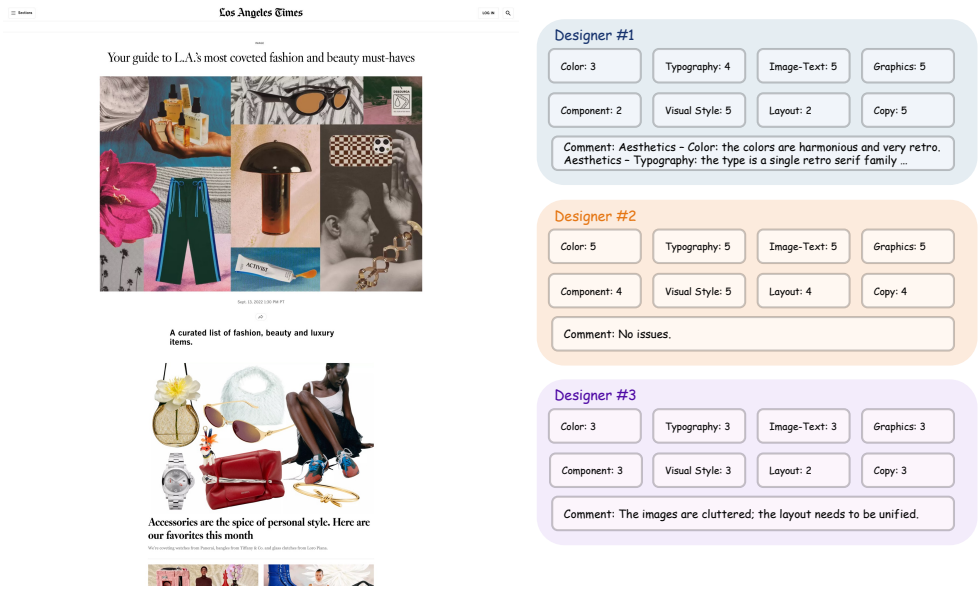

Figure 30: Representative human annotation example.

professional UI aesthetic guidelines that can be mapped to local visual elements or page structures and operationalize them as executable degradation rules. Rules whose effects are visually insignificant or difficult to attribute reliably are excluded. We run rule detectors over all 1,395 reference UIs to assess applicability, identify candidate target elements, and detect naturally occurring violations. A violation is instantiated only when the required structural or semantic conditions are present and the target region satisfies the corresponding principle. Each degraded instance therefore has an explicit target, an associated design principle, and a traceable visual modification.

In total, we define 13 controlled violation rules spanning five aesthetic dimensions: Typography, Layout and Reading Flow, Spacing, Visual Style Consistency, and Color. Table 9 summarizes the design constraint represented by each rule, the primary property being manipulated, and the resulting aesthetic violation.

Rule Applicability and Coverage. Different aesthetic rules require different page structures, semantic roles, and local design contexts, and are therefore not applicable to every reference UI. For each rule, we first perform an applicability check on the original page. A controlled violation is constructed only when the required candidate structure is present and the corresponding region of the reference UI already satisfies the target design principle.

For example, Peer Font Size and Peer Font Weight require comparable peer text elements; Container Radius Consistency requires a group of visually consistent peer containers; Table Text Alignment applies only to pages containing an appropriate tabular structure; and Semantic Color Stability and Fitts Edge Zone require more specific semantic or layout contexts. If the required element is absent, or if the reference UI already violates the target principle, the corresponding rule is not instantiated on that page.

After passing the applicability check, a degradation must additionally be locally executable, render reliably, and have an explicitly identifiable affected region. Across the candidate reference UIs, the 13 rules identify 77,277 valid candidate elements. Some rules are applicable to a broad range of pages, whereas others naturally arise only under more specific structural or semantic conditions. Consequently, the rule and dimension distributions in the formal evaluation cohort are determined by the actual applicability of each rule to the reference UIs, while preserving a well-defined design context and a verifiable clean-to-degraded correspondence for every controlled defect.

Degradation Execution and Validation. Each degradation modifies a small subset of eligible elements, typically around 20% of the target group, to keep the intervention localized and attributable.

Table 9: The 13 controlled aesthetic violation rules used in our benchmark. A rule is instantiated only when the reference UI satisfies its required structural or semantic context. T, L, S, V, and C denote Typography, Layout & Reading Flow, Spacing, Visual Style Consistency, and Color, respectively.

