COMMERCEVIBE: LEARNING TO DESIGN E-COMMERCE CREATIVES AS EXECUTABLE VISUAL CODE VIA DUAL-FEEDBACK REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

High-quality e-commerce creatives are essential for presenting products and conveying marketing messages. Recent diffusion models enable scalable creative generation and produce visually compelling images, but their flattened raster outputs often contain distorted text and inconsistent product details, requiring refinement before deployment. Moreover, without explicit structure, the resulting creatives are difficult to edit and reuse, while complex design requirements remain challenging to encode as verifiable training signals. To address these challenges, we present **CommerceVibe**, which represents creatives as executable visual code and formulates generation as conditional HTML/CSS program synthesis. Given product images, design requirements, and product information, it produces renderable, editable, and reusable creatives. We further introduce dual-feedback reinforcement learning, in which rule-based feedback evaluates rendered programs for text readability, product visibility, and layout validity, while visual feedback from a vision-language model (VLM) assesses rendered creatives against input specifications across six perceptual and commercial dimensions. Together, these complementary feedback signals improve both constraint satisfaction and perceptual quality. We perform supervised fine-tuning (SFT) of Qwen3.5-9B on over 28,000 e-commerce examples, followed by dual-feedback reinforcement learning. On a 1,300-case benchmark, the optimized **CommerceVibe** model achieves a weighted score of 94.0/100, compared with 87.3 for the SFT-only variant, and outperforms the three evaluated external models under the same one-pass HTML/CSS generation protocol. A blind evaluation by five e-commerce design experts provides independent evidence of improved output quality. **CommerceVibe** supports controllable and editable e-commerce creative generation through executable HTML/CSS.

1 INTRODUCTION

E-commerce creatives combine supplied product images, marketing copy, and graphic elements within a limited canvas. To be production-ready, a creative must preserve product appearance, present the required information accurately, and organize the content clearly. However, creating such materials manually for large product catalogs, frequent campaigns, and multiple output formats is difficult, while fixed template-based generation methods limit design diversity. These challenges call for generation methods that are reliable, controllable, and directly editable.

Diffusion models have substantially improved visual realism and controllable image synthesis through various conditioning strategies (Rombach et al., 2022; Saharia et al., 2022; Zhang et al., 2023; Li et al., 2023a), enabling large-scale creative generation. However, the resulting images may still contain incorrect text, altered product features, or fabricated selling points. Such issues are not merely visual defects, as they may misrepresent products, mislead customers, and make the creatives unsuitable for direct deployment. Moreover, diffusion models generate flattened raster images in which products, text, and backgrounds are not preserved as separate, editable elements. As a re-

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

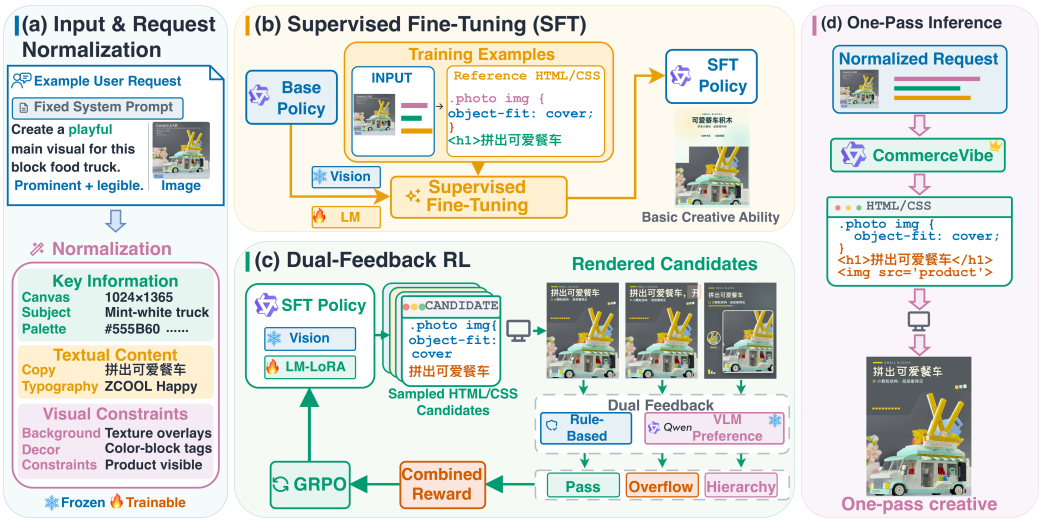

Figure 1: Overview of **CommerceVibe**. Supervised fine-tuning establishes basic creative generation ability, while dual-feedback reinforcement learning combines rule-based feedback with VLM-based preference feedback to optimize the policy through GRPO. The resulting policy generates executable HTML/CSS creatives in one pass.

sult, modifying textual content, replacing product images, or adjusting layouts often requires costly refinement or complete regeneration.

Structured visual representations preserve the identity of individual products, text blocks, and graphic elements, enabling element-level editing and reuse (Cheng et al., 2025; Wang et al., 2025b; Liu et al., 2026a). Beyond editability, executable representations further provide a way to verify whether generated creatives satisfy explicit constraints. An HTML/CSS document naturally supports such verification: after rendering, its visible content, image references, element positions, and boundaries can be inspected automatically. This allows errors such as text overflow, element occlusion, and out-of-bounds elements to be measured by rule-based checks and assigned severity scores (Pan et al., 2026).

However, converting these measurements into effective training signals for structured e-commerce creative generation raises two challenges. **(1)** The set of checks and their weights must align with the number of supplied product images. For example, with one product image, the product should be visually prominent. With multiple product images, all supplied products should be included in the creative, and occlusion between products must be detected. **(2)** Rule-based checks can identify structural errors such as text overflow, element occlusion, and out-of-bounds elements, but cannot fully judge the visual quality of a creative. Learned preference evaluators assess the rendered creative as a whole (Xu et al., 2023), but are less precise about element-level errors. These limitations motivate combining rule-based feedback for explicit structural constraints with learned preference feedback for overall visual quality.

To address these challenges, **CommerceVibe** represents e-commerce creative generation as conditional HTML/CSS program synthesis (Figure 1). A fixed preprocessing procedure is first used to convert each free-form user request into a consistent structured format. The normalized request, supplied product images, and product information are then provided to the policy under a fixed system instruction that defines the generator role and required output structure. The policy generates one complete HTML/CSS document in a single decoding pass. Supplied product images are directly referenced rather than regenerated, and text is represented as native HTML elements rather than synthesized in a raster image, so that product appearance is preserved and text distortion is avoided.

CommerceVibe is trained in two stages. First, the Qwen3.5-9B backbone (Qwen Team, 2026) is fine-tuned on 28,568 quality-controlled examples to establish a basic ability to generate e-commerce creatives. Second, the SFT policy is optimized with GRPO (Shao et al., 2024) under dual-feedback

reinforcement learning to further improve compliance with design requirements and overall visual quality.

To systematically evaluate model performance in e-commerce creative generation, we construct a 1,300-case benchmark using product images collected from real-world e-commerce scenarios, with cases containing varying numbers of input images. On this benchmark, **CommerceVibe** is compared with three leading external models—GPT-5.5 (OpenAI, 2026), Claude Opus 4.8 (Anthropic, 2026), and Gemini 3.5 Flash (Google DeepMind, 2026)—under the same one-pass HTML/CSS generation protocol. **CommerceVibe** achieves the highest weighted score in this comparison. An independent blind evaluation conducted by five e-commerce design experts further supports the overall creative quality of **CommerceVibe** and shows that the automatic evaluation is moderately associated with expert judgments. These results demonstrate the effectiveness of executable HTML/CSS generation for e-commerce creatives.

Our contributions are:

- We introduce **CommerceVibe**, which represents e-commerce creative generation as conditional HTML/CSS program synthesis, producing editable and executable creatives that directly reuse supplied product images, render text as native HTML elements, and provide verifiable structures for reinforcement learning optimization.
- We develop a dual-feedback reinforcement learning approach, which combines rule-based checks for Text, Product, and Layout errors with VLM-based preference feedback for visual and commercial quality.
- We construct a 1,300-case benchmark for e-commerce creative generation and show that **CommerceVibe** outperforms leading external models in both automatic evaluation and independent expert assessment.

2 Related Work

2.1 E-Commerce Creative Generation

Template-based systems place supplied assets into predefined layouts, whereas generative approaches support product staging, poster generation, and product–background synthesis (Chen et al., 2021; Ku et al., 2023; Lin et al., 2023; Cao et al., 2024; Wang et al., 2025a). Multi-stage pipelines and recent joint-control systems further coordinate layouts and generated backgrounds with supplied product subjects, glyphs, and visual styles (Li et al., 2023b; Chen et al., 2025a; Gao et al., 2026; Qin et al., 2026; Gao et al., 2025; Chen & Yang, 2026; Chen et al., 2026). Feedback- and behavior-driven methods improve reliability or commercial effectiveness through human annotations, click-through objectives, or personalization (Du et al., 2024; Yang et al., 2024; Fan et al., 2026; Xu et al., 2026). These methods improve the visual diversity and commercial relevance of e-commerce creatives, but their evaluation systems mainly assess the final image as a whole rather than the individual elements within it. Without a structured representation of these elements, errors such as distorted text and incorrect product features are difficult to detect and trace back to a specific requirement.

2.2 Structured and Executable Visual Code

Structured visual representations keep products, text, and graphic elements separate, enabling element-level editing and reuse. Prior work has generated HTML layouts, JSON design specifications, Figma and SVG components, and HTML posters (Tang et al., 2024; Cheng et al., 2025; Wang et al., 2025b; Liu et al., 2026a). PosterVerse uses an MLLM-powered HTML engine for scalable commercial-poster typography (Liu et al., 2026a). DesignAsCode formulates graphic design as HTML/CSS synthesis and repairs visual defects through render-and-reflect iterations (Liu et al., 2026d). Among these representations, HTML/CSS serves as a bridge between editable source code and the final rendered output: the same document can be rendered for inspection and revised at the element level (Guo et al., 2026). However, a document can be correctly rendered while using the wrong product image, clipping required text, or placing supplied images improperly. Verification therefore needs to compare the rendered document with the supplied assets and requirements. In CommerceVibe, rule profiles instantiate checks and weights by input-image count, and some checks

can be performed only after rendering. This routing is a task-specific design; alternative weighting schemes are not compared.

2.3 Verifiable Rule-Based and Preference Feedback

In rendering-aware reinforcement learning, structured graphics are optimized with validity, fidelity, semantic, and geometric signals; AeSlides, for example, converts slide-layout violations into verifiable rewards (Rodriguez et al., 2025; Chen et al., 2025b; Xing et al., 2025; Li et al., 2026; Pan et al., 2026). In contrast, learned evaluators have been used to provide supervision for qualities that are difficult to fully encode as rules, including rendered equivalence, typography, and aesthetics (Xu et al., 2023; Liu et al., 2026c; Lai et al., 2026). For example, AesthetiQ applies preference optimization to graphic layout generation and evaluates aesthetic quality with an MLLM (Patnaik et al., 2025). The two feedback types are complementary but individually incomplete. Rule-based feedback is able to identify explicit structural errors, but the required checks and weights vary with the number of supplied product images. Learned preference feedback assesses the visual quality of a rendered creative as a whole, but may miss localized errors such as missing products or overlapping elements. CommerceVibe assigns the two signals different roles. Rules target localized constraints; the fixed VLM Judge scores holistic quality.

