

# BannerAgency: Advertising Banner Design with Multimodal LLM Agents

Heng Wang<sup>†</sup>, Yotaro Shimose, Shingo Takamatsu

Sony Group Corporation

{heng.wang, yotaro.shimose, shingo.takamatsu}@sony.com

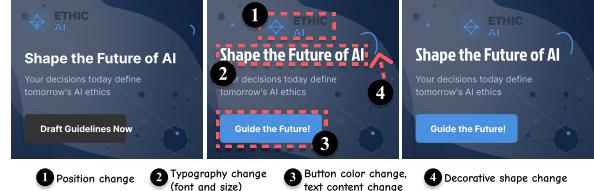
## Abstract

Advertising banners are critical for capturing user attention and enhancing advertising campaign effectiveness. Creating aesthetically pleasing banner designs while conveying the campaign messages is challenging due to the large search space involving multiple design elements. Additionally, advertisers need multiple sizes for different displays and various versions to target different sectors of audiences. Since design is intrinsically an iterative and subjective process, flexible editability is also in high demand for practical usage. While current models have served as assistants to human designers in various design tasks, they typically handle only segments of the creative design process or produce pixel-based outputs that limit editability. This paper introduces a training-free framework for fully automated banner ad design creation, enabling frontier multimodal large language models (MLLMs) to streamline the production of effective banners with minimal manual effort across diverse marketing contexts. We present BannerAgency, an MLLM agent system that collaborates with advertisers to understand their brand identity and banner objectives, generates matching background images, creates blueprints for foreground design elements, and renders the final creatives as editable components in Figma or SVG formats rather than static pixels. To facilitate evaluation and future research, we introduce BannerRequest400, a benchmark featuring 100 unique logos paired with 400 diverse banner requests. Through quantitative and qualitative evaluations, we demonstrate the framework’s effectiveness, emphasizing the quality of the generated banner designs, their adaptability to various banner requests, and their strong editability enabled by this component-based approach.

Method	DALL-E3	OpenCOLE	BannerAgency
Sample			
Multimodal Input	✗	✗	✓
Size Variation	✗	✗	✓
Brand Logo:			

✗: not supported    ○: partially supported.    ✓: supported.

(a) Left: multimodal banner request. Right: Algorithm comparison.



(b) An example of element-level editing.

Figure 1: Algorithm comparison and BannerAgency’s editability in banner ad image generation.

## 1 Introduction

Advertising banners constitute an instrumental medium in digital marketing campaigns. These graphical displays efficiently capture viewer attention, communicate strategic messaging, and enhance brand visibility across competitive market environments (Robinson et al., 2007; Landa, 2016; Chen et al., 2021; Vaddamanu et al., 2022). Unlike generic images, banner ad images are visually harmonized compositions of multiple creative assets including backdrops, brand-related logos or product images, click-to-action (CTA) buttons, campaign texts, and other decorative elements such as shapes and patterns (Maheshwari et al., 2019; Vempati et al., 2020). Campaign texts further require typography design choices, contributing to a vast search space that demands significant time and effort (Du et al., 2024; Wang et al., 2022). In addition, advertisers have various size requirements for different display needs (Bannerwise, 2022) and

<sup>†</sup>Corresponding author: Heng Wang (heng.wang@sony.com, hengwang.vision@gmail.com)

diverse design requirements tailored to specific target audiences and purposes (community, 2025). As design remains inherently creative and subjective, ensuring human interaction and editing capability is crucial for practical applications (Guo et al., 2021; Jia et al., 2023).

Current models accelerate this banner design process by automating separate aspects: product background generation (Wang et al., 2022; Yang et al., 2024a; Fong and See, 2024; Du et al., 2024; Chen et al., 2025), template-based layout matching (Yang et al., 2016; Express, 2025; Canva, 2025; Alibaba, 2020; Li and Yang, 2023), layout generation (Hu et al., 2021; Vempati et al., 2020; Zhang et al., 2017; Maheshwari et al., 2019; Lin et al., 2025), and automatic typography design (Choi et al., 2018; Iluz et al., 2023; Park et al., 2024). Advanced scene-text rendering models (Chen et al., 2024; Liu et al., 2024; Tuo et al., 2024; Zhao et al., 2024) and text-to-image (T2I) models such as DALL-E3 (Betker et al., 2023) are suboptimal for banner design due to their pixel-based generation paradigm, which produces outputs that are difficult to edit and often contains incoherent text as in Fig. 1a. Although TypeR (Shimoda et al., 2024) addresses the "typos" from T2I models through detection and correction of incorrectly rendered text, it still constrains the design process to a pixel-based approach, limiting subsequent editability. Recent works (Jia et al., 2023; Inoue et al., 2024; Wang et al., 2024) explore automatic poster generation from prompts, but banner design requires handling both structured requests and unstructured brand identity like logos. Their data dependency and fine-tuning further limit adaptability to diverse banner sizes and requests. Although these works allow text modification through typography rendering, they merge decorative elements with images, restricting editability (see Fig.1a, where COLE-based(Jia et al., 2023; Inoue et al., 2024) output's black rectangle cannot be removed).

To bridge these gaps and deliver a fully automatic yet editable banner design solution, we introduce BannerAgency, a training-free framework powered by multimodal large language models (MLLMs) (OpenAI, 2024; Anthropic, 2024; Liu et al., 2023). Our fundamental idea is to utilize MLLMs as AI agents (Wu et al., 2024; Zhang et al., 2024; Islam et al., 2024; Qian et al., 2024) that simulate how a human design team works collaboratively from conception to implementation (community, 2025), leveraging the emergent design knowl-

edge inherited from web-scale training data (Lin et al., 2023a; Huang et al., 2024; Wang et al., 2025). With tool calling and memory sharing capabilities of LLM agents (Gupta and Kembhavi, 2023; Lu et al., 2024; Hu et al., 2024; Xu et al., 2024), we effectively navigate the extensive design search space through our specialized agent architecture. The system comprises: (1) a *Strategist* that constrains design possibilities by establishing brand guidelines and campaign objectives; (2) a *Background Designer* responsible for visual backdrop generation; (3) a *Foreground Designer* that creates blueprints for banner assets with engaging content and harmonized styling and positioning; and (4) a *Developer* that transforms these specifications into editable components via Figma plugin code or SVG generation. Unlike pixel-based methods that produce rasteric outputs, BannerAgency generates designs directly within professional tools, enabling seamless editing of major design components—text, typography, buttons, shapes, and layouts—within a familiar interface used by professional designers as illustrated in Fig. 1b.

To catalyze further exploration to advertising banner creation, we introduce BannerRequest400, a dedicated benchmark featuring 100 uniquely created logos paired with 400 diverse banner requests spanning multiple target audiences and primary purposes. Both quantitative and qualitative assessments confirm BannerAgency's capabilities in generating high-quality banner designs that effectively adapt to various requirements while offering unparalleled editing flexibility through its component-based output format.