Dimension	Violation Rule	Description
T	**Peer Font Weight**	Changes the weight of a peer text element, breaking consistency among texts at the same visual level.
T	**Peer Font Size**	Changes the size of a peer text element, disrupting the local typographic hierarchy.
T	**Font Family Count**	Introduces an additional font family, increasing unnecessary variation in the page typography.
T	**Text Scale Count**	Introduces an additional font-size level, increasing complexity in the existing typographic scale.
T	**Table Text Alignment**	Changes text or numeric alignment within a table, breaking its alignment consistency.
L	**Vertical Alignment**	Offsets a peer element from its original vertical alignment.
L	**Gutenberg Information Region**	Alters the spatial or sequential organization of information elements under the Gutenberg reading pattern.
L	**Fitts Edge Zone**	Perturbs elements associated with page-edge regions, disrupting their original edge-layout relationship.
S	**Peer Spacing Uniformity**	Perturbs peer elements to break their originally uniform spacing.
S	**Container Row Spacing**	Changes spacing between rows or elements within comparable containers.
V	**Container Radius Consistency**	Changes the radius of a peer container, making it inconsistent with comparable components.
C	**Peer Accent Color**	Changes the accent color of a peer component, breaking color consistency among comparable components.
C	**Semantic Color Stability**	Perturbs a color with a stable semantic role, deviating from the page's semantic color system.

For example, Peer Font Size perturbs selected text elements at the same hierarchy level, while Peer Accent Color changes the accent color of a subset of comparable components. Professional UI designers iteratively review the rule definitions and perturbation parameters. We construct both single-violation and compositional cases, with mild and severe degradation levels.

Each modified page is re-rendered and automatically validated before inclusion. A candidate is retained only if the intended violation is introduced with a perceptible magnitude, remains localized without unintended layout changes or unrelated aesthetic violations, and yields a detected violation set matching the injected recipe. Candidates that fail these checks are discarded. Each retained instance contains the reference UI, the degraded UI, and precise violation annotations identifying the affected dimensions and regions. These instances and annotations are shared by Aesthetic Diagnosis and Aesthetic Repair.

Diagnosis and Repair Cohorts. Based on these controlled violation rules, we construct the formal Aesthetic Diagnosis and Aesthetic Repair tasks. To enable a controlled comparison between diagnosis and repair, Tasks 2 and 3 share exactly the same 660 underlying reference UI instances and the same 1,209 controlled aesthetic defects.

For Task 2, each degraded UI is paired with its corresponding clean control, resulting in 660 UI pairs for evaluating whether a model can identify and localize the injected aesthetic issues from visual inputs. Task 3 uses the same 660 degraded UIs and defect annotations to construct 660 repair instances, evaluating whether a model can correct the corresponding issues by modifying

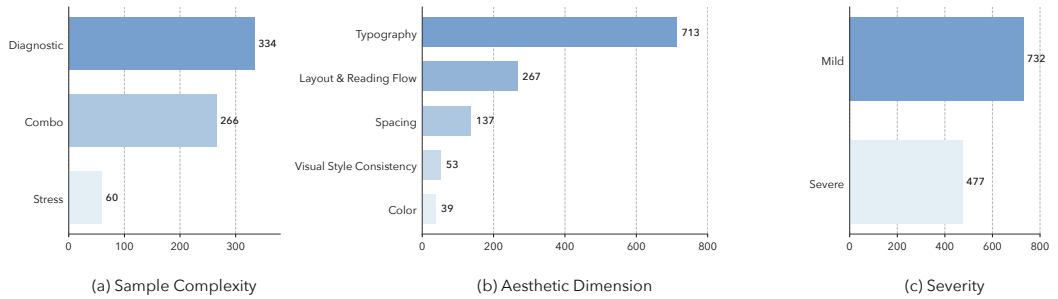

Figure 31: Distribution of the formal evaluation cohort shared by Tasks 2 and 3. (a) Complexity distribution of the 660 underlying UI instances. (b) Distribution of the 1,209 controlled aesthetic defects across five aesthetic dimensions. (c) Distribution of mild and severe defects.

the executable page. Sharing the same underlying pages and defect ground truth allows the two capabilities to be evaluated under closely matched data conditions.

Defect Complexity Splits. We further organize the shared Diagnosis–Repair cohort into three regimes according to violation multiplicity, each targeting a distinct diagnostic challenge. The *diagnostic* split contains exactly one injected violation, enabling rule-level attribution of detection failures to a specific aesthetic rule. The *combo* split contains 2–3 co-occurring violations and tests search completeness, i.e., whether a model continues to identify the remaining defects after detecting one issue. The *stress* split contains 4–6 simultaneous violations and probes recall under defect-dense interfaces. Together, these splits progress from isolated rule recognition to compositional diagnosis and high-density defect retrieval. As shown in Figure 31, the 660 instances are further organized into three levels according to defect composition. The *diagnostic* split contains 334 instances with a single controlled defect, providing a basic setting for isolated aesthetic issues. The *combo* split contains 266 instances with multiple simultaneous defects, while the remaining 60 *stress* instances contain more complex defect combinations.

