PosterDuet: Co-Evolving Design Generation and Reward Optimization for Product Poster Synthesis

Junlong Wu[1,*], **Pengcheng Wei**[2,*], **Jia Sun**[3,†], **Huaiqing Wang**[3], **Weixuan Zeng**[3], **Zijun Li**[3], **Honglie Wang**[3], **Yongrui Heng**[3], **Boheng Zhang**[3], **Dewen Fan**[3], **Fan Yang**[3], **Tingting Gao**[3], **Houde Liu**[1,†], **Qianqian Gan**[3]

[1] **Shenzhen International Graduate School,Tsinghua University**
[2] **School of Software, Tsinghua University**
[3] **Kuaishou Technology**
[*] **Equal contribution.** [†] **Corresponding authors.**

Abstract

Automatic e-commerce poster generation requires jointly optimizing product fidelity, visual composition, text rendering, and commercial appeal, yet existing staged pipelines restrict mutual adaptation among these elements. We reformulate this task as *holistic product-aware poster editing*: given only a raw product image and structured metadata, the goal is to produce the final poster directly without segmentation masks, glyph images, or layout priors. We propose PosterDuet, a closed-loop framework coupling a VLM-based *Designer* that generates holistic design prompts with a unified *Renderer* that synthesizes the poster in a single editing pass. To enable both modules to co-evolve, we introduce *alternating dual-level optimization*: the Designer is refined via Target Split Relative Optimization (TSRO), which decomposes the reward into rotating semantic targets and compares against a frozen baseline; the Renderer is refined via DiffusionNFT under the same reward signal; and the two branches alternate when improvement saturates. Optimization is guided by two dedicated reward models, namely a CTR preference model that incorporates multi-granularity image–text encoding and adopts a two-stage bucket-then-ranking training strategy, and a generative quality reward model that provides per-dimension scores and natural-language critiques. Extensive experiments demonstrate state-of-the-art performance on automatic metrics, human preference evaluation, and online A/B testing with significant CTR improvement on a live commercial platform.

1 Introduction

Automatic e-commerce poster generation requires jointly optimizing product fidelity, visual composition, text rendering, and commercial appeal. On major platforms, each product may need dozens of promotional creatives tailored to different channels and campaigns; manual design at this scale is prohibitively expensive, making automated generation an urgent practical need. Existing methods decompose this task into staged sub-problems, including product segmentation, layout prediction, glyph rasterization, and conditional synthesis Wang et al. [2025a], Chen et al. [2025a], Gao et al. [2025], Fan et al. [2025], but this decomposition introduces three fundamental limitations (Figure 1, bottom): (i) segmentation assumes a clean product–background separation, which fails for hand-held, worn, or context-entangled products where what to retain is a *design* choice; (ii) text content and typography must be fully specified before generation. For example, PosterMaker Gao et al. [2025] requires pre-rendered glyph images as conditional input, and AutoPP Fan et al. [2025] depends on externally supplied text images. These requirements treat promotional copy as a *mandatory, fixed* prior and prevent the system from autonomously crafting copy that adapts to the visual context;

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 1: **Top:** Example posters generated by PosterDuet. **Bottom:** Architectural comparison among PosterMaker Gao et al. [2025] , AutoPP Fan et al. [2025] , and our PosterDuet . The radar chart and feature table highlight that PosterDuet uniquely achieves CTR guidance, self-improvement, and alternating training while requiring only a product image and metadata.

and (iii) feed-forward pipelines fix layout in advance, precluding the reciprocal interaction between product appearance and scene composition that human designers naturally exploit.

We reformulate the task as *holistic product-aware poster editing*: given only a raw product image and structured metadata (selling points, category, etc.), the goal is to produce the final poster directly, without relying on segmentation masks, glyph images, or predefined text content. To realize this formulation, we propose PosterDuet , a closed-loop framework comprising a VLM-based *Designer* that automatically generates holistic design prompts from raw selling points, and a unified *Renderer* that synthesizes the poster conditioned on both the prompt and the raw image. A central challenge in optimizing this two-module pipeline is credit assignment: a poor poster may result from a suboptimal plan, a weak renderer, or both. Rather than optimizing a single module in isolation or training both jointly end-to-end, we introduce an *alternating dual-level optimization* mechanism: the Designer is refined via Target Split Relative Optimization (TSRO), which decomposes the reward into rotating semantic targets and compares against a frozen baseline to isolate per-component improvements; the Renderer is refined via DiffusionNFT Zheng et al. [2025] under the same reward signal. The two branches alternate when reward improvement saturates, enabling design and rendering to co-evolve without mutual interference.

A key challenge is defining reward signals that capture both aesthetic quality and commercial effectiveness. On the *commercial* side, we build a dedicated CTR preference model with a multi-granularity image–text architecture trained on real user-behavior pairs; empirically, we find generic VLMs perform near chance on pairwise CTR prediction, confirming the need for domain-specific supervision. On the *quality* side, we employ a generative reward model that provides per-dimension quality scores together with natural-language critiques, enabling the Designer to iteratively refine its prompts based on concrete feedback rather than opaque scalar signals. Importantly, we observe that CTR and quality are near-orthogonal, validating the necessity of separate reward models rather than a single conflated objective. The two rewards jointly drive both TSRO and DiffusionNFT. Additionally,

before entering the closed-loop stage, we warm up the Renderer with OCR-guided RL to ensure baseline text fidelity for the dense Chinese characters common in e-commerce posters.

Our contributions are as follows:

- We reformulate e-commerce poster generation as a holistic product-aware editing problem and propose a complete Designer–Renderer pipeline that produces advertising posters directly from raw product images.
- We introduce an alternating dual-level optimization framework in which the Designer is optimized with TSRO and the Renderer with DiffusionNFT, enabling the two modules to co-evolve under both CTR-oriented and visual-quality rewards.
- We design two complementary reward models to guide poster optimization: a CTR preference model that achieves strong pairwise accuracy on poster comparisons, and a generative quality reward model that assesses posters along multiple visual and semantic dimensions while providing critiques for closed-loop prompt refinement.

2 Related Work

2.1 E-commerce Poster Generation

Early poster generation methods rely on predefined templates or layout representations to compose design elements Lin et al. [2023], Zhou et al. [2022], Li et al. [2023a], Hsu et al. [2023]. With the advance of diffusion-based generation Rombach et al. [2022], Podell et al. [2023], Black Forest Labs [2024] and controllable editing Zhang et al. [2023], Brooks et al. [2023], Ye et al. [2023], Wu et al. [2025], recent works decompose poster synthesis into staged sub-tasks: product background generation Cao et al. [2024], Wang et al. [2025a], layout planning Li et al. [2023b], Chen et al. [2025a], text rendering with glyph controls Gao et al. [2025], Anonymous [2026b,a], and full-pipeline automation Fan et al. [2025]. While effective in constrained settings, these pipelines fix each stage independently, preventing mutual adaptation. Our work removes this decomposition: we condition a unified image editor directly on the raw product image without segmentation masks, glyph images, or layout boxes.

2.2 Reward-driven Optimization for Generative Models

Learned reward models provide scalable supervision for generative models. Generic aesthetic rewards, including ImageReward Xu et al. [2023], PickScore Kirstain et al. [2023], HPS v2 Wu et al. [2023], and EditReward Wu et al. [2026], capture human preference for visual quality and text-image alignment. To render accurate text, particularly dense logographic glyphs, methods such as TextDiffuser Chen et al. [2023] and AnyText Tuo et al. [2024] introduce character-aware modules, while OCR-based evaluation Gao et al. [2025], Cui et al. [2025] serves as a post-hoc quality signal. For e-commerce, the objective further extends to click effectiveness. CG4CTR Yang et al. [2024a] and CAIG Chen et al. [2025b] optimize generated creatives toward CTR via offline surrogate models, DEPO Sun et al. [2025] applies short-trajectory preference optimization to background generation conditioned on CTR signals, and AutoPP Fan et al. [2025] leverages online A/B feedback. On the optimization side, policy-gradient methods such as DDPO Black et al. [2024] and DPOK Fan et al. [2023], preference alignment through Diffusion-DPO Wallace et al. [2024], reward backpropagation through AlignProp Prabhudesai et al. [2024], and online RL methods such as DiffusionNFT Zheng et al. [2025], Flow-GRPO Liu et al. [2025], and DanceGRPO Xue et al. [2025] demonstrate that diffusion models can be steered toward diverse downstream objectives. Our framework combines CTR reward with general-quality reward under a unified alternating scheme.

2.3 VLM-based Planning and Iterative Refinement

Recent works employ VLMs or LLMs as high-level planners for poster design: PnR Li et al. [2023b] and T-Stars Chen et al. [2025a] generate layout specifications, while PosterLLaVA Yang et al. [2024b] and BannerAgency Wang et al. [2025b] produce structured design instructions. However, these planners are decoupled from the downstream renderer. The plan is frozen before generation and cannot be revised based on generation outcomes. Our closed-loop framework bridges these two lines:

the VLM Designer receives natural-language critiques from a generative reward model conditioned on the rendered poster, enabling design and rendering to co-evolve under unified optimization.

3 Method

We propose PosterDuet, a closed-loop framework for e-commerce poster generation (Figure 2). Given a raw product image and structured metadata, it first uses a VLM-based Designer to generate a holistic design prompt, then feeds both the prompt and the original image into a unified Renderer to produce the final poster. An alternating dual-level reward mechanism jointly optimizes both modules, each receiving CTR and general-quality signals from a shared reward model.

