

Mirror in the Model: Ad Banner Image Generation via Reflective Multi-LLM and Multi-modal Agents

Zhao Wang^{1*}, Bowen Chen^{1,2*}, Yotaro Shimose^{1*},

Sota Moriyama^{1,3}, Heng Wang¹, Shingo Takamatsu¹

¹Sony Group Corporation, Japan

²The University of Tokyo, Japan

³The Graduate University for Advanced Studies, Japan

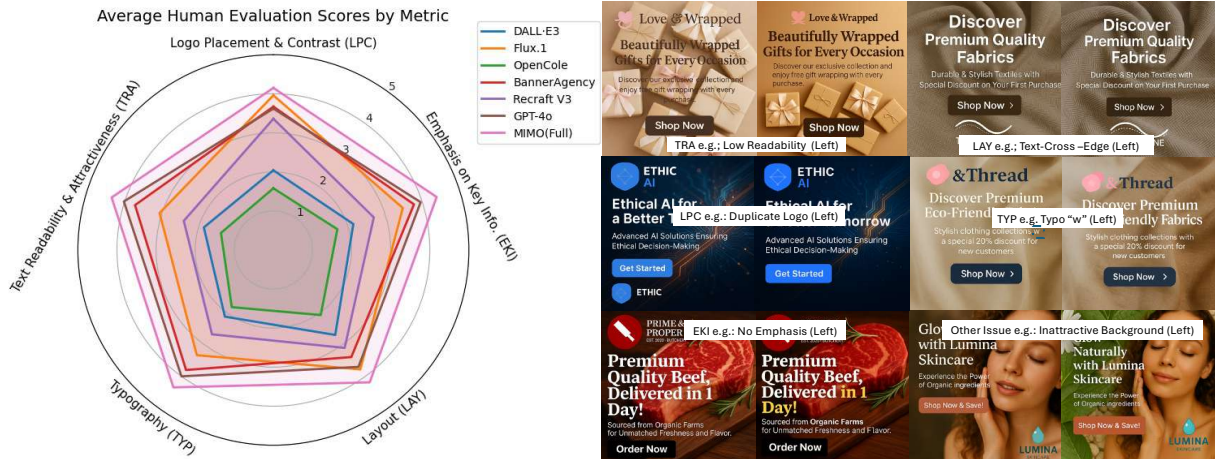


Figure 1: **Left:** Radar chart showing human evaluations across baselines and our **MIMO** for six major design criteria (evaluated on 20 samples), all of which are critical factors in the commercialization of banner ads. **Right:** Six examples of common errors alongside corresponding MIMO refinement. Even the strongest baseline, GPT-4o, produces minor yet critical flaws that are unacceptable in commercial applications.

Abstract

Recent generative models such as GPT-4o have shown strong capabilities in producing high-quality images with accurate text rendering. However, commercial design tasks like advertising banners demand more than visual fidelity—they require structured layouts, precise typography, consistent branding and etc. In this paper, we introduce **MIMO (Mirror In-the-Model)**, an agentic refinement framework for automatic ad banner generation. MIMO combines a hierarchical multi-modal agent system (MIMO-Core) with a coordination loop (MIMO-Loop) that explores multiple stylistic directions and iteratively improves design quality. Requiring only a simple natural language based prompt and logo image as input, MIMO automatically detects and corrects multiple types of errors during generation. Experiments show that MIMO significantly outper-

forms existing diffusion and LLM-based baselines in real-world banner design scenarios. ¹

1 Introduction

Recent advances in generative modeling have greatly improved image synthesis quality. Text-to-image generative models such as DALL-E (OpenAI, 2023) and Flux.1 series (Labs, 2024) produce high-fidelity visuals. However, these models still face problems such as instruction adherence and text rendering. Hybrid approaches such as OpenCOLE (Inoue et al., 2024) and BannerAgency (Wang et al., 2025a) have proposed using text-to-image generative models for the background, while employing LLMs to control the layout and attributes of foreground elements such as typography to overcome the weakness of text-to-image generative models. The most recently multimodal LLMs such as GPT-4o (OpenAI, 2025)

* indicates equal contribution. Bowen Chen and Sota Moriyama completed this work during their internship at Sony. Corresponding author: Zhao.Wang@sony.com

¹The code of this research is at <https://github.com/sony/mimo>

achieve superior semantic–visual alignment (Zhang et al., 2025), outperforming existing text-to-image models like Recraft V3 (Recraft Inc., 2024) and Flux.1 Pro (Labs, 2024) in character-level accuracy, instruction adherence, and text rendering.

However, commercial tasks such as poster or advertising-banner design demand more than visual fidelity—they require structured layout, precise typography, and consistent brand styling. To assess whether existing models meet these standards, we conducted a pretest by manually annotating 20 samples from each baseline (Flux.1 Kontext Max (Labs, 2025), Recraft V3 (Recraft Inc., 2024), OpenCOLE (Inoue et al., 2024), and BannerAgency (Wang et al., 2025a)). We tracked frequent design errors, including *Logo Placement & Contrast(LPC)*, *Emphasis on Key Info.(EKI)*, *Text Readability & Attractiveness(TRA)*, *Typography(TYP)* and *Layout(LAY)*. As illustrated in Fig. 1, these baselines exhibit diverse and recurring failure modes, highlighting their limitations for real-world advertising applications. Some visualized failures are shown in Fig. 4 in Appendix A.

Recent breakthroughs in agentic, self-refinement frameworks—such as SELF-REFINE, which enables LLMs to provide feedback and iteratively refine outputs (Madaan et al., 2023), and TalkHier, a hierarchical multi-agent feedback system for structured task correction (Wang et al., 2025b)—highlight the potential of feedback-driven, step-by-step generation refinement. In the vision domain, Idea2Img (Yang et al., 2024) leverages GPT-4V to iteratively edit and refine images through prompt generation, selection, and feedback loops. COLE (Jia et al., 2024) fine-tunes LLaVA (Liu et al., 2023) to provide visual feedback for refining typographic layout and attributes. However, neither method adopts an agentic workflow that actively refines the intermediate visual output in a structured and goal-driven manner.

Motivating Question: *Can we design an agentic module within banner generation models that automatically detects and corrects the visual errors shown at Figure 1 while also enhancing overall design quality?*

This leads to our proposed **MIMO (Mirror In-the-Model)** architecture. MIMO automatically and continuously inspects intermediate banner drafts, identifies flawed regions—such as misaligned text, missing logos, or inconsistent style—and iteratively refines them, effectively mir-

roring expert human design review until the output meets professional advertising standards. Specifically, MIMO introduces four key innovations:

1. **MIMO-Core:** a hierarchical, multimodal multi-agent system based on LLMs, which dynamically and autonomously refines visual elements through iterative design–evaluate–revise loops.
2. **MIMO-Loop:** a high-level coordination layer that simultaneously initiates multiple MIMO-Core instances to explore different stylistic directions. It adopts a multi-agent voting protocol to identify and discard poor designs, while enabling information sharing across cores to boost overall design quality.
3. **Minimal User Burden:** MIMO requires only a simple natural language prompt and a logo image as user input. All refinement, decision-making, and stylistic exploration are handled automatically—greatly lowering the barrier for commercial deployment.

2 Methodology

We consider the task of generating high-quality ad banner images for a product R , given a user-issued prompt u and a logo image $\ell \in \mathbb{R}^{H \times W \times 3}$. The prompt u serves as the primary input for ad generation and may range from a minimal description (e.g., a few keywords like “summer sale” or “premium skincare”) to more detailed advertiser intent or constraints (e.g., “include 30% OFF,” “show product clearly,” “make it premium”), depending on user preference. The logo ℓ must be faithfully embedded into the final design output.

MIMO aims to generate a set of n stylistically diverse ad banner images $\{x_1, x_2, \dots, x_n\}$, where each $x_i \in \mathbb{R}^{H' \times W' \times 3}$ satisfies both the semantic intent encoded in u and the visual branding constraint enforced by ℓ . MIMO requires only a simple natural language prompt u and a logo image ℓ as user input. All refinement, decision-making, and stylistic exploration are handled automatically—greatly lowering the barrier for commercial deployment. Figure 2 illustrates the overall structure of the MIMO architecture.

2.1 MIMO-Core: Dynamic Multi-modal, Multi-agents Generate-Refine Pipeline

The **MIMO-Core** subsystem (light blue area in Figure 2) is a hierarchical multi-agent pipeline that generates a single high-quality banner image through iterative creation, evaluation, and revision.