The 1,209 defects span all five aesthetic dimensions: 713 in Typography, 267 in Layout and Reading Flow, 137 in Spacing, 53 in Visual Style Consistency, and 39 in Color. In terms of degradation strength, 732 defects are categorized as mild and 477 as severe. The number of degraded instances is naturally imbalanced across violation rules because not every degradation can be validly applied to every real-world UI. We apply a degradation only when the original UI satisfies the rule-specific applicability conditions, rather than forcing each rule onto a predefined number of pages. Consequently, rules that are applicable to more reference UIs yield more degraded instances, while those with more restrictive applicability conditions yield fewer.

The shared cohort establishes a direct correspondence between Tasks 2 and 3: the former evaluates whether a model can detect and localize a given aesthetic issue, whereas the latter evaluates whether it can take effective corrective action on the same issue. Because evaluating capabilities in isolation can overlook their interactions (Wu et al., 2026), we hold the underlying UI instances, defect definitions, and defect distribution fixed to analyze the alignment between principle-level judgment and UI repair. For the cross-task Judgment–Action Association (JAA) analysis, correct principle-level judgment requires both correct degradation detection and exact attribution of the affected aesthetic dimensions; localization accuracy is evaluated separately and is not included in J_i.

Perceptual Comparison between Controlled Degradations and Real-World Defects. To examine whether our controlled degradations exhibit visual patterns similar to naturally occurring UI defects, we conduct a small-scale blind comparison study. For each of the 13 controlled violation rules, we collect five naturally occurring defect cases with clear visual evidence from public webpages, product bug reports, and official community forums, yielding 65 real-world defects in total. Each real-world case is paired with a controlled degradation from our benchmark corresponding to the same violation rule, resulting in 65 real–controlled pairs. Three non-expert participants independently evaluate all pairs. For each pair, the two UIs are presented in randomized left–right order, and participants are unaware of their sources and the degradation procedure. They are asked only to identify which UI issue was artificially introduced.

If the controlled degradations contained obvious synthetic artifacts, participants should be able to identify them substantially above the 50% chance level. The average source-identification accuracy is 56.4%, only modestly above chance. This result suggests that our controlled degradations do not exhibit readily identifiable synthetic artifacts and are perceptually similar to naturally occurring UI defects of the same type.

C.3 Requirement Reconstruction

For Text-to-UI Generation, we use GPT-5.4 to reconstruct an English natural-language requirement from each reference UI. The requirement takes the form of a product requirements document describing the page's purpose, information structure, major functional modules, and overall visual direction. Exact colors, font sizes, coordinates, and spacing values are omitted, leaving these visual implementation choices to the evaluated model.

Each requirement is accompanied by objectively verifiable checklist items and reviewed by professional UI designers. The requirement and checklist specify the functional intent and content to be preserved without prescribing the precise visual parameters of the reference page. This provides a common specification for comparing generated UIs while allowing different visual implementations.

At evaluation time, the model receives the reconstructed requirement, original textual content, and image assets, without access to the reference screenshot. Although the requirement is in English, visible text retains the language of the original page. The model determines the layout, visual hierarchy, typography, spacing, color, and component organization. Asset selection and presentation are described in Appendix D.2.

D Experimental Details

D.1 Model and Inference Settings

We evaluate a diverse set of frontier proprietary models and open-weight multimodal models. Within each task, all models use the same task instructions, input format, and evaluation protocol. All experiments use English prompts with temperature set to 0. The maximum output length is set to 2,000 tokens for Tasks 1 and 2 and 8,000 tokens for Task 3. Since Task 4 requires models to generate complete executable HTML pages, we do not impose an explicit maximum output length for this task. For API failures or invalid outputs, each sample is retried up to three times.

All open-weight models are evaluated on a single NVIDIA A800 GPU with 80 GB of memory, using a batch size of 8. Except for hardware-specific configurations, the open-weight and proprietary models use the same prompts, input preprocessing, and task settings.