3.1 Problem Formulation

We formulate e-commerce poster generation as *holistic product-aware poster editing*. Given a raw product image I_s and structured metadata $\mathbf{m}$, the goal is to produce a poster I_g directly:

$$I_g = \mathcal{G}_\phi\big(\pi_\theta(I_s, \mathbf{m}),\ I_s\big), \tag{1}$$

where π_θ is the Designer that generates a design prompt y, and $\mathcal{G}_\phi$ is the Renderer that synthesizes the poster conditioned on both y and I_s. Unlike prior formulations that require a pre-computed subject mask M_s, glyph image T, and background prompt P, our formulation takes only the raw image and metadata as input, delegating all image decisions to the learned pipeline itself.

3.2 Pipeline Composition

Product-aware Designer. The Designer π_θ is a vision-language model that takes the raw product image I_s and metadata $\mathbf{m}$ as input, and autoregressively generates a holistic design prompt y that jointly specifies background, layout, promotional copy, and typography in one coherent description. By unifying all design decisions into a single prompt, the Designer ensures cross-element consistency under a shared product-aware context. We first train π_θ with SFT on curated $(I_s, \mathbf{m}, y)$ triplets.

Unified Poster Renderer. The Renderer $\mathcal{G}_\phi$ synthesizes the final poster in a single forward pass conditioned on I_s and the prompt y. Two challenges arise in this domain: (i) preserving product identity under redesigned display, and (ii) faithfully rendering dense Chinese text. We address them through a two-stage training: *product-consistent SFT* that adapts shallow DiT blocks to bind the reference product to the target layout via flow-matching, followed by *OCR-guided RL* that refines text accuracy using DiffusionNFT with a text-fidelity reward(Appendix A).

3.3 Alternating Optimization with Dual Rewards

3.3.1 CTR Preference Model

We adopt a two-stage training strategy to build a reliable CTR reward model Chen et al. [2025b], Wang et al. [2023]. As shown in Fig. 3, the model uses independent encoders to jointly encode global and local visual features from the full image and 4×4 patches, totaling 17 visual units, together with multi-granularity text features from the title and three-level category, totaling 5 text units. These features are fused through cross-attention and mapped to a scalar score via an MLP head.

In Stage 1, we discretize CTR values into K buckets and train with standard K-way cross-entropy to build an initial representation of click attractiveness. In Stage 2, for same-product pairs (x_i, x_j) with CTR values $c_i > c_j$, the model is trained with a gap-weighted ranking loss combined with pointwise regression:

$$\mathcal{L}_{stage2} = \frac{1}{M} \sum_{(i,j)\in\mathcal{P}} w_{ij} \log\Big(1 + e^{-(s_i - s_j)}\Big) + \frac{\lambda}{N} \sum_{k=1}^{N} (s_k - c_k)^2, \tag{2}$$

where $w_{ij} = \min(1, |c_i - c_j|/\gamma)$ emphasizes pairs with larger CTR gaps, c_k is the observed CTR of sample k, and the regression term preserves absolute score calibration.

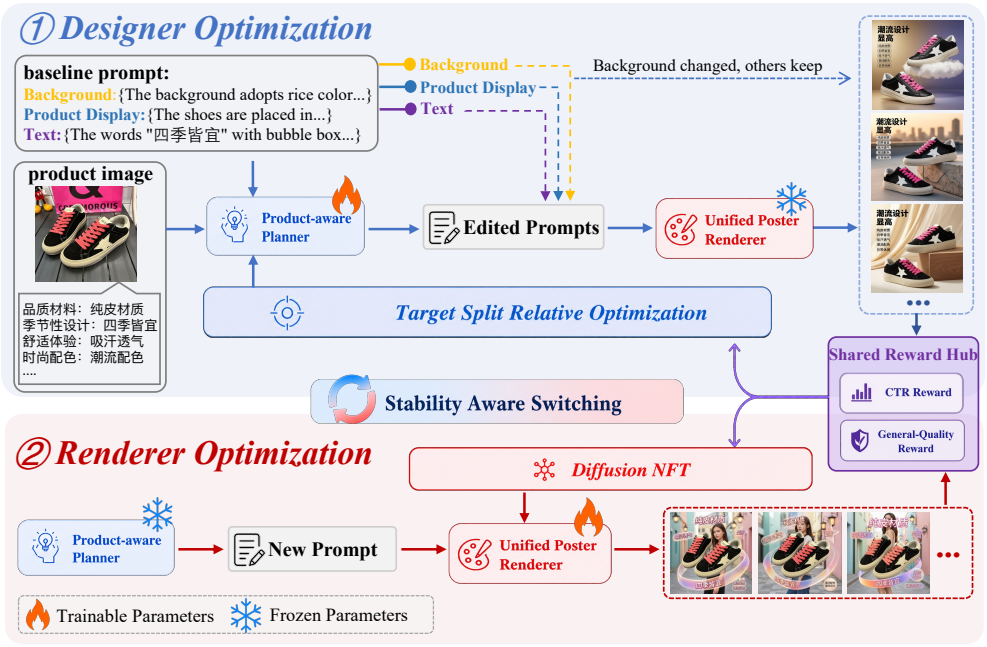


Figure 2: Overall framework of PosterDuet: the Designer and Renderer are alternately optimized under shared CTR and quality rewards.

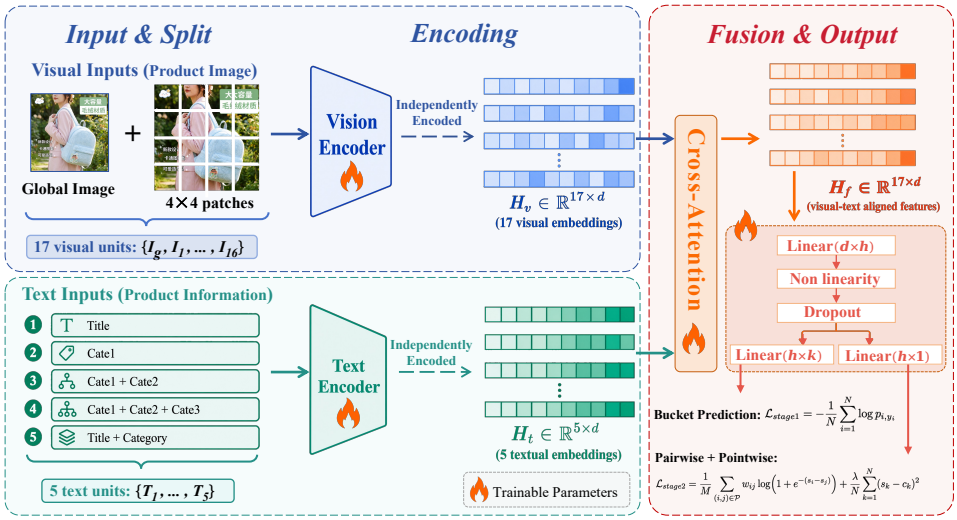


Figure 3: Architecture of the CTR preference model with multi-granularity image–text encoding and cross-attention fusion.

3.3.2 Alternating Dual-Level Optimization

Optimizing only the Designer suffers from reward ambiguity across semantic components and instability introduced by the frozen Renderer; optimizing only the Renderer is fundamentally upper-bounded by the fixed design capability. We therefore alternate between the two: at each training step only one branch is trainable, and we switch when the current branch's reward gain saturates (Algorithm 1 in Appendix A.4). Concretely, we maintain a moving-average reward $\bar{m}_t$ over a sliding window of size W. When the marginal gain $\Delta_t = \bar{m}_t - \bar{m}_{t-1} < \varepsilon$ persists for H consecutive evaluations, we switch the active branch and reset the counter. A robustness analysis in Appendix **??** shows that moderate perturbations of these values yield small variance in final metrics. The overall framework is illustrated in Fig. 2.

Since optimizing toward CTR alone risks reward hacking such as visually attractive but semantically incoherent posters Chen et al. [2025b], we combine CTR with a general-quality reward and further decompose the signal differently for the Designer and the Renderer.

Designer Optimization. A design prompt jointly controls $|\mathcal{T}|=3$ semantic dimensions (background, product display, text). Standard GRPO computes a single holistic reward for the entire output and normalizes across a candidate group; while effective for single-objective generation, it cannot disentangle which dimension caused a reward change. As a result, the per-dimension gradient signal scales as $\sim 1/|\mathcal{T}|$ and is dominated by cross-dimension noise, leading to sparse feedback and slow, unstable convergence.

We address this with **Target Split Relative Optimization (TSRO)**, whose core mechanism is *dimension-rotating reward decomposition*: at each optimization step, we sample one focal target $t \in \mathcal{T}$ and construct a reward that amplifies the contribution of dimension t while attenuating the others, transforming a sparse multi-objective problem into a sequence of dense single-objective sub-problems that rotate across steps. Concretely, the focused reward is:

$$R_{main}(I_{g,n}(p), t) = \lambda_{ctr} S_{ctr}(I_{g,n}(p)) + \lambda_{gen} \sum_{i \in \mathcal{T}} w_{t,i}\, S_i(I_{g,n}(p)), \tag{3}$$

where $w_{t,i}$ assigns dominant weight to the focal target t and $I_{g,n}(p)$ is the n-th poster rendered from prompt p. Over a full rotation through $\mathcal{T}$, every dimension receives dense, high-signal feedback in its focal step while still receiving non-zero gradient in others, ensuring that no dimension stagnates or regresses.