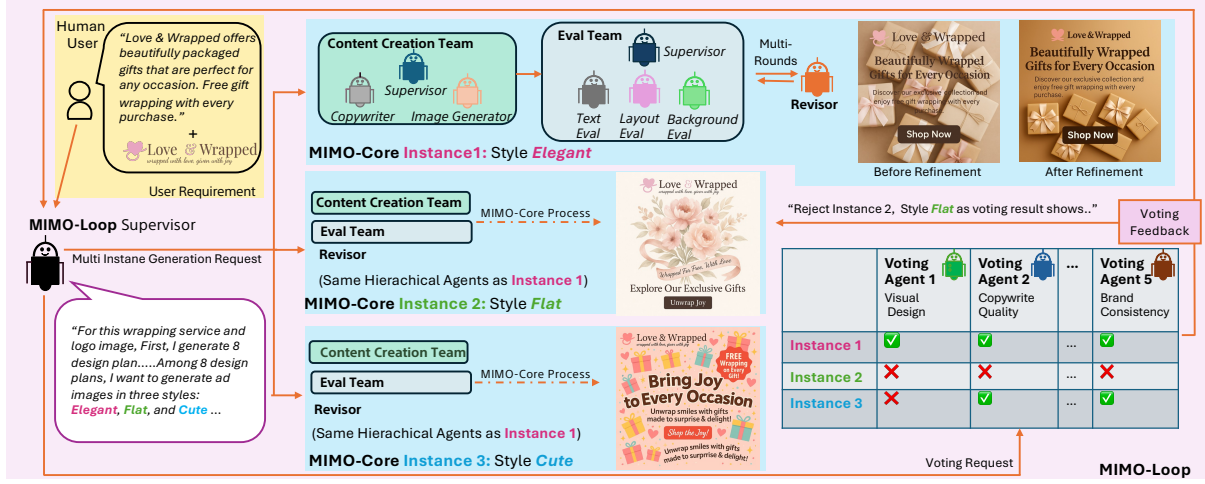


Figure 2: System overview of our proposed MIMO framework. **MIMO-Core** handles a dynamic refinement process including content generation, evaluation, and revision via a hierarchy of agents. **MIMO-Loop** wraps around the core to introduce style prompting and multi-agent voting, and iterative refinement for diversity-aware generation.

It mimics the structure of human design teams, where agent roles are divided across content creation, quality evaluation, and targeted refinement.

Agent Structure. We define the agent set:

$$\mathcal{A}_{\text{core}} = \{\mathcal{A}_{\text{sup}}, \mathcal{A}_{\text{create}}, \mathcal{A}_{\text{eval}}, \mathcal{A}_{\text{rev}}\}$$

with:

- $\mathcal{A}_{\text{sup}}$: MIMO-Core Supervisor.
- $\mathcal{A}_{\text{create}} = \{\mathcal{A}_{\text{cs}}, \mathcal{A}_{\text{cw}}, \mathcal{A}_{\text{ig}}, \mathcal{A}_{\text{lay}}\}$: Content Creation Team with a supervisor and three agents: copywriter, image generator, and layout planner.
- $\mathcal{A}_{\text{eval}} = \{\mathcal{A}_{\text{es}}, \mathcal{A}_{\text{text}}, \mathcal{A}_{\text{bg}}, \mathcal{A}_{\text{lay}}\}$: Evaluation Team with a supervisor and agents for text, background, and layout critique.
- $\mathcal{A}_{\text{rev}}$: Graphic Revisor Agent.

Interaction Protocol. At each step t , the system maintains a banner draft $x^{(t)} \in \mathbb{R}^{H' \times W' \times 3}$ and shared context memory $\mathcal{M}^{(t)}$. The supervisor selects an active subteam:

$$\mathcal{A}_{\text{active}}^{(t)} = \text{Route}(x^{(t)}, \mathcal{M}^{(t)})$$

which performs one of three actions:

- **Create:** Generate an initial draft via $\mathcal{A}_{\text{create}}$.
- **Evaluate:** Return textual feedback via $\mathcal{A}_{\text{eval}}$.
- **Revise:** Refine banner draft to incorporate feedback via $\mathcal{A}_{\text{rev}}$.

Evaluation agents produce feedback messages:

$$f^{(t)} = \{(a_i, \text{comment}_i)\}_{i=1}^k, \quad a_i \in \mathcal{A}_{\text{eval}}$$

which are aggregated by the Evaluation Team Supervisor and routed to $\mathcal{A}_{\text{sup}}$. The supervisor then determines whether to re-enter the creation phase, initiate revision, or perform another evaluation.

Multimodal Reasoning. All agents operate over multimodal input:

$$(u, \ell, x^{(t)}, \mathcal{M}^{(t)})$$

where u is the prompt, ℓ is the logo image, and $x^{(t)}$ is the current visual draft. Agents reason not only over text but also over rendered designs, enabling grounded, perceptual feedback (e.g., “CTA overlaps background” or “font lacks emphasis”).

This iterative loop continues:

$$x^{(t)} \rightarrow \text{Eval} \rightarrow \text{Revise} \rightarrow x^{(t+1)}$$

until the banner is deemed complete by the supervisor $\mathcal{A}_{\text{sup}}$.

2.2 MIMO-Loop: Outer-Loop for Style Diversity and Robust Selection

The **MIMO-Loop** subsystem builds on MIMO-Core by introducing a diversity-enhancing outer loop that explores multiple stylistic directions and iteratively refines them through agent-based evaluation and elimination. It increases the robustness and expressiveness of the final ad banner while preserving brand fidelity and user intent.

Style Prompting. Given a logo-prompt pair (ℓ, u) , a language model generates k distinct stylistic

prompts:

$$\mathcal{S} = \{s_1, s_2, \dots, s_k\}$$

Each style s_i reflects attributes such as product category, color palette, and branding tone. A style selection agent selects a subset of n styles:

$$\mathcal{S}^* = \{s_{i_1}, s_{i_2}, \dots, s_{i_n}\} \subseteq \mathcal{S}$$

based on the compatibility between style descriptions and the visual-semantic content of (ℓ, u) .

Parallel MIMO-Core Execution. Each selected style prompt $s_j \in \mathcal{S}^*$ is fed into an independent MIMO-Core instance, resulting in a set of n refined banner outputs:

$$\begin{aligned} \mathcal{X}^{(0)} &= \{x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\} \\ x_j^{(0)} &\leftarrow \text{MIMO-Core}(u, \ell, s_j) \quad \text{for } s_j \in \mathcal{S}^* \end{aligned}$$

Multi-Agent Judging. Let $\mathcal{J} = \{\text{Judge}_1, \dots, \text{Judge}_5\}$ denote five specialized JudgeAgents, each evaluating every $x_j^{(t)} \in \mathcal{X}^{(t)}$ with respect to a distinct criterion (e.g., visual design, copywriting quality, brand consistency, user experience, or technical fidelity). Each judge issues a tuple:

$$r_j^{(t)} = (\text{vote}, \text{reason})$$

$\text{vote} \in \{\text{RECOMMENDED}, \text{REJECTED}\}, \quad \text{reason} \in \mathcal{T}$

Feedback Aggregation and Elimination. Let $\#\text{REJECTED}(x_j^{(t)})$ denote the number of judges who vote REJECTED for candidate $x_j^{(t)}$. We eliminate the candidate with the highest rejection count, i.e., $\mathcal{X}^{(t+1)} = \mathcal{X}^{(t)} \setminus \{x_{j^*}^{(t)}\}$, where $j^* = \arg \max_j \#\text{REJECTED}(x_j^{(t)})$.

Iterative Refinement Loop. This process continues iteratively:

$$\mathcal{X}^{(t)} \xrightarrow{\text{Judge}} \text{Feedback} \xrightarrow{\text{Refine}} \mathcal{X}^{(t+1)}$$

until $|\mathcal{X}^{(t)}| = 1$, at which point the final banner $x^* \in \mathcal{X}^{(t)}$ is selected as the winner.

The complete procedural steps of both the MIMO-Core and MIMO-Loop pipelines—including agent initialization, hierarchical routing, style diversification, and iterative refinement—are detailed in Algorithm 1 and Algorithm 2 in Appendix A.

Logo and Product Image Authenticity. To ensure the authenticity of logos and product images in the generated banners, we use external inpainting tools (Recraft Inc., 2024) for manual replacement. Please refer to Appendix D for further details.

3 Experimental Result

Datasets. We use the *BannerRequest400* dataset from the BannerAgency (Wang et al., 2025a), which contains 100 real-world advertising prompts with product metadata, logo images, and human-designed banners. The dataset spans categories such as electronics, cosmetics, and home goods. Each sample includes a product name, marketing description, campaign constraints, and logo—supporting evaluation of both text-to-image generation and layout-aware design.

Baselines. We compare MIMO framework with five baselines covering latest text-to-image models, multi-modal LLM and hybrid approaches.

- **DALL·E 3** (OpenAI, 2023): A proprietary text-to-image generative model that produces images from natural language prompts.
- **Flux.1 Kontext Max** (Labs, 2025): The latest flow-matching model in the Flux.1 family. It offers enhanced prompt adherence and controllability.
- **Recraft V3** (Recraft Inc., 2024): A commercial text-to-image model specialized in text rendering.
- **OpenCole** (Inoue et al., 2024): An open-source hybrid system that combines the background generation of diffusion models with the reasoning ability of language models.
- **BannerAgency** (Wang et al., 2025a): A multi-modal LLM agent framework that generates advertisement banners using a static agent workflow.
- **GPT-4o** (OpenAI, 2025): OpenAI’s latest multimodal model capable of generating and editing images from natural language prompts with strong semantic-visual alignment.