D.2 Input Preprocessing and Asset Organization

Visual Inputs for Tasks 1–3. Tasks 1–3 use full-page webpage screenshots as visual inputs. Since some webpages are substantially taller than a typical single-image input, we adopt a unified long-page processing strategy that preserves both global page structure and local visual details. The webpages are rendered with a viewport width of 1440 pixels. Let h denote the height of a full-page screenshot. When $h \leq 1800$ pixels, the full screenshot is directly provided to the model without slicing. When $h > 1800$ pixels, the screenshot is divided vertically into slices of 1800 pixels in height, with an overlap of 180 pixels between adjacent slices, corresponding to a stride of 1620 pixels. The slice starting positions are therefore $0, 1620, 3240, \ldots$. If the final slice produced by this fixed stride does not align with the bottom of the page, we additionally include a bottom-aligned slice starting at $h - 1800$, ensuring that the complete page is covered. As a result, the overlap between the final two slices may be larger than 180 pixels.

For webpages requiring multiple slices, we additionally provide a full-page overview image, obtained by proportionally resizing the original screenshot to a height of 1500 pixels. The overview image preserves the overall layout, information structure, and relative positions of different page regions, while the high-resolution slices retain local details such as typography, spacing, alignment, and component styling. The model first receives the full-page overview, followed by all slices in top-to-bottom order. Each slice is explicitly labeled with its position in the sequence.

The same visual preprocessing protocol is used across Tasks 1–3. For Task 2, the clean and degraded versions are processed independently using the same slicing strategy, and their presentation order is randomized before being given to the model. Task 3 uses the same visual representation of the degraded UI as Task 2, together with the corresponding HTML source code and the structured diagnosis produced by the same model in Task 2.

Asset Inputs for Task 4. Task 4 provides each model with an English webpage requirement, the visible textual content from the original page, and a set of visual assets extracted from the corresponding Reference UI. Since a webpage may contain a large number of duplicated, decorative, or low-information images, we apply a fixed asset filtering procedure before generation. The procedure includes geometric filtering, duplicate removal, near-duplicate clustering, filtering of primarily decorative assets, and relevance-based ranking of the remaining candidates. At most 32 visual assets are retained for each webpage.

Each retained asset is assigned a deterministic identifier, `V01`, `V02`, . . ., `V32`, and is made available through a corresponding local relative path. To provide direct visual access to the selected assets, we organize them into at most two 4×4 contact sheets, each containing up to 16 assets. Each entry is labeled with its asset identifier, filename, and original image dimensions.

We additionally provide a compact textual description for each selected asset, including its identifier, relative path, width, height, and a short semantic caption. The visual contact sheets allow the model to inspect the appearance of the candidate assets, while the textual descriptions provide explicit correspondence between their semantic content and local file paths. The generated HTML can therefore reference the selected assets using the provided relative paths.

All webpage requirements in Task 4 are written in English, while the visible text extracted from the original webpage is preserved in its source language. Therefore, although the generation instructions are consistently given in English, the target webpage content may span different languages. Models do not have access to the Reference UI screenshot during generation; the layout, visual hierarchy, typography, spacing, color, and component organization must be determined from the provided requirement, textual content, and visual assets.

For webpages with no usable visual assets after filtering, the model is provided with an empty asset list and is not allowed to introduce external images or online placeholder services. For webpages with available assets, the generated page may only reference the local visual resources provided for that sample.

E More Experimental Analysis

E.1 Validation and Calibration of GPT-5.4 as an Automatic Aesthetic Judge

To provide an automatic aesthetic assessment for Task 4, we employ GPT-5.4-2026-03-05 as the aesthetic judge. This model is not included among the evaluated models in the main experiments. Before applying it to generated UIs, we first examine its agreement with professional human judgments using the designer-annotated data from Task 1.

Specifically, we follow the same eight-dimensional aesthetic rubric, image-input protocol, and inference setting as in Task 1. GPT-5.4 assigns a score from 1 to 5 for each of the eight aesthetic dimensions: color, typography, graphics and imagery, layout, component consistency, visual-style consistency, copy quality, and image–text fit. We compare these ratings against the cleaned professional-designer Mean Opinion Scores (MOS) using Spearman Rank Correlation (SRCC) and Mean Absolute Error (MAE).

As shown in Table 10, the raw GPT-5.4 ratings exhibit moderate ranking agreement with professional designers, with a macro-average SRCC of 0.570. However, the raw ratings also show substantial systematic overestimation: the macro-average MAE is 0.950 and the mean signed error is $+0.808$. The bias is consistently positive across all eight dimensions, indicating that GPT-5.4 tends to assign more lenient numerical ratings than professional designers.

Human-Anchored Calibration. To correct this systematic scoring bias, we perform post-hoc calibration using the existing Task 1 human annotations. For each aesthetic dimension d, we independently fit a monotonically non-decreasing isotonic regression function $f_d(\cdot)$ that maps the raw

Table 10: Five-fold cross-validation of GPT-5.4 aesthetic-score calibration against professional-designer MOS on Task 1. Bias denotes the mean signed error (*prediction − MOS*); values closer to zero indicate better calibration. Macro denotes the equally weighted average across the eight aesthetic dimensions.