To further prevent non-focal dimensions from degrading during focused optimization, we introduce a *relative reward baseline*: we compare against the frozen SFT policy π_{θ_0} and penalize regressions below its performance:

$$R_{rel}^{(n)} = \eta_+ [\Delta_n - m_+]_+ - \eta_- [-\Delta_n - m_-]_+, \tag{4}$$

where $\Delta_n = R_{main}(I_{g,n}(p), t) - R_{main}(I_{g,n}(p_{base}), t)$ and asymmetric margins m_+, m_- ensure that degradation below the baseline is penalized more heavily than improvement is rewarded, acting as a safety constraint. The per-sample reward averages over N renderings: $R_i = \frac{1}{N} \sum_{n=1}^{N} (R_{main}^{(n)} + \lambda_{rel}\, R_{rel}^{(n)})$. We sample G candidate prompts per input, compute group-normalized advantages $\hat{A}_i = (R_i - \mu_G)/\sigma_G$, and optimize with a clipped surrogate and KL penalty:

$$\mathcal{L}_{TSRO}(\theta) = -\mathbb{E}\left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|p_i|} \sum_{\tau=1}^{|p_i|} \min\big(r_{i,\tau}\, \hat{A}_i,\ \mathrm{clip}(r_{i,\tau}, 1-\epsilon, 1+\epsilon)\, \hat{A}_i\big) - \beta\, D_{\mathrm{KL}}(\pi_\theta \| \pi_{ref}) \right], \tag{5}$$

where $r_{i,\tau} = \pi_\theta(y_\tau \mid y_{<\tau}, x) / \pi_{\theta_{old}}(y_\tau \mid y_{<\tau}, x)$.

Renderer Optimization. Visual factors are more entangled at the image realization level. We therefore use a unified reward, and optimize $\mathcal{G}_\phi$ directly via DiffusionNFT Zheng et al. [2025]:

$$R_{\mathrm{uni}}(\hat{I}) = \lambda_{\mathrm{ctr}} S_{\mathrm{ctr}}(\hat{I}) + \lambda_{\mathrm{gen}} \sum_{i \in \mathcal{T}} \bar{w}_i S_i(\hat{I}), \tag{6}$$

where $\bar{w}_i$ are fixed dimension weights. DiffusionNFT samples K posters for each condition (I_s, y). Note that this reward applies during the closed-loop alternating stage; the earlier OCR-guided RL (Appendix A) is used as a warm-up to ensure baseline text fidelity before joint optimization begins.

4 Experiments

4.1 Implementation Overview

The Designer uses Qwen3-VL-8B, trained with full-parameter SFT then full-parameter RL. The Renderer is built on Qwen-Image-Edit-2511, also trained with SFT followed by DiffusionNFT-based RL. The CTR preference model adopts SigLIP2-Giant; the general-quality reward model uses a

Table 1: Comparison of different models on our test set for pairwise CTR preference prediction. †: released checkpoint trained on the public dataset.

Model	PairAcc	PairWAcc	ItemTopAcc	ItemSpearman
Qwen3-VL	46.7	47.1	29.2	0.172
CAIG†	47.8	49.6	33.3	0.191
Claude-opus-4.6	48.7	51.7	36.8	0.172
GPT-4o	51.6	52.7	39.8	0.198
Gemini-3.1-pro	55.2	59.2	42.4	0.230
Qwen3-VL-MLP	68.2	74.0	51.1	0.394
Ours	**70.1**	**76.0**	**53.0**	**0.415**

Table 2: Quantitative comparison under different settings.

Method	NED↓	Overlap↓	CLIP↑	VQ↑					PC↑			
				Overall	Aesth.	Layout	Product	Commercial	Overall	Construct	Key Detail	Material
Holistic Product-aware Poster Editing Setting												
Flux.1 Fill	0.907	0.175	0.321	32.53	33.51	39.25	36.70	24.78	53.09	69.75	50.21	54.36
Flux.2	0.386	0.041	0.371	78.91	79.27	80.43	83.29	74.72	75.53	77.12	73.51	81.17
Qwen Image Edit 2509	0.376	0.144	0.360	69.61	69.51	72.44	71.95	67.19	69.70	74.45	66.58	78.33
Qwen Image Edit 2511	0.302	0.066	0.366	74.45	74.25	76.38	76.62	71.83	75.85	78.05	72.93	81.59
GPT-4o	0.445	0.148	0.363	77.58	77.14	78.35	80.76	75.98	67.72	77.94	58.17	82.54
Ours	**0.251**	**0.031**	**0.379**	**81.24**	**80.95**	**80.63**	**84.26**	**79.66**	**85.51**	**85.43**	**85.13**	**87.08**
Using the Prior Work Setting												
Poster Maker	0.239	0.189	0.320	47.30	49.69	51.95	49.42	41.81	89.68	91.34	87.97	93.19
GPT-4o w/ Text Image	0.439	0.134	0.352	76.44	76.27	77.38	78.94	74.26	81.92	83.71	80.43	86.38
Ours	**0.214**	**0.048**	**0.375**	**79.26**	**77.81**	**79.36**	**80.74**	**78.68**	**92.13**	**92.33**	**91.71**	**94.98**

separate Qwen3-VL-8B instance. For CTR training, we collect over 2M same-product image pairs from a commercial e-commerce platform, filtered for temporal CTR consistency across 7/14/30-day windows, with 50K pairs held out for testing. For offline evaluation, we use GPT-5.4 as an independent judge to score Visual Quality (VQ) and Product Consistency (PC); this evaluator is strictly separate from the Qwen3-VL-8B-based reward model used during training, differing in both model family and scoring dimensions. Full dataset construction and compute details are in Appendix B.

4.2 CTR Preference Model Evaluation

We evaluate pairwise CTR preference prediction accuracy using four metrics: PairAcc (pair-level accuracy), PairWAcc (gap-weighted pair accuracy), ItemTopAcc (whether the top-ranked image per product is correct), and ItemSpearman (rank correlation per product). Formal definitions are in the Appendix C.1. Table 1 compares our CTR preference model against general-purpose multimodal models on our test set. General-purpose VLMs perform near chance level, and even the strongest baseline remains far below our model. We also include Qwen3-VL-MLP, which appends an MLP head to Qwen3-VL's visual representation and trains on our data; while this narrows the gap considerably, our dedicated multi-granularity architecture still outperforms it and is $8\times$ more efficient in both training and inference throughput, making it far more practical for the tight reward-scoring loop required by RL. Additional evaluation on a public creative-ranking benchmark is provided in Appendix C.2. We also conduct comprehensive ablations of the CTR preference model's architecture, training strategy, and backbone choice (full results in Appendix C.3).

4.3 Main Results

Table 2 presents the quantitative results under our proposed holistic product-aware poster editing setting. Unlike prior protocols that assume access to text priors, this setting requires a model to simultaneously render promotional text, organize the poster layout, align with the semantic instruction, and preserve the target product appearance. Under this more challenging setting, our method achieves the best performance on the primary metrics(Figure 4). The lower half of Table 2 evaluates all methods under the protocol used by prior work, where text priors are available. This setting enables accurate segmentation of product images and allows for direct comparison with previous poster generation and editing methods. Our method again achieves the best overall performance on most metrics. These results indicate that, even when explicit text priors are provided, our method renders text more accurately, arranges visual elements more coherently, and maintains stronger semantic consistency



Figure 4: Qualitative comparison of different methods under the same prompt. Red boxes highlight text rendering errors, blue boxes indicate inconsistencies in product details or structure, and yellow boxes mark abnormal visual artifacts such as unrealistic colors. Compared with other methods, our method better preserves product appearance, textual information, and fine-grained visual consistency, achieving the best overall visual quality.

Table 3: Quantitative comparison under different optimization settings.

Method	NED↓	Overlap↓	CLIP↑	VQ↑					PC↑			
				Overall	Aesth.	Layout	Product	Commercial	Overall	Construct	Key Detail	Material
Ablation Study on Reward Signal Optimization												
No Optimization	0.273	0.058	0.366	75.17	75.75	78.26	76.67	73.21	83.64	83.05	83.93	84.59
Optimize generation model only	0.268	0.052	0.369	76.04	76.28	78.61	77.35	74.02	84.32	84.31	84.07	84.64
Optimize VLM only	0.259	0.041	0.371	77.81	77.68	79.74	79.82	75.84	84.47	84.24	84.01	85.51
Optimize VLM before generation model	0.254	0.035	0.373	78.61	78.32	80.21	80.67	76.81	84.65	84.44	84.13	85.91
Reward Composition Ablation												
CTR reward only	0.750	0.150	0.249	72.00	72.02	76.27	75.10	67.87	68.29	79.57	62.12	73.51
General-quality reward only	0.253	0.033	0.374	**81.91**	77.21	**80.89**	81.03	78.82	**85.90**	**86.27**	84.02	**88.21**
Full model (dual reward + joint opt.)	**0.251**	**0.031**	**0.379**	81.24	**80.95**	80.63	**84.26**	**79.66**	85.51	85.43	**85.13**	87.08

than competing approaches such as Poster Maker and GPT-4o. Therefore, the effectiveness of our method generalizes across both the proposed holistic setting and the prior-work setting, yielding a better balance among text rendering, visual quality, and product faithfulness. We further conduct a pairwise blind human evaluation with five professional annotators, with details and results reported in Appendix B.3.

4.4 Ablation Study and Analysis

Optimization Settings Ablation. The ablation results highlight the importance of the proposed alternating dual-level optimization and dual-reward formulation. Compared with optimizing a single branch or relying on a single reward signal, the full framework establishes a more balanced and effective optimization loop between semantic planning and visual rendering. The Designer is encouraged to produce more reward-aligned and commercially meaningful editing instructions, while the Renderer progressively improves its ability to translate these instructions into visually appealing, product-aware posters. By alternating optimization once the reward gain saturates, the method avoids overfitting one module in isolation and instead enables the two components to co-adapt, so that improved design plans provide clearer rendering guidance and a stronger Renderer further amplifies the benefit of high-quality instructions. Meanwhile, the reward composition ablation shows

that the two reward signals play complementary roles: optimizing only for CTR tends to exploit click-through patterns at the cost of generation fidelity and visual quality, reflecting a form of reward hacking, whereas relying solely on general-quality reward preserves visual quality but lacks sufficient commercial relevance and semantic alignment. The dual-reward design therefore prevents either objective from dominating and encourages a better trade-off between aesthetic quality, instruction faithfulness, and commercial effectiveness. Overall, these ablations validate that both the alternating optimization strategy and the balanced reward composition are essential for robust and commercially effective poster generation.