Implementation Details. For the commercial models (e.g., DALL·E 3, Recraft V3, Flux.1 Kontext Max), we use their public API to generate outputs under default or recommended settings. Prompt design for each baseline model is detailed in Appendix I. We use GPT-4o as the backbone LLM for MIMO, with the decoding temperature fixed at 0 across all experiments. MIMO is implemented using the LangGraph (LangChain, 2024). All the prompts we used in MIMO are detailed in Appendix F, G and I.

Tool Usage. MIMO only employs two tools:

- **MultiModal I/O Tool:** Interfaces with the GPT-4o vision model for both image generation and visual input. It enables agents to perceive and analyze images directly, which is critical for layout and design tasks.
- **Output Tool:** A simple tool used to save internal

Method	TAA	LPS	CTAE	CPYQ	BIS	AQS	LPC	EKI	LAY	TYP	TRA
DALL-E3	3.17 ^{0.19}	1.78 ^{0.21}	4.18 ^{0.20}	2.48 ^{0.18}	3.06 ^{0.22}	3.10 ^{0.20}	3.10 ^{0.35}	2.95 ^{0.40}	3.25 ^{0.38}	3.00 ^{0.42}	3.05 ^{0.39}
Flux.1	3.36 ^{0.28}	3.67 ^{0.26}	4.28 ^{0.25}	2.64 ^{0.27}	3.09 ^{0.25}	3.12 ^{0.26}	4.54 ^{0.70}	3.96 ^{1.72}	4.27 ^{1.13}	3.99 ^{1.78}	4.00 ^{1.67}
OpenCole	2.10 ^{0.20}	1.20 ^{0.22}	2.30 ^{0.21}	2.40 ^{0.20}	2.25 ^{0.19}	2.50 ^{0.18}	2.85 ^{0.41}	2.70 ^{0.45}	2.90 ^{0.44}	2.60 ^{0.50}	2.75 ^{0.48}
BannerAgency	3.30 ^{0.45}	3.10 ^{0.42}	3.40 ^{0.40}	3.25 ^{0.43}	3.20 ^{0.41}	3.35 ^{0.44}	3.96 ^{1.74}	3.64 ^{1.40}	3.80 ^{1.21}	3.83 ^{1.17}	3.80 ^{1.15}
Recraft V3	3.85 ^{0.29}	3.75 ^{0.27}	3.90 ^{0.26}	3.72 ^{0.28}	3.80 ^{0.30}	3.88 ^{0.27}	4.46 ^{0.93}	3.86 ^{1.95}	4.07 ^{1.79}	3.88 ^{2.01}	3.90 ^{1.90}
GPT-4o	4.15 ^{0.30}	4.10 ^{0.28}	4.20 ^{0.27}	4.10 ^{0.29}	4.15 ^{0.30}	4.18 ^{0.28}	4.97 ^{0.04}	4.90 ^{0.14}	4.88 ^{0.19}	4.81 ^{0.46}	4.84 ^{0.26}
MIMO-Core(-)	4.48 ^{0.21}	4.41 ^{0.20}	4.29 ^{0.19}	4.37 ^{0.22}	4.50 ^{0.18}	4.53 ^{0.20}	4.95 ^{0.06}	4.93 ^{0.08}	4.92 ^{0.08}	4.89 ^{0.09}	4.90 ^{0.08}
MIMO-Core	4.62 ^{0.18}	4.60 ^{0.19}	4.45 ^{0.20}	4.58 ^{0.21}	4.66 ^{0.22}	4.71 ^{0.19}	4.97 ^{0.05}	4.96 ^{0.06}	4.96 ^{0.05}	4.95 ^{0.06}	4.95 ^{0.05}
MIMO(Full)	4.83 ^{0.17}	4.81 ^{0.18}	4.65 ^{0.19}	4.70 ^{0.18}	4.78 ^{0.19}	4.88 ^{0.18}	4.98 ^{0.03}	4.97 ^{0.04}	4.97 ^{0.04}	4.96 ^{0.05}	4.96 ^{0.04}

Table 1: Comparison of average metric scores across methods (higher is better). **MIMO-Core(-)** = MIMO without refinement; **MIMO-Core** = MIMO with refinement; **MIMO(Full)** = MIMO with refinement and loop.

thoughts or logs from agents. It does not return any output, and is mainly for memory tracking during agent communication.

Evaluation Metrics. We evaluate image quality using the following six criteria proposed by BannerAgency(Wang et al., 2025a) for fair comparison including: *TAA* (*Target Audience Alignment*), *LPS* (*Logo Placement Score*), *CTAE* (*Call-to-Action Effectiveness*), *CPYQ* (*Copy Quality*), *BIS* (*Brand Integration Score*) and *AQS* (*Aesthetic Quality Score*).

To check if MIMO can handle the problems mentioned in Sec. 1, we designed the following metrics: *Logo Placement & Contrast (LPC)*: Checks if the logo is easy to see, clearly placed, and stands out from the background. It does not evaluate the authenticity of the logo, as most baseline methods do not support this functionality.

Emphasis on Key Information (EKI): Looks at whether important messages like discounts or key features are highlighted clearly.

Layout (LAY): Evaluates the arrangement of elements, including spacing and alignment, for a clean and balanced design.

Typography (TYP): Checks if the font style, size, and consistency make the text easy to read and visually appealing.

Text Readability & Attractiveness (TRA):

Measures how easy the text is to read and whether it looks good within the design.

3.1 Comparison with Baselines

Table 1 compares the average metric scores of various methods across 13 evaluation criteria. Overall, our proposed MIMO variants outperform all baselines. Among them, MIMO(Full) achieves the highest scores across all metrics, demonstrating the effectiveness of both refinement and loop-based optimization. The MIMO-Core version, which includes refinement but excludes the loop mechanism, still performs strongly and consistently surpasses

MIMO-Core(-) (the variant without refinement), highlighting the critical role of iterative revision. Notably, while GPT-4o performs competitively in aesthetic and layout-related metrics, it falls short in structured design elements such as Logo Placement (LPC) and Emphasis on Key Information (EKI), where MIMO excels. Other baseline models perform poorly across most dimensions, particularly in structured or brand-sensitive attributes. These results underscore the importance of integrating hierarchical refinement and adaptive feedback loops for generating commercially viable ad creatives. A detailed visualized comparison with baselines is shown in Fig 4 in Appendix A. Visualization of MIMO-Core Refinement and MIMO-Loop Stylishment are shown in Fig 5 and 6 in Appendix B.

3.2 Human Evaluation

Vision-language models often struggle to fully understand image content, as noted in (Rahmanzadehgervi et al., 2025). BannerAgency (Wang et al., 2025a) showed a strong correlation between human evaluations and GPT-4o-based evaluations using their proposed metrics. To verify the reliability of GPT-4o-based evaluation on our new metrics, we collected scores from 12 human evaluators on 20% of the prompts. Table 2 shows the average human scores, which align well with machine-based evaluations, achieving a Spearman’s rank correlation of **0.85**. MIMO outperforms all baselines, confirming the validity of our evaluation method.

While GPT-based evaluations generally rank models correctly, human evaluators reveal performance gaps more clearly. For instance, weaker models like OpenCole and DALL-E3 receive much lower scores from humans compared to stronger models such as Flux.1 or BannerAgency. In contrast, GPT-4o scores tend to be closer across models. This suggests that human evaluation offers more precise and detailed insights, especially for

Method	LPC	EKI	LAY	TYP	TRA
DALL-E3	2.04	2.15	2.69	2.11	1.87
Flux.1	3.96	3.48	3.78	3.33	3.06
OpenCole	1.58	1.72	2.06	1.81	1.41
BannerAgency	3.67	3.78	3.39	3.80	3.72
Recraft V3	3.36	2.70	3.10	2.67	2.41
GPT-4o	3.62	3.96	3.70	4.00	4.02
MIMO(Full)	4.15	4.39	4.19	4.35	4.35

Table 2: Average metric scores from human evaluators

capturing qualitative differences in generation quality.

3.3 Ablation Study: Single- vs. Multi-Agent

To assess the importance of decomposing the ad generation process into multiple agents, we conducted a controlled ablation study comparing a single-agent setup against MIMO-Core framework. The single-agent setting processes all requirements via a single, comprehensive prompt (including all detailed prompts in multi-agent based MIMO-Core setting to ensure fair comparison), while the multi-agent system distributes the generation and evaluation tasks across specialized roles. Each configuration was evaluated over three independent runs. Table 3 shows the averaged metric scores and standard deviations across the three runs. This result highlights the limitation of the single-agent setup: simply packing all requirements into one prompt yields weaker performance. In contrast, the multi-agent MIMO framework assigns tasks to specialized agents in stages—MIMO-Core handles layout, copywriting, and evaluation separately, while MIMO-Loop further refines style—leading to more accurate and higher-quality outputs.