Our contributions can be summarized as follows:

- We propose BannerAgency, a training-free MLLM agent system that processes minimal multimodal inputs (brand logos, advertiser requirements) and autonomously generates all necessary creative assets as harmonized banner designs.
- Simulating professional design workflows—from strategy development to technical implementation, BannerAgency leverages MLLMs' extensive world knowledge to navigate the complex design space while ensuring complete editability through component-based implementation in design tools such as Figma or as SVG format.
- To catalyze research in agent-driven advertis-

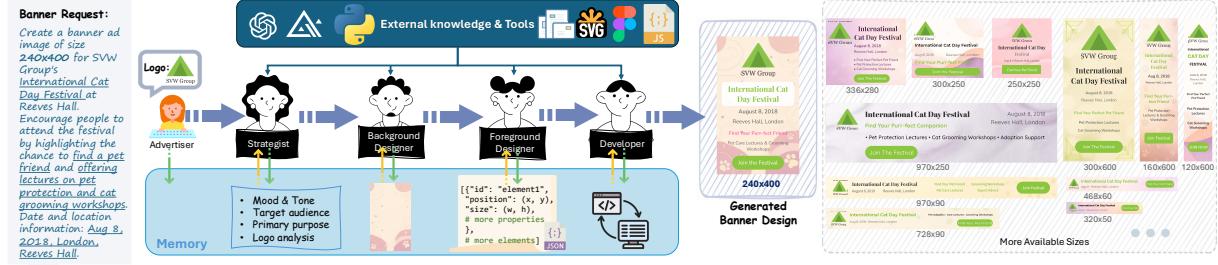


Figure 2: **BannerAgency overview.** Given a logo and request, BannerAgency begins with the *Strategist* analyzing banner objectives, followed by the *Background Designer* creating matching backgrounds, then the *Foreground Designer* producing element blueprints, and concludes with the *Developer* rendering the final design as editable components. With access to external knowledge, tool-calling capabilities, and shared memory, BannerAgency enables context-aware, harmonized decisions and supports multiple banner sizes if requested.

ing banner design, we introduce BannerRequest400, a curated benchmark featuring 100 uniquely created logos paired with 400 diverse banner requests spanning multiple target audiences and primary purposes.

- Through comprehensive evaluation, we exhibit BannerAgency’s effectiveness in generating high-quality, versatile, and editable banner designs that can serve either as final outputs or as starting points for subjective refinement.

## 2 Related Work

### 2.1 Graphic Design Generation

Current approaches to graphic design generation face significant limitations. Pixel-based text-to-image models (Betker et al., 2023; BlackForestLab, 2024; Rombach et al., 2022; Zhang et al., 2023) require cumbersome editing, while layout generation research (Zheng et al., 2019; O’Donovan et al., 2014; Lee et al., 2020; Lin et al., 2024a; Inoue et al., 2023b,a; Shabani et al., 2024; Cheng et al., 2024; Lin et al., 2023b; Iwai et al., 2024; Weng et al., 2024; Yun et al., 2025) has not adequately addressed simultaneous element generation and arrangement (Kikuchi et al., 2024). Existing solutions in prompt-based poster generation (Jia et al., 2023; Inoue et al., 2024; Wang et al., 2024) rely on domain-specific training, accept only unimodal text input (not multimodal inputs like logos), and lack flexibility. While LayoutPrompt (Lin et al., 2023a) offers a training-free approach, it addresses only layout without end-to-end capabilities. We propose a training-free agent framework that leverages MLLMs to simultaneously generate and arrange design elements in editable code representation from multimodal inputs, enabling flexible banner creation without domain-specific fine-tuning.

### 2.2 Multimodal LLM Agents

MLLMs (Liu et al., 2023; Zhu et al., 2024; Abdin et al., 2024; Li et al., 2023) have recently spurred an emerging wave of multimodal agent research (Durante et al., 2024; Xie et al., 2024; He et al., 2024; Lin et al., 2024b; Li et al., 2024; Koh et al., 2024). These agents—autonomous entities that leverage tool-calling capabilities, planning, memory, and goal-directed reasoning—have been applied to various design domains including presentation slides generation (Zheng et al., 2025; Ge et al., 2025), image manipulation (Wang et al., 2025; Yang et al., 2024b), infographic design (Huang et al., 2024; Chen and Huang, 2024), and sketch creation (Vinker et al., 2024). Our work extends this evolving frontier by leveraging MLLMs’ emergent capabilities to pioneer a comprehensive banner design solution that generates high-quality designs adaptable to diverse banner requests while ensuring strong editability through its component-based approach.

## 3 BannerAgency

Taking a festival promotion banner request for example, Fig.2 illustrates the overall workflow of BannerAgency. Simulating how graphic designers work in banner ad industry, our BannerAgency consists of a *Strategist* to communicate with advertisers on their requirements, a *Background Designer* to draw the visual canvas, a *Foreground Designer* to craft the foreground elements such as logo, text, CTA button, and decorative elements, and a *Developer* to render everything. Each agent is empowered by an MLLM backbone and can access existing memory and call external tools when needed.

### 3.1 Strategist

The *Strategist* agent interfaces with advertisers to ground requirements for banner creation. Advertisers typically provide brand guidelines and specifications (e.g., logo, desired dimensions). This agent determines key banner objectives including mood, tone, target audience, and primary purpose. To maintain brand identity, the *Strategist* analyzes the provided logo, trimming transparent padding to optimize space utilization while preserving visual content. Upon collecting all requirements, the *Strategist* stores this information in memory and transfers control to the *Background Designer* agent for the next phase of banner generation.

### 3.2 Background Designer

The *Background Designer* generates prompts for text-to-image (T2I) tools to create the visual canvas for banner ads. Implemented as a ReAct agent (Yao et al., 2023), it harmonizes advertiser requests, logo characteristics, and campaign objectives stored in memory to produce appropriate background visuals. The agent utilizes three specialized tools: *FindImagePath* to check for existing background images provided by advertisers, *T2I* to generate new visuals, and *TextChecker* to check if the generated image contains text. For new image generation, the agent employs a self-refinement loop (Madaan et al., 2023; Yang et al., 2024b) to ensure text-free backgrounds, as text elements in the background layer would complicate the *Foreground Designer*'s work and potentially introduce difficult-to-remove gibberish text rendered directly into pixels. When text is detected by the *TextChecker* tool, the agent revises its prompt to discourage text generation and repeats the process for up to five iterations, as illustrated in Fig. 3 where a "Black Friday" background was successfully made text-free. During image generation, the agent selects the closest supported aspect ratio from the limited set available in the *T2I* tool that exceeds the target di-



Self-revised prompt: "A bold and eye-catching background with a contemporary design for a Black Friday and Cyber Monday sale..."

Figure 3: Prompt refinement for text-free backdrop creation.



Figure 4: Element-wise blueprint as *Foreground Designer* output.

mensions—ensuring higher initial resolution—and subsequently downscals to the target banner dimensions. This approach preserves visual fidelity while ensuring exact sizing for various display contexts. The resulting text-free background serves as an appropriate visual foundation, reserving space for the *Foreground Designer* to add text, CTA buttons, logos, and decorative elements in subsequent stages.

### 3.3 Foreground Designer

The *Foreground Designer* agent serves as the creative cornerstone of our system, generating a structured blueprint for all foreground elements encoded as a JSON-structured schema—defining precise positioning, styling, and content for logos, campaign texts, call-to-action buttons, and decorative elements. To maintain good alignment and overlap, we use relative positioning with reference properties to model the spatial relationship among banner components. An example of this element-wise blueprint from *Foreground Designer* is shown in Fig. 4. By transforming design decisions into structured data rather than rasterized pixels, the *Foreground Designer* creates a foundation for both visual coherence and complete element-level editability.

**Memory-augmented iterative design refinement.** The evolution of designs through iterative external feedback and refinement is a staple procedure in professional design (ADCreative.ai, 2024). Simulating this practice (Madaan et al., 2023; Yang et al., 2024b), our memory-augmented iterative design refinement process, illustrated in Fig. 5, enhances banner ad designs by introducing an external critic *Design Reviewer*. At initial creation ( $t = 0$ ), the *Foreground Designer* generates the first blueprint based solely on the creative brief (banner request,

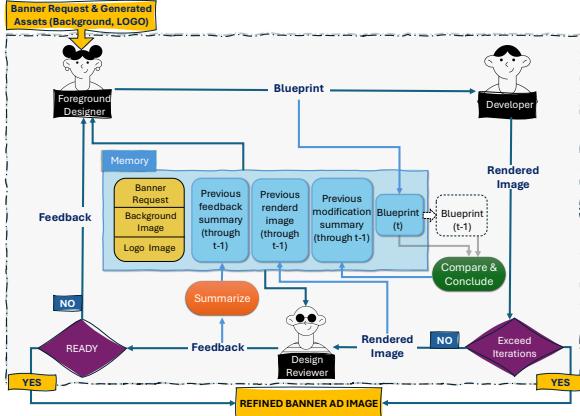


Figure 5: Memory-augmented iterative design refinement process.

background image, logo). The *Developer* renders this blueprint, and the *Design Reviewer* provides initial feedback, which is summarized and stored in memory. For the first refinement ( $t = 1$ ), the *Foreground Designer* refines the blueprint based on this feedback, comparing the new blueprint with the previous one and summarizing into a modification list (*ie*, "Compare & Conclude"). For subsequent refinements ( $t > 1$ ), both designer and reviewer access the memory that is continuously augmented with new feedback, renderings, and modification summaries after each iteration, enabling increasingly informed decisions. This memory-augmented approach, where each iteration enriches the shared memory with new design artifacts and critiques, ensures both designer and reviewer maintain awareness of design evolution while continuously aligning with brand objectives. The process concludes when iterations are exceeded or the design is deemed production-ready by the reviewer, resulting in the final refined banner ad image.