TSRO and Switching Strategy Ablation. We ablate the internal design choices of TSRO and the alternating switching criterion. The ablation validates our design hierarchy: dimension-rotating reward decomposition (target split) is the core mechanism, while relative reward serves as an auxiliary stabilizer.

Table 4: Ablation of TSRO components and switching strategy. All variants use alternating dual-level optimization; differences are in the Designer branch only.

Variant	NED↓	Overlap↓	CLIP↑	VQ↑					PC↑			
				Overall	Aesth.	Layout	Product	Commercial	Overall	Construct	Key Detail	Material
TSRO Component Ablation												
No target split (GRPO)	0.308	0.050	0.297	75.82	73.46	72.76	75.89	72.48	81.85	81.23	79.76	83.68
No relative reward (R_{main} only, λ_{rel}=0)	0.271	0.042	0.374	78.23	77.91	79.81	80.04	76.51	84.63	77.33	80.92	82.73
λ_{rel}=0.1	0.268	0.040	0.375	81.47	79.83	80.21	80.31	76.72	84.79	82.38	**85.42**	83.81
Switching Strategy Ablation												
Fixed alternating (K=1)	0.294	0.068	0.312	72.24	69.78	73.41	75.21	71.87	79.58	80.98	77.12	83.54
Fixed alternating (K=5)	0.323	0.051	0.351	74.86	74.52	77.93	75.64	73.42	82.87	77.06	80.21	81.78
Fixed alternating (K=10)	0.301	0.036	0.376	77.71	73.42	72.19	76.53	72.08	82.89	72.41	78.01	80.85
Switching Hyperparameter Sensitivity												
W=3, H=2, ε=0.3	0.256	0.033	0.361	80.87	80.12	80.41	83.58	79.02	85.23	**85.67**	84.78	86.54
W=8, H=5, ε=1.0	0.268	0.034	0.376	80.53	80.46	**80.92**	84.01	79.41	85.08	84.91	85.37	**87.22**
Full model (W=5, H=3, ε=0.5)	**0.251**	**0.031**	**0.379**	**81.24**	**80.95**	80.63	**84.26**	**79.66**	**85.51**	85.43	85.13	87.08

No target split produces the largest degradation across all metrics. This directly demonstrates the central claim: when the entire prompt receives a single holistic reward—equivalent to standard GRPO applied to structured output, the per-dimension gradient signal is too sparse to drive meaningful improvement in any individual semantic component. The model converges to a local optimum where easy-to-improve dimensions dominate gradient flow while harder dimensions stagnate.

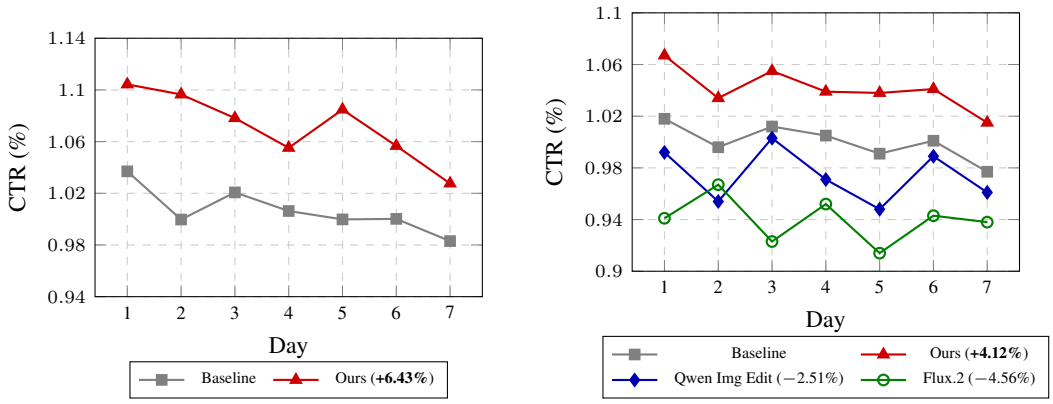
No relative reward (λ_{rel}=0) shows moderate degradation: without the baseline comparison as a safety constraint, non-focal dimensions occasionally regress during focused optimization steps. The safety constraint needs sufficient strength (λ_{rel}=0.5) to consistently protect all dimensions simultaneously.

Fixed alternating schedules show that premature switching (K=1) causes oscillatory instability comparable to removing target split entirely, while longer schedules (K=5, 10) progressively approach the full model but still underperform the adaptive criterion. The adaptive switching mechanism handles per-category variance: simple products saturate quickly while complex scenes require longer optimization horizons, making a fixed schedule systematically suboptimal for at least one subset.

Switching hyperparameter robustness. The adaptive criterion uses W=5 H=3, and ε=0.5. As shown in Table 4, varying these parameters across a wide range ($W \in [3, 8]$, $H \in [2, 5]$, $\varepsilon \in [0.3, 1.0]$) yields final metrics within small of our default. Notably, both alternative settings surpass the full model on certain individual metrics, indicating that the performance variations are due to minor trade-offs rather than systematic degradation. This robustness stems from the nature of the saturation signal itself: reward curves plateau sharply once a branch exhausts its current improvement capacity, making detection reliable across a broad range of thresholds. The ε threshold is defined relative to the reward scale (percentage points), making it agnostic to product category.

4.5 Online A/B Testing

To validate real-world commercial impact, we deploy the CTR preference model on a major e-commerce platform for image selection optimization. We select a set of seed products and use each product's current main display image (head image) as the baseline. For each seed product, we retrieve



(a) Image selection via CTR preference model.

(b) Full pipeline generation comparison.

Figure 5: Online A/B tests over 7 consecutive days on a live e-commerce platform. (a) Image selection: replacing head images with the highest CTR-scored candidate from the same-product cluster. (b) Image generation: replacing head images with posters generated by our full pipeline vs. competing generation methods.

all candidate images from its same-product cluster (SPU-level) and score both the baseline and all candidates using our CTR preference model. A replacement is triggered only when a candidate's predicted CTR score strictly exceeds that of the baseline; in such cases, the candidate with the highest predicted score is used as the new display image. The experiment runs for 7 consecutive days on live traffic. Figure 5 reports daily CTR over a 7-day period for both experiments.

Part I: Image Selection. As shown in Figure 5a, the CTR-guided image replacement achieves an average relative CTR improvement of **+6.43%**, with daily gains ranging from +4.54% to +9.68%. The baseline group serves approximately 102M impressions and 1.03M clicks over the period, while the experimental group receives 56.7M impressions and 607K clicks. The consistent gap confirms that the CTR preference model trained with our two-stage bucket-then-ranking strategy provides reliable commercial value.

Part II: Image Generation. We further evaluate the full PosterDuet pipeline by replacing product head images with posters generated end-to-end from only the raw product image. We compare against two strong generation baselines: Qwen Image Edit and Flux.2, both conditioned on similar inputs. As shown in Figure 5b, PosterDuet achieves an average relative CTR improvement of **+4.12%** over the original head images, significantly outperforming Qwen Image Edit (-2.51%) and Flux.2 (-4.56%). The negative CTR impact of both baselines suggests that generic generation quality alone is insufficient—effective poster synthesis requires both visual appeal and commercial relevance, which our reward-driven co-evolution framework explicitly optimizes.

5 Conclusion

We presented PosterDuet, a closed-loop framework that couples a VLM-based Designer with a unified Renderer for holistic product-aware poster generation. Unlike staged pipelines that fix design decisions before rendering, our framework allows the Designer to refine its instructions based on feedback from the generated posters, enabling reciprocal adaptation between design planning and visual synthesis. Through alternating dual-level optimization with TSRO and DiffusionNFT, the two modules co-evolve under shared CTR and quality rewards, aligning visual quality with commercial effectiveness. Natural-language critiques further provide actionable guidance for revising design decisions in response to rendering outcomes. Both offline evaluations and online A/B tests confirm state-of-the-art poster quality and significant CTR gains on live traffic. These results highlight the value of jointly optimizing design and rendering with outcome-based feedback for automated product poster generation.

Acknowledgements

This work was supported by the Shenzhen Science and Technology Program (Grant No. RCJC20210706091946001) and the Shenzhen Science and Technology Program (Grant No. ZDCY20250901104207008).

References

Anonymous. Innoads-composer: Efficient condition composition for e-commerce poster generation. *arXiv preprint*, 2026a.

Anonymous. Posteromni: Generalized artistic poster creation via task distillation and unified reward feedback. *arXiv preprint*, 2026b.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Black Forest Labs. Flux.1. `https://huggingface.co/black-forest-labs/FLUX.1-dev`, 2024. Text-to-image generation model.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

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*, 2024. doi: 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)*, pages 5583–5591, 2025a.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xingye Chen, Wei Feng, Zhenbang Du, Weizhen Wang, Yanyin Chen, Haohan Wang, Linkai Liu, Yaoyu Li, Jinyuan Zhao, Yu Li, Zheng Zhang, Jingjing Lv, Junjie Shen, Zhangang Lin, Jingping Shao, Yuanjie Shao, Xinge You, Changxin Gao, and Nong Sang. Ctr-driven advertising image generation with multimodal large language models. In *Proceedings of the ACM Web Conference 2025*, pages 2262–2275, 2025b. doi: 10.1145/3696410.3714836.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL `https://arxiv.org/abs/2507.05595`.