Taken together, prior work has established creative generators, editable visual representations, and rendering-aware feedback. However, optimization of one-pass HTML/CSS generation for e-commerce creatives from supplied product images and user requirements remains underexplored. **CommerceVibe** uses task-specific rule feedback for localized constraints and a fixed VLM Judge for perceptual qualities.

3 CommerceVibe

This section describes how **CommerceVibe** generates e-commerce creatives as HTML/CSS documents and how rule-based and VLM-based feedback are used for optimization.

3.1 HTML/CSS Creative Generation

Let $r = (\mathcal{I}, u, m)$ denote the supplied product images, free-form user request, and optional product information. Before generation, a fixed preprocessing procedure converts the free-form request into a consistent structured format, $\tilde{u} = \mathrm{Normalize}(\mathcal{I}, u, m)$. The normalized request is provided as input context and is not updated during model optimization.

A fixed system instruction s defines the generator role and required output structure; its full text is provided in the supplementary material. The policy context is therefore given by $x = (s, \mathcal{I}, \tilde{u}, m)$. With $n = |\mathcal{I}|$, inputs with $n = 1$ are referred to as *Single*, and those with $n \geq 2$ as *Multi*. Only the parameters of the LLM policy are optimized.

The policy generates a sequence $c = (y_1, \ldots, y_T)$ that contains a complete HTML/CSS document and ends with an end-of-turn token. Each token is generated from the input context and the preceding tokens:

$$\pi_\theta(c \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}). \tag{1}$$

Each SFT example consists of an input r and a reference HTML/CSS document c^*. The policy is trained by minimizing the negative log-likelihood of target tokens:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathrm{E}_{(r,c^*)\sim\mathcal{D}_{\mathrm{SFT}}} \left[\sum_{t=1}^{T^*} \log \pi_\theta(y_t^* \mid x, y_{<t}^*) \right]. \tag{2}$$

Only the tokens in the reference HTML/CSS document and the end-of-turn token are supervised. SFT therefore learns to imitate reference documents but does not directly optimize the properties of rendered creatives.

During reinforcement learning, K candidate documents are sampled from the policy, $c_k \sim \pi_\theta(\cdot \mid x)$. Each candidate is rendered and captured once by a predefined browser-based procedure. The

procedure adapts to the document size and waits for all design elements and images to load before capturing the screenshot. It produces $z_k = \mathrm{Render}(c_k)$, which contains the DOM, element boxes, visible text, visible image references, and the final screenshot. These rendered outputs are used only to compute the two reward signals.

3.2 Rule-Based Checks for Rendered Creatives

Rule-based checks are divided into three common error types that are unacceptable in e-commerce creatives: Text, Product, and Layout errors, as illustrated in Figure 2. This division separates the main types of production errors and allows each to be measured from the rendered document.

Text Readability checks whether required information can be read clearly. It detects text overflow and overlap from rendered text boxes and clipping containers, and evaluates font size and luminance contrast for different text roles so that text remains legible on mobile screens. It also identifies text placed over a product, which can obscure product details. **Product Visibility** checks whether the supplied products are correctly used and clearly shown. Visible `img` elements are matched to the supplied product images by URL. The Single profile evaluates the visible area of the matched product image, whereas the Multi profile aggregates the visible areas of all matched product images. For transparent product images, crop and pairwise occlusion are measured using alpha masks; rendered geometry is used for opaque images. **Layout Validity** checks whether elements remain within the canvas, whether the visual center is balanced, and whether lower-page whitespace is appropriate, so that the creative has a clear and reasonable composition.

Rule settings are selected according to the number of supplied product images. For a Single input, the main product is expected to be visually prominent. For a Multi input, all supplied products are expected to appear, and occlusion is evaluated when at least two products have sufficient visible evidence. Each setting specifies the applicable checks, thresholds, weights, and critical violations. After rendering, only checks with the required evidence are applied, and the remaining scores are aggregated with their corresponding weights.

3.3 Rule Reward and Safeguards

Before rendering, malformed or incomplete HTML receives a negative reward. After rendering, hard gates reject outputs with an empty page, missing required product images, or severe product undercoverage. A soft text-sparsity penalty is applied when copy is largely omitted, preventing candidates from bypassing Text checks by leaving out required information.

For candidates that pass the hard gates, scores are computed only for the active dimensions and aggregated according to the selected rule setting. Critical errors, including text overflow, text overlap, elements outside the canvas, and product occlusion beyond the specified threshold, incur an additional penalty that increases with the number of errors. The resulting score is normalized as the rule reward R_{rule}. The three safeguards and their penalties are used only during reinforcement learning; automatic evaluation retains the same rule dimensions and weighted aggregation. Detailed detector definitions, rule settings, weights, and reward construction are provided in Supplementary Sections "Partially Conditional Rule Verifier" and "Reward Construction and Gates."

3.4 VLM Preference Feedback and GRPO

Figure 2 summarizes the two feedback branches. Rule-based checks identify explicit structural errors but cannot fully assess visual hierarchy, style coherence, product presentation, or marketing expression. A fixed external vision-language model, Qwen3-VL-Plus, serves as the Judge and is not updated during training. It evaluates the visual and commercial quality of the rendered creative along six dimensions: **visual appeal**, **product presentation**, **perceptual readability**, **marketing relevance**, **commercial usability**, and **copy faithfulness**. The Judge receives a screenshot of the rendered creative, the supplied product images, the normalized request, product information, and a scoring prompt that specifies the 1–5 criteria for each dimension. The six ratings are aggregated with low-rating penalties and normalized as the preference reward R_{pref}. The complete prompt, scoring criteria, and reward mapping are provided in Supplementary Section "Preference Judge and Combined Reward."

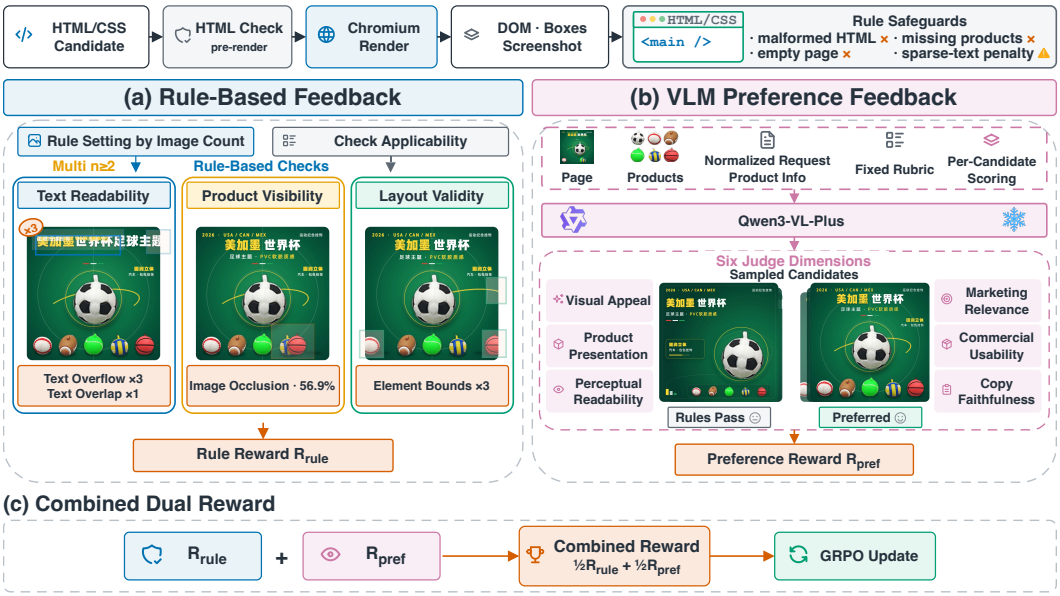

Figure 2: Dual-feedback reward design. (a) Rule-based feedback measures Text, Product, and Layout errors from rendered documents. (b) VLM-based preference feedback assesses visual and commercial quality independently of rule outputs. (c) The two terminal rewards are combined with equal weights and used for GRPO optimization.

Method	Overall ($N = 1{,}300$)				Single ($N = 1{,}140$)				Multi ($N = 160$)			
	$S_{\text{rule}}\uparrow$	$S_{\text{pref}}\uparrow$	$S_{\text{overall}}\uparrow$	$S_{\text{expert}}\uparrow$	$S_{\text{rule}}\uparrow$	$S_{\text{pref}}\uparrow$	$S_{\text{overall}}\uparrow$	$S_{\text{expert}}\uparrow$	$S_{\text{rule}}\uparrow$	$S_{\text{pref}}\uparrow$	$S_{\text{overall}}\uparrow$	$S_{\text{expert}}\uparrow$
External models												
GPT-5.5	42.4	37.7	80.1	78.9	42.1	37.6	79.7	77.9	44.7	38.4	83.1	<u>86.8</u>
Claude Opus 4.8	41.5	36.3	77.7	78.4	41.0	36.3	77.3	78.4	44.6	36.0	80.6	78.7
Gemini 3.5 Flash	42.1	33.1	75.3	75.9	41.8	33.2	75.0	75.8	44.8	32.6	77.4	76.2
***CommerceVibe** variants*												
Qwen3.5-9B	41.1	29.0	70.1	59.0	41.2	28.9	70.1	58.3	40.6	29.5	70.1	64.1
+ SFT	48.0	39.3	87.3	81.6	48.2	39.2	87.4	82.1	<u>46.2</u>	40.0	86.2	78.2
+ Rule-RLVR	<u>48.9</u>	<u>43.4</u>	<u>92.4</u>	<u>87.3</u>	<u>49.3</u>	<u>43.8</u>	<u>93.1</u>	<u>87.9</u>	<u>46.2</u>	40.6	86.8	83.0
+ Preference-RL	48.5	41.9	90.4	84.4	48.9	42.0	90.9	85.7	46.1	<u>41.0</u>	<u>87.1</u>	75.5
+ Rule+Preference RL	**49.2**	**44.8**	**94.0**	**90.0**	**49.4**	**44.9**	**94.3**	**90.4**	**47.8**	**44.1**	**91.9**	**87.3**

Table 1: Overall and Single/Multi results. S_{rule} and S_{pref} are out of 50; the other scores are out of 100. Indentation denotes SFT initialization; bold and underline mark the best and second-best scores in each column.

All three reinforcement-learning variants are initialized from the same SFT policy and trained on the same data with identical sampling and optimization settings. They differ only in the feedback used for optimization: rule-based reinforcement learning with verifiable rewards (Rule-RLVR), VLM-based Preference-RL, or their combination. For each input, the rewards of the K sampled candidates are standardized to obtain within-group relative advantages:

$$\widehat{A}_k = \frac{R_k - \text{mean}(R_{1:K})}{\text{std}(R_{1:K}) + \epsilon}. \tag{3}$$

The policy is updated with the standard clipped GRPO objective and reference-policy KL regularization.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data. Each SFT example consists of product images, associated product information, and a user design request collected from real-world e-commerce design scenarios. These inputs are provided to the model, while a reference HTML/CSS document serves as the SFT target. After quality filtering, the SFT corpus contains 28,568 training examples and 3,174 validation examples. The same normalized input contexts are reused in all RL runs.

Existing public benchmarks for poster and product-webpage generation do not jointly evaluate the strict requirements of e-commerce creatives, including the preservation of supplied product images and required copy, text readability, product visibility, and layout validity (Liu et al., 2026a;b). We therefore construct a 1,300-case benchmark for this task and use it to systematically evaluate model performance. It contains 1,140 Single-image and 160 Multi-image cases and covers 34 common e-commerce categories, including home goods, beauty products, consumer electronics, food and beverages, apparel, and pet products. The benchmark is separated from the training data at the product level, with a unique product set for each case. User design requests are normalized using the same procedure as in training. The full category distribution is provided in the supplementary material.