3.4 Ablation Study: Number of Evaluation Agents in the MIMO-Loop

In MIMO-Loop, we deploy five evaluation agents to vote on the quality of banners generated from different styles. We examine how performance is affected by the number of evaluation agents, the number of styles, and the importance of each evaluation criterion.

As shown in Figure 3, performance improves as the number of evaluation agents increases. Using

Setting	TAA	LPS	CTAE	CPYQ	BIS	AQS
Single	4.01 ^{0.09}	4.05 ^{0.07}	3.89 ^{0.10}	3.85 ^{0.11}	4.29 ^{0.07}	4.01 ^{0.09}
Multi	4.37 ^{0.18}	4.59 ^{0.06}	4.77 ^{0.17}	4.66 ^{0.04}	4.76 ^{0.13}	4.79 ^{0.05}

Table 3: Ablation Study: GPT-4o Scores for Single-Agent vs. Multi-Agent (Mean^{Std} across 3 runs)

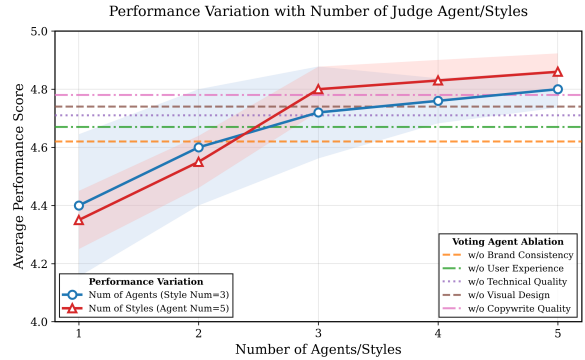


Figure 3: We measure average performance by varying the number of agents or styles (solid lines), while keeping the other fixed. We also run an ablation study by removing one voting agent at a time (dotted lines).

only one agent results in lower and more variable scores, suggesting that a single agent cannot provide a comprehensive assessment. Adding more agents not only improves performance but also reduces variance, leading to more consistent results. We also observe that increasing the number of styles boosts performance, with a larger gain from 1 to 3 styles than from 3 to 5. Finally, in the voting-agent ablation study, we find that removing the Brand Consistency agent causes the most significant drop in performance, underscoring the importance of preserving logo integrity. In contrast, removing the Copywriting Quality agent has minimal impact.

4 Related Work

Image Generation & Ad Design. Text-to-image models such as DALL-E 3 (OpenAI, 2023), Flux.1 Pro (Labs, 2024) and Recraft V3 (Recraft Inc., 2024) achieve high visual fidelity but often struggle with structured layout, typography accuracy, and brand-style coherence—especially in banner or poster designs. Multimodal LLM-driven frameworks, including OpenCOLE (Inoue et al., 2024), AutoPoster (Lin et al., 2023), and BannerAgency (Wang et al., 2025a), improve text rendering and blueprint creation but still exhibit recurring flaws in alignment, emphasis, and brand consistency, as our pretest and Figure 1 confirm.

Agentic Refinement. Text-only systems like SELF-REFINE (Madaan et al., 2023) use an LLM to generate feedback and make iterative revisions without extra training. TalkHier (Wang et al., 2025b) builds on this by introducing a structured, multi-agent setup with communication rules, shared context, and unbiased evaluation.

tion. Idea2Img (Yang et al., 2024) leverages GPT-4V to iteratively edit and refine images through prompt generation, selection, and feedback loops. COLE (Jia et al., 2024) fine-tunes LLaVA (Liu et al., 2023) to provide visual feedback for refining typographic layout and attributes.

Limitations

While MIMO delivers substantial improvements in banner quality, layout fidelity, and brand alignment via multi-agent coordination, it incurs a higher computational cost. Compared to GPT-4o’s one-shot generation (\$0.21 per image), our full pipeline (MIMO-Core + Loop) may cost up to \$2.87 per banner. A detailed cost breakdown is presented in Appendix E. Considering that a visually compelling banner can significantly boost user engagement and conversion, we leave it to practitioners and advertisers to judge whether this cost–quality trade-off is justified in their specific use cases.

Another limitation is the reliance on a manual logo and product image replacement process in this paper. We found that this is the only way to ensure 100% authenticity of the logo and product image. Automatic replacement methods, such as mask-based inpainting, often fail to preserve the fine details of the original logo and product design. These methods can unintentionally distort elements such as shadows, gradients, or positioning, resulting in visually inconsistent or incorrect integration within the banner. Therefore, to maintain brand integrity, we adopt a manual approach for the replacement.

References

- Inc. Flux.1 AI. Ai image prompt collection. <https://flux1.ai/prompts>. Accessed: 2025-06-20.
- Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. 2024. *Opencole: Towards reproducible automatic graphic design generation*. *Preprint*, arXiv:2406.08232.
- Peidong Jia, Chenxuan Li, Yuhui Yuan, Zeyu Liu, Yichao Shen, Bohan Chen, Xingru Chen, Yinglin Zheng, Dong Chen, Ji Li, Xiaodong Xie, Shanghang Zhang, and Baining Guo. 2024. *Cole: A hierarchical generation framework for multi-layered and editable graphic design*. *Preprint*, arXiv:2311.16974.
- Black Forest Labs. 2024. Flux. <https://github.com/black-forest-labs/flux>.
- Black Forest Labs. 2025. Flux.1-kontext max. <https://bfl.ai/models/flux-kontext>. Accessed: 2025-06-20.
- LangChain. 2024. Langgraph: State machine for llm applications. <https://github.com/langchain-ai/langgraph>. Accessed: 2025-06-26.
- Jinpeng Lin, Min Zhou, Ye Ma, Yifan Gao, Chenxi Fei, Yangjian Chen, and et al. 2023. Autoposter: A highly automatic and content-aware design system for advertising poster generation. In *ACM Multimedia*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. *Visual instruction tuning*. *Preprint*, arXiv:2304.08485.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- OpenAI. 2023. DALL-E 3 System Card. <https://cdn.openai.com/papers/dall-e-3.pdf>.
- OpenAI. 2025. Gpt-4o report. <https://openai.com/ja-JP/index/introducing-4o-image-generation/>. Accessed: 2025-06-09.
- Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. 2025. *Vision language models are blind: Failing to translate detailed visual features into words*. *Preprint*, arXiv:2407.06581.
- Recraft. 2025. How to craft prompts for accurate, ai-generated images. <https://recraft.ai/blog/>. Accessed: 2025-06-20.
- Recraft Inc. 2024. Recraft v3. <https://www.recraft.ai>. Web-based text-to-image model emphasizing long-text rendering, anatomical correctness, and top-rank performance on design-quality leaderboards.
- Heng Wang, Yotaro Shimose, and Shingo Takamatsu. 2025a. Banneragency: Advertising banner design with multimodal llm agents. *arXiv*.
- Zhao Wang, Sota Moriyama, Wei-Yao Wang, Briti Gangopadhyay, and Shingo Takamatsu. 2025b. Talk structurally, act hierarchically: A collaborative framework for llm multi-agent systems. *arXiv preprint arXiv:2502.11098*.
- Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2024. *Idea2img: Iterative self-refinement with gpt-4v(ision) for automatic image design and generation*. *Preprint*, arXiv:2310.08541.

Tianyu Zhang, Xinyu Wang, Zhenghan Tai, Lu Li, Jijun Chi, Jingrui Tian, Hailin He, and Suyuchen Wang. 2025. Strict: Stress test of rendering images containing text. *arXiv preprint arXiv:2505.18985*. Evaluates GPT-4o, Gemini 2.0, Recraft V3, FLUX 1.1 Pro on text-rendering accuracy and instruction compliance.

A Algorithm of MIMO-Core and MIMO-Loop; Visualized Comparison with Baselines

Algorithm 1 presents the full procedure of the MIMO-Core pipeline, which governs the generation of a high-quality banner image through iterative collaboration among hierarchical multi-agent teams. The algorithm reflects dynamic team selection, internal routing within content creation and evaluation subteams, and multimodal feedback refinement. Each agent acts over shared memory (default setting in Langraph (LangChain, 2024))and rendered visual drafts, and the loop continues until the supervisor determines that the output meets quality standards.