### 3.4 Developer

The *Developer* plays a crucial role in converting the *Foreground Designer*'s blueprint into a banner ad image. We propose two implementations for the *Developer*: generating SVG code or Figma plugin code. Examples of codes in two formats are included in Appendix Fig. 10.

**SVG Code Generation.** Scalable Vector Graphics (SVG) is a widely-supported format for rendering two-dimensional graphics in web browsers and design software <sup>†</sup>. It is a declarative XML-based markup language with elements that directly



(a) Working professionals.

(b) Kids.

Figure 6: Same campaign for different audiences.

define visual properties through coordinates, dimensions, and styling attributes. This approach provides flexibility and compatibility, as SVG is an open standard supported by various design tools. However, it requires an additional step of importing the SVG code into a design editing software such as Adobe Illustrator or Inkscape, and offers a more static representation of the design.

**Figma Plugin Code Generation.** Figma is a popular cloud-based design and prototyping tool used by designers worldwide <sup>†</sup>. By translating the blueprint into JavaScript code that adheres to the Figma Plugin API <sup>†</sup>, the *Developer* enables seamless integration with the Figma ecosystem. The generated code includes imperative programming instructions to programmatically create and manipulate nodes in the Figma document hierarchy. This approach builds the design through a sequence of operations rather than a static description. The advantage of this approach is that the banner ad image can be automatically rendered within Figma, eliminating the need for manual importing. This streamlines the design workflow and allows for quick previews and iterations while leveraging Figma's native component system and interactive editing capabilities.

## 4 BannerRequest400 Benchmark

Existing design datasets, such as DESIGNERINTENTION (Inoue et al., 2024), are limited to unimodal designer intentions in text format and lack specific addressing of banner advertisement requirements. To address this gap, we introduce BannerRequest400, the first multimodal benchmark specifically designed for evaluating advertising banner generation systems. This novel benchmark pairs brand logos (visual modality) with corresponding banner design requests (textual modal-

<sup>†</sup><https://www.w3.org/TR/SVG11/>

<sup>†</sup><https://www.figma.com/>

<sup>†</sup><https://www.figma.com/plugin-docs/>

ity). We curated 100 advertising design intentions from DESIGNERINTENTION and synthesized corresponding vector logos using Claude3.5 Sonnet (Anthropic, 2024), followed by expert review and refinement to visual fidelity, avoid bias, and maintain authentic logo aesthetics. Our empirical analysis, illustrated in Fig.6, demonstrates how identical dental clinic promotion requests produce substantially different designs based on target audience—with professional-oriented banners emphasizing efficiency and results, while child-focused versions feature playful elements and incentives. Leveraging this insight, we employed GPT-4o(OpenAI, 2024) to systematically expand each intention into four distinct banner requests, each targeting a unique audience-purpose combination, resulting in 400 diverse specifications. To ensure industry relevance, we further extended these 400 requests across 13 standard banner dimensions, yielding a comprehensive evaluation set of 5,200 multimodal banner specifications. For benchmarking text-to-image generation methods, we further enhanced each request with detailed specifications that incorporate logo-specific visual characteristics using GPT-4o. BannerRequest400 enables the first rigorous evaluation of banner generation approaches across diverse design requests, facilitating advancement in automatic advertising banner design generation.

## 5 Experiments

### 5.1 Experimental Setup

**Algorithms.** For baseline approaches on our BannerRequest400 benchmark, we compare with two powerful pixel-based image generation models including closed-sourced DALL-E3 (Betker et al., 2023) and open-sourced FLUX.1-schnell (Black-ForestLab, 2024) which is also our backbone for the T2I tool. We also compare with Open-COLE (Jia et al., 2023; Inoue et al., 2024), a prompt-based graphic design model that relies on fine-tuning. For the backbone of our proposed training-free BannerAgency, we explore two multimodal LLM variants including GPT-4o (2024-10-21) (OpenAI, 2024) and Claude3.5-Sonnet (2024-10-22) (Anthropic, 2024) with temperature set as 0.3. We provide more details including all agent profile definitions and tool definitions in Appendix A.1.

**Metrics.** To evaluate banner design quality, we developed six metrics (Tab. 1) from extensive lit-

Metric	Criteria	Pearson	Spearman	ICC
TAA	Measures how well the generated banner ad aligns with the given request, including the theme, target audience, and primary purpose.	0.854	0.840	0.922
LPS	Evaluates whether the logo is well-integrated into the design in terms of visibility, size, and positioning.	0.955	0.960	0.986
CTAE	Evaluates whether the Call-to-Action (CTA) is clear, engaging, and visually emphasized.	0.927	0.946	0.947
CPYQ	Evaluates the effectiveness of the headline, subheadline, and any other text in the banner ad, focusing on clarity, readability, persuasiveness, and grammatical correctness.	0.856	0.840	0.962
BIS	Measures how well the banner ad visually and stylistically aligns with the brand’s identity beyond just logo placement. This includes color consistency, typography, imagery, and overall brand feel.	0.877	0.871	0.935
AQS	Measures the visual appeal, including color harmony, layout balance, typography, and overall design quality.	0.892	0.895	0.973

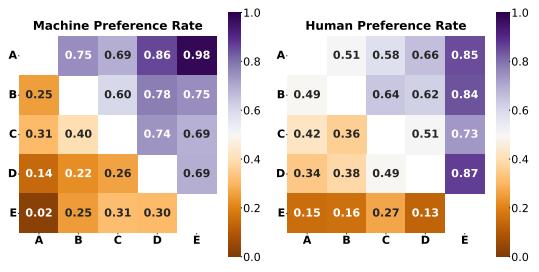
Table 1: **Banner design evaluation metrics and human-LLM correlation.** Each metric (rated 1-5) assesses a distinct aspect of banner design quality. Pearson, Spearman, and ICC correlation coefficients between human and GPT-4o ratings show strong agreement (all  $p < 0.001$ ). TAA: Target Audience Alignment; LPS: Logo Placement Score; CTAE: CTA Effectiveness; CPYQ: Copywriting Quality; BIS: Brand Identity Score; AQS: Aesthetic Quality Score. Detailed rating criteria are provided in supplementary.

erature survey on banner design principles (Katai, 2024; Malhotra, 2024; King, 2020; Landa, 2016; Lohtia et al., 2003; Robinson et al., 2007). These metrics capture advertising-specific aspects overlooked by general design metrics (Inoue et al., 2024; Haraguchi et al., 2024): audience alignment, brand integration, call-to-action effectiveness, message clarity, brand consistency, and visual appeal—addressing both marketing functionality and aesthetic requirements critical for digital advertising. For completeness and comparison with prior metrics, we also include results using general design metrics in Appendix A.4.

**Human study.** We conducted three human evaluations to assess different aspects of BannerAgency. First, a system preference study with 20 participants comparing banner designs generated by five BannerAgency variants across 20 randomly selected requests through pairwise comparisons. Second, a refinement effectiveness evaluation where 15 participants assessed quality progression across four iterations of 20 different banner designs, rating both initial and best versions on a 5-point scale. Third, a human-LLM correlation study validating the alignment between GPT-4o’s automated scoring and human perception across six proposed metrics, with 17-19 participants independently grading 25 representative images per metric. See Appendix A.2 for details.