Compared Methods. Compared methods include the base Qwen3.5-9B policy, the SFT policy, three RL variants, and three external models—GPT-5.5 (OpenAI, 2026), Claude Opus 4.8 (Anthropic, 2026), and Gemini 3.5 Flash (Google DeepMind, 2026). The external comparison focuses on multimodal models capable of generating complete HTML/CSS documents under the same instruction. For each case, all methods receive the same normalized request, product images, and product information, and generate one complete creative without iterative correction or candidate reranking. Unless otherwise specified, **CommerceVibe** refers to the Rule+Preference RL policy.

Metrics. Automatic evaluation is performed offline using the same rule dimensions as reinforcement learning, except that safeguards and penalties are not applied. For each case, scores for the active rule dimensions are aggregated into $S_{\text{rule}} \in [0, 50]$. Preference scores are produced by Qwen3.8-Max, a separate evaluator from the Qwen3-VL-Plus Judge used during training. It rates the six preference dimensions on a 1–5 scale; the ratings are rescaled and aggregated into $S_{\text{pref}} \in [0, 50]$. The overall score is defined as $S_{\text{overall}} = S_{\text{rule}} + S_{\text{pref}} \in [0, 100]$. For the reported automatic results, we average scores from three API calls for each external model, three inference runs for Qwen3.5-9B, and three checkpoints trained with different seeds for each RL variant; SFT is evaluated once.

To complement automatic evaluation, an independent blind evaluation of the benchmark is conducted by five e-commerce design experts with 1–5 years of professional experience. For each case, outputs from all eight methods are anonymized and presented in random order, while all automatic scores are withheld. The ratings are aggregated with predefined weights into $S_{\text{expert}} \in [0, 100]$.

Implementation. The Qwen3.5-9B policy is fine-tuned for three epochs with an effective batch size of 32, a learning rate of 1×10^{-5}, and a maximum sequence length of 16,384. Each GRPO variant starts from this SFT policy and uses $K = 8$, a maximum completion length of 3,584, a maximum total length of 20,480, temperature 0.7, a learning rate of 5×10^{-5}, and an effective completion batch size of 64. Training uses 8 NVIDIA H20 GPUs with 96 GB of memory each; the fixed Qwen3-VL-Plus Judge uses temperature 0. The offline Qwen3.8-Max evaluator uses temperature 0 with thinking disabled.

4.2 MAIN RESULTS

Overall performance and external comparison. Table 1 reports reward-aligned automatic scores, followed by independent expert evaluation. The overall score increases from 70.1 for the base policy to 87.3 after SFT, and Rule+Preference RL achieves the highest score of 94.0. Among the external models, GPT-5.5 performs best, followed by Claude Opus 4.8 and Gemini 3.5 Flash. As shown in Figure 3, **CommerceVibe** achieves the strongest results across all three rule families and

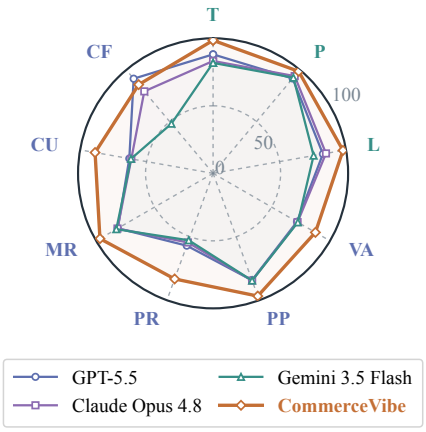

Figure 3: Nine-dimensional comparison with external models. T/P/L denote the three rule families, and VA/PP/PR/MR/CU/CF denote the six preference dimensions. All dimensions are normalized to 0–100.

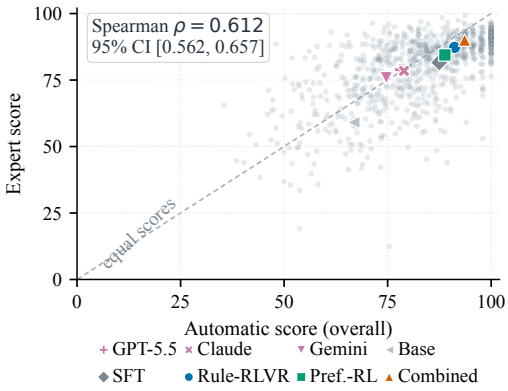

Figure 4: Association between automatic and expert scores on the benchmark. Gray points and colored markers denote individual cases and method means, respectively. The inset gives Spearman ρ with a 95% case-cluster bootstrap CI; the diagonal marks equality.

five of the six preference dimensions. GPT-5.5 retains an advantage in copy faithfulness. Claude Opus 4.8 performs relatively well in Product and Layout, whereas Gemini 3.5 Flash achieves comparable rule compliance but lower preference quality. The base policy mainly uses supplied product images and basic text, resulting in relatively high rule compliance but limited visual quality. In contrast, the external models and **CommerceVibe** generate more complex visual elements, which place greater demands on spatial organization and boundary control. The final **CommerceVibe** policy combines richer designs with the strongest structural and perceptual performance.

Complementary feedback across input complexity. All three RL variants improve upon the shared SFT policy across all reported metrics on Single-image cases, while Rule+Preference RL achieves the highest scores on both Single- and Multi-image cases. Relative to the stronger single-feedback variant in each setting, Rule+Preference RL gains 4.8 points on Multi-image cases and 1.2 points on Single-image cases. This larger Multi-image gain suggests stronger complementarity between the two feedback sources within the evaluated benchmark.

Expert and qualitative validation. The blind expert evaluation provides independent support for the automatic results, as shown in Figure 4. **CommerceVibe** achieves the highest overall expert score of 90.0 and ranks first on both the Single-image and Multi-image subsets. The expert ratings show high inter-rater reliability, with $\mathrm{ICC}(\mathrm{A}, 5) = 0.811$.

Expert scores provide clearer separation among the methods, particularly for the base policy, whose simple outputs satisfy many rule-based checks but receive substantially lower expert ratings, consistent with the preceding analysis. The higher expert score of **CommerceVibe** compared with Rule-RLVR suggests that VLM-based preference feedback complements rule-based feedback in improving perceptual quality. Figure 5 provides representative comparisons across Single- and Multi-image cases. The examples show that **CommerceVibe** more consistently preserves required information, presents supplied products clearly, and maintains coherent layouts, while producing more polished creatives with fewer localized errors than the compared methods.

4.3 Rule-Family Ablations

Table 2 shows that all three rule families contribute to their intended objectives. Removing Layout rules causes the largest within-family degradation, highlighting the importance of explicit geometric constraints. Removing Product rules causes a smaller decrease in Product score, suggesting that rule feedback refines the strong product-presentation capability learned during SFT.

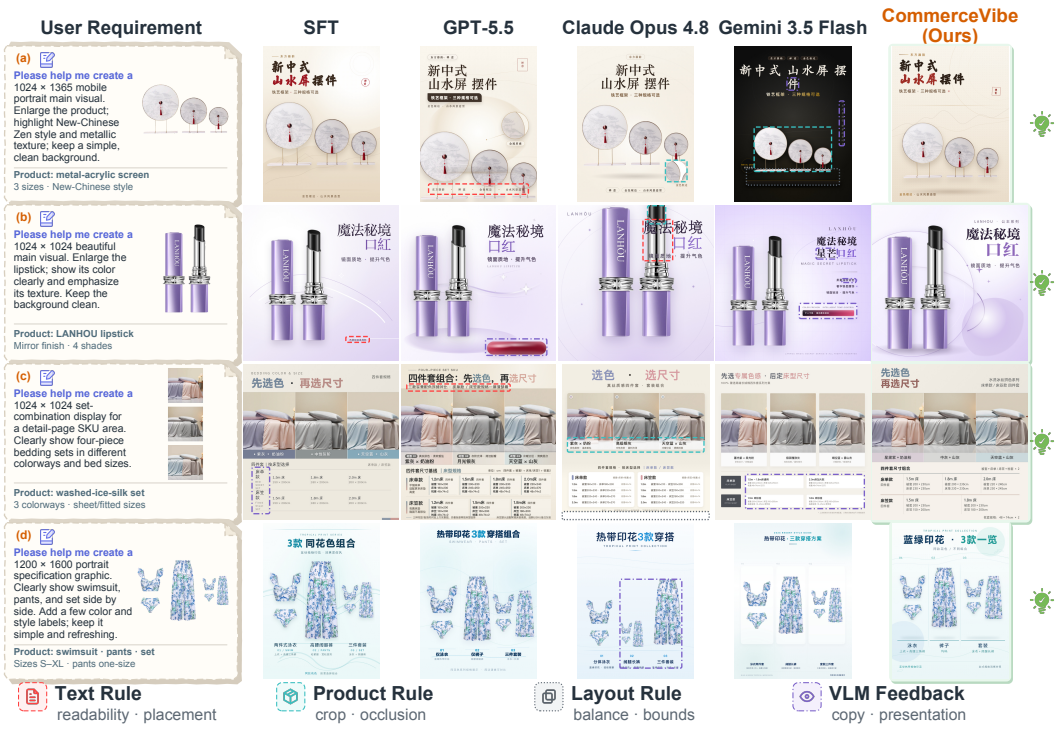

Figure 5: Qualitative comparison on two Single-image and two Multi-image cases. Paper cards summarize user requirements. Colored and patterned outline overlays localize representative issues: red, cyan, gray, and violet denote Text, Product, Layout, and VLM feedback, respectively. The underlying model outputs are otherwise unmodified; green checks indicate that no localized issue was identified in the audited categories.

Variant	Text	ΔProd.	ΔLayout	Δ	ΔS_{pref}	$\Delta S_{\text{overall}}$
Rule-RLVR	–	–	–	–	–	
w/o Text rules	**−2.15**	+1.50	+1.45	−0.9	−1.6	
w/o Product rules	−0.12	**−0.50**	+0.38	0.0	−0.1	
w/o Layout rules	+0.49	+0.33	**−4.47**	−0.1	−0.2	

Table 2: Rule-family ablation on 1,300 cases. Values are paired changes from Rule-RLVR (0–100); bold marks the removed family.

Removing Product rules also slightly lowers the Text score, possibly because product prominence and placement affect the space available to copy.

5 Conclusion

In this paper, we present **CommerceVibe**, an e-commerce creative generation model that produces executable HTML/CSS documents, enabling direct editing and structural verification. To optimize **CommerceVibe**, we develop a dual-feedback reinforcement learning approach that combines rule-based feedback for Text, Product, and Layout with VLM-based preference feedback. On our 1,300-case benchmark, **CommerceVibe** outperforms the three evaluated external models under the same one-pass HTML/CSS generation protocol and generates creatives that satisfy design requirements while maintaining high visual quality. Future work will develop suitable comparison protocols for raster-generation systems and iterative refinement pipelines, evaluate the approach with additional policy backbones and broader Multi-image settings, and extend it toward closed-loop self-improvement through iterative generation, evaluation, and refinement.

REFERENCES

Anthropic. Introducing Claude Opus 4.8. https://www.anthropic.com/news/claude-opus-4-8, May 2026. Accessed: 2026-07-29.