Algorithm 1: MIMO-Core: Dynamic Multimodal Generate-Refine Pipeline

Input: Prompt u , Logo image ℓ
Output: Final banner image $x^{(T)}$

- 1 Initialize shared memory $\mathcal{M}^{(0)} \leftarrow \{u, \ell\}$;
- 2 Initialize visual draft $x^{(0)} \leftarrow \emptyset$, step $t \leftarrow 0$;
- 3 Initialize agent hierarchy::
- 4 MIMO-Core Supervisor $\mathcal{A}_{\text{sup}}$;
- 5 Content Creation Team $\mathcal{A}_{\text{create}} = \{\mathcal{A}_{\text{cs}}, \mathcal{A}_{\text{cw}}, \mathcal{A}_{\text{ig}}, \mathcal{A}_{\text{layout}}\}$;
- 6 Evaluation Team $\mathcal{A}_{\text{eval}} = \{\mathcal{A}_{\text{es}}, \mathcal{A}_{\text{text}}, \mathcal{A}_{\text{bg}}, \mathcal{A}_{\text{lay}}\}$;
- 7 Graphic Revisor Agent $\mathcal{A}_{\text{rev}}$;
- 8 **while** $\mathcal{A}_{\text{sup}}$ decides not terminated **do**
- 9 Supervisor $\mathcal{A}_{\text{sup}}$ selects active team: $\mathcal{A}_{\text{team}}^{(t)} \leftarrow \text{Route}(x^{(t)}, \mathcal{M}^{(t)})$;
- 10 **if** $\mathcal{A}_{\text{team}}^{(t)} = \mathcal{A}_{\text{create}}$ **then**
- 11 Team supervisor $\mathcal{A}_{\text{cs}}$ selects agent(s)::
- 12 $\mathcal{A}_{\text{create}}^{(t)} \leftarrow \text{Route}_{\text{create}}(\mathcal{M}^{(t)})$;
- 13 Active creation agent(s) generate content::
- 14 $x^{(t+1)} \leftarrow \text{Create}(u, \ell, \mathcal{M}^{(t)})$;
- 15 **else if** $\mathcal{A}_{\text{team}}^{(t)} = \mathcal{A}_{\text{eval}}$ **then**
- 16 Team supervisor $\mathcal{A}_{\text{es}}$ selects agent(s)::
- 17 $\mathcal{A}_{\text{eval}}^{(t)} \leftarrow \text{Route}_{\text{eval}}(x^{(t)})$;
- 18 Active evaluation agents return feedback::
- 19 $f^{(t)} \leftarrow \text{Evaluate}(x^{(t)}, \mathcal{M}^{(t)})$;
- 20 Update shared memory::
- 21 $\mathcal{M}^{(t+1)} \leftarrow \mathcal{M}^{(t)} \cup f^{(t)}$;
- 22 **else if** $\mathcal{A}_{\text{team}}^{(t)} = \mathcal{A}_{\text{rev}}$ **then**
- 23 Graphic Revisor applies revisions::
- 24 $x^{(t+1)} \leftarrow \text{Revise}(x^{(t)}, f^{(t)}, \mathcal{M}^{(t)})$;
- 25 $t \leftarrow t + 1$;
- 26 **return** $x^{(T)}$;

Algorithm 2 presents the full procedure of the MIMO-Loop subsystem, which builds on MIMO-Core to enhance design diversity and robustness. It begins by generating multiple style prompts based on the initial input (u, ℓ) , then selects a subset of stylistically distinct directions. Each style prompt is processed independently by a parallel instance of MIMO-Core. The resulting banners are evaluated by a panel of JudgeAgents, who issue both a vote and a reason. The most-rejected banner is iteratively removed, and feedback is passed to the corresponding MIMO-Core agents for refinement. This elimination-and-refinement loop continues until a single final banner is selected.

Algorithm 2: MIMO-Loop: Style Diversity and Iterative Selection

Input: Prompt u , Logo image ℓ , total style pool size k , number of selected styles n

Output: Final banner image x^*

```

1 Generate  $k$  stylistic prompts:
    
$$\mathcal{S} \leftarrow \{s_1, s_2, \dots, s_k\}$$


2 Select a subset of  $n$  styles:
    
$$\mathcal{S}^* \leftarrow \text{SelectStyles}(\mathcal{S}, u, \ell)$$


3 Initialize style candidates via parallel MIMO-Core calls:;
4 foreach  $s_j \in \mathcal{S}^*$  do
5    $x_j^{(0)} \leftarrow \text{MIMO-Core}(u, \ell, s_j)$ ;
6  $\mathcal{X}^{(0)} \leftarrow \{x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\}$ , step  $t \leftarrow 0$ ;
7 while  $|\mathcal{X}^{(t)}| > 1$  do
8   JudgeAgents evaluate each  $x_j^{(t)} \in \mathcal{X}^{(t)}$  and issue:
        
$$r_j^{(t)} = (\text{vote}, \text{reason})$$

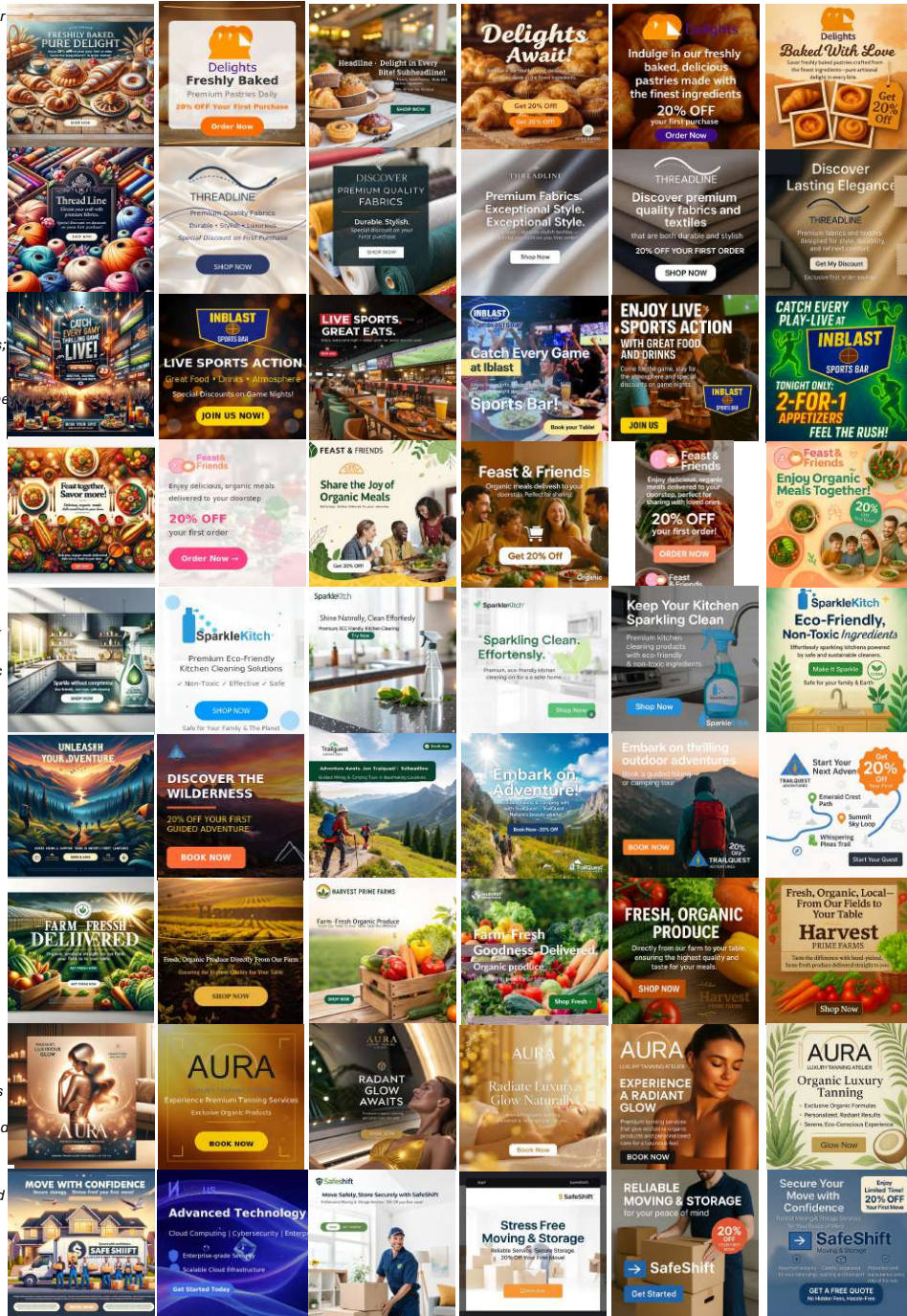

9   Count REJECTED votes for each candidate;
10  Identify index:
        
$$j^* \leftarrow \arg \max_j \# \text{REJECTED}(x_j^{(t)})$$


11  Eliminate worst candidate:
        
$$\mathcal{X}^{(t+1)} \leftarrow \mathcal{X}^{(t)} \setminus \{x_{j^*}^{(t)}\}$$


12  Pass feedback to MIMO-Core instance  $j^*$  for refinement;
13   $t \leftarrow t + 1$ ;
14 return final banner  $x^* \in \mathcal{X}^{(t)}$ 

```

1. "Delights. Indulge in our freshly baked, delicious pastries made with the finest ingredients; enjoy a special 20% discount on your first purchase!"
2. "Threadline. Discover premium quality fabrics and textiles that are both durable and stylish, with a special discount on your first purchase."
3. "Inblast Sports Bar. Enjoy live sports action with great food and drinks; come for the game, stay for the atmosphere and special discounts on game nights."
4. "Feast & Friends. Enjoy delicious, organic meals delivered to your doorstep, perfect for sharing with loved ones. Get 20% off your first order!"
5. SparkleKitch. SparkleKitch offers premium kitchen cleaning products. Highlight the eco-friendly and non-toxic ingredients for a safe and effective cleaning experience.
6. TrailQuest Adventures. Embark on thrilling outdoor adventures with TrailQuest's guided hiking and camping tours. Book now and get 20% off your first adventure!
7. "Harvest Prime Farms. Fresh, organic produce directly from our farm to your table, ensuring the highest quality and taste for your meals."
8. "AURA Luxury Tanning Atelier. Experience premium tanning services with exclusive organic products and personalized care for a luxurious feel."
9. "SafeShift. SafeShift offers reliable moving and storage services to ensure your belongings are transported securely and stored safely, now with a 20% discount on your first move."