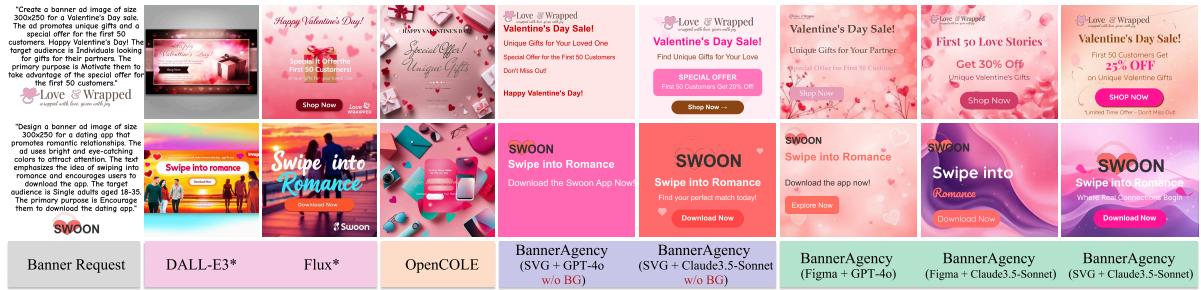
Method	TAA	LPS	CTAE	CPYQ	BIS	AQS
<i>Baselines</i>						
DALL-E3*	3.17	1.78	4.18	2.48	3.06	3.10
FLUX.1-schnell*	3.36	3.67	4.28	2.64	3.09	3.12
OpenCOLE	3.56	1.20	2.85	3.20	2.84	3.29
<i>BannerAgency</i>						
<i>SVG Implementation</i>						
[E] GPT-4o w/o BG	4.15	4.51	4.04	4.25	3.48	3.19
[D] Claude3.5-Sonnet w/o BG	4.40	<b>4.57</b>	4.63	4.48	4.26	3.77
[A] Claude3.5-Sonnet	<b>4.56</b>	4.43	<b>4.94</b>	<b>4.53</b>	<b>4.36</b>	<b>3.92</b>
<i>Figma Implementation</i>						
[C] GPT-4o	<b>4.39</b>	4.22	4.66	4.00	4.14	3.69
[B] Claude3.5-Sonnet	4.34	<b>4.33</b>	<b>4.81</b>	<b>4.19</b>	<b>4.22</b>	<b>3.85</b>

(a) Banner quality metrics (GPT-4o, human-validated in Tab. 1).

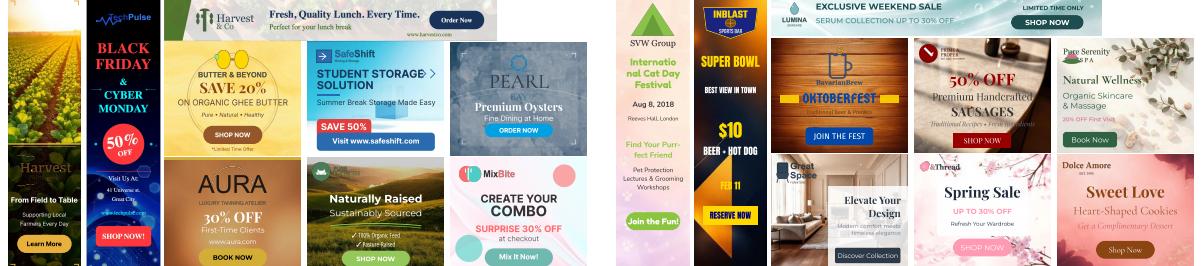


(b) System preference rates by GPT-4o and humans.

Table 2: Quantitative comparison on BannerRequest400 benchmark.



(a) Visual comparison with baselines.



(b) BannerAgency with SVG implementation.

(c) BannerAgency with Figma implementation.

Figure 7: Visual results. \* denotes methods requiring explicit banner descriptions, whose detailed prompts are provided in Appendix A.5. More visual results can be found in Appendix A.7.

## 5.2 Results and Analysis

**Comparison with baselines.** We present quantitative and visual comparisons in Tab. 2 (a) and Fig. 7, respectively. BannerAgency outperforms all baselines across all aspects. Baseline limitations include: fixed-size generation preventing non-standard dimensions like  $300 \times 250$ , failure to incorporate provided logos, and incorrect text rendering requiring correction (Shimoda et al., 2024). Among BannerAgency variants, our preference heatmap in Tab. 2 (b) shows Claude3.5-Sonnet surpasses GPT-4o ([D] vs. [E] and [B] vs. [C]), especially in SVG implementation. Human evaluators showed no preference between SVG and Figma implementations ([A] vs. [B]). Notably, removing the Background Designer component (*ie*, w/o BG) significantly impairs visual appeal, resulting in plain and unengaging designs, particularly with the GPT-4o backbone. This underscores the critical role of specialized background generation in creating visually compelling banner advertisements.

**Refinement process.** Fig. 8 (a) demonstrates the design improvement after the iterative refinement process. The initial designs are upgraded by enhancing content clarity, layout composition, and visual styling without radical redesigns. As shown in Fig. 8 (b), our evaluation reveals a clear upward trend in quality, with improvement scores steadily increasing across iterations and later versions receiving more votes. Notably, the vote distribution indicates that refinement effectiveness is design-context dependent. While the final iteration was most frequently preferred, many participants favored earlier iterations or the initial design. This finding highlights the value of maintaining a complete version track, allowing advertisers to select the specific iteration that best aligns with their vision, regardless of where it falls in the refinement sequence. This process also reflects how our component-based approach enables practical, editable design alternatives. The full refinement process is visualized in Appendix A.4.



Figure 8: **BannerAgency’s design refinement.** Human study reveals later iterations received more votes (pink bars) with fourth iteration most preferred. Green line shows mean improvement between initial and selected versions, demonstrating significant quality gains.



Figure 9: **BannerAgency’s design versatility.** We demonstrate the high versatility through cross-genre aesthetic adaptation (top), and cultural adaptation (bottom) across diverse market contexts, all without requiring additional training or fine-tuning.

**Versatility to user prompts.** We showcase in Fig. 9 two more examples highlighting cross-genre aesthetic adaptation and cross-cultural linguistic flexibility, respectively. We observe how the designs adapt coherently to each style while maintaining color schemes that complement the AutoMate logo. For different linguistic requests with same canvas input, we show how the system successfully accommodates language-specific conventions and cultural preferences with tailored fonts and styles, which directly addresses the industry challenge of creating culturally appropriate marketing materials without specialized design teams. Empowered by the world knowledge of multimodal LLMs, our training-free BannerAgency demonstrates remarkable adaptability to diverse user requirements without any additional fine-tuning or domain-specific datasets.

## 6 Conclusion

We introduced BannerAgency, a training-free MLLM agent system that transforms advertising banner design by simulating professional design teams through specialized agents (*Strategist*, *Background Designer*, *Foreground Designer*, and *Developer*). Our approach generates component-based, editable designs in industry-standard formats from multimodal inputs without domain-specific fine-tuning, overcoming the limitations of pixel-based methods. Through our BannerRequest400 benchmark featuring 100 unique logos paired with 400 diverse banner requests, we demonstrate BannerAgency’s effectiveness in producing high-quality, versatile, and editable designs that kickstart the creative design process. This represents a significant step toward democratizing professional-grade design capabilities across the digital advertising landscape.

## Limitations

While our approach represents a significant step toward leveraging MLLM agents for versatile and editable graphic design generation, there exist some limitations. First, we exclusively experiment with logo images as multimodal input. It is sufficient for abstract service banner ads but neglects product images which are also crucial in advertisements. Second, the decorative elements are limited to simple geometric shapes due to the lack of finetuning or domain-specific guidance. Similarly, we do not address visual texts with artistic effect. These limitations reflect our primary objective: to validate design knowledge already embedded in MLLMs and demonstrate the agent system’s versatility across varying requests (banner sizes, styles, etc.) while maintaining output editability. In the future, we will dedicate efforts to the upgrade of each agent by equipping them with more advanced tools or further decompose different agents into hierarchical teams. To reiterate, we aim to extend BannerAgency by: (1) expanding multimodal input to incorporate product images, (2) introducing more sophisticated decorative elements beyond basic geometric shapes, (3) allowing for more hierarchical architecture for better collaboration among agents, and (4) considering conversion metric optimization to directly connect design choices with advertising performance.