Tingfeng Cao, Junsheng Kong, Xue Zhao, Wenqing Yao, Junwei Ding, Jinhui Zhu, and Jiandong Zhang. Product2IMG: Prompt-free e-commerce product background generation with diffusion model and self-improved LMM. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10774–10783. ACM, 2024. doi: 10.1145/3664647.3680753. URL https://doi.org/10.1145/3664647.3680753.

Hongyu Chen, Min Zhou, Jing Jiang, Jiale Chen, Yang Lu, Zihang Lin, Bo Xiao, Tiezheng Ge, and Bo Zheng. T-Stars-Poster: A framework for product-centric advertising image design. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 5583–5591. ACM, 2025a. doi: 10.1145/3746252.3761554. URL https://doi.org/10.1145/3746252.3761554.

Jin Chen, Tiezheng Ge, Gangwei Jiang, Zhiqiang Zhang, Defu Lian, and Kai Zheng. Efficient optimal selection for composited advertising creatives with tree structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3967–3975, 2021. doi: 10.1609/aaai.v35i5.16516. URL https://ojs.aaai.org/index.php/AAAI/article/view/16516.

SiXiang Chen, Jianyu Lai, Jialin Gao, Tian Ye, Haoyu Chen, Hengyu Shi, Shitong Shao, Yunlong Lin, Song Fei, Zhaohu Xing, Yeying Jin, Junfeng Luo, Xiaoming Wei, and Lei Zhu. PosterCraft: Rethinking high-quality aesthetic poster generation in a unified framework. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10010467.

Yamei Chen, Haoquan Zhang, Yangyi Huang, Zeju Qiu, Kaipeng Zhang, Yandong Wen, and Weiyang Liu. Symbolic graphics programming with large language models, 2025b. URL https://arxiv.org/abs/2509.05208.

Yiyun Chen and Weikai Yang. Refadgen: High-fidelity advertising image generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(4):3138–3146, 2026. doi: 10.1609/aaai.v40i4.37307. URL https://ojs.aaai.org/index.php/AAAI/article/view/37307.

Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic design with large multimodal model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(3):2473–2481, 2025. doi: 10.1609/aaai.v39i3.32249. URL https://ojs.aaai.org/index.php/AAAI/article/view/32249.

Zhenbang Du, Wei Feng, Haohan Wang, Yaoyu Li, Jingsen Wang, Jian Li, Zheng Zhang, Jingjing Lv, Xin Zhu, Junsheng Jin, Junjie Shen, Zhangang Lin, and Jingping Shao. Towards reliable advertising image generation using human feedback. In *Computer Vision – ECCV 2024*, volume 15078 of *Lecture Notes in Computer Science*, pp. 399–415. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72661-3_23. URL https://doi.org/10.1007/978-3-031-72661-3_23.

Jiahao Fan, Yuxin Qin, Wei Feng, Yanyin Chen, Yaoyu Li, Ao Ma, Yixiu Li, Li Zhuang, Haoyi Bian, Zheng Zhang, Jingjing Lv, Junjie Shen, and Ching Law. Autopp: Towards automated product poster generation and optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(5):3768–3776, 2026. doi: 10.1609/aaai.v40i5.37377. URL https://ojs.aaai.org/index.php/AAAI/article/view/37377.

Jialu Gao, Mithun Das Gupta, Qun Li, Raveena Kshatriya, Andrew D. Wilson, Keng-hao Chang, and Balasaravanan Thoravi Kumaravel. Multi-object advertisement creative generation, 2026. URL https://arxiv.org/abs/2603.13745.

Yifan Gao, Zihang Lin, Chuanbin Liu, Min Zhou, Tiezheng Ge, Bo Zheng, and Hongtao Xie. PosterMaker: Towards high-quality product poster generation with accurate text rendering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8083–8093. IEEE, 2025. doi: 10.1109/CVPR52734.2025.00757. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gao_PosterMaker_Towards_High-Quality_Product_Poster_Generation_with_Accurate_Text_Rendering_CVPR_2025_paper.html.

Google DeepMind. Gemini 3.5 Flash: Model card. https://deepmind.google/models/model-cards/gemini-3-5-flash/, May 2026. Accessed: 2026-07-29.

Junrong Guo, Shancheng Fang, Yadong Qu, and Hongtao Xie. Seeing is improving: Visual feedback for iterative text layout refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25893–25903, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Guo_Seeing_is_Improving_Visual_Feedback_for_Iterative_Text_Layout_Refinement_CVPR_2026_paper.html.

Yueh-Ning Ku, Mikhail Kuznetsov, Shaunak Mishra, and Paloma de Juan. Staging e-commerce products for online advertising using retrieval assisted image generation. In *Proceedings of the Workshop on Data Mining for Online Advertising (AdKDD 2023)*, 2023. URL https://ceur-ws.org/Vol-3556/adkdd23-ku-staging-ceur-paper.pdf.

Jianyu Lai, Sixiang Chen, Jialin Gao, Hengyu Shi, Zhongying Liu, Fuxiang Zhai, Junfeng Luo, Xiaoming Wei, Lujia Wang, and Lei Zhu. PosterReward: Unlocking accurate evaluation for high-quality graphic design generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7762–7772, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lai_PosterReward_Unlocking_Accurate_Evaluation_for_High-Quality_Graphic_Design_Generation_CVPR_2026_paper.html.

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

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023a. URL https://openaccess.thecvf.com/content/CVPR2023/html/Li_GLIGEN_Open-Set_Grounded_Text-to-Image_Generation_CVPR_2023_paper.html.

Zhaochen Li, Fengheng Li, Wei Feng, Honghe Zhu, Yaoyu Li, Zheng Zhang, Jingjing Lv, Junjie Shen, Zhangang Lin, Jingping Shao, and Zhenglu Yang. Planning and rendering: Towards product poster generation with diffusion models, 2023b. URL https://arxiv.org/abs/2312.08822.

Jinpeng Lin, Min Zhou, Ye Ma, Yifan Gao, Chenxi Fei, Yangjian Chen, Zhang Yu, and Tiezheng Ge. Autoposter: A highly automatic and content-aware design system for advertising poster generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1250–1260, 2023. doi: 10.1145/3581783.3611930. URL https://arxiv.org/abs/2308.01095.

Junle Liu, Peirong Zhang, Yuyi Zhang, Pengyu Yan, Hui Zhou, Xinyue Zhou, Fengjun Guo, and Lianwen Jin. PosterVerse: A full-workflow framework for commercial-grade poster generation with HTML-based scalable typography. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(9):7197–7205, 2026a. doi: 10.1609/aaai.v40i9.37656. URL https://ojs.aaai.org/index.php/AAAI/article/view/37656.

Zhihong Liu, Siqi Kou, Zheng Li, Ye Ma, Quan Chen, Peng Jiang, Kai Yu, and Zhijie Deng. ProductWebGen: Benchmarking multimodal product webpage generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 9462–9473. ACM, 2026b. doi: 10.1145/3770855.3817507. URL https://doi.org/10.1145/3770855.3817507.

Ziyu Liu, Shengyuan Ding, Xinyu Fang, Xuanlang Dai, Penghui Yang, Jianze Liang, Jiaqi Wang, Kai Chen, Dahua Lin, and Yuhang Zang. Visual-ERM: Reward modeling for visual equivalence, 2026c. URL https://arxiv.org/abs/2603.13224.

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

OpenAI. GPT-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, April 2026. Accessed: 2026-07-29.

Yiming Pan, Chengwei Hu, Xuancheng Huang, Can Huang, Mingming Zhao, Yuean Bi, Xiaohan Zhang, Aohan Zeng, and Linmei Hu. AeSlides: Incentivizing aesthetic layout in LLM-based slide generation via verifiable rewards, 2026. URL https://arxiv.org/abs/2604.22840.

Sohan Patnaik, Rishabh Jain, Balaji Krishnamurthy, and Mausoom Sarkar. AesthetiQ: Enhancing graphic layout design via aesthetic-aware preference alignment of large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23701–23711, 2025. doi: 10.1109/CVPR52734.2025.02207. URL https://openaccess.thecvf.com/content/CVPR2025/html/Patnaik_AesthetiQ_Enhancing_Graphic_Layout_Design_via_Aesthetic-Aware_Preference_Alignment_of_CVPR_2025_paper.html.

Yuxin Qin, Ke Cao, Haowei Liu, Ao Ma, Fengheng Li, Honghe Zhu, Zheng Zhang, Run Ling, Wei Feng, Xuanhua He, Zhanjie Zhang, Zhen Guo, Haoyi Bian, Jingjing Lv, Junjie Shen, and Ching Law. InnoAds-Composer: Efficient condition composition for e-commerce poster generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32988–32999, 2026. URL https://arxiv.org/abs/2603.05898.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026. Accessed: 2026-07-29.

Juan Rodriguez, Haotian Zhang, Abhay Puri, Rishav Pramanik, Aarash Feizi, Pascal Wichmann, Arnab Mondal, Mohammad R. Samsami, Rabiul Awal, Perouz Taslakian, Spandana Gella, Sai Rajeswar Mudumba, David Vazquez, Chris Pal, and Marco Pedersoli. Rendering-aware reinforcement learning for vector graphics generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60496–60534. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/57126328c3b40cf618a34f1c5df24d8a-Abstract-Conference.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. LayoutNUWA: Revealing the hidden layout expertise of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a6a1e4c756d700d9aedcc1896a7e6fb0-Abstract-Conference.html.

Haohan Wang, Wei Feng, Yaoyu Li, Zheng Zhang, Jingjing Lv, Junjie Shen, Zhangang Lin, and Jingping Shao. Generate e-commerce product background by integrating category commonality and personalized style. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025a. doi: 10.1109/ICASSP49660.2025.10888013. URL https://doi.org/10.1109/ICASSP49660.2025.10888013.

Heng Wang, Yotaro Shimose, and Shingo Takamatsu. Banneragency: Advertising banner design with multimodal llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4304–4329. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.emnlp-main.214. URL https://aclanthology.org/2025.emnlp-main.214/.

Ximing Xing, Ziteng Xue, Yandong Guan, Jing Zhang, Dong Xu, and Qian Yu. Reason-SVG: Enhancing structured reasoning for vector graphics generation with reinforcement learning, 2025. URL https://arxiv.org/abs/2505.24499.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Yexing Xu, Wei Feng, Shen Zhang, Haohan Wang, Yuxin Qin, Yaoyu Li, Ao Ma, Yuhao Luo, Lu Wang, Xudong Ren, Haoran Wang, Run Ling, Zheng Zhang, Jingjing Lv, Junjie Shen, Ching Law, Longguang Wang, and Yulan Guo. Design your ad: Personalized advertising image and text generation with unified autoregressive models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 472–483, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xu_Design_Your_Ad_Personalized_Advertising_Image_and_Text_Generation_with_CVPR_2026_paper.html.

Hao Yang, Jianxin Yuan, Shuai Yang, Linhe Xu, Shuo Yuan, and Yifan Zeng. A new creative generation pipeline for click-through rate with stable diffusion model. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 180–189. ACM, May 2024. doi: 10.1145/3589335.3648315. URL https://doi.org/10.1145/3589335.3648315.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_Adding_Conditional_Control_to_Text-to-Image_Diffusion_Models_ICCV_2023_paper.html.

A SUPPLEMENTARY MATERIAL

This appendix describes the data construction, rule-based checks and rewards, training implementation, and expanded evaluation of **CommerceVibe**. Each complete HTML/CSS candidate is rendered once to obtain visible text, image elements, geometry, and the final screenshot for reward computation. These rendered outputs are not returned to the policy.