Prompt

DALLE E3

BannerAgency

Recraft V3

Flux .1 Kontext Max

GPT4o

Our MIMO

Figure 4: Visual comparison between our method and baseline models. Even the strongest baseline—GPT-4o, the latest image generation model—exhibits notable failures across various prompts: poor layout design (Prompt 6), misleading ad text (Prompt 2, where “20% off” is omitted despite being specified), typographical errors (e.g., “ecclusive” in Prompt 8), low logo contrast (Prompts 1 & 7), incorrect aspect ratio (Prompt 4), and prompt misinterpretation (Prompt 9, where a storage-moving service is mistaken for luggage moving). In contrast, our proposed MIMO framework—combining multi-agent refinement (MIMO-Core) with stylistic enhancement (MIMO-Loop)—consistently produces visually appealing and prompt-faithful banner images.

B Visualization of MIMO-Core Refinement and MIMO-Loop Stylishment

Figure 5 illustrates the iterative refinement process of the MIMO-Core pipeline. Starting from an initial ad draft generated by the content creation team, the system undergoes multiple evaluation and revision cycles. Each round addresses specific issues—such as poor logo visibility, unbalanced layout, or weak CTA emphasis—based on textual feedback from evaluation agents. The resulting improvement highlights the effectiveness of hierarchical refinement in producing commercially viable designs.

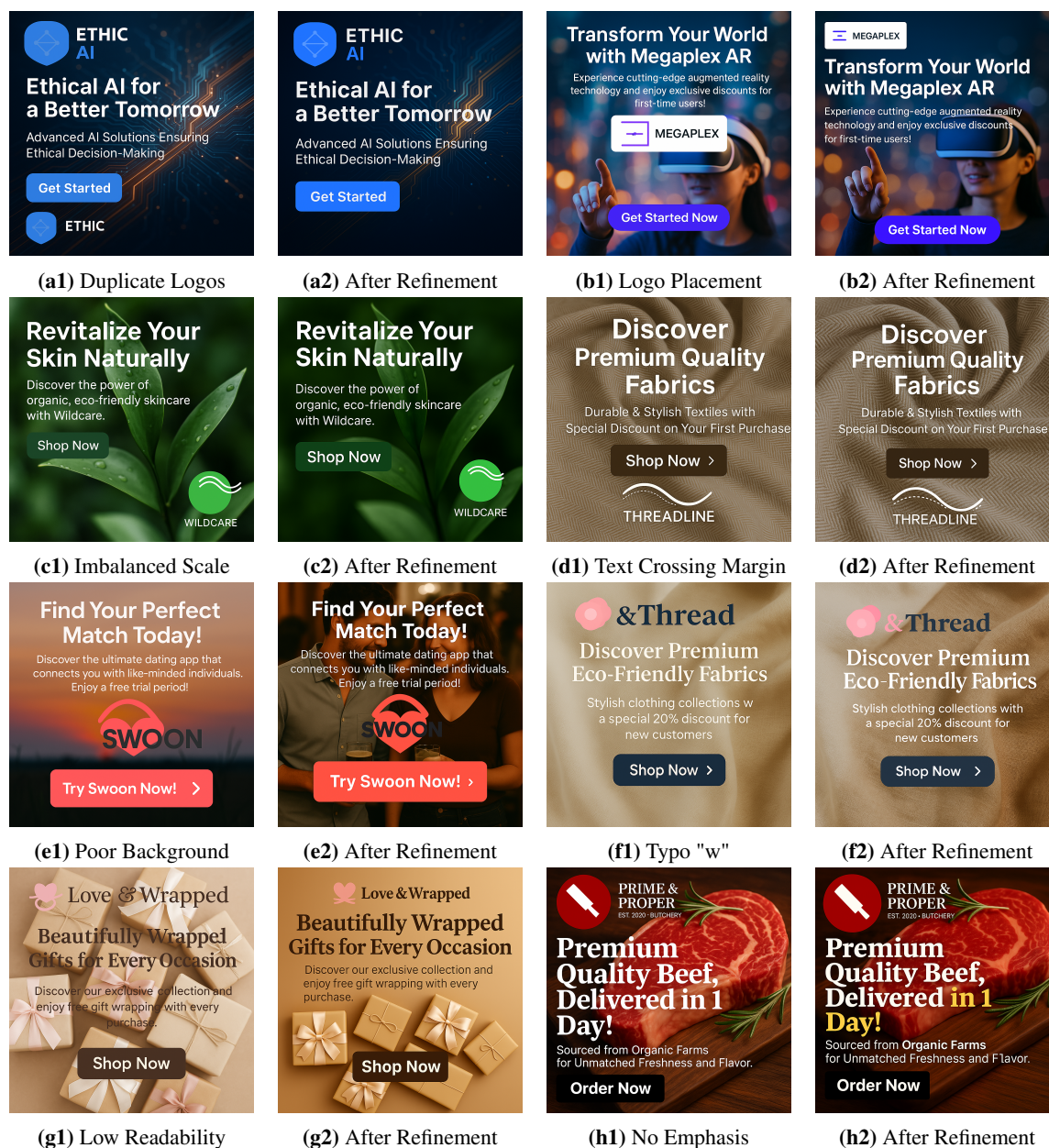


Figure 5: Visualization of our system’s automatic refinements. It effectively corrects subtle but important issues in ad designs, including logo placement, layout balance, background relevance, and text clarity.

Figure 6 showcases the stylistic diversity generated by MIMO-Loop. For a single product prompt and logo, multiple design styles are instantiated in parallel, producing banners that vary in tone, layout, and visual aesthetics (e.g., minimalist vs. vibrant, modern vs. traditional). This visualization demonstrates MIMO-Loop’s ability to generate style-aligned outputs while retaining core brand identity and campaign intent. The comparison further shows how different stylistic framings can shift emphasis and design focus, providing valuable choices for downstream selection and human feedback.