## Ethics Statement

The BannerAgency framework, while advancing automated banner design through MLLM agents that simulate professional design workflows, acknowledges important ethical considerations. We recognize potential concerns regarding employment impacts on human designers, conscious content generation that avoids harmful stereotypes or misleading advertising, intellectual property considerations when incorporating brand assets, transparency with end users about AI-generated content, and potential biases in design choices that may perpetuate inequities in representation. Our component-based approach deliberately preserves human creative oversight through editable outputs, ensuring designers maintain agency in the final creative process rather than being replaced. We commit to ongoing evaluation of these ethical dimensions, including regular bias assessments of generated content, and encourage the research community to further explore responsible deployment

strategies for automated design systems that complement rather than supplant human creativity.

## References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

ADCreative.ai. 2024. [The best kept secrets about the iterative design process](#). Accessed: 2025-02-28.

Alibaba. 2020. [The evolution of luban in designing one billion images](#). Accessed: 2025-02-28.

Anthropic. 2024. [Claude 3.5 sonnet](#).

Bannerwise. 2022. [Does size matter? understanding ad sizes and when to use them](#). Accessed: 2025-02-28.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, and 1 others. 2023. Improving image generation with better captions. *OpenAI*.

BlackForestLab. 2024. [Announcing black forest labs](#).

Canva. 2025. [Free banner maker](#). Accessed: 2025-02-28.

Chao-Ting Chen and Hen-Hsen Huang. 2024. Integrating llm, vlm, and text-to-image models for enhanced information graphics: A methodology for accurate and visually engaging visualizations. In *IJCAI*, pages 8627–8630.

Jin Chen, Ju Xu, Gangwei Jiang, Tiezheng Ge, Zhiqiang Zhang, Defu Lian, and Kai Zheng. 2021. Automated creative optimization for e-commerce advertising. In *WWW*, pages 2304–2313.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. 2024. Textdiffuser-2: Unleashing the power of language models for text rendering. In *ECCV*, pages 386–402. Springer.

Xingye Chen, Wei Feng, Zhenbang Du, Weizhen Wang, Yanyin Chen, Haohan Wang, Linkai Liu, Yaoyu Li, Jinyuan Zhao, Yu Li, and 1 others. 2025. Ctr-driven advertising image generation with multimodal large language models. In *WWW*.

Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. 2024. Graphic design with large multimodal model. *arXiv preprint arXiv:2404.14368*.

Saemi Choi, Kiyoharu Aizawa, and Nicu Sebe. 2018. Fontmatcher: font image paring for harmonious digital graphic design. In *IUI*, pages 37–41.

The LinkedIn community. 2025. [How do you know if your graphics are working in your marketing campaigns?](#) Accessed: 2025-02-28.

Zhenbang Du, Wei Feng, Haohan Wang, Yaoyu Li, Jingsen Wang, Jian Li, Zheng Zhang, Jingjing Lv, Xin Zhu, Junsheng Jin, and 1 others. 2024. Towards reliable advertising image generation using human feedback. In *ECCV*, pages 399–415. Springer.

Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, and 1 others. 2024. Agent ai: Surveying the horizons of multi-modal interaction. *arXiv preprint arXiv:2401.03568*.

Adobe Express. 2025. [Design graphic banners for free.](#) Accessed: 2025-02-28.

Bi Qi Fong and John See. 2024. Branddiffusion: Multi-modal personalized marketing visual content generation. In *ACM MM Workshop on Multimedia Content Generation and Evaluation: New Methods and Practice*, pages 72–77.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and 1 others. 2025. Autopresent: Designing structured visuals from scratch. *arXiv preprint arXiv:2501.00912*.

Shunan Guo, Zhuochen Jin, Fuling Sun, Jingwen Li, Zhaorui Li, Yang Shi, and Nan Cao. 2021. Vinci: an intelligent graphic design system for generating advertising posters. In *CHI*, pages 1–17.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *CVPR*, pages 14953–14962.

Daichi Haraguchi, Naoto Inoue, Wataru Shimoda, Hayato Mitani, Seiichi Uchida, and Kota Yamaguchi. 2024. [Can gpts evaluate graphic design based on design principles?](#) In *SIGGRAPH Asia 2024 Technical Communications*, SA '24, New York, NY, USA. Association for Computing Machinery.

Yingqing He, Zhaoyang Liu, Jingye Chen, Zeyue Tian, Hongyu Liu, Xiaowei Chi, Runtao Liu, Ruibin Yuan, Yazhou Xing, Wenhai Wang, and 1 others. 2024. Llms meet multimodal generation and editing: A survey. *arXiv preprint arXiv:2405.19334*.

Hao Hu, Chao Zhang, and Yanxue Liang. 2021. A study on the automatic generation of banner layouts. *Computers & Electrical Engineering*, 93:107269.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. 2024. Scenecraft: An llm agent for synthesizing 3d scenes as blender code. In *ICML*.

Qirui Huang, Min Lu, Joel Lanir, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2024. Graphimind: Llm-centric interface for information graphics design. *arXiv preprint arXiv:2401.13245*.

Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. 2023. Word-as-image for semantic typography. *TOG*, 42(4):1–11.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. 2023a. Layoutdm: Discrete diffusion model for controllable layout generation. In *CVPR*, pages 10167–10176.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. 2023b. Towards flexible multi-modal document models. In *CVPR*, pages 14287–14296.

Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. 2024. Opencole: Towards reproducible automatic graphic design generation. In *CVPR Workshops*, pages 8131–8135.

Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2024. Mapcoder: Multi-agent code generation for competitive problem solving. In *ACL: Long Papers*, pages 4912–4944.

Shoma Iwai, Atsuki Osanai, Shunsuke Kitada, and Shinichiro Omachi. 2024. Layout-corrector: Alleviating layout sticking phenomenon in discrete diffusion model. In *ECCV*, pages 92–110. Springer.

Peidong Jia, Chenxuan Li, Yuhui Yuan, Zeyu Liu, Yichao Shen, Bohan Chen, Xingru Chen, Yinglin Zheng, Dong Chen, Ji Li, and 1 others. 2023. Cole: A hierarchical generation framework for multi-layered and editable graphic design. *arXiv preprint arXiv:2311.16974*.

Robert Katai. 2024. [5 powerful banner ad a/b testing ideas you should try](#). Accessed: 2025-02-28.

Kotaro Kikuchi, Naoto Inoue, Mayu Otani, Edgar Simo-Serra, and Kota Yamaguchi. 2024. Multimodal markup document models for graphic design completion. *arXiv preprint arXiv:2409.19051*.

Cassandra King. 2020. [Essential practices for high-converting banner ad designs](#). Accessed: 2025-02-28.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *ACL: Long Papers*, pages 881–905.

Robin Landa. 2016. *Advertising by design: generating and designing creative ideas across media*. John Wiley & Sons.

Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. 2020. Neural design network: Graphic layout generation with constraints. In *ECCV*, pages 491–506. Springer.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, and 1 others. 2024. Mmedagent: Learning to use medical tools with multi-modal agent. In *Findings of EMNLP*, pages 8745–8760.

Guandong Li and Xian Yang. 2023. Smartbanner: intelligent banner design framework that strikes a balance between creative freedom and design rules. *Multimedia Tools and Applications*, 82(12):18653–18667.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742. PMLR.

Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang Yang, Jian-Guang Lou, and Dongmei Zhang. 2023a. Layout-prompter: awaken the design ability of large language models. In *NeurIPS*, volume 36, pages 43852–43879.