B SFT DATA CONSTRUCTION

The SFT corpus consists of product images, associated product information, and user design requests collected from real-world e-commerce design scenarios. Each raw example therefore comprises the user request, available product images, structured product information when available, and a reference HTML/CSS target. A fixed system instruction constrains the model role and the structure of the generated response.

Request normalization. Free-form user requests vary substantially in organization and level of detail. Before HTML/CSS generation, a fixed rewriting procedure extracts the supplied requirements into a structured representation. The procedure introduces no additional product facts, remains fixed throughout SFT, RL, and evaluation, and is shared by all **CommerceVibe** variants. The schematic pseudocode below summarizes the normalization structure without reproducing a complete policy input.

Quality control removes failed or incomplete examples. The data loader rejects embedded base64 payloads and short single-image wrappers, removes repeated image URLs within an example, and retains only input–target pairs.

The training split contains 28,568 quality-controlled examples: 23,726 with Single-image inputs and 4,842 with Multi-image inputs. A separate validation split contains 3,174 examples, yielding an approximately 9:1 train-to-validation ratio. Figure S1 reports both the Single/Multi composition of the training split and the cardinality distribution within its Multi subset.

B.1 GENERATOR SYSTEM INSTRUCTION

The following English translation preserves the content of the fixed system instruction used to define the generator role and constrain the response:

> **Role.** You are an e-commerce visual-design HTML generator.
>
> **Input.** A product design specification containing product images, design requirements, a layout plan, and related information.
>
> **Output.** One complete HTML file that can be rendered directly.
>
> **Requirements.** Output only HTML code, without explanations or Markdown wrapping. Do not output reasoning or emit `<think>` or `</think>` tags. Reference the product-image URLs supplied in the input through `<img>` elements. Strictly follow all dimensions, layouts, colors, and copy requirements in the design specification. Present every listed selling point, text item, and information module, including size charts, without omission or reduction.

C TASK AND BENCHMARK

For product images I, free-form request u, and optional structured product information m, request normalization gives $\tilde{u} = \mathrm{Normalize}(I, u, m)$. The policy context is $x = (s, I, \tilde{u}, m)$, where s is the fixed generator system instruction above. A shared multimodal policy generates one complete HTML/CSS document $c \sim \pi_\theta(c \mid x)$. We define Single by $|I| = 1$ and Multi by $|I| \geq 2$. This definition uses the number of input product images, not the number of image elements produced by the candidate.

The benchmark is constructed independently using product images, associated product information, and user design requests collected from real-world e-commerce scenarios. It contains 1,300 cases:

Input: product images $\mathcal{I}$, free-form request u, and optional product information m.
`PRIMARY_IDEA` ← summarize the requested visual concept, comparison goal, and reading order.
`SOURCE_ANCHORS` ← extract only visual and product evidence grounded in $(\mathcal{I}, m)$.
`CORE_PROOF` ← bind each required product, variant, and claim to its supporting source evidence.
`COMPOSITION_AND_TITLE` ← specify layout order, title placement, product-safe regions, background, and blocking constraints.
`DESIGN_PAYOFF` ← define `PRIMARY_TREATMENT`, `ELEMENT_SYSTEM`, `VISIBLE_GAIN`, and `ORNAMENT`.
`TYPOGRAPHY` ← specify font family and weight, hierarchy, colors, background treatment, and small-screen readability.
`RISK_CONSTRAINTS` ← record design risks, source-evidence strength, and forbidden changes.
`PROTECTED_RELATIONS`, `SUCCESS_CONDITION`, and `PRIMARY_RELATION` ← preserve required relations and define acceptance conditions.
`FALLBACK_LAYOUT` ← define a simpler composition used only when the primary layout cannot preserve the required relations.
Output: $\tilde{u} = \texttt{Phase2Plan}(\texttt{PRIMARY_IDEA}, \ldots, \texttt{FALLBACK_LAYOUT})$, with no unsupported product facts.

Schematic pseudocode for request normalization. Each field preserves supplied requirements and source-grounded relations while making the generation constraints explicit.

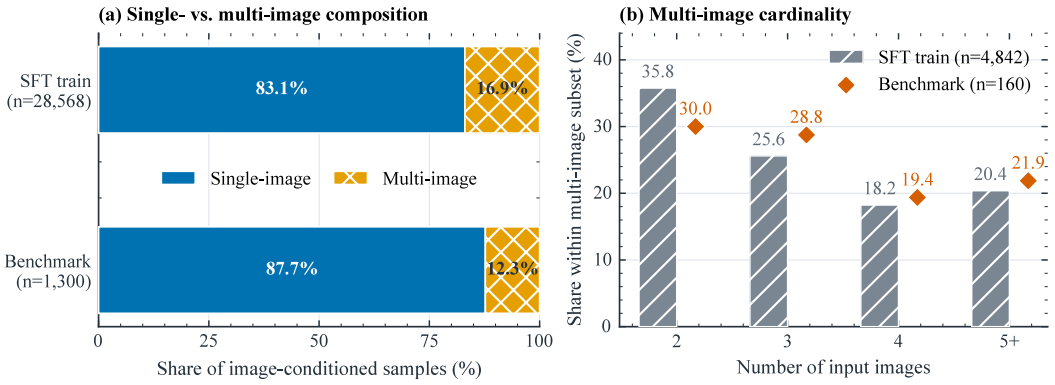

Figure S1: Training and evaluation distributions by input product-image count.

1,140 Single cases (87.69%) and 160 Multi cases (12.31%). At the product level, the benchmark is disjoint from all SFT and RL data, and products do not repeat across benchmark cases. Within the Multi stratum, 48, 46, 31, 16, 8, 6, and 5 cases contain 2, 3, 4, 5, 6, 7, and 8 input images, respectively. Results are reported both overall and by stratum. Rule settings are selected solely by the number of input product images.

D Partially Conditional Rule Verifier

The verifier organizes continuous measurements into Text, Product, and Layout families, but it does not apply one fixed vector of rules to every example. The selected profile determines which dimensions are present, their weights, critical dimensions, font thresholds, prominence semantics, and selected detector settings. The rendered state then determines whether image-dependent measurements are applicable. Table S2 separates these continuous families from the safeguards that precede reward aggregation.

Category	Cases	Category	Cases
Shoes	69	Underwear	67
Storage and cleaning supplies	58	Home improvement and building materials	42
Household essentials	52	Fashion accessories and jewelry	56
Women's clothing	54	Pet and gardening products	51
Sports and outdoors	50	Kitchenware and tableware	47
Home appliances	47	Children's clothing	48
Home textiles and decor	47	Consumer electronics and computers	44
Bags and leather goods	44	Food and beverages	43
Men's clothing	42	Automotive supplies	41
Beauty and cosmetics	41	Toys	40
Office and stationery	40	Personal care and household cleaning	37
Packaging	34	Agriculture	35
Lighting	34	Media and broadcasting	31
Fresh food and catering	27	Safety and protective equipment	25
Electrical equipment	21	Automotive and motorcycle parts	15
Instruments and meters	12	Adult products	3
Textiles and leather	2	Hardware and tools	1

Table S1: Category distribution of the 1,300-case benchmark. The 34 categories sum to 1,300 cases.

Component	Dimensions	Conditional behavior
Text	text overflow, text overlap, text on product, minimum font size, contrast	Weights and font thresholds differ between Single and Multi. Text-on-product is evaluated only when a visible `<img>` is available.
Product	subject prominence, image crop, image occlusion	Prominence uses the largest matched image for Single and summed matched-image area for Multi. Crop uses alpha-aware subject pixels when available. Occlusion is applicable only with at least two eligible rendered images.
Layout	element bounds, visual balance, bottom blank	Bounds and balance settings differ between Single and Multi. Multi uses ink-level text mass and tighter balance thresholds, whereas Single uses text boxes and excludes image backing panels. Bottom blank is activated and weighted only in rendered states for which it applies.
Safeguards	HTML validity, empty body, zero visible images, URL coverage, sparse text, critical violations	These checks act before or after continuous aggregation. They provide fixed negative rewards or additive penalties and are not members of the Text, Product, or Layout ablations.

Table S2: Conditional rule families and safeguards. Inapplicable image-dependent detectors are removed from aggregation, and the remaining applicable weights are renormalized within the selected profile; shared safeguards are excluded from the three rule-family ablations.

D.1 Profile Routing

The reward path routes a case only by the number of input product images:

$$p = \rho(|I|) = \begin{cases} \textsc{Single}, & |I| = 1, \\ \textsc{Multi}, & |I| \geq 2. \end{cases} \tag{S1}$$

The generation request and product information remain conditioning inputs to the policy and the Judge, but neither selects the rule setting. Table S3 gives the two weight vectors selected by this cardinality branch.

For Single, an applicable occlusion detector receives weight 3 while preserving the profile total. The implementation transfers up to two points from prominence, followed by one point from contrast, font size, or text-on-product as needed. Multi declares weight 4 directly. Bottom blank follows the same state-conditioned aggregation: it receives a weight only when applicable, and otherwise

Profile	Ov	Ol	Tp	Bd	Pr	Cr	Fs	Ct	Io	Ba	Bb	Critical dimensions
SINGLE	9	9	6	9	9	2	2	2	–	2	–	Ov, Ol, Bd, Io
MULTI	10	10	7	9	2	3	2	1	4	2	–	Ov, Ol, Bd, Io

Table S3: Single- and Multi-profile rule weights. Each row has nominal weight 50 before rendered-state adjustments. Abbreviations are: Ov, overflow; Ol, text overlap; Tp, text on product; Bd, element bounds; Pr, prominence; Cr, crop; Fs, font size; Ct, contrast; Io, image occlusion; Ba, visual balance; Bb, bottom blank. A dash denotes no nominal profile weight; detectors such as Single-image occlusion and bottom blank receive state-conditioned weights only when applicable.

Profile	Title	Subtitle	Normal	Fallback	Prominence	W/H	Other settings
SINGLE	56/.060/30	28/.025/15	22/.020/8	16/.014/20	.20 (max)	.78	box balance; strict bounds
MULTI	48/.050/30	28/.025/15	20/.018/8	14/.012/20	.20 (sum)	.70	ink balance; clipped exclusion

Table S4: Profile-specific font and prominence thresholds. Prominence mode `max` uses the largest matched input image, while `sum` uses the summed area of matched input images.

contributes neither a score nor a weight. Thus, the profile depends only on the input count, whereas detector applicability depends on the rendered candidate.

D.2 THRESHOLDS AND DETECTOR SEMANTICS

Table S4 reports the profile-dependent font and prominence settings. A font entry has the form absolute pixels / canvas-short-side ratio / deduction. A text element is penalized when it falls below both size thresholds. Prominence uses a smooth score below the area threshold; an image whose width or height reaches the listed canvas ratio receives a score of at least 80.

Image crop receives full credit up to 30% crop and zero at 50%, with smooth interpolation in between. For transparent images, the detector estimates cropped non-transparent subject pixels; otherwise it uses rendered-box geometry. Image occlusion considers visible, loaded, nontrivial `<img>` elements, excludes full-bleed scene images, and becomes applicable only when at least two eligible images remain. Its safe and zero-score overlap ratios are 0.02 and 0.18, with a minimum overlap area of 80 pixels. When alpha masks can be read, overlap is measured on non-transparent pixels; otherwise, the implementation falls back to box geometry.