Dimension	MAE ↓		Bias → 0		SRCC ↑	
	Raw	Cal.	Raw	Cal.	Raw	Cal.
Color	0.95	**0.61**	+0.86	0.00	0.55	0.48
Typography	0.70	**0.55**	+0.52	0.00	0.60	0.54
Graphics & Imagery	0.98	**0.61**	+0.85	0.00	0.59	0.52
Layout	0.71	**0.58**	+0.50	0.00	0.66	0.60
Component Consistency	1.03	**0.58**	+0.95	0.00	0.62	0.55
Visual Style Consistency	1.00	**0.60**	+0.87	0.00	0.58	0.54
Copy Quality	1.05	**0.69**	+0.84	0.00	0.36	0.30
Image–Text Fit	1.17	**0.64**	+1.07	0.00	0.60	0.55
Macro	**0.95**	**0.61**	**+0.81**	**0.00**	**0.57**	**0.51**

1–5 GPT-5.4 rating to the corresponding continuous professional-designer MOS. Each mapping is constrained to the range $[1, 5]$, and the eight dimensions are calibrated independently.

All calibration functions are learned exclusively from Task 1. We first evaluate the calibration procedure using five-fold cross-validation: for each fold, the calibration mapping is fitted only on the training split and evaluated on the held-out split. After validation, we refit the eight calibration functions using all 1,395 successfully aligned Task 1 examples and freeze these mappings before applying them to Task 4. No Task 4 generation results, Pairwise Win Rate, or other Task 4 evaluation signals are used to fit or select the calibration functions.

Calibration Validation. Table 10 reports pooled out-of-fold results from the five-fold cross-validation. Calibration consistently reduces MAE across all eight aesthetic dimensions, lowering the macro-average MAE from 0.950 to 0.607, corresponding to a 36.1% reduction. Meanwhile, the macro-average signed bias decreases from $+0.808$ to approximately zero, indicating that the systematic leniency of the raw judge scores is effectively removed.

The macro-average SRCC decreases moderately from 0.570 to 0.508 after calibration. The reduction is largely attributable to additional ties introduced by the piecewise-constant isotonic mappings. Our calibration is intended to improve numerical agreement with the professional-designer rating scale rather than to optimize ranking performance.

Based on this validation, we apply the frozen dimension-specific calibration functions to the GPT-5.4 ratings in Task 4. For each valid generated UI, we first calibrate its eight dimension-level ratings independently and then average the calibrated scores to obtain the *Aesthetic Judge Score*. Invalid or failed generations receive a score of zero, and the model-level score is computed over the same fixed evaluation denominator for all models. The resulting metric therefore provides a human-anchored automatic assessment of generated UI aesthetics, while Pairwise Win Rate serves as a complementary measure of relative holistic preference.

E.2 Additional Fine-Grained Analyses

Effect of Defect Complexity. We first examine how model performance changes as multiple aesthetic violations co-occur within the same interface. As shown in Figure 32, increasing defect complexity has substantially different effects on coarse detection and fine-grained diagnosis. Degradation detection remains stable or even improves from single-violation to compositional and stress settings, suggesting that an interface becomes easier to recognize as problematic when more violations are present. In contrast, attribution, exact-chain diagnosis, and repair degrade consistently as the number of simultaneous defects increases. For example, Claude Opus 5 maintains high detection accuracy, while its ECS decreases from 0.374 to 0.139 and further to 0.017; its Repair Pass similarly drops from 0.859 to 0.680 and 0.517. GPT-5.6 Sol exhibits the same pattern, with ECS decreasing from 0.311 to 0.079 and 0.050, and Repair Pass from 0.850 to 0.466 and 0.400. These results indi-

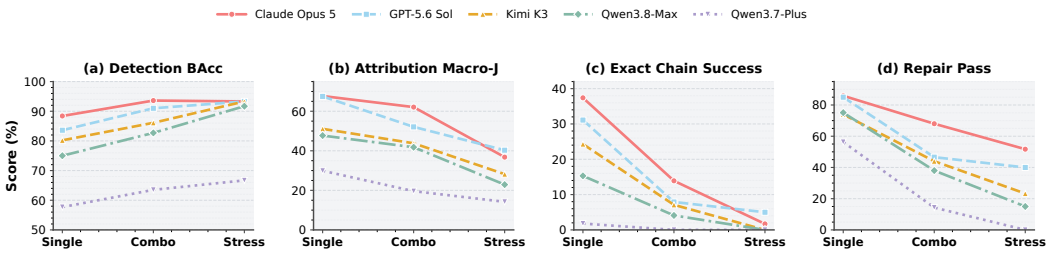

Figure 32: Performance under increasing defect complexity. We compare single-violation (*Diagnostic*), compositional (*Combo*), and high-density (*Stress*) settings across degradation detection, violation attribution, exact-chain diagnosis, and aesthetic repair. Detection remains comparatively stable as more defects are introduced, whereas fine-grained diagnosis and repair become substantially more difficult.