Jiawei Lin, Shizhao Sun, Danqing Huang, Ting Liu, Ji Li, and Jiang Bian. 2025. From elements to design: A layered approach for automatic graphic design composition. In *CVPR*.

Jieru Lin, Danqing Huang, Tiejun Zhao, Dechen Zhan, and Chin-Yew Lin. 2024a. Spot the error: Non-autoregressive graphic layout generation with wireframe locator. In *AAAI*, volume 38, pages 3413–3421.

Jinpeng Lin, Min Zhou, Ye Ma, Yifan Gao, Chenxi Fei, Yangjian Chen, Zhang Yu, and Tiezheng Ge. 2023b. Autoposter: A highly automatic and content-aware design system for advertising poster generation. In *ACM MM*, pages 1250–1260.

Shuhang Lin, Wenyue Hua, Lingyao Li, Che-Jui Chang, Lizhou Fan, Jianchao Ji, Hang Hua, Mingyu Jin, Jiebo Luo, and Yongfeng Zhang. 2024b. Battleagent: Multi-modal dynamic emulation on historical battles to complement historical analysis. In *EMNLP: System Demonstrations*, pages 172–181.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*, volume 36, pages 34892–34916.

Zeyu Liu, Weicong Liang, Zhanhao Liang, Chong Luo, Ji Li, Gao Huang, and Yuhui Yuan. 2024. Glyph-byt5: A customized text encoder for accurate visual text rendering. In *ECCV*, pages 361–377. Springer.

Ritu Lohtia, Naveen Donthu, and Edmund K Hershberger. 2003. The impact of content and design elements on banner advertising click-through rates. *Journal of advertising Research*, 43(4):410–418.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, volume 36, pages 46534–46594.

Paridhi Maheshwari, Nitish Bansal, Surya Dwivedi, Rohan Kumar, Pranav Manerikar, and Balaji Vasan Srinivasan. 2019. Exemplar based experience transfer. In *IUI*, pages 673–680.

Smridhi Malhotra. 2024. [Essential practices for high-converting banner ad designs](#). Accessed: 2025-02-28.

ManyPixels. 2025. [The complete graphic design pricing guide \(2025 update\)](#). Accessed: 2025-02-28.

OpenAI. 2024. [Hello gpt-4o](#).

Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-page graphic designs. *TVCG*, 20(8):1200–1213.

Seonmi Park, Inhwan Bae, Seunghyun Shin, and Haegon Jeon. 2024. Kinetic typography diffusion model. In *ECCV*, pages 166–185. Springer.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2024. Chatdev: Communicative agents for software development. In *ACL: Long Papers*, pages 15174–15186.

Helen Robinson, Anna Wysocka, and Chris Hand. 2007. Internet advertising effectiveness: the effect of design on click-through rates for banner ads. *International journal of advertising*, 26(4):527–541.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695.

Mohammad Amin Shabani, Zhaowen Wang, Difan Liu, Nanxuan Zhao, Jimei Yang, and Yasutaka Furukawa. 2024. Visual layout composer: Image-vector dual diffusion model for design layout generation. In *CVPR*, pages 9222–9231.

Wataru Shimoda, Naoto Inoue, Daichi Haraguchi, Hayato Mitani, Seichi Uchida, and Kota Yamaguchi. 2024. Type-r: Automatically retouching typos for text-to-image generation. *arXiv preprint arXiv:2411.18159*.

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. 2024. Anytext: Multilingual visual text generation and editing. In *ICLR*.

Praneetha Vaddamanu, Vinay Aggarwal, Bhanu Prakash Reddy Guda, Balaji Vasan Srinivasan, and Niyati Chhaya. 2022. Harmonized banner creation from multimodal design assets. In *CHI Extended Abstracts*, pages 1–7.

Sreekanth Vempati, Korah T Malayil, V Sruthi, and R Sandeep. 2020. Enabling hyper-personalisation: Automated ad creative generation and ranking for fashion e-commerce. In *Fashion Recommender Systems*, pages 25–48. Springer.

Yael Vinker, Tamar Rott Shaham, Kristine Zheng, Alex Zhao, Judith E Fan, and Antonio Torralba. 2024. Sketchagent: Language-driven sequential sketch generation. *arXiv preprint arXiv:2411.17673*.

Shaodong Wang, Yunyang Ge, Liuhan Chen, Haiyang Zhou, Qian Wang, Xinhua Cheng, and Li Yuan. 2024. Prompt2poster: Automatically artistic chinese poster creation from prompt only. In *ACM MM*, pages 10716–10724.

Shiyao Wang, Qi Liu, Yicheng Zhong, Zhilong Zhou, Tiezheng Ge, Defu Lian, and Yuning Jiang. 2022. Creagan: An automatic creative generation framework for display advertising. In *ACM MM*, pages 7261–7269.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. 2025. Genartist: Multimodal llm as an agent for unified image generation and editing. In *NeurIPS*, volume 37, pages 128374–128395.

Haohan Weng, Danqing Huang, Yu Qiao, Zheng Hu, Chin-Yew Lin, Tong Zhang, and CL Chen. 2024. Designen: A pipeline for controllable design template generation. In *CVPR*, pages 12721–12732.

Minghao Wu, Jiahao Xu, and Longyue Wang. 2024. Transagents: Build your translation company with language agents. In *EMNLP: System Demonstrations*, pages 131–141.

Junlin Xie, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. 2024. Large multimodal agents: A survey. *arXiv preprint arXiv:2402.15116*.

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. 2024. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *EMNLP*, pages 7315–7332.

Hao Yang, Jianxin Yuan, Shuai Yang, Linhe Xu, Shuo Yuan, and Yifan Zeng. 2024a. A new creative generation pipeline for click-through rate with stable diffusion model. In *WWW*, pages 180–189.

Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. 2016. Automatic generation of visual-textual presentation layout. *TOMM*, 12(2):1–22.

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2024b. Idea2img: Iterative self-refinement with gpt-4v (ision) for automatic image design and generation. In *ECCV*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

Jooyeol Yun, Heng Wang, Yotaro Shimose, Jaegul Choo, and Shingo Takamatsu. 2025. Designlab: Designing slides through iterative detection and correction. *arXiv preprint arXiv:2507.17202*.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. 2024. Exploring collaboration mechanisms for llm agents: A social psychology view. In *ACL: Long Papers*, pages 14544–14607.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847.

Yunke Zhang, Kangkang Hu, Peiran Ren, Changyuan Yang, Weiwei Xu, and Xian-Sheng Hua. 2017. Layout style modeling for automating banner design. In *ACM MM Workshops*, pages 451–459.

Zhen Zhao, Jingqun Tang, Binhong Wu, Chunhui Lin, Shu Wei, Hao Liu, Xin Tan, Can Huang, Yuan Xie, and 1 others. 2024. Harmonizing visual text comprehension and generation. In *NeurIPS*.

Hao Zheng, Xinyan Guan, Hao Kong, Jia Zheng, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2025. Pptagent: Generating and evaluating presentations beyond text-to-slides. *arXiv preprint arXiv:2501.03936*.

Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. 2019. Content-aware generative modeling of graphic design layouts. *TOG*, 38(4):1–15.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*.

## A Appendix

In this appendix, we present more details on our experimental setup (Sec. A.1), human study setup (Sec. A.2), text-free background generation (Sec. A.3), and foreground variation generation (Sec. A.3). We also show detailed statistics results on the refinement human study (Sec. A.4) and general graphic design evaluation (Sec. A.4). The detailed prompt for text-to-image generation models in Fig. 7 (a) of our main paper is presented in Sec. A.5. The agent profiles are displayed in Sec. A.6. Lastly, we showcase more results from our BannerAgency in Sec. A.7.