Visual balance is always computed, but its representation depends on the profile. Multi uses ink-level text geometry, text weight 0.70, and smooth thresholds 0.12–0.30. Single uses text boxes, text weight 1.0, thresholds 0.15–0.45, and removes decorative backing panels that largely coincide with product images. When applicable, bottom blank is computed from rendered geometry, with full credit below 5% blank area and zero at 15%. It enters the active weight vector with raw weight 1 and is then included in the profile-total renormalization. Bottom blank is not critical, so a failure loses only its weighted contribution and incurs no additive critical penalty.

The visible-image filter counts unique `<img>` URLs only when the element is not hidden, has opacity at least 0.05, has rendered dimensions of at least 40×40 pixels, and intersects the canvas. CSS `background-image` resources do not count. Input coverage matches input URLs to visible rendered `<img>` URLs. When an image-dependent detector is inapplicable, its dimension is removed from the aggregation; it contributes neither a score nor a weight. The weights of the remaining applicable dimensions are renormalized to the continuous total of the selected Single or Multi profile after any rendered-state adjustment.

E REWARD CONSTRUCTION AND GATES

Let p be the routed profile, $A_{p,s}$ the dimensions applicable to profile p in rendered state s, $q_d \in [0, 100]$ the score of dimension d, and $w_d(p, s)$ its profile/state-specific weight after renormalization

over $A_{p,s}$. The nominal continuous score is

$$B_{\text{rule}} = \sum_{d \in A_{p,s}} \frac{w_d(p,s) q_d}{100}. \tag{S2}$$

Equation S2 gives the continuous rule score before additive penalties. For a rule-family ablation, let $D_{p,s}$ be all dimensions present before ablation and $A_{p,s} \subseteq D_{p,s}$ the remaining enabled dimensions. The implementation preserves the original total weight by applying

$$\alpha = \frac{\sum_{d \in D_{p,s}} w_d}{\sum_{d \in A_{p,s}} w_d}, \qquad B^{\text{abl}}_{\text{rule}} = \sum_{d \in A_{p,s}} \frac{\alpha w_d q_d}{100}. \tag{S3}$$

Dimensions disabled through `RLVR_DISABLED_DIMS` or `RLVR_DISABLED_DIM_GROUPS` are excluded from both the weighted score and critical-violation penalties.

Critical-violation penalty. A critical dimension is penalized twice. Its score q_d is first set to zero before weighted aggregation, removing its own weighted contribution. An additional penalty is then computed from its violation count $c_d \geq 1$:

$$P_d = b_d + e_d(c_d - 1). \tag{S4}$$

The configured base penalty b_d and per-extra penalty e_d are:

Dimension	b_d	e_d
`text_overflow`	3	3
`text_overlap`	3	3
`element_bounds`	3	3
`subject_prominence`	3	0
Other/default	3	3

`forbidden_price_info` uses the default $(3, 3)$ setting. Thus, dimensions with $(b_d, e_d) = (3, 3)$ incur penalties of 3, 6, and 9 for one, two, and three violations, whereas the subject-prominence penalty remains 3.

Image occlusion uses a separate nonlinear penalty. For overlap ratio r, let

$$v = \text{clip}\left(\frac{r - 0.02}{0.18 - 0.02}, 0, 1\right). \tag{S5}$$

With h violating image pairs, its penalty is

$$P_{\text{occlusion}} = 6 + 20v + 4\max(0, h - 1). \tag{S6}$$

Each critical dimension is included once after de-duplication, and penalties from different critical dimensions are added without decay or an upper cap:

$$P_{\text{critical}} = \sum_{d \in C_{p,s}} P_d, \tag{S7}$$

where the image-occlusion term uses Eq. S6. For example, two text-overflow violations contribute 6 points and one element-bounds violation contributes 3 points, for an additive penalty of 9, in addition to zeroing both dimension scores.

The penalized score passed to the reward mapping is

$$\begin{aligned} \widetilde{S}_{\text{rule}} &= B_{\text{rule}} - P_{\text{critical}} - P_{\text{sparse}}, \\ R_{\text{rule}} &= \text{clip}\left(\frac{\widetilde{S}_{\text{rule}} - 40}{10}, -1, 1\right). \end{aligned} \tag{S8}$$

The intermediate score $\widetilde{S}_{\text{rule}}$ is not clipped and may be negative; only the final reward mapping is clipped, reaching -1 when $\widetilde{S}_{\text{rule}} \leq 30$. Visible body text containing fewer than 10 characters

incurs $P_{\text{sparse}} = 25$ for both Single and Multi. This is a soft penalty: scoring continues rather than returning immediately.

Before continuous scoring, malformed outputs receive fixed rewards: HTML shorter than 50 characters receives -1.0, non-HTML text receives -0.8, and a candidate without a closing body or HTML tag receives -0.5. After rendering, an empty body or no visible `<img>` receives -1.0. Product-coverage gates depend on input cardinality: a Single candidate receives -1.0 when its sole input URL is missing, while a Multi candidate receives -1.0 when at least two distinct input URLs are missing.

F Preference Judge and Combined Reward

The fixed preference scorer used during training is Qwen3-VL-Plus. Under the summary-free protocol, the Judge receives the rendered screenshot, normalized request, structured product information, and input product images. It receives neither rule scores nor rule issues. The Judge is not updated during SFT or RL.

The Judge assigns an integer score $s_j \in \{1, \ldots, 5\}$ to visual appeal, product presentation, perceptual readability, marketing relevance, commercial usability, and copy faithfulness. The first five dimensions have weight 8 and copy faithfulness has weight 10. With $n(s) = 25(s-1)$, the preference score is

$$V = \sum_j \frac{w_j n(s_j)}{100} - \sum_{j:s_j \le 2} \lambda_j (5 - s_j), \tag{S9}$$

where $\lambda_j = 12$ for copy faithfulness and 8 otherwise. The mapped preference reward is

$$R_{\text{pref}} = \text{clip}\left(\frac{V - 25}{12.5}, -1, 1\right). \tag{S10}$$

After the shared validity and content gates, Rule-RLVR uses R_{rule}, Preference-RL uses R_{pref} from Eq. S10, and Rule+Preference RL uses

$$R_{\text{comb}} = \tfrac{1}{2} R_{\text{rule}} + \tfrac{1}{2} R_{\text{pref}}. \tag{S11}$$

F.1 VLM Judge Scoring Prompt

The following English rendering records the complete scoring instruction, input fields, and anchors supplied to the fixed Judge.

> You are a strict evaluator of e-commerce creatives. Evaluate the final rendered creative against the user request, product information, and reference product images. Judge only the supplied evidence. Return one integer score from 1 to 5 for each of the following six dimensions, together with a concise reason. Do not add dimensions or return an overall score.
>
> **Input fields.** User request: [USER REQUEST]. Product information: [PRODUCT INFORMATION]. Reference product images: [REFERENCE IMAGES]. Rendered creative: [SCREENSHOT].
>
> **1. Visual appeal (`visual_appeal`).** 5: polished, harmonious, and suitable as a high-quality commercial creative; 4: attractive with only minor generic or locally unrefined details; 3: acceptable but visually ordinary; 2: visibly rough, with color, texture, or decoration that harms the composition; 1: cheap-looking, chaotic, or commercially unusable.
>
> **2. Product presentation (`product_presentation`).** For Single inputs, the product should form a clear primary focus. For Multi inputs, the supplied set should be clearly covered, visually coordinated, and separable without requiring one product to dominate. 5: clear, complete, attractive, and naturally integrated; 4: clear and complete with minor integration or quality limitations; 3: visible but ordinary in scale, position, angle, or integration; 2: insufficiently prominent or incomplete for Single, or incompletely covered, poorly separated, or awkwardly coordinated for Multi; 1: severely cropped, missing, obscured, or otherwise unable to support product understanding.
>
> **3. Perceptual readability (`perceptual_`**

readability). 5: clear hierarchy and reading path, immediately readable core information, balanced composition, and orderly Multi-image arrangement; 4: mostly clear hierarchy with only minor effort required for secondary information; 3: understandable but with a weak focal hierarchy, marginal type size or contrast, or slightly uneven arrangement; 2: competing hierarchy, partially unreadable overlap or overflow, image occlusion, or visibly disordered arrangement; 1: failed composition or unreadable core information.

4. Marketing relevance (`marketing_relevance`). 5: style, copy, color, typography, and graphic language strongly match the product category, target audience, and intended scene; 4: coherent and relevant with minor weaknesses; 3: broadly relevant but generic or slightly collage-like; 2: noticeable mismatch or conflicting visual styles; 1: off-topic or visually incoherent for the product and intended use.

5. Commercial usability (`commercial_usability`). 5: directly suitable for a business candidate pool; 4: usable after minor revision; 3: usable as a draft but requiring substantial refinement; 2: low business usability because of multiple visible problems; 1: unusable.

6. Copy faithfulness (`copy_faithfulness`). 5: all visible titles, selling points, prices, sizes, and other copy agree with the supplied information and contain no visible corruption; 4: a minor noncritical difference that does not affect understanding; 3: a visible but noncritical inconsistency; 2: an incorrect title, price, size, or central selling point; 1: extensive incorrect, corrupted, or hallucinated copy.

Account for visible cropping, overlap, overflow, occlusion, or malformed text through the relevant perceptual dimension; do not create separate defect fields. Return strict JSON containing the six dimension keys, an integer `score` and one-sentence `reason` for each key, and an `issues` list for copy faithfulness. Return no Markdown or explanatory text outside the JSON object.

G Training and Evaluation Implementation

Table S6 summarizes the core settings for the Qwen3.5-9B policy. Beyond the entries in the table, SFT uses a cosine schedule with warmup ratio 0.1, weight decay 0.01, gradient clipping at 1.0, BF16, gradient checkpointing, and seed 42. SFT performs a full language-model update with LoRA disabled; LoRA with rank 8 is used for all RL variants. The vision encoder is frozen, and DeepSpeed ZeRO-2 with CPU optimizer offload uses `sft/ds_config_zero2_offload.json`. The maximum image-pixel budget is 100,000. During SFT, structured product information is independently dropped with probability 0.3 to model requests in which users do not provide product information.

We implement all GRPO runs with MS-SWIFT 4.3.0 through the `swift rlhf` entry point with `rlhf_type=grpo`. Each RL variant is trained in three independent runs with seeds 42, 25, and 999. All runs use a cosine schedule with warmup ratio 0.01, KL coefficient 0.001, gradient clipping at 1.0, BF16, gradient checkpointing, and DeepSpeed ZeRO-2. Rollout generation is accelerated with vLLM 0.17.1 in colocated mode with memory utilization 0.4. The maximum image-pixel budget is 100,000 and thinking is disabled. Reported automatic scores are averaged across the three independent runs using their fixed step-900 checkpoints; step 900 is selected before benchmark evaluation and independently of benchmark scores. Figures S2, S6, and S7 visualize the seed-42 runs.

For benchmark and standalone inference by the **CommerceVibe** policies, the system prompt is identical to the training system prompt and is used unchanged. Decoding uses no temperature and no top-p sampling, with a maximum of 8,192 output tokens. These inference settings are shared by the SFT and all RL variants; GRPO rollout sampling uses the separate training setting reported above.

Offline preference evaluation uses Qwen3.8-Max (`qwen3.8-max`) with temperature 0 and thinking disabled. This evaluator is separate from the Qwen3-VL-Plus Judge used to produce preference rewards during training. It receives the same input fields and applies the same six evaluation dimensions.