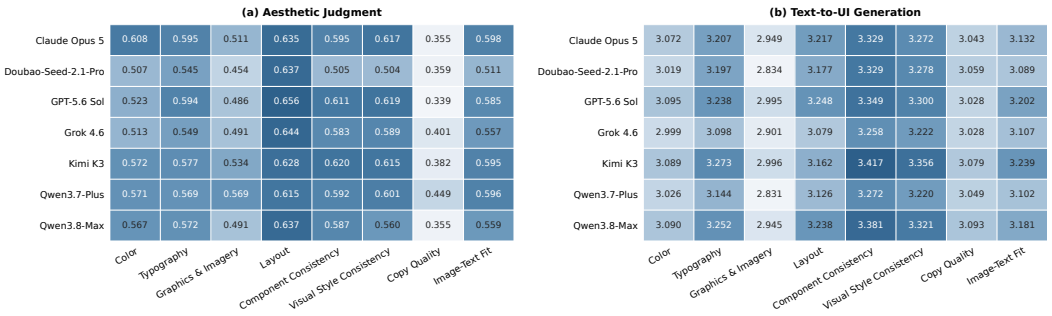

Figure 33: Dimension-wise aesthetic capability profiles. **Left:** SRCC with professional-designer MOS for Aesthetic Scoring. **Right:** human-calibrated aesthetic scores for successfully rendered Text-to-UI generations. The two panels use the same eight-dimensional perceptual rubric but characterize scoring and generation, respectively.

cate that the principal challenge under compositional defects is not recognizing that an interface is aesthetically problematic, but exhaustively identifying and correcting the interacting violations.

Dimension-Wise Aesthetic Capability Profiles. Figure 33 further decomposes aesthetic performance along the eight perceptual dimensions shared by Aesthetic Scoring and Text-to-UI Generation. For Aesthetic Scoring, the strongest frontier models show consistently higher agreement with professional designers on *layout* (SRCC 0.615–0.656), while *copy quality* is consistently more difficult (SRCC 0.339–0.449). This suggests that models capture global spatial organization more reliably than the contextual appropriateness and quality of interface copy.

A different profile emerges for generated interfaces. Among successfully rendered generations, *component consistency* and *visual-style consistency* generally receive the highest calibrated aesthetic scores, whereas *graphics and imagery* is consistently among the weakest dimensions. For example, GPT-5.6 Sol scores 3.349 on component consistency but 2.995 on graphics and imagery, while Kimi-K3 scores 3.417 and 2.996, respectively. These dimension-level results show that aggregate aesthetic scores conceal systematic differences in the types of aesthetic knowledge that current models can judge and realize.

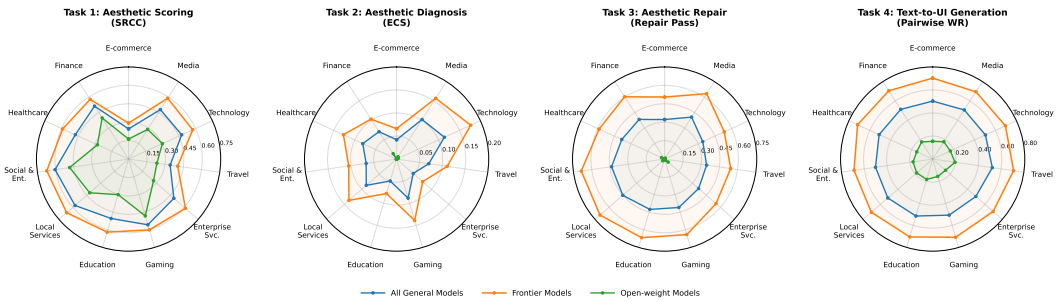

Figure 34: Absolute performance across UI industries. Group-average performance across 11 industries for Aesthetic Scoring (SRCC), Aesthetic Diagnosis (ECS), Aesthetic Repair (Repair Pass), and Text-to-UI Generation (Pairwise Win Rate). Each panel uses the original scale of its corresponding metric and should be compared within task.