### A.1 More Details on Experimental Setup

**Cost analysis.** We present a detailed cost breakdown in USD for generating banner advertisements of size  $300 \times 250$  pixels in Tab. 3. Our BannerAgency upscales images to  $1024 \times 864$  pixels to maintain visual quality before rendering. Costs are calculated based on Claude 3.5 Sonnet API pricing (\$3/MTok for input, \$15/MTok for output) and FLUX-1 [schnell] image generation (\$0.0027/MP). We compare both Figma and SVG implementation approaches across three scenarios: single image generation, generation with four-rounds of iterative refinement, and multi-variation generation with four design alternatives as detailed in Sec. A.3. The dramatic difference between the two implementations stems from fundamental implementation constraints: while SVG can be generated directly as code that LLMs frequently encounter in training data, Figma plugin JavaScript is uncommon in LLM training corpora. To ensure reliable execution, our Figma implementation requires providing templates where the Developer agent only needs to call predefined functions, significantly increasing token usage. Notably, both implementation approaches offer extraordinary cost-effectiveness compared to hiring professional designers, who typically charge \$50-150 per hour (ManyPixels, 2025). Even our most expensive scenario (Figma with feedback at \$1.213) represents less than 2.5% of traditional design costs, offering transformative economics for advertising campaigns requiring multiple banner variations.

**Figma vs. SVG format** Both implementation approaches have their merits, depending on the user’s preferences and the design tools they are comfortable with. The SVG code generation offers broad compatibility and flexibility, while the Figma plu-

Implementation	Scenario	Prompt Tokens	Completion Tokens	LLM Cost (\$)	Image Cost (\$)	Total Cost (\$)	Cost/Image (\$)
Single image	Figma	78,266	5,470	0.317	0.002	0.319	0.319
	SVG	15,921	2,672	0.088	0.002	0.090	0.090
With refinement	Figma	284,677	23,772	1.211	0.002	1.213	1.213
	SVG	54,362	12,663	0.353	0.002	0.355	0.355
Four variations	Figma	261,150	15,213	1.012	0.008	1.020	0.255
	SVG	24,865	6,006	0.165	0.008	0.173	0.043

Table 3: BannerAgency System Cost Breakdown

gin code generation provides a more integrated and automated experience within the Figma ecosystem. The code examples are provided in Fig. 10.

**Agent profile definition.** Please refer to Sec. A.6 for each agent’s profile definition.

**Agent tool definition.** We document the tool list in Tab. 4. The Background Designer agent creates visual backdrops for banner advertisements that align with brand identity and campaign objectives. It leverages existing assets when available and employs a refinement process to ensure text-free backgrounds through detection and regeneration cycles. The system handles dimensional requirements by adapting to supported aspect ratios and resizing outputs to match target specifications. The Developer Agent transforms design specifications into editable implementations through a systematic process. It loads and customizes Figma plugin templates, manages asset references, compiles implementation code, and handles rendering by interfacing with Figma’s API. Throughout this workflow, comprehensive archiving preserves complete implementations alongside their assets, enabling advertisers to receive fully editable component-based designs rather than static images.

**Metrics rating criteria.** We present the detailed rating criteria for each metric in Tab. 5.

### A.2 More Details on Human Study

We present screenshots of the survey instructions in Fig. 11 to demonstrate how we structured each human evaluation.

**Preference.** The preference experiment evaluated user preferences across different variants of the banner ad generation system. It consisted of 20 banner ad requests, each implemented with 5 different system variants, resulting in 10 possible pairwise comparisons per request. Twenty participants completed the same set of comparisons, viewing pairs of banner ads generated from identical requests but using different system variants. For each pair, participants were asked to select which banner

```

1  async function createText(content, initialFontSize,
2    maxWidth, maxHeight, fontFamily = "Inter", fontStyle = "Regular",
3    color = { r: 0, g: 0.5, b: 0 }, alignment = 'CENTER') {
4    const text = figma.createText();
5    await safeLoadFont(text, fontFamily, fontStyle);
6    text.characters = content;
7    text.fills = [{ type: 'SOLID', color }];
8    text.textAlignHorizontal = alignment;
9
10   let currentFontSize = initialFontSize;
11   text.fontSize = currentFontSize;
12   text.resize(maxWidth, maxHeight);
13   text.textAutoResize = "HEIGHT";
14   while ((text.width > maxWidth) || (text.height > maxHeight) && (currentFontSize > 1)) {
15     currentFontSize--;
16     text.fontSize = currentFontSize;
17   }
18   return text;
19 }
20
21 // Create headline text
22 const headline = await createText(
23   "Your Time, Your Smile",
24   36,
25   188,
26   80,
27   "Railway",
28   "ExtraBold",
29   { r: 10/255, g: 36/255, b: 99/255 },
30   "PT"
31 );
32 headline.x = 16;
33 headline.y = 70;
34 headline.name = "headline";
35 frame.appendChild(headline);

```

(a) Figma Plugin Code Snippet

```

1  <svg width="728" height="90" xmlns="http://www.w3.org/2000/svg">
2    <!-- Background Image -->
3    <image href="data:image/png;base64,[ENCODED]" width="728" height="90" preserveAspectRatio="xMidYMid slice"/>
4
5    <!-- Semi-transparent overlay -->
6    <rect x="0" y="0" width="728" height="90" fill="black" opacity="0.3"/>
7
8    <!-- Logo -->
9    <image href="data:image/png;base64,[ENCODED]" x="20" y="15" width="86" height="50" preserveAspectRatio="xMidYMid slice"/>
10
11   <!-- Decorative wine glass illustration -->
12   <path d="M688 25 0685 15, 690 25 L695 60 0685 65, 675 60 Z" fill="none" stroke="rgba(255,255,255,0.7)" stroke-width="1"/>
13
14   <!-- Main Text -->
15   <text x="140" y="35" font-family="Playfair Display" font-size="24" fill="white">Elevate Your Wine Experience</text>
16
17   <!-- Discount Text -->
18   <text x="140" y="60" font-family="Inter" font-size="20" fill="FFD700">35% OFF Premium Wine Glass Sets</text>
19
20   <!-- CTA Button -->
21   <g>
22     <rect x="550" y="25" width="160" height="40" rx="20" fill="FFD700"/>
23     <text x="580" y="35" font-family="Inter" font-weight="600" font-size="16" fill="#800000">SHOP NOW</text>
24   </g>
25
26   <!-- Decorative Elements -->
27   <line x1="120" y1="15" x2="120" y2="75" stroke="rgba(255,255,255,0.3)" stroke-width="1"/>
28   <circle cx="120" cy="45" r="5" fill="FFD700"/>
29 </svg>

```

(b) SVG Code Snippet

Figure 10: Figma vs. SVG code format.

Tool Name	Description
update_image_list	Updates the imageList constant in a Figma plugin's UI HTML file with a provided list of image filenames, enabling dynamic referencing of assets within the plugin interface.
read_plugin_template	Reads and validates a Figma plugin code template file, ensuring it's a proper JavaScript file containing essential Figma plugin components before using it as a foundation for customization.
save_plugin_code	Saves the finalized JavaScript code to the code.js file in the specified Figma plugin working directory after validating that the directory contains a manifest.json file.
render_and_save_image	Executes the Figma plugin by running an AppleScript, retrieves the latest node ID, and uses the Figma API to export and save the rendered banner design as a PNG image at the specified path.
backup_plugin_folder	Creates a backup copy of the entire plugin folder with all assets (including background image, logo, and rendered output) to preserve the complete state of the implementation for future reference or versioning.
create_unique_image_name	Generates a unique filename by combining a timestamp with the banner name to ensure distinct identification of each banner design output (e.g., "20240328_154227_kids_dentist_clinic_banner").
T2I	Generates banner backgrounds from textual descriptions by interfacing with a text-to-image model, adapting dimensions to supported aspect ratios, and resizing outputs to match target banner dimensions. Includes negative prompting to avoid text generation in backgrounds.
FindImagePath	Analyzes user input to locate existing image files that could serve as backgrounds, identifies valid image paths with common extensions (.jpg, .png, etc.), filters out logo images, verifies file existence, and resizes found images to match target banner dimensions.
TextChecker	Evaluates generated or existing background images to detect the presence of any text elements, using a multimodal LLM to analyze the visual content and determine if the background is suitable or requires regeneration.