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, 2025. URL `https://arxiv.org/abs/2512.21921`.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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, 2025. URL `https://arxiv.org/abs/2504.06632`.

Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6018–6026, 2023.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Fengheng Li, An Liu, Wei Feng, Honghe Zhu, Yaoyu Li, Zheng Zhang, Jingjing Lv, Xin Zhu, Junjie Shen, Zhangang Lin, and Jingping Shao. Relation-aware diffusion model for controllable poster layout generation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023a. doi: 10.1145/3583780.3615028.

Zhaochen Li, Fengheng Li, Wei Feng, Honghe Zhu, An Liu, Yaoyu Li, Zheng Zhang, Jingjing Lv, Xin Zhu, Junjie Shen, Zhangang Lin, Jingping Shao, and Zhenglu Yang. Planning and rendering: Towards end-to-end product poster generation. *arXiv preprint arXiv:2312.08822*, 2023b. doi: 10.48550/arXiv.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*, pages 1250–1260, 2023. doi: 10.1145/3581783.3611930.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl, 2025. URL `https://arxiv.org/abs/2505.05470`.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. In *European Conference on Computer Vision*, 2024.

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*, pages 10684–10695, 2022.

Shuai Sun et al. Depo: Enhancing e-commerce image background generation with short-trajectory direct expected preference optimization. *arXiv preprint*, 2025.

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Gao, and Jiawei Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Dong Wang, Kavé Salamatian, Yunqing Xia, Weiwei Deng, and Qi Zhang. Bert4ctr: An efficient framework to combine pre-trained language model with non-textual features for ctr prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5039–5050, 2023. doi: 10.1145/3580305.3599780.

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, 2025a. URL `https://arxiv.org/abs/2312.13309`.

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 (EMNLP)*, 2025b.

Shichen Wang, Junqi Jin, Jian Xu, Zhibo Zhu, Yanhua Cheng, Hao Li, and Weiwen Liu. A hybrid bandit model with visual priors for creative ranking in display advertising. In *Proceedings of the Web Conference 2021 (WWW)*, pages 2324–2335, 2021. doi: 10.1145/3442381.3449857.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL `https://arxiv.org/abs/2508.02324`.

Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhu Chen. Editreward: A human-aligned reward model for instruction-guided image editing. In *International Conference on Learning Representations*, 2026.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. In *arXiv preprint arXiv:2306.09341*, 2023.

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, 2023.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. Dancegrpo: Unleashing grpo on visual generation, 2025. URL `https://arxiv.org/abs/2505.07818`.

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 on Web Conference 2024*, pages 180–189, 2024a. doi: 10.1145/3589335.3648315.

Tao Yang, Yingmin Luo, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. Posterllava: Constructing a unified multi-modal layout generator with llm. *arXiv preprint arXiv:2406.02884*, 2024b.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

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*, pages 3836–3847, 2023.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process, 2025. URL `https://arxiv.org/abs/2509.16117`.

Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pages 4995–5001, 2022. doi: 10.24963/ijcai.2022/692.

A Training Details of the Core Generation Pipeline

A.1 Designer Supervised Fine-Tuning

Given a raw product image I_s and its metadata $\mathbf{m}$, a template function constructs the instruction $X_{instr} = f_{instr}(\mathbf{m})$ encoding design rules and style constraints. The Designer π_θ models the design prompt as an autoregressive token sequence:

$$\pi_\theta(y \mid I_s, X_{instr}) = \prod_{i=1}^{N} p_\theta(y_i \mid I_s, X_{instr}, y_{<i}) \,, \tag{7}$$

where $y = (y_1, \ldots, y_N)$ jointly encodes background, layout, promotional text, and decorative elements. We fine-tune π_θ on curated triplets $\mathcal{D} = \{(I_s^{(k)}, \mathbf{m}^{(k)}, y^{(k)})\}_{k=1}^{K}$ by minimizing the standard cross-entropy loss:

$$\mathcal{L}_{plan}(\theta) = -\mathbb{E}_{(I_s,\mathbf{m},y)\sim\mathcal{D}} \sum_{i=1}^{N} \log p_\theta(y_i \mid I_s, X_{instr}, y_{<i}) \,. \tag{8}$$

A.2 Renderer Stage I: Product-Consistent SFT

We curate a high-quality dataset $\mathcal{D}_{edit} = \{(I_s^{(k)}, y^{(k)}, I_{post}^{(k)})\}_{k=1}^{K}$ of (product, prompt, poster) triplets and fine-tune the Renderer $\mathcal{G}_\phi$ under the flow-matching objective:

$$\mathcal{L}_{SFT}(\phi) = \mathbb{E}_{(I_s,y,I_{post}),\, t,\, \epsilon} \left\| v_\phi(z_t, t \mid I_s, y) - (\epsilon - z_0) \right\|_2^2 , \tag{9}$$

where z_0 is the latent of I_{post} and z_t its noised version at timestep t.

To retain the pretrained prior while adapting to the poster domain, we fine-tune only the first $L/3$ DiT blocks and freeze the remainder together with the text encoder and VAE. This shallow-block adaptation strategy dominates the binding between the reference product and the target layout, yielding strong product consistency without catastrophic forgetting (see ablation in Section 4.4).

A.3 Renderer Stage II: OCR-Guided Reinforcement

Chinese text remains difficult for diffusion models due to its large character set and dense stroke structure. Since glyph-level errors are weakly constrained by the flow-matching objective, we introduce an OCR-based reward and optimize the Renderer with DiffusionNFT.

Given a generated poster $\hat{I} \sim \mathcal{G}_\phi(I_s, y)$, we extract target text snippets $\mathcal{T}(y) = \{t_i\}_{i=1}^{N_t}$ from the quoted spans in y, and run PaddleOCR-v5 to obtain recognized strings $\mathcal{O}(\hat{I}) = \{o_j\}_{j=1}^{N_o}$. Both sets are normalized (whitespace removal, case folding). For each target t_i, we first check substring inclusion (similarity $= 1$ if matched), otherwise compute normalized Levenshtein similarity:

$$s_i = \max_j \left(1 - \frac{\mathrm{Lev}(\tilde{t}_i, \tilde{o}_j)}{\max(|\tilde{t}_i|, |\tilde{o}_j|)} \right), \tag{10}$$

where $\tilde{\cdot}$ denotes normalization. Targets with $s_i < \tau$ are counted as missed ($s_i \leftarrow 0$). The final reward combines a similarity term and a coverage term:

$$R_{\mathrm{OCR}}(\hat{I}, y) = \underbrace{\frac{1}{N_t} \sum_{i=1}^{N_t} s_i \mathbf{1}[s_i \geq \tau]}_{similarity} \cdot \underbrace{\max\left(0,\ 1 - \frac{N_t^{\mathrm{miss}} + N_o^{\mathrm{unmatched}}}{\max(N_t, N_o)}\right)}_{coverage}, \tag{11}$$

where N_t^{miss} is the number of missed targets and $N_o^{\mathrm{unmatched}}$ is the number of OCR strings unmatched by any target. This reward encourages faithful rendering of required text while penalizing missing and hallucinated strings. R_{OCR} is used as the raw reward in DiffusionNFT, normalized within each prompt group and converted to the optimality probability for gradient weighting.

Algorithm 1: Alternating Dual-Level Optimization

Input: training set $\mathcal{D}$, Designer π_θ, Renderer $\mathcal{G}_\phi$, window size W, patience H, threshold ε
Output: optimized parameters θ and ϕ

Initialize active branch $a \leftarrow$ DESIGN;
Initialize metric queue $\mathcal{Q} \leftarrow \emptyset$ and stable counter $c \leftarrow 0$;
while *training not converged* **do**
  $(I_s, \mathbf{m}) \sim \mathcal{D}$;
  $X_{\text{instr}} = f_{\text{instr}}(\mathbf{m})$;
  if $a =$ DESIGN **then**
    $\theta, m_t \leftarrow \text{TSRO}(\theta; \phi, I_s, X_{\text{instr}})$;
  else
    $\phi, m_t \leftarrow \text{NFT}(\phi; \theta, I_s, X_{\text{instr}})$;
  Push m_t into $\mathcal{Q}$ and keep the latest W values;
  if $|\mathcal{Q}| = W$ **then**
    $\bar{m}_t = \frac{1}{W} \sum_{j=t-W+1}^{t} m_j$;
    $\Delta_t = \bar{m}_t - \bar{m}_{t-1}$;
    if $\Delta_t < \varepsilon$ **then**
      $c \leftarrow c + 1$;
    else
      $c \leftarrow 0$;
  if $c \geq H$ **then**
    $a \leftarrow$ RENDER if $a =$ DESIGN, otherwise $a \leftarrow$ DESIGN;
    $\mathcal{Q} \leftarrow \emptyset$, $c \leftarrow 0$;

A.4 Alternating Dual-Level Optimization Algorithm

B Experimental Details

B.1 Dataset Preparation

CTR Prediction Dataset. The CTR prediction data is collected from Kuaishou e-commerce platform. We retain images with page views (PV) $> 2{,}000$ and construct same-product image pairs. For the test set, we extract CTR records over 7-day, 14-day, and 30-day windows and keep only pairs whose relative CTR ordering remains consistent across all three periods. Training pairs from the same product cluster as any test pair are removed to prevent leakage. We further filter obviously low-quality images (e.g., excessively large overlaid text, black backgrounds) to align the CTR signal with general aesthetic quality. The final dataset contains over 2M training pairs and 50K test pairs. A public-dataset variant with PV threshold 100 yields 400K/120K train/test samples.