What Limits Successful Repair? To identify the source of repair failures, Table 11 decomposes the official Repair Pass criterion into target correction (F), collateral preservation (C), content preservation (P), and visual-scope compliance (S). For most frontier models, collateral and content preservation are already high, while target correction and visual-scope compliance remain more restrictive. Qwen3.7-Plus, for example, achieves 0.995 collateral preservation and perfect content preservation, but only 0.371 target-fix success, resulting in a Repair Pass of 0.342. Similarly, Doubao-Seed-2.1-Pro preserves collateral and content at 0.985 and 0.992, respectively, but reaches only 0.498 on target correction. In contrast, Claude Opus 5 achieves substantially stronger target correction (0.826) and visual-scope compliance (0.824). These results suggest that repair failures are driven primarily by the difficulty of correctly resolving the intended aesthetic defect while keeping the modification appropriately localized, rather than by widespread corruption of unaffected content.

Table 11: Decomposition of Aesthetic Repair performance. F: target fix; C: collateral preservation; P: content preservation; S: visual-scope compliance.

Model	F	C	P	S	Repair
Claude Opus 5	0.83	0.97	0.97	0.82	**0.76**
Doubao-Seed-2.1-Pro	0.50	0.99	0.99	0.70	0.45
GPT-5.6 Sol	0.69	0.95	0.95	0.75	0.66
Grok 4.6	0.74	0.94	0.94	0.77	0.70
Kimi-K3	0.59	0.77	0.77	0.64	0.57
Qwen3.7-Plus	0.37	**1.00**	**1.00**	0.77	0.34
Qwen3.8-Max	0.62	0.98	0.99	0.71	0.55

E.3 Performance across Industries

We further examine how model performance varies across 11 UI industries. We first characterize the overall industry-level patterns across model groups, then inspect individual frontier models, and finally analyze whether these differences persist after controlling for model strength and where they emerge along the diagnosis–repair pipeline.

Industry sensitivity differs substantially across tasks. Figure 34 provides a group-level view of industry variation. For frontier models, Aesthetic Scoring and Repair vary substantially across industries: SRCC ranges from 0.293 on E-commerce to 0.673 on Social & Entertainment, while Repair Pass ranges from 0.505 on E-commerce to 0.696 on Local Services. In contrast, Text-to-UI Generation is considerably more stable, with frontier-model Pairwise Win Rate remaining around 0.69–0.72 across industries. Open-weight models show a large absolute gap, particularly in Diagnosis and Repair, where performance is close to the floor.

Group-level trends coexist with substantial model-specific variation. Figure 35 further decomposes the frontier-model average into individual models. Some industry patterns are shared across models: for example, Aesthetic Scoring is generally weaker on E-commerce and stronger on Local Services and Social & Entertainment. However, Diagnosis and Repair also exhibit substantial model-specific variation. In particular, ECS differs sharply across models, with some models remaining relatively strong across multiple industries while others approach zero on many categories. By comparison, Text-to-UI Generation is more stable across industries within the same model, and variation is dominated more by differences between models.