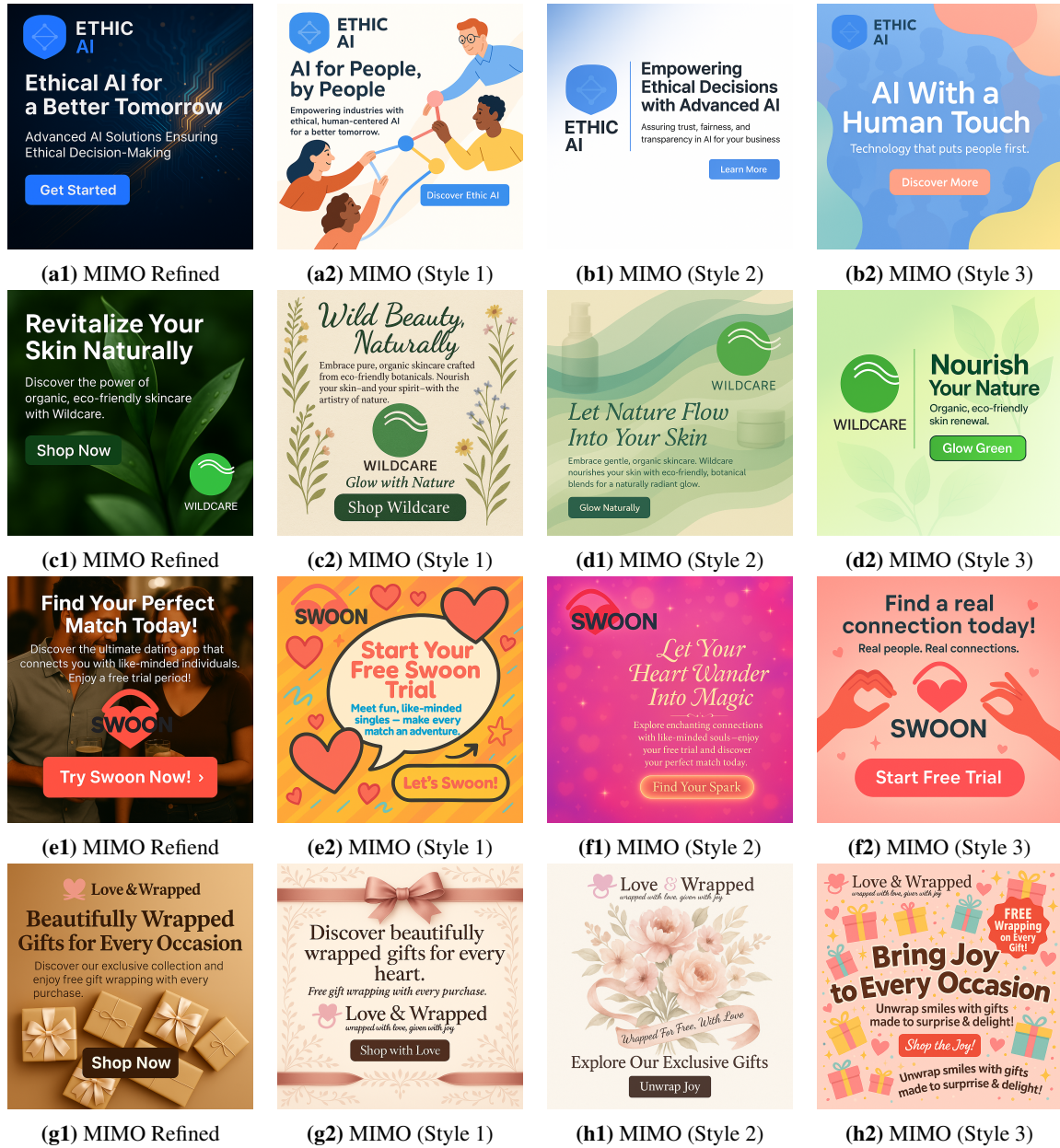


Figure 6: Visualization of MIMO-Loop with the different style of ad banners.

C Hierarchical Multi-Agent Team Prompt for Ad Image Generation

To support the multi-agent setting in our system, we designed a hierarchical agent framework where the task is decomposed into content creation, evaluation, and revision teams. Each team includes specialized agents responsible for different aspects of the ad generation pipeline, and the system follows a coordination protocol governed by return signals (e.g., FINISH, Default) and revision limits.

The agent teams are organized as follows:

- **ContentCreationTeam:** Includes Copywriter and ImageResearcher for generating textual and visual elements.
- **EvaluationTeam:** Includes TextContentEvaluator, LayoutEvaluator, and BackgroundImageEvaluator, each focusing on a single evaluation dimension.
- **GraphicRevisor:** Revises the ad banner based on the structured feedback from EvaluationTeam.

The system supports a maximum of three image revisions. Once this threshold is reached, the system terminates with the return signal "FINISH".

Below is the full hierarchical agent prompt structure used in our implementation:

Hierarchical Agent Prompt Structure

```
{
  "team": {
    "team": "Default",
    "return": "FINISH",
    "prompt": "Make a **single** catchy square AD banner image for {item}.
      Make sure that there is a **CTA button icon** and **logo** in the banner.
      Furthermore, make sure that the banner stands out when small, and has a good background image.
      If the Ad banner image has already been revised multiple times, ensure at most 3 revisions are the limitations.
      If over 3 revisions, just return 'FINISH' and stop the process.
      Final output MUST be in the form of: {\\"images\\": [...], \\"instructions\\": [...], \\"current_content\\": [...]}.",
    "additional_prompt": "Ensure the AD banner is visually appealing and the text is persuasive.
      You must evaluate the image before reporting back to FINISH.
      Furthermore, make sure to remind the evaluators that there is a revised image whenever there is one.",
    "ContentCreationTeam": {
      "team": "ContentCreation",
      "return": "Default",
      "prompt": "Create the text content and find high-quality images for the product we are advertising.
        You only need to generate 1 image and the text to go with it.
        If there is a reference image, you need to pass the path to Image Researcher.",
      "additional_prompt": "Ensure the texts you created and images are cohesive and high-quality,
        and the path to the image is properly returned.",
      "Copywriter": {
        "prompt": "You are a Copywriter responsible for creating the text content for the AD banner,
          including the headline, subheadline, and CTA text.
          You can use tools to find information about the product.
          CTA text must be short, catchy and concise.
          Do not generate untrue, misleading or incorrect information.",
        "tools": [4]
      },
      "ImageResearcher": {
        "prompt": "You are an Image Researcher responsible for finding high-quality images of the product.
          Follow the instructions from the supervisor.
          You need to pass detailed ad image generation request to the tool.
          Only generate 1 image at a time.
          Only 1 logo in the image.
          All text must be high contrast and visible.",
        "tools": [4, 12]
      }
    },
    "EvaluationTeam": {
      "team": "Evaluation",
      "return": "Default",
      "prompt": "Review the text content and image design of the AD banner for clarity, persuasiveness, correctness, and visual appeal.
        You must evaluate the text and image separately.
        If nothing is wrong, say 'No changes needed'.
        If over 3 revisions, return 'FINISH'.",
      "additional_prompt": "Check whether the image is revised.
        When given a revised image path, evaluate again.",
    }
  }
}
```



```

    "TextContentEvaluator": {
      "prompt": "You are a Text Content Evaluator responsible for reviewing the text for clarity, persuasiveness, and correctness.  

        Make sure the text is **visible and readable**.  

        Must use tools to check the text rendered in the image.  

        Only focus on text (not layout/background).",
      "tools": [4, 12]
    },

    "LayoutEvaluator": {
      "prompt": "You are a Layout Evaluator responsible for reviewing element placement and overall visual flow.  

        Give layout-specific suggestions only (not text or background).",
      "tools": [4, 12]
    },

    "BackgroundImageEvaluator": {
      "prompt": "You are an Image Evaluator responsible for judging if the background is suitable.  

        Give suggestions on better imagery if needed.  

        Only focus on background (not text/logo).",
      "tools": [4, 12]
    }
  },

  "GraphicRevisor": {
    "prompt": "You are a Graphic Revisor responsible for modifying the AD image per evaluation feedback.  

      Use the most recently generated image.  

      Give clear instructions to the editing tool.",
    "tools": [4, 12]
  }
},
"intermediate_output_desc": "Dictionary format.",
"int_out_format": "Dictionary"
}

```

D Inpainting for Logo and Product Image Replacement



Figure 7: Inpainting example for logo replacement.

In this section, we describe how to post-process the generated banner image for the logo and product image inpainting to guarantee the logo’s correctness.

For a generated banner image, while we input the logo and product image for generation, due to the nature of generative models, the correctness of them in the banner image is not guaranteed. This creates a fatal weakness in using generative models for real-world business banner image generation, as the logo generated through iterations may not accurately capture the details of the original logo and product image. However, for a banner image, a wrong logo and product image banner is certainly not usable. To address this issue, we use inpaint to do the replacement to ensure the their correctness.

We show an example in the Figure 7. For this figure, the first generated final banner image has a wrong logo which is not the same with the actual one. We first ask an inpaint model to inpaint the original logo to fit the background image. After that, we insert the original transparent logo at the same inpaint area so that the size can fit with the original logo design generated by the MIMO method. In this way, we are able to replace the generated logo and product image with the actual ones while ensuring the original design from the MIMO is kept.

E Token and Cost Analysis

We analyze the token usage and cost of our MIMO framework compared to a GPT-4o baseline, based on the latest Azure OpenAI pricing for GPT-4o’s multimodal model. Text input and output tokens are billed at \$0.00004 and \$0.00008 per token, respectively. For image generation, GPT-4o consumes approximately 1,105 output tokens per image at 512×512 resolution and medium detail, resulting in a per-image cost of \$0.0884.

GPT-4o Baseline performs one-shot banner generation using approximately 1,000 text tokens (500 input, 500 output) and one image, for a total cost of:

- Text: $(500 \times \$0.00004) + (500 \times \$0.00008) = \$0.06$
- Image: \$0.0884
- **Total: \$0.15**

MIMO-Core⁻ (a simplified one-round version) uses about 4,000 text tokens (2,000 input, 2,000 output) and one image:

- Text: $(2,000 \times \$0.00004) + (2,000 \times \$0.00008) = \$0.24$
- Image: \$0.0884
- **Total: \$0.33**

MIMO-Core includes three rounds of generation, evaluation, and refinement, totaling approximately 10,800 text tokens (5,400 input, 5,400 output) and three images:

- Text: $(5,400 \times \$0.00004) + (5,400 \times \$0.00008) = \$0.65$
- Image: $3 \times \$0.0884 = \0.265
- **Total: \$0.91**

MIMO-Core + Loop, which runs three parallel MIMO-Core branches, a judging phase (five agents $\times$ three candidates), and one final refinement, consumes about 42,000 text tokens (21,000 input, 21,000 output) and four images:

- Text: $(21,000 \times \$0.00004) + (21,000 \times \$0.00008) = \$2.52$
- Image: $4 \times \$0.0884 = \0.3536
- **Total: \$2.87**

While MIMO introduces higher computational costs (ranging from 2–20 $\times$ over the baseline), it consistently yields higher-quality banners, stronger visual fidelity, and more robust multi-agent evaluations—justifying its use in high-stakes advertising scenarios.