SFT Dataset. We build a large-scale image–text editing dataset for product-oriented image editing. Each sample in the dataset consists of a source image, an edited image, an editing instruction prompt, and a quality assessment of the edited result. The images are obtained from a major e-commerce platform, while the editing prompts and quality assessments are generated using Gemini 3.1 Pro. To improve the reliability of the dataset for supervised fine-tuning, we conduct data filtering to remove low-quality examples. Specifically, we discard samples in which the source and edited images exhibit poor semantic or visual consistency, as well as samples containing blurred or otherwise degraded images. This filtering process helps ensure that the retained samples provide reliable supervision for learning high-quality and consistent product image edits. After filtering, the resulting dataset contains approximately 200K image–text editing samples. We use this dataset for three downstream training objectives. First, it serves as supervised fine-tuning data for the image editing generation model, aiming to enhance the consistency of products before and after editing. Second, it is used to fine-tune a designer model to capture high-quality poster design priors, including background generation, layout arrangement, and visual composition. Third, the same dataset is used to train a quality reward model that predicts the editing quality of generated images.

Table 5: Human preference evaluation (% of Ours vs. each baseline, n=200, 5 annotators).†: $p < 0.01$ (two-sided binomial test on Win vs. Lose counts).

Ours vs.	Win	Tie	Lose
Flux.2	77.5†	9.0	13.5
Qwen Edit 2511	71.0†	12.5	16.5
GPT-4o	62.0†	13.0	25.0

B.2 Model and Training Configuration

Designer. We initialize the designer from Qwen3-VL-8B and first perform full-parameter supervised fine-tuning on the filtered 200K image–text editing dataset for 5 epochs with a learning rate of 1e-5 on 8×H200 GPUs. We then conduct full-parameter reinforcement learning using a curated subset of 960 prompts, which is sufficient to achieve stable preference alignment in our setting. The RL stage is run on 16×A800 GPUs, with 8 GPUs serving as generation workers.

Renderer. We initialize the renderer from Qwen-Image-Edit-2511 and first perform full-parameter supervised fine-tuning on the filtered 200K image–text editing dataset for 5 epochs with a learning rate of 5e-6 on 8×H200 GPUs. We then apply DiffusionNFT-based reinforcement learning using a dedicated 5,760-prompt preference-alignment subset, where 4,800 prompts are used with the OCR-based reward and 960 prompts are used with the CTR and quality rewards. We set the NFT coefficient $\beta = 0.1$, clipping range 2×10^{-4}, advantage clipping range 5.0, learning rate 8×10^{-5} with AdamW, weight decay 10^{-4}, and EMA decay 0.9. For each prompt, we sample $K = 16$ candidate edited images and compute rewards according to the corresponding alignment objective. We generate candidate images using 25 inference steps with guidance scale 4.

CTR Preference Model. Visual encoder: siglip2. Text encoder: same architecture. Both encoders are fine-tuned end-to-end during Stage-2 ranking training.

Quality Reward Model. We instantiate a separate Qwen3-VL-8B model, without parameter sharing with the Designer, to predict per-dimension quality scores for edited images. It is trained with full-parameter supervised fine-tuning on the filtered 200K image–text editing dataset for 10 epochs using a learning rate of 5e-6 on 8×H200 GPUs.

B.3 Human Evaluation Protocol

We conduct a pairwise blind preference study to complement automatic metrics.

Setup. We randomly sample 200 product images from our held-out test set, stratified across 5 major categories (apparel, electronics, food & beverage, home & furniture, beauty & personal care; 40 images each). For each product, we compare our method against baselines: Flux.2, Qwen Image Edit 2511 and GPT-4o. Five annotators with e-commerce visual design experience participate. For each comparison pair, the two posters are displayed side-by-side with randomized left/right positions. Annotators are asked a single question: *"Which poster would you prefer to use for commercial deployment?"* and select one of three options: *Left*, *Right*, or *Tie*. Annotators are blind to which system produced each poster.

Results. We aggregate by majority vote (at least 3 out of 5 agree) and report win/tie/lose rates in Table 5. Statistical significance is assessed via two-sided binomial test.

C CTR Preference Model Analysis

C.1 Evaluation Metrics

In Stage 1, the model is evaluated by bucket accuracy. Let $\mathcal{B} = \{B_1, \ldots, B_K\}$ denote the predefined CTR buckets. For an image i, we denote by $b(c_i)$ the ground-truth bucket index determined by its

CTR, and by $\hat{b}(s_i)$ the predicted bucket index. The Stage-1 accuracy is defined as

$$\text{BucketAcc} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[\hat{b}(s_i) = b(c_i)\right],$$

where N is the number of evaluated images.

In Stage 2, we evaluate the CTR preference model from both pair-level and item-level perspectives. For pair-level evaluation, given an image pair (i, j), the predicted preference is considered correct if the predicted score difference has the same direction as the ground-truth CTR difference. The average PairAccuracy is computed as

$$\text{PairAcc} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}}\mathbb{I}\left[(s_i - s_j)(c_i - c_j) > 0\right],$$

where $\mathcal{P}$ denotes the evaluated image-pair set. Since image pairs with larger CTR gaps provide more reliable preference supervision, we further report weighted PairAccuracy. The gap-aware weight is defined as $w_{ij} = \min\left(1, \frac{|c_i - c_j|}{\gamma}\right)$, where $\gamma = 0.009$. The weighted PairAccuracy is then calculated as

$$\text{PairWAcc} = \frac{\sum_{(i,j)\in\mathcal{P}} w_{ij}\mathbb{I}\left[(s_i - s_j)(c_i - c_j) > 0\right]}{\sum_{(i,j)\in\mathcal{P}} w_{ij}}.$$

For item-level evaluation, we consider all candidate images belonging to the same item. For item q, let $\{c_{q,k}\}_{k=1}^{n_q}$ and $\{s_{q,k}\}_{k=1}^{n_q}$ denote the ground-truth CTR values and predicted scores of its n_q candidate images. ItemTopAccuracy measures whether the image with the highest predicted score is also the image with the highest ground-truth CTR:

$$\text{ItemTopAcc} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\mathbb{I}\left[\arg\max_k s_{q,k} = \arg\max_k c_{q,k}\right],$$

where $\mathcal{Q}$ denotes the evaluated item set. In addition, we report ItemSpearman to measure the overall ranking consistency within each item. It is computed as the average Spearman rank correlation between the predicted scores and the ground-truth CTR values:

$$\text{ItemSpearman} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\rho\left(\{s_{q,k}\}_{k=1}^{n_q}, \{c_{q,k}\}_{k=1}^{n_q}\right),$$

where $\rho(\cdot,\cdot)$ denotes the Spearman rank correlation.

C.2 Effect Verification on Public Data

We further evaluate our method on a public benchmark for creative ranking in display advertising, namely the *CreativeRanking* dataset Wang et al. [2021]. CreativeRanking is a large-scale benchmark constructed for creative ranking research, containing over 1.7M ad creatives from 500K products.

Since this public benchmark only supports image-based input in our evaluation setting and does not provide the full multimodal product-side information required by our original model design, we adapt our model to a simplified vision-only variant. Specifically, the model is reduced to a *vision encoder + MLP* architecture, where the text branch and multimodal fusion module are removed.

As shown in Table 6, our vision-only variant still outperforms all baselines including task-specific methods (CG4CTR, CAIG), demonstrating that the proposed two-stage bucket-then-ranking training strategy generalizes effectively even without the multimodal fusion components. We additionally report a cross-domain variant trained exclusively on our proprietary data and evaluated directly on the public test set without any fine-tuning, which achieves 55.6 PairAcc—already competitive with CAIG trained on in-domain data. Since CAIG releases only model checkpoints without training code, we evaluate their released model directly on our proprietary test set for cross-domain comparison (Table 1), where it achieves 47.8 PairAcc.

Table 6: Results on the public CreativeRanking dataset. †: trained on our proprietary data only (no public training set).

Model	PairAcc
Qwen3-VL	47.9
Qwen3-VL-Trained	51.2
GPT4o	50.2
LLaVA	52.8
VAM	52.2
CG4CTR	53.1
CAIG	56.2
Ours (Cross-domain†)	55.6
Ours (Vision-only)	**57.1**

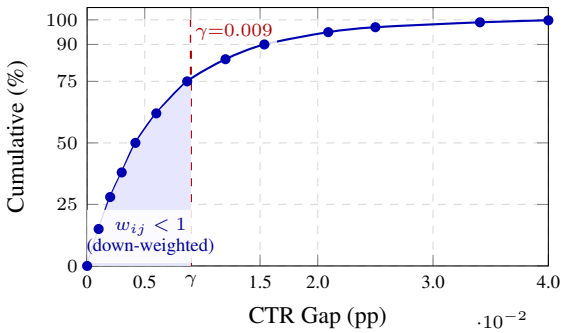


Figure 6: Empirical CDF of pairwise CTR gaps. Pairs left of γ ($\sim$75%) are down-weighted.

C.3 Ablation Studies

Model Architecture. We compare the proposed multi-granularity cross-attention architecture with several simplified variants (Table 7). Incorporating product-side text significantly improves over the vision-only baseline, confirming that CTR prediction benefits from multimodal input. However, directly concatenating visual and textual embeddings *degrades* performance below the vision-only model, indicating that naive fusion introduces noise. Separating image and text into multi-granularity streams resolves this conflict; further adding cross-attention consistently improves all metrics.

Table 7: Ablation study on the image-text processing architecture.