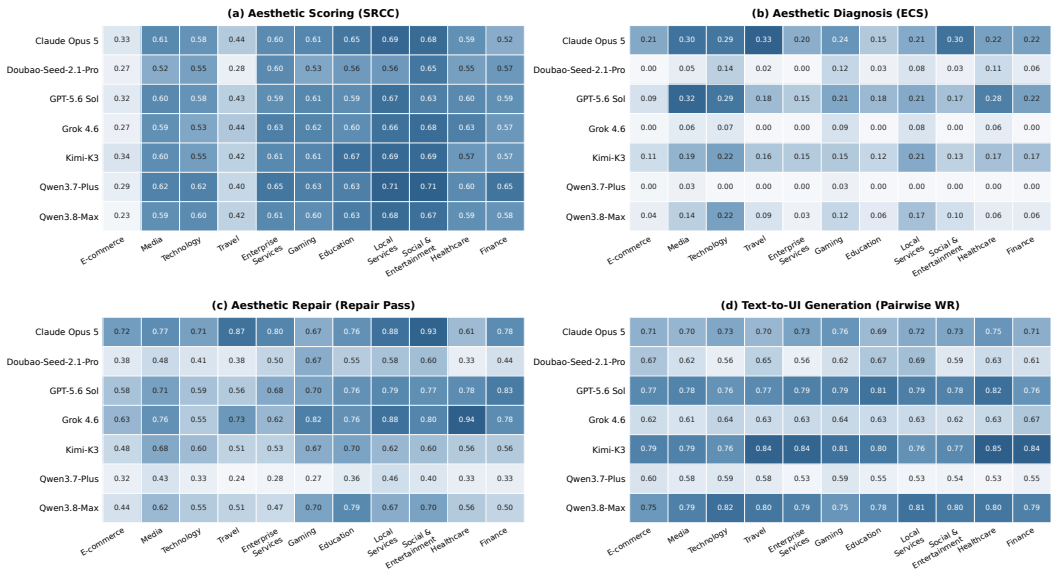

Figure 35: Fine-grained performance of individual frontier models across 11 UI industries. We report industry-wise results for (a) Aesthetic Scoring (SRCC), (b) Aesthetic Diagnosis (ECS), (c) Aesthetic Repair (Repair Pass), and (d) Text-to-UI Generation (Pairwise Win Rate). Cell values show the original metric scores, while color scales are independently normalized within each task for clearer comparison.

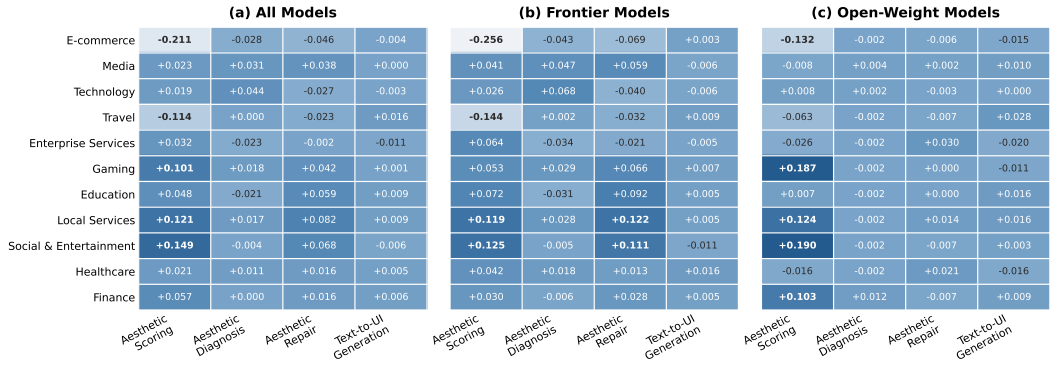

Figure 36: Model-normalized industry effects across the four tasks. Each cell reports the mean within-model deviation from overall performance: $\Delta_{g,t} = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}(S_{m,g,t} - S_{m,\text{overall},t})$. Positive values indicate above-overall performance and negative values indicate below-overall performance.

Industry effects remain after controlling for model strength. As shown in Figure 36, frontier models remain consistently below their overall level on E-commerce in Scoring (-0.256), Diagnosis (-0.043), and Repair (-0.069), while Local Services and Social & Entertainment show positive shifts in both Scoring and Repair. Similar trends appear across all general-purpose models. By contrast, Text-to-UI Pairwise Win Rate changes little across industries, with most normalized deviations within about ± 0.02. Open-weight models preserve some industry structure in Scoring, but floor effects largely obscure such variation in Diagnosis and Repair.

Industry differences become more pronounced at finer levels of aesthetic understanding. Figure 37 shows that frontier-model degradation detection remains relatively stable across industries, with BAcc around 0.74–0.85. Larger variation appears in attribution and exact-chain diagnosis, and persists in downstream repair. Open-weight models similarly retain non-trivial coarse detection ability but drop sharply on attribution, ECS, and Repair. This suggests that industry-specific challenges arise mainly when models must identify and act on specific aesthetic problems, rather than when simply recognizing that an interface is degraded.

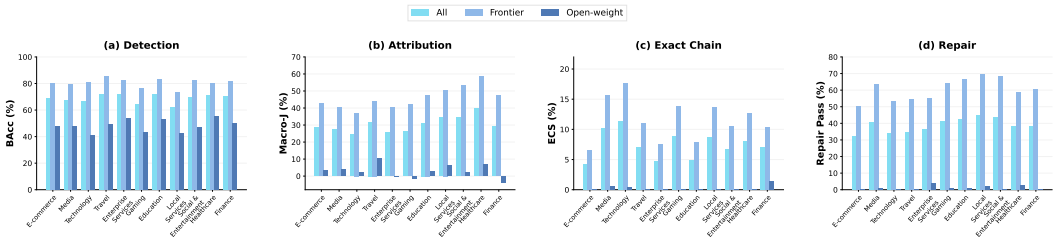

Figure 37: Industry-wise performance from coarse diagnosis to corrective action. We report degradation detection (BAcc), violation attribution (Macro-J), exact diagnosis-chain success (ECS), and final Repair Pass for the three model groups across industries.