External-model inference. GPT-5.5, Claude Opus 4.8, and Gemini 3.5 Flash are evaluated on all 1,300 benchmark cases through their APIs between July 16, 2026 and July 17, 2026. All three models use the same fixed generator system instruction as the **CommerceVibe** policies and receive the same normalized request, product images, and product information. Their model-specific re-

Item	Setting
Models	GPT-5.5; Claude Opus 4.8; Gemini 3.5 Flash
Evaluation	1,300 cases; July 16–17, 2026 (UTC+8)
Input	Shared system instruction, normalized request, product images, and product information
Sampling payload	Temperature and top-p omitted; no reasoning/thinking parameter; no seed parameter
Output limit	8,192 tokens
Image encoding	Base64 data URLs; maximum 100,000 pixels per image
Invalid responses	At most two retries after the initial call; increasing delay capped at 30 seconds; no response feedback or reranking
Post-processing	Uniform removal of content through `</think>` when present

Table S5: Common API inference settings for the three external models.

Item	Setting
Policy backbone	Qwen3.5-9B multimodal model
Parameterization	SFT: full language-model update; RL: LoRA (rank 8); vision encoder frozen
SFT	28,568 training and 3,174 validation examples; 3 epochs; per-device batch 1×8 GPUs $\times$ grad. accum. $4 =$ effective batch 32; LR 1×10^{-5}; max length 16,384; max pixels 100,000
GRPO (all variants)	Three independent runs (seeds 42, 25, 999); LR 5×10^{-5}; temperature 0.7; effective batch of 64 completions (8 prompt groups with $K = 8$)
Terminal reward	Rule-RLVR: R_{rule}; Preference-RL: R_{pref}; Rule+Preference RL: $(R_{\text{rule}} + R_{\text{pref}})/2$
RL data and length	Identical normalized training contexts across variants; max completion 3,584; total max length 20,480
Renderer	Headless Chromium through Playwright
Training preference Judge	Fixed Qwen3-VL-Plus; temperature 0; summary-free protocol
Offline preference evaluator	Qwen3.8-Max; temperature 0; thinking disabled
Hardware	Eight NVIDIA H20 GPUs (96 GB each)

Table S6: Training and evaluation settings for SFT and the three GRPO variants.

quest profiles differ only in the model identifier and request header. The system instruction requires one complete, directly renderable HTML file without explanations, Markdown fences, or hidden reasoning.

Table S5 summarizes the common API settings. Temperature, top-p, reasoning or thinking controls, and a random seed are not included in the API payload; the APIs do not expose a seed through this evaluation interface. Each supplied image is converted to a base64 data URL with a maximum pixel count of 100,000. If a response is invalid, at most two retries are made, giving at most three attempts per case. Retries use a linearly increasing delay capped at 30 seconds and do not provide the preceding response or error message to the model. Thus, they handle invalid API outputs without iterative correction or candidate reranking. Although the system instruction explicitly prohibits reasoning traces and `<think>` tags, a uniform `clean_html()` step removes any content through a closing `</think>` tag if one is returned.

G.1 RL Training Dynamics

Figure S2 reports the seed-42 terminal-reward trajectories of the three main RL variants through training step 900, with each scalar objective shown on its native scale. The displayed trend uses a debiased exponential moving average with the W&B smoothing setting set to 1, corresponding to an internal weight of 0.999.

Figure S7 complements the reward curves with KL-divergence and clipping diagnostics from the same three seed-42 variant runs.

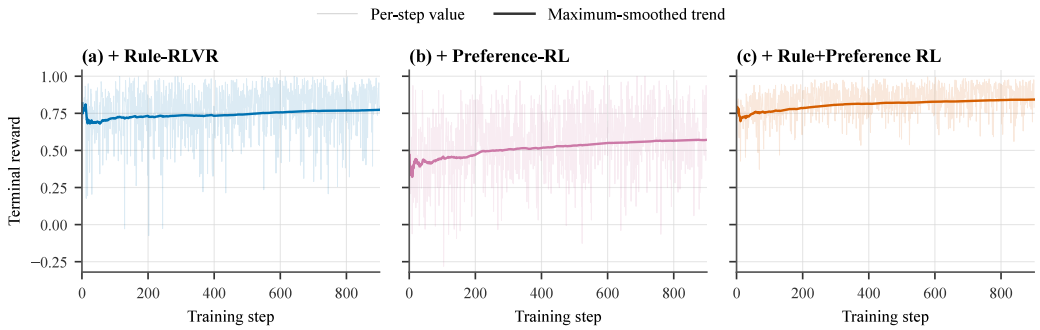

Figure S2: Seed-42 terminal rewards for the three RL variants through step 900. Light lines show per-step values; dark lines show debiased EMA trends using the maximum W&B smoothing setting.

	Single			Multi		
Method	Text	Product	Layout	Text	Product	Layout
Qwen3.5-9B	87.2	89.0	74.7	80.3	91.3	75.5
+ SFT	96.1	98.6	95.2	94.4	89.4	95.3
+ Rule-RLVR	98.5	98.9	97.4	94.3	91.1	91.6
+ Preference-RL	98.2	98.3	94.4	94.1	92.5	93.9
+ Rule+Preference RL	98.8	98.9	98.0	96.7	96.3	95.5

Table S7: Rule-family scores for 1,140 Single and 160 Multi cases. Each family is the equal-weight mean of its applicable dimensions; higher is better.

Automatic evaluation reuses the rendered-state component scores but reports a pure-weighted view. For the Rule branch, active profile weights are normalized to 50 and aggregated without the additive critical-violation or sparse-text penalties in Eq. S8; detector-produced zeros remain zero. For the VLM branch, the six weighted dimensions are summed without the low-score penalties in Eq. S9. The reported Text, Product, and Layout family scores are equal-weight means over applicable dimensions in the corresponding family, while S_{rule} uses the profile-weighted B_{rule}. Both retain profile-dependent dimension presence, thresholds, detector semantics, and applicability handling.

H Expanded Automatic Results

Table S7 provides the rule-family scores omitted from the compact main-paper table. These measurements use the pure-weighted evaluation view described above.

Figures S3 and S4 provide the fine-grained measurements underlying the aggregate rule and preference scores. The rule breakdown contains every detector dimension reported for the Single and Multi profiles; detector applicability varies with the input profile and rendered state, and the two strata are therefore reported separately. The VLM breakdown reports the unweighted mean of the original 1–5 ratings for the six preference dimensions.

H.1 Rule-Family Ablation Effects

Figure S5 visualizes the paired changes summarized in the main ablation table. Positive off-family changes reflect redistribution among jointly optimized dimensions.

Figure S6 reports the seed-42 training rewards of the three ablation policies; the paired held-out scores are reported in the main-paper ablation table.

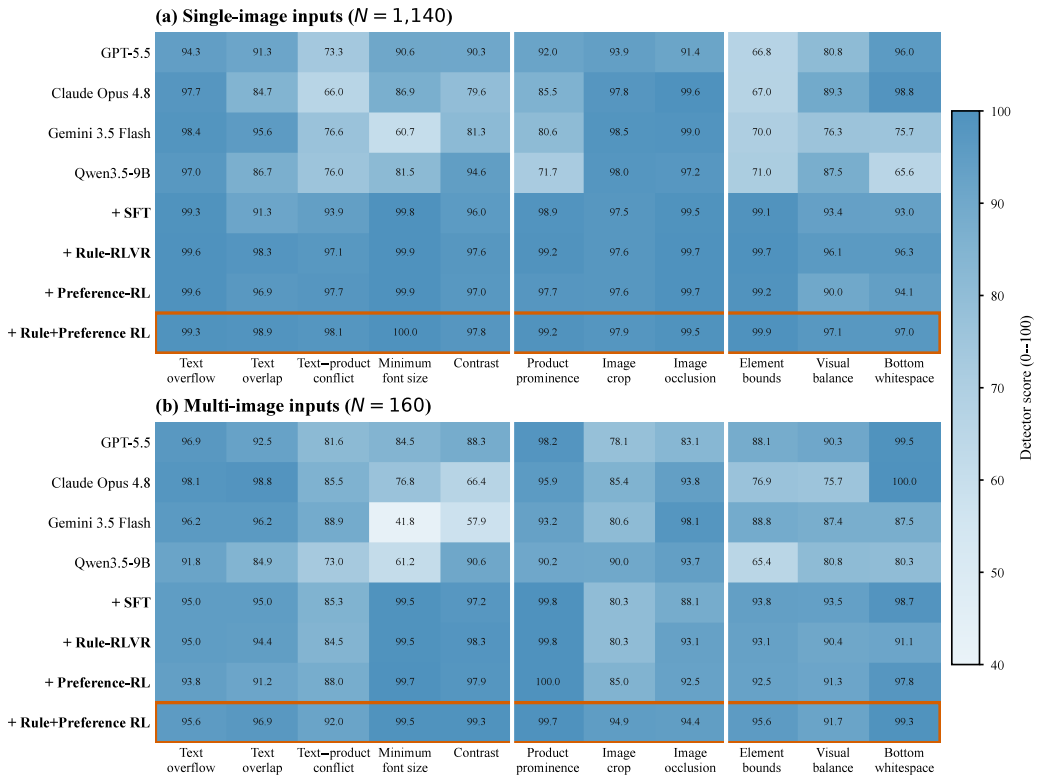

Figure S3: Detector-level rule scores for 1,140 Single and 160 Multi cases. Values are means on a 0–100 scale; the orange outline marks + Rule+Preference RL.

H.2 Additional RL Optimization Diagnostics

Figure S7 reports KL divergence and clipping-region ratio for the seed-42 runs of the three main GRPO variants. The small updates are consistent with the strong SFT initialization. Earlier tuning trials with larger learning rates or stronger KL settings produced more severe reward hacking and degraded instruction following, suggesting partial forgetting. Preference-RL has the lowest clipping ratio, whereas Rule+Preference RL has the largest, but still small, KL trend.

I Expert Evaluation

Evaluation protocol. We conduct an expert evaluation on the benchmark. For each case, one fixed render from each of the eight methods is used: the three RL variants use outputs from their seed-42 checkpoints, Qwen3.5-9B and the three external models use the second of their three inference outputs, and SFT uses its single inference output. These fixed choices are applied uniformly across all cases without candidate selection or reranking. Each render is assigned an anonymous identifier and presented in random order. Five experts, denoted A–E, evaluate the anonymized renders. All have one to five years of professional experience in e-commerce creative production. The evaluation interface provides the product images, normalized request, product information, and rendered candidate; method identities, verifier outputs, VLM Judge scores, and aggregate automatic scores are withheld.