F LLM Evaluation Prompt

To enable structured and consistent assessments of banner quality, we follow (Wang et al., 2025a) a unified evaluation prompt for LLM-based comparison. Given two ad images from the same campaign, the model is instructed to rate and justify each image across six critical design dimensions: audience alignment, logo visibility, CTA effectiveness, copywriting quality, brand consistency, and overall aesthetics. Each criterion is scored on a 0–5 scale with detailed reasoning, and the final judgment is returned in a standardized JSON format for programmatic evaluation.

LLM Evaluation Prompt

Compare two ad images that are from the same ad campaign. You must choose which is better. Explain why the first or second ad image is better using the following six criteria. For each metric, rate the first and second image on a scale from 0 (very poor) to 5 (excellent), and explain the reasoning for each rating.

The ratings should be based on the following scale:

"0": "Very poor – The element is missing, unreadable, or completely fails in its purpose. For example, no CTA, broken layout, or irrelevant audience targeting."

"1": "Poor – The element exists but is confusing, hard to notice, or seriously misaligned with goals. For instance, logo is tiny and placed in a non-visible area, or the message is unclear."

"2": "Fair – The element is somewhat effective but has noticeable problems. For example, the CTA is present but lacks emphasis, or the visuals are cluttered or off-brand."

"3": "Good – The element is mostly effective with minor issues. For example, the layout works but the font size is slightly small, or the copy is clear but could be more persuasive."

"4": "Very good – The element is clear, aligned with the goal, and visually sound. It may need very small refinements, such as slight enlargement of text or better spacing."

"5": "Excellent – The element is perfect in both design and function. No changes are needed. It meets its communication goal effectively and aligns with professional visual standards."

Return your response in the following JSON format:

```
{
  "TAA": { "image_1_score": 0, "image_1_reason": "Your explanation here.", "image_2_score": 0, "image_2_reason": "Your explanation here." },
  "LPS": { "image_1_score": 0, "image_1_reason": "Your explanation here.", "image_2_score": 0, "image_2_reason": "Your explanation here." },
  "CTAE": { "image_1_score": 0, "image_1_reason": "Your explanation here.", "image_2_score": 0, "image_2_reason": "Your explanation here." },
  "CPYQ": { "image_1_score": 0, "image_1_reason": "Your explanation here.", "image_2_score": 0, "image_2_reason": "Your explanation here." },
  "BIS": { "image_1_score": 0, "image_1_reason": "Your explanation here.", "image_2_score": 0, "image_2_reason": "Your explanation here." },
  "AQS": { "image_1_score": 0, "image_1_reason": "Your explanation here.", "image_2_score": 0, "image_2_reason": "Your explanation here." }
}
```

Evaluation Metric Descriptions:

- TAA (Target Audience Alignment): How well the ad aligns with the theme, audience, and purpose.
- LPS (Logo Placement Score): Visibility, size, and placement of the logo.
- CTAE (Call-to-Action Effectiveness): Clarity and visual emphasis of the CTA.
- CPYQ (Copy Quality): Clarity, grammar, and persuasiveness of the headline and other text.
- BIS (Brand Integration Score): Visual and stylistic alignment with the brand identity (beyond logo).
- AQS (Aesthetic Quality Score): Overall visual appeal, color harmony, balance, and typography.

In addition to pairwise LLM-based comparisons, we design a six-way evaluation protocol tailored to our structured ad design assessment needs. This prompt asks the model to evaluate six ad images from the same campaign across six granular criteria: logo placement, emphasis on key information, layout, typography, text readability, and background relevance. Each metric is scored on a scale from 1 (unacceptable) to 5 (commercial-ready), with justifications provided for every image and criterion. This comprehensive and fine-grained evaluation aligns with our goal of supporting scalable SSA (Sponsored Search Advertising) design diagnostics and quality control at the campaign level.

Six-Way Multi-Criteria Evaluation Prompt

Compare six ad images that are from the same ad campaign.

You must assign scores to all six images using the following six criteria:

1. **Logo Placement & Contrast (LPC)** – Checks logo visibility, sizing, and whether it contrasts with the background.
2. **Emphasis on Key Information (EKI)** – Assesses if key content like discounts or product names are visually emphasized.
3. **Layout (LAY)** – Evaluates the structure, spacing, and alignment of visual elements.
4. **Typography (TYP)** – Considers font clarity, consistency, hierarchy, and branding alignment.
5. **Text Readability & Attractiveness (TRA)** – Rates clarity and aesthetic of text given contrast and composition.

Scoring Guidelines:

- 1 – Critical error: unusable in commercial settings (e.g., unreadable text, missing logo).
- 3 – Moderate issues: usable with revisions (e.g., poor layout or weak visual emphasis).
- 5 – Strong: commercially ready with clean, clear, and goal-aligned design.

Provide your response in JSON format, assigning a score and a reason for each image per criterion.

G Single-Agent Prompt Used in Ablation Study

The following prompt was used in the single-agent setting during the ablation study (Section 4.X). It compresses all responsibilities into a single instruction and serves as a baseline to evaluate the benefit of role decomposition in our multi-agent design.

Single-Agent Prompt for Ad Image Generation

I need you to make a **single** catchy square AD banner image for {item}.

Make sure that there is a **CTA button icon** and **logo** in the banner. Furthermore, make sure that the banner stands out when small, and has a good background image.

Ensure the AD banner is visually appealing and the text is persuasive.

Creating the text content for the AD banner, including the headline, subheadline, and CTA text. CTA text used in CTA button must be short, catchy, and concise. Do not generate untrue, misleading, or incorrect information.

Only generate 1 image at a time. Only generate 1 logo in the image. All text (including logo text, CTA text, text in the image, etc.) in the image needs to be high contrast and visible.

Ensure the text content is clear, persuasive, and correct. Make sure the text in the image is **visible and readable** with no typos.

Text must be high-contrast enough to the background image.

Review the layout of the AD banner for proper placement of elements and overall effectiveness for an AD image.

The background image needs to be suitable. Make sure to give pointers to what you think is a suitable background image, if any adjustments are required.

H Style Prompting

To enable diverse and brand-aligned banner generation, we introduce a style prompting mechanism as the first stage of the MIMO-Loop framework. This stage leverages an LLM to infer multiple stylistic directions given the advertiser’s prompt and logo.

Stylized Prompt Set. Given a logo–prompt pair (ℓ, u) , a language model generates k stylistic descriptions:

$$\mathcal{S} = \{s_1, s_2, \dots, s_k\}$$

where each style prompt s_i is a short textual description representing a unique visual theme (e.g., minimalist, futuristic, or elegant). These styles are conditioned on logo characteristics (e.g., shape, color palette) and prompt semantics (e.g., product category, tone).

Style Selection. Out of the k generated styles, a style selection agent chooses a subset $\mathcal{S}^*$ of n promising candidates:

$$\mathcal{S}^* = \{s_{i_1}, s_{i_2}, \dots, s_{i_n}\} \subseteq \mathcal{S}$$

The selection is based on compatibility between the prompt–logo pair (ℓ, u) and each stylistic candidate s_i , aiming to preserve brand alignment while encouraging stylistic diversity.

Style Prompting Instruction. The instruction used to query the LLM for diverse styles is provided below.

Style Prompting Instruction

Given the following product information and brand logo description, generate **five distinct style prompts** that describe alternative visual directions for designing an ad banner.

Each style prompt should be:

- A concise one-sentence description (10–20 words).
- Visually grounded: refer to layout, mood, color palette, or design theme.
- Coherent with the brand tone and product characteristics.
- Diverse: avoid overlapping ideas or redundant phrasing across styles.

Use the product information and logo cues to infer appropriate stylistic themes. For example, if the logo is elegant and monochrome, one style could be “minimalist black-and-white luxury layout”; if the product is tech-related, another might be “futuristic neon over dark background.”

Product Description: [Insert here]

Logo Description: [Insert here]

Return your response as: { "style_1": "...", "style_2": "...", "style_3": "...", "style_4": "...", "style_5": "..." }

I Baseline prompt design

Text-to-image generative models often require carefully designed prompts to produce high-quality outputs. To bridge the gap between advertising-specific prompts and the formats expected by different generative models, we implemented an agent, T2IPromptGenerator, which automatically reformulates advertising prompts into model-compatible inputs. The agent operates in a few-shot manner, guided by a system prompt that provides instructions along with several examples of the desired prompt format to align outputs with user intent.

System Prompt for T2IPromptGenerator

You are an image prompt generator for web advertisement. Given the user’s request about the image or explanation about their product, you will generate a detailed prompt for the image generation API.

When generating the prompt, make sure:

- There is a **CTA button icon** and **logo** in the banner.
- Include text contents for the AD banner, including the headline, subheadline, and CTA text.
- CTA text used in CTA button must be short catchy and concise
- The banner stands out when small
- The banner has a good background image

Here are some examples of prompts:

- Example 1
- Example 2
- Example 3

We picked prompt examples from the AI Image Prompt Collection ([Flux.1 AI](#)) for Flux, the official prompt crafting guide ([Recraft, 2025](#)) for Recraft V3, and the intention-to-design-plan dataset ([Inoue](#)

et al., 2024) for OpenCOLE. For DALL-E 3, as we could not find a suitable source of examples, we manually crafted them.