Setting	Image&Text Split	Vision Enc.	Text Enc.	Cross Attn.	PairAcc	PairWAcc	ItemTopAcc	ItemSpearman
Vision-only	✗	✓	✗	✗	67.1	72.7	44.7	0.388
Direct Concat	✗	✓	✓	✗	66.4	72.0	39.9	0.385
Multi-gran. Concat	✓	✓	✓	✗	68.8	74.8	51.8	0.393
Ours	✓	✓	✓	✓	**70.1**	**76.0**	**53.0**	**0.415**

Training Strategy and Ranking Objective. We analyze the effect of training strategy under the fixed proposed architecture (Table 8). Key findings: **Stage-1 initialization.** Bucket pretraining consistently improves over training from scratch. 3-bucket initialization outperforms 9-bucket, suggesting that overly fine-grained discretization introduces label noise. **Ranking objective.** Replacing the gap-aware pairwise loss (Loss-1) with an item-level listwise loss (Loss-2: $\mathcal{L}^{(q)}_{\text{item}} = \log(1 + \sum_{c_{q,i} > c_{q,j}} w_{ij} \exp((s_{q,j} - s_{q,i})/T))$, where batches are organized at the item level with K images per item) consistently degrades results, confirming that pairwise supervision better exploits the CTR gap structure. **Pointwise weight λ.** $\lambda = 0.2$ yields the best trade-off; larger values over-regularize toward absolute CTR regression. **Encoder fine-tuning.** Tuning the vision encoder alone recovers most of the full-tuning performance, indicating it is the information bottleneck for CTR preference modeling.

Model Backbones. We compare visual backbones to disentangle encoder capacity from the proposed CTR-specific design (Table 9). SigLIP2-based encoders outperform DINOv3-Base. Scaling from SigLIP2-Base to SigLIP2-Giant yields only +0.2pp PairAcc, confirming that gains stem from architecture and training strategy rather than encoder capacity.

C.4 Analysis of CTR Gap Distribution

We analyze the CTR gap distribution of image pairs to understand the role of the gap-aware weight $w_{ij} = \min(1, |c_i - c_j|/\gamma)$ in our ranking loss. As shown in Figure 6, the cumulative distribution is heavily left-skewed: the median gap is only 0.42pp, and over 75% of pairs fall below $\gamma = 0.009$—without explicit weighting, the majority of gradient signals would come from noisy near-tie pairs.

Our gap-aware weight w_{ij} addresses this by continuously down-weighting near-tie pairs and up-weighting high-confidence pairs, providing an *implicit* curriculum that adapts to the gap distribution

Table 8: Training strategy ablation of the CTR preference model.

Ablation Group	Stage-1	Loss	pointwise	Tune Vision	Tune Text	PairAcc	PairWAcc	ItemTopAcc	ItemSpearman
Stage-1 pretraining and ranking objective									
Training	None	Loss-1	0.2	✓	✓	68.3	74.4	51.3	0.396
Training	None	Loss-2	–	✓	✓	66.4	72.1	39.9	0.385
Training	3 buckets	Loss-2	–	✓	✓	67.4	72.7	50.8	0.373
Training	9 buckets	Loss-1	0.2	✓	✓	69.5	75.5	51.8	0.398
Training	9 buckets	Loss-2	–	✓	✓	66.3	71.8	39.1	0.383
Pointwise auxiliary weight in Loss-1									
λ	*same as final*	Loss-1	0.1	✓	✓	69.5	75.6	52.7	0.401
λ	*same as final*	Loss-1	0.5	✓	✓	68.0	73.2	51.0	0.383
Encoder fine-tuning strategy									
Freeze	*same as final*	*same as final*	*same as final*	✗	✗	65.4	70.1	47.8	0.331
Freeze	*same as final*	*same as final*	*same as final*	✓	✗	69.2	75.2	51.5	0.401
Freeze	*same as final*	*same as final*	*same as final*	✗	✓	65.8	70.4	48.0	0.340
Final model	**3 buckets**	**Loss-1**	**0.2**	✓	✓	**70.3**	**76.3**	**53.3**	**0.420**

Table 9: Backbone ablation of the CTR preference model.

Backbone	PairAcc	PairWAcc	ItemTopAcc	ItemSpearman
SigLIP2-Base	70.1	76.0	53.0	0.415
DINOv3-Base	68.7	74.2	51.1	0.391
SigLIP2-Giant	**70.3**	**76.3**	**53.3**	**0.420**

without requiring hard filtering. To verify this, we train models on different gap-based subsets (hard filtering at P0/P25/P50/P75 thresholds) and evaluate across gap levels (Table 10). Performance differences across training subsets are minimal (<1pp), confirming that w_{ij} already captures the informative signal from large-gap pairs: explicit data filtering provides no additional benefit because the loss weight has already suppressed the contribution of unreliable small-gap pairs. This validates that using all available pairs (P0) with the gap-aware weight is the optimal strategy—it retains maximum data coverage while the soft weighting mechanism ensures training is dominated by trustworthy supervision.

D Reward Signal Analysis

D.1 What Design Dimensions Drive CTR?

To understand which visual design dimensions most strongly influence CTR, we conduct an LLM-based taxonomy analysis on our preference data. Specifically, we sample 1,000 image pairs where a VLM-generated editing prompt describes the transition from the lower-CTR image to the higher-CTR one. We focus on the 468 pairs where the *edited* image achieves higher CTR (i.e., the editing direction aligns with CTR improvement), and use GPT-4o to classify each prompt into exactly one of seven predefined design categories based on the *dominant* change.

Table 11 provides representative examples for each category, illustrating the concrete editing actions that characterize each design dimension.

As shown in Figure 7, the distribution reveals a clear hierarchy. **Composition/Framing** (25.0%)—encompassing cropping, camera angle, viewpoint, and subject positioning—is the single most impactful dimension, followed by **Text/Promotional Elements** (17.7%) and **Display Mode** (17.3%). Together, these three layout-level attributes account for 60.0% of all CTR improvements, while pixel-level attributes (Lighting/Color 10.9%, Image Quality 6.0%) contribute only 16.9%.

This finding provides empirical support for two design choices in our framework: (1) the Designer's target split in TSRO decomposes generation into layout structure (background, product display, text)—precisely the dimensions that dominate CTR impact; (2) pure image quality enhancement contributes minimally to CTR, validating our use of a *separate* generative quality reward rather than conflating quality with CTR in a single signal.

Table 10: PairAcc under different training and evaluation CTR-gap thresholds.

Training / Evaluation	P0	P25	P50	P75
P0	**70.1**	73.9	**77.7**	81.5
P25	69.6	73.7	77.4	81.1
P50	70.0	**74.0**	77.6	**81.7**
P75	69.3	73.1	76.9	81.3

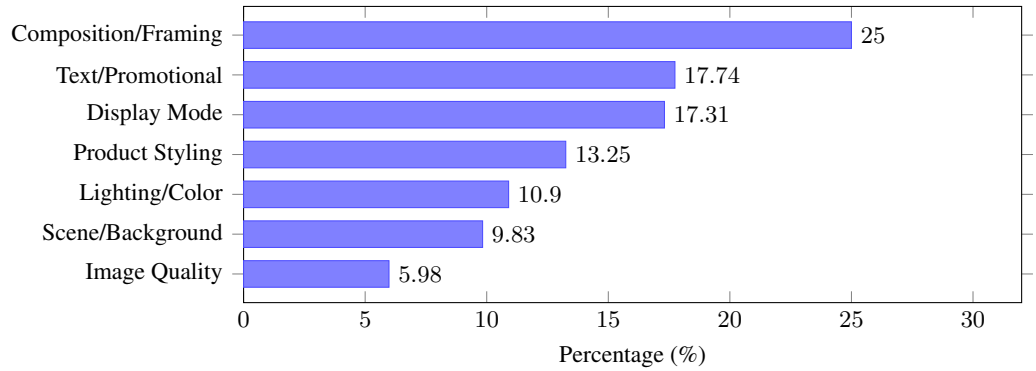


Figure 7: Distribution of dominant design dimensions in CTR-improving edits ($n = 468$). Categories are determined by GPT-4o single-label classification of VLM-generated editing prompts.

D.2 CTR vs. Generative Quality: Reward Orthogonality

A core assumption of our alternating dual-level optimization is that CTR appeal and generative quality capture *different* aspects of poster effectiveness, and therefore require separate reward signals. To empirically verify this, we score 100 generated posters with both the CTR preference model (producing $S_{\text{ctr}} \in [0, 1]$) and the generative quality reward model (GPT-5.4-based, producing per-dimension scores on a 0–100 scale across background/layout, text correctness, product fidelity, and a weighted total).

As shown in Table 12, the Spearman correlation between CTR score and every quality dimension is negligible ($|\rho| \leq 0.15$, all $p > 0.13$). In contrast, the quality dimensions are strongly correlated with each other ($\rho = 0.31$–0.81, $p < 0.002$), indicating they form a coherent quality manifold that is nearly *orthogonal* to CTR.

Concretely, we observe posters with high quality yet low CTR (e.g., a phone case poster scoring 97.7 on quality but only 0.005 on CTR—technically faithful but visually unremarkable in a competitive feed), and conversely, posters with high CTR yet mediocre quality (e.g., a jewelry poster scoring 0.971 on CTR but only 76.7 on quality due to text rendering errors). These cases illustrate that:

- Optimizing quality alone cannot improve CTR—a poster may be technically flawless yet lack feed-level visual salience.
- Optimizing CTR alone risks reward hacking—the model may learn visually "clickable" patterns that degrade text fidelity or product consistency.

This orthogonality provides the fundamental justification for our dual-reward design: CTR and quality must be *jointly* optimized as complementary signals rather than collapsed into a single scalar, and the alternating scheme ensures neither signal dominates at the expense of the other.

E Prompt Templates

We provide the full prompt templates used in the three core stages of our framework: Designer generation (§E.1), TSRO target-split revision (§E.2), and generative quality evaluation (§E.3).