Ratings and aggregation. Experts rate every applicable dimension on a five-point Likert scale. For the rule-aligned defect dimensions, the anchors are 1 (severe), 2 (major), 3 (moderate), 4 (minor), and 5 (no problem). These dimensions are text overflow, text overlap, text on product, minimum font size, text–background contrast, product prominence, product crop, inter-product occlusion, element bounds, visual balance, and bottom whitespace. For the preference dimensions, the

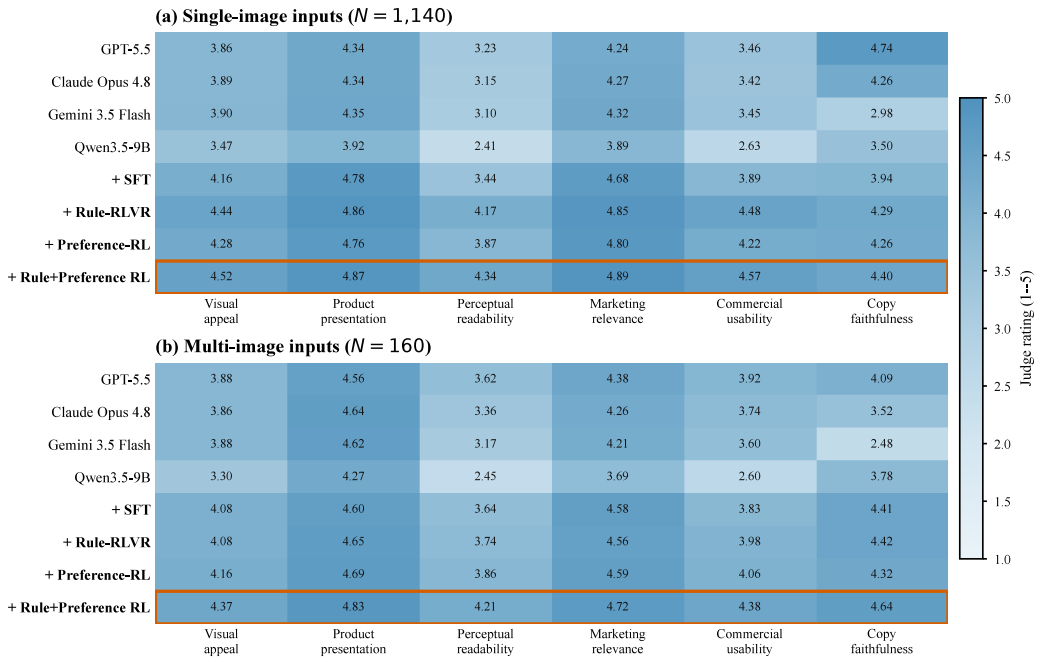

Figure S4: VLM preference dimensions for 1,140 Single and 160 Multi cases. Values are unweighted mean ratings on the original 1–5 scale; the orange outline marks + Rule+Preference RL.

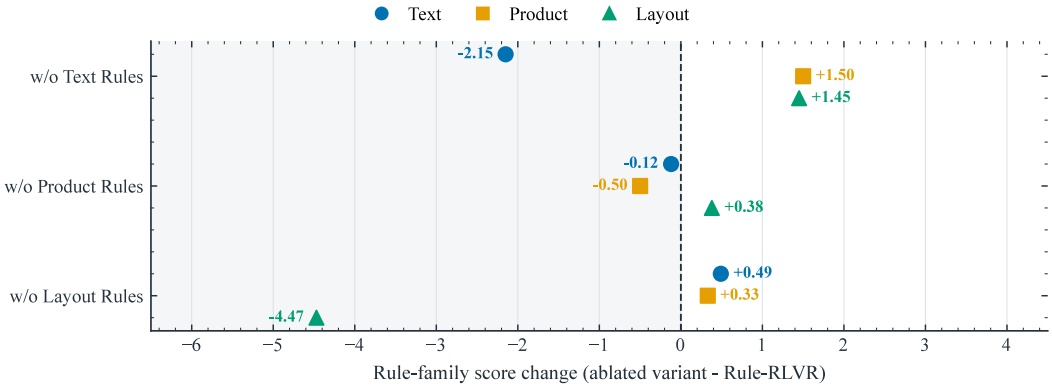

Figure S5: Paired changes in Text, Product, and Layout scores after removing each rule family. Values are means relative to Rule-RLVR on the same 1,300 cases; negative values indicate degradation.

anchors are 1 (very poor), 2 (poor), 3 (fair), 4 (good), and 5 (very good). The six dimensions are visual appeal, product presentation, perceptual readability, marketing relevance, commercial usability, and copy faithfulness. The interface presents a rule-aligned dimension only when it is active for the case; otherwise it is marked unavailable rather than assigned a low score.

Each raw rating $\ell \in \{1, \ldots, 5\}$ is mapped to $z = (\ell - 1)/4 \in [0, 1]$. For every expert–case–method tuple, rule-aligned ratings use the profile-conditioned active dimensions and weights of the automatic rule evaluator; unavailable dimensions are excluded and the remaining weights are renormalized. The six preference ratings use the fixed preference-dimension weights. The two mapped branch scores are equally weighted and rescaled to $S_{\text{expert}} \in [0, 100]$. Structural gate overrides and automatic low-score penalties are not included in this human composite.

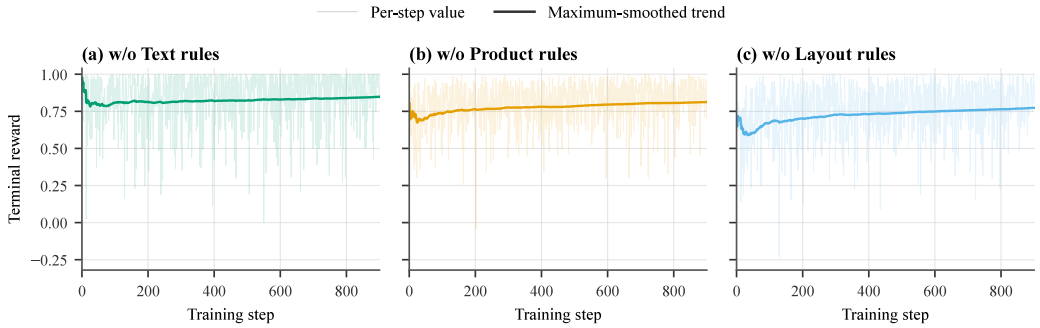

Figure S6: Seed-42 terminal rewards for the three rule-family ablations through step 900. Light lines show per-step values; dark lines show debiased EMA trends using the maximum W&B smoothing setting.

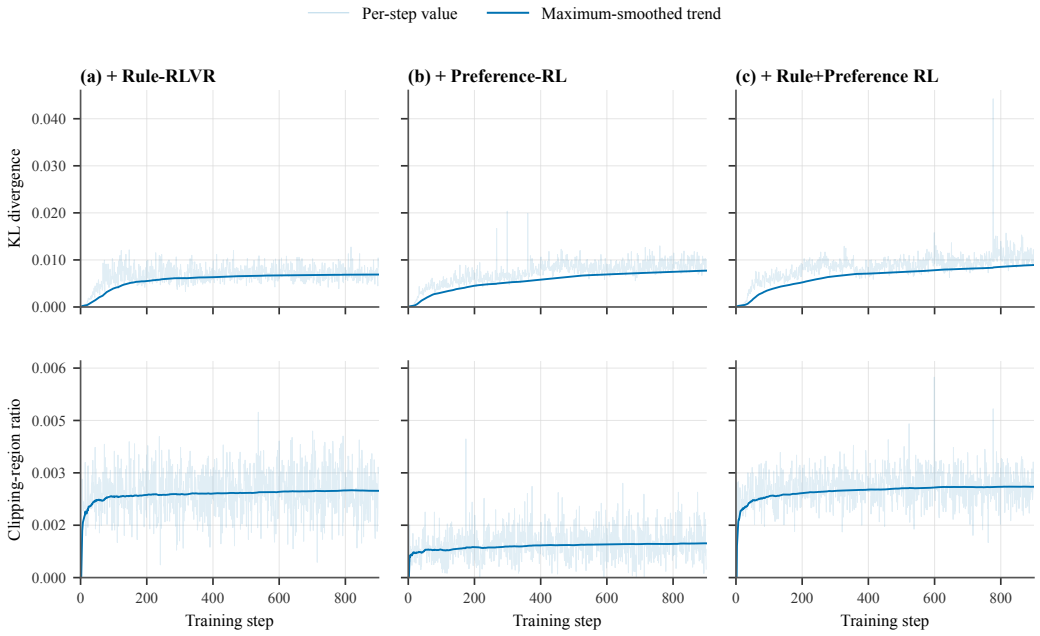

Figure S7: Seed-42 GRPO diagnostics through step 900. Rows show KL divergence and mean clipping-region ratio; light and dark lines denote per-step values and debiased EMA trends using the maximum W&B smoothing setting.

Statistical analysis. The case is the statistical unit for method means, confidence intervals, paired tests, and correlations. The five expert composites are first averaged within each case–method pair, followed by an unweighted mean over the 1,300 benchmark cases. Each 95% confidence interval is the percentile interval from 10,000 bootstrap replicates of the 1,300 case identifiers; all eight method scores associated with a resampled case are retained together. Inter-rater agreement is measured using a two-way random-effects, absolute-agreement, average-measures intraclass correlation. The resulting $\mathrm{ICC}(\mathrm{A}, 5)$ is 0.811.

Rule+Preference RL is compared with SFT, Rule-RLVR, Preference-RL, and GPT-5.5 using asymptotic two-sided paired Wilcoxon signed-rank tests. Tests of automatic and expert scores use the same 1,300 benchmark cases for each pair. Zero differences are omitted by the signed-rank statistic, and p-values are Holm-corrected across the four prespecified comparisons within each evaluation source. Automatic and expert scores are associated at the case level by Spearman correlation. A 10,000-replicate case-cluster bootstrap, which resamples case identifiers while retaining all eight methods,

Method	$S_{\text{expert}} \uparrow$	95% CI
External models		
GPT-5.5	78.9	[76.9, 80.9]
Claude Opus 4.8	78.4	[76.2, 80.5]
Gemini 3.5 Flash	75.9	[73.7, 78.0]
***CommerceVibe** variants*		
Qwen3.5-9B	59.0	[56.3, 61.7]
+ SFT	81.6	[79.9, 83.2]
+ Rule-RLVR	87.3	[85.9, 88.5]
+ Preference-RL	84.4	[82.8, 85.8]
+ Rule+Preference RL	**90.0**	**[88.5, 91.3]**

Table S8: Five-expert scores on the 1,300-case benchmark. S_{expert} is reported on a 0–100 scale with 95% case-bootstrap confidence intervals.

Comparator	$\Delta S_{\text{overall}}$	p_{Holm}	ΔS_{expert}	p_{Holm}
+ SFT	$+6.7$	1.3×10^{-91}	$+8.4$	2.4×10^{-14}
+ Rule-RLVR	$+1.6$	9.1×10^{-13}	$+2.8$	4.2×10^{-8}
+ Preference-RL	$+3.6$	3.1×10^{-39}	$+5.6$	2.1×10^{-13}
GPT-5.5	$+13.9$	9.0×10^{-165}	$+11.1$	1.5×10^{-12}

Table S9: Paired comparisons with Rule+Preference RL on the 1,300-case benchmark. Differences are Rule+Preference RL minus the comparator. Reported p-values are from asymptotic two-sided Wilcoxon signed-rank tests with Holm correction.

gives $\rho = 0.612$ with a 95% confidence interval of $[0.562, 0.657]$. The median method-specific correlation is 0.385 (range: 0.075–0.492).

Table S8 reports the method means and case-bootstrap confidence intervals. Figure S8 provides the corresponding visual summary.

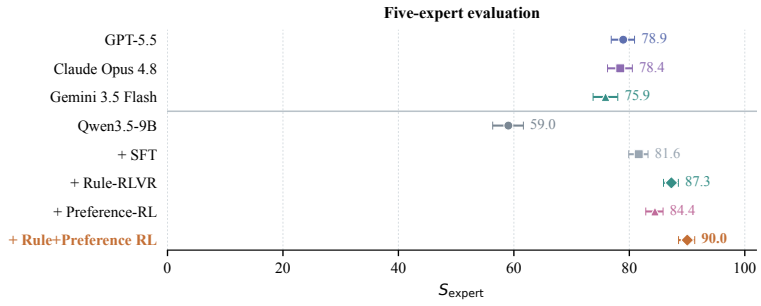

Figure S8: Five-expert mean scores and 95% case-bootstrap confidence intervals for all eight methods on the benchmark.