Table 11: Category taxonomy with representative editing examples from the classified prompts.

Category	**%**	**Representative Edit Description**
Composition/Framing	25.0	Reframe the shot through tighter cropping on the product, shift to a centered symmetric layout, or adjust the camera angle to a 45° overhead viewpoint.
Text/Promotional	17.7	Add a bold price tag and "limited-time offer" banner in the upper-left corner; reorganize scattered text into a clean left-aligned hierarchy.
Display Mode	17.3	Switch from a flat-lay product shot to a model-worn lifestyle scene; present multiple color variants in a split-panel grid layout.
Product Styling	13.3	Change the displayed colorway from black to seasonal coral; add coordinated accessories (scarf, bag) around the main garment.
Lighting/Color	10.9	Raise overall brightness and shift the color tone from cool blue to warm golden; increase saturation to make the product pop.
Scene/Background	9.8	Replace the plain white studio backdrop with a contextual kitchen environment; switch to a gradient pastel background.
Image Quality	6.0	Improve sharpness and reduce compression artifacts while keeping composition and scene unchanged.

Table 12: Spearman correlation (ρ) between CTR predicted score and generative quality dimensions ($n = 100$). None reaches statistical significance ($p > 0.13$), confirming near-orthogonality.

Quality Dimension	ρ **vs. CTR**	p**-value**
Background & Layout	-0.151	0.133
Text Correctness	0.045	0.660
Product Fidelity	0.118	0.242
Total Quality Score	0.025	0.803

E.1 Designer Generation Prompt

The following prompt is used during Designer SFT and inference to generate holistic design descriptions from a product image and selling points.

System Prompt: Designer Generation

You are tasked with designing a promotional e-commerce homepage image for a product, aimed at Chinese online shoppers. You are provided with a product image and a set of key selling points in text. Generate a detailed description for creating a visually compelling product highlight image suitable for online shopping platforms. Follow these rules:

1. Strictly follow all provided selling points; do not invent or omit features or benefits.
2. Display the product in a context appropriate to its type:
   - For wearable items (clothing, shoes, jewelry, bags), show a human model wearing or holding the product.
   - For lifestyle items (home goods, electronics, gadgets), show the product in a realistic usage scenario or styled environment.
   - For media content (videos, digital products), emphasize visual presentation such as screens, thumbnails, or dynamic elements.
3. Focus on overall composition: placement of the product and supporting elements, foreground and background structure, and visual hierarchy that naturally guides the viewer's attention.
4. Describe background and scene elements that enhance the product: lifestyle settings, minimal or abstract studio backgrounds, props, textures, or subtle environmental cues.
5. Include decorative and promotional elements such as lighting effects, highlights, text banners, icons, geometric shapes, or sparkles to draw attention to key selling points without overpowering the product.
6. Emphasize the product's details, colors, textures, and usage context while making selling points clearly readable.
7. Suggest a vibrant, modern, and professional color scheme and visual style suitable for Chinese e-commerce homepage banners.
8. Output your result as a single, clear paragraph directly usable by a designer or AI image-generation system.
9. Remove the original text from the image while keeping the product unchanged.

E.2 TSRO Target-Split Revision Prompt

During alternating optimization, the Designer revises only a designated semantic target while keeping other components stable. Template variables are shown in `blue`.

System Prompt: Target-Split Revision

You are tasked with revising an existing promotional e-commerce homepage image prompt for a product, aimed at Chinese online shoppers. You are provided with a product image, a set of key selling points in text, and an existing full prompt that is already usable. Your job is NOT to rewrite the whole prompt freely. Instead, revise only one designated part while keeping the rest of the prompt as stable as possible.
Follow these rules:

1. Strictly follow all provided selling points; do not invent or omit features or benefits.
2. Preserve the product identity, product category, and overall commercial intent of the original prompt.
3. Only revise the designated target part:
   - Allowed to change: `{target_editable}`
   - Must stay unchanged: `{target_keep}`
   - Revision goal: `{effective_instruction}`
4. Keep the overall prompt style suitable for Chinese e-commerce homepage banners: visually appealing, commercial, energetic, and directly usable by an AI image-generation system.
5. Do not casually rewrite unrelated sentences. Make the smallest effective edit needed to improve the design goal.
6. Keep the prompt as one complete, coherent paragraph.
7. Output only the final revised full prompt wrapped in `<revised_prompt>...</revised_prompt>`.

Input: selling points = `{selling_points}`; original prompt = `{original_prompt}`

E.3 Generative Quality Reward Prompt

The quality reward model receives four inputs and returns per-dimension scores together with actionable revision advice for the next iteration.The total score `total_score_100` is computed as a

weighted sum of the three dimensions, but the weights vary depending on the active TSRO target (Section 3.3) to focus supervision on the dimension most relevant to the current revision goal.

System Prompt: Quality Evaluation

You are an e-commerce poster quality evaluator. You will receive four inputs:

- **[Product Photo]:** the real reference image of the product
- **[Selling Points]:** key product selling points
- **[Generation Prompt]:** the prompt used to generate the poster
- **[Final Poster]:** the generated poster image

Evaluate the [Final Poster] across three dimensions and provide a revision suggestion for the generation prompt.

1. Background & Layout (`bg_layout`) — Does the background and layout complement the product without overshadowing it? Is the overall design aesthetically appealing?

- 90–100: Background, layout, and color palette perfectly match the product and selling points; professional composition with strong visual appeal.
- 75–89: Generally harmonious with minor issues (e.g., slightly bland background, imperfect whitespace); usable as-is.
- 50–74: Acceptable but unremarkable; weak connection between background and product; unclear visual hierarchy.
- 25–49: Obvious problems—mismatched background, product not prominent, cluttered layout, or color clashes.
- 1–24: Background completely disconnected from product; chaotic layout; unusable.

2. Text Correctness (`text_correctness`) — Are all rendered text elements in the poster accurate, complete, and legible?

- 90–100: All text is correct, sharp, and fully readable.
- 75–89: Text is mostly correct with only very minor blurriness at edges.
- 50–74: A few questionable characters or slightly unclear words; needs manual review.
- 25–49: Clear misspellings, garbled characters, or multiple illegible areas.
- 1–24: Widespread text errors, gibberish, or unreadable content.

3. Product Fidelity (`product_fidelity`) — Does the product in the poster faithfully match the original product photo in shape, color, texture, and details?

- 90–100: Highly faithful—color, structure, material, and logos all accurately reproduced.
- 75–89: Overall consistent with minor deviations that do not affect recognition.
- 50–74: Recognizable as the same product but with noticeable detail differences.
- 25–49: Key attributes deviate (wrong color, deformation, structural errors); product identity is unstable.
- 1–24: Severely distorted or unrecognizable; unusable.

Output format (strict JSON, no text outside):

```
{"scores": {"bg_layout": <int>, "text_correctness": <int>, "product_fidelity":
<int>, "total_score_100": <float>}, "prompt_revision": {"advice": "<concise,
actionable paragraph for next-iteration improvement>"}}
```

F More Generated Examples

Figure 8 presents additional poster examples generated by our full PosterDuet pipeline across diverse product categories. These results demonstrate the generalization capability of our system in handling various visual styles, layout structures, and promotional copy.

G Limitations

OCR-RL yields diminishing returns on dense small-font Chinese text; stroke-level fidelity for rare glyphs remains an open diffusion challenge. What's more, the current pipeline handles single-product posters; multi-product composite layouts (e.g., category-level promotional banners) are not yet supported.



Figure 8: More poster examples generated by PosterDuet.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state three contributions (closed-loop Designer–Renderer framework, CTR preference model, and TSRO algorithm) that are all supported by experimental results in Sections 4–5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations section is provided in Appendix G, discussing OCR-RL bottlenecks on dense text and single-platform CTR model limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results (theorems or proofs). All mathematical content describes model architectures and optimization objectives.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Full model architectures, training hyperparameters, dataset sizes, and optimization details are provided in Section 4.1 and Appendix B. The alternating optimization algorithm is detailed in Algorithm 1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The training data is collected from a commercial e-commerce platform and cannot be publicly released due to business confidentiality. We plan to release model code upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details including learning rates, epochs, GPU configurations, dataset sizes, and data filtering criteria are provided in Section 4.1 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the high computational cost of full-pipeline RL training (requiring 16×A800 GPUs), we do not report error bars over multiple runs. However, we provide 7-day online A/B test results showing consistent daily gains, which demonstrate the stability of our approach.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: GPU types (H200, A800) and counts are specified for each training stage in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. All data is collected from a commercial platform with proper authorization, and human annotators are compensated fairly.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses positive impacts (reducing manual design labor, improving e-commerce efficiency) in the introduction. Potential negative impacts (e.g., generating misleading advertisements) are mitigated by the CTR model being trained on genuine user engagement signals rather than deceptive patterns.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The models are designed specifically for e-commerce poster generation and are not released publicly at this stage. The system generates commercial posters from product images, posing minimal risk for misuse beyond its intended domain.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All pre-trained models (Qwen3-VL, Qwen-Image-Edit, SigLIP2) and tools (PaddleOCR) are properly cited. The training data is proprietary and collected with platform authorization.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new public assets (datasets or model checkpoints) at this stage.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The human evaluation protocol is described in Appendix B.3, including the number of annotators, evaluation criteria, randomization procedure, and annotator qualifications.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The human evaluation involves professional annotators providing preference judgments on generated posters, which poses no risk to participants and does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper uses a VLM (Qwen3-VL-8B) as the core Designer component and GPT-5.4 as the offline evaluation judge. Both usages are clearly described in Sections 3.2 and 4.1.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.