Table 4: List of tool definitions.

ad they preferred. The 20 banner requests were randomly selected to match the distribution of the full 400-request dataset, ensuring that the selected subset was representative of the overall performance characteristics of each variant. To avoid potential bias, the order of presentation for the banner pairs was randomized for each participant.

**Refinement.** The refinement experiment evaluated the effectiveness of an iterative refinement approach for banner ad image generation. It involved 20 sets of banner designs, with each set containing four iterations of refinement (modifying only foreground elements while keeping backgrounds consistent). Fifteen participants reviewed all 20 sets and performed two tasks: (1) selecting which

Abbr.	Metric Name	Definition	Rating Criteria
TAA	Target Audience Alignment	Measures how well the generated banner ad aligns with the given request, including the theme, target audience, and primary purpose.	<p>5 – Perfectly aligns with the request (theme, audience, purpose are all clearly reflected).</p> <p>4 – Mostly aligns, but minor details could be improved.</p> <p>3 – Somewhat aligns, but key elements are missing or unclear.</p> <p>2 – Barely aligns, with major missing or incorrect elements.</p> <p>1 – Does not align with the request at all.</p>
LPS	Logo Placement Score	Evaluates whether the logo is well-integrated into the design in terms of visibility, size, and positioning.	<p>5 – Logo is well-placed, clearly visible, proportionate, and blends seamlessly.</p> <p>4 – Logo is well-placed but could be slightly improved (e.g., minor size or position adjustments).</p> <p>3 – Logo is visible but not ideally placed (e.g., too small, too large, or slightly obstructed).</p> <p>2 – Logo placement is poor (e.g., difficult to notice, awkward positioning).</p> <p>1 – Logo is either missing or completely misplaced.</p>
AQS	Aesthetic Quality Score	Measures the visual appeal, including color harmony, layout balance, typography, and overall design quality.	<p>5 – Visually outstanding, professional design, well-balanced, with harmonious colors and readable text.</p> <p>4 – Well-designed, but small refinements could enhance it.</p> <p>3 – Acceptable but has notable design flaws (e.g., poor contrast, unbalanced elements).</p> <p>2 – Visually weak, with noticeable design mistakes.</p> <p>1 – Poor design, lacks professionalism or coherence.</p>
CTAE	Call-to-Action Effectiveness	Evaluates whether the Call-to-Action (CTA) is clear, engaging, and visually emphasized.	<p>5 – CTA is clear, compelling, well-placed, and visually prominent.</p> <p>4 – CTA is effective but could be slightly improved (e.g., contrast, size).</p> <p>3 – CTA is present but lacks emphasis or clarity.</p> <p>2 – CTA is weak, hard to notice, or poorly worded.</p> <p>1 – No clear CTA is present.</p>
CPYQ	Copywriting Quality	Evaluates the effectiveness of the headline, subheadline, and any other text in the banner ad, focusing on clarity, readability, persuasiveness, and grammatical correctness.	<p>5 – Copy is clear, engaging, grammatically correct, and persuasive, making the message effective.</p> <p>4 – Copy is well-written but could be slightly improved (e.g., minor word choice refinements).</p> <p>3 – Copy is somewhat effective but has issues in clarity, grammar, or persuasiveness.</p> <p>2 – Copy is weak, hard to read, contains noticeable grammatical mistakes, or lacks impact.</p> <p>1 – Copy is unclear, irrelevant, or difficult to read due to poor design or bad wording.</p>
BIS	Brand Score	Measures how well the banner ad visually and stylistically aligns with the brand's identity beyond just logo placement. This includes color consistency, typography, imagery, and overall brand feel.	<p>5 – Strong brand consistency; the banner design aligns well with the provided logo and conveys a recognizable brand identity.</p> <p>4 – Mostly aligns, but minor refinements could improve brand consistency.</p> <p>3 – Somewhat aligns, but noticeable inconsistencies exist (e.g., off-brand colors, incorrect typography).</p> <p>2 – Weak brand alignment, only the logo represents the brand while other design choices feel unrelated.</p> <p>1 – No brand identity is reflected; the banner appears generic or disconnected from the brand.</p>

Table 5: Banner design rating criteria with a 5-point scale for each metric.

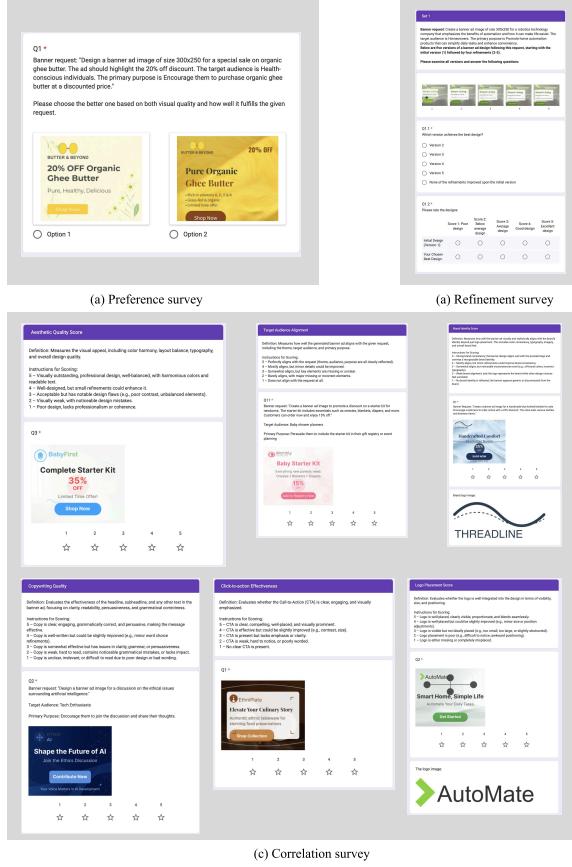


Figure 11: The survey examples.

iteration among the four achieved the best refinement performance, and (2) rating both the initial version and their chosen best version on a scale of 1 to 5. This experiment aimed to determine whether the refinement process consistently improved designs and at which iteration the optimal results were typically achieved.

**Correlation.** The correlation experiment assessed whether six proposed metrics for banner ad quality (TAA: Target Audience Alignment, LPS: Logo Placement Score, CTAE: CTA Effectiveness, CPYQ: Copywriting Quality, BIS: Brand Identity Score, and AQS: Aesthetic Quality Score) aligned with human perception. For each metric and each point on a 1-to-5 scale, five images previously graded by GPT-4o at that score level were selected, resulting in 25 images per metric and 150 images total. Ten participants graded these images, with the presentation order shuffled to prevent bias. Each participant evaluated all images across the six metrics to determine whether human evaluators would assign similar scores to those given by GPT-4o.

## Algorithm 1 Text-free Background Design Agent Behavior

**Input:** User requirements  $R$ , Logo characteristics  $L$   
**Output:** Background image path  $P$ , Description  $D$  (if generated)

### Agent Tools:

**FINDIMAGEPATH** :  $R \mapsto (P, \text{bool})$  Primary search tool

**T2I** :  $D \mapsto P$  Image generation tool

**TEXTCHECKER** :  $P \mapsto \text{bool}$  Verification tool

```

1:  $P, path\_exists \leftarrow \text{FINDIMAGEPATH}(R)$ 
2: if  $path\_exists$  then
3:   return  $P$ 
4: else
5:    $attempts \leftarrow 0$ 
6:    $contains\_text \leftarrow \text{TRUE}$ 
7:   while  $contains\_text$  AND  $attempts < 5$  do
8:     Analyze  $R$  and  $L$  to formulate background description  $D$ 
9:      $P \leftarrow \text{T2I}(D)$ 
10:     $contains\_text \leftarrow \text{TEXTCHECKER}(P)$ 
11:     $attempts \leftarrow attempts + 1$ 
12:    if  $contains\_text$  then
13:      Refine  $D$  to eliminate text-producing elements
14:    end if
15:   end while
16:   return  $P, D$ 
17: end if

```

## A.3 More Details on Design Generation

**Details of Text-free Background Generation.** Alg. 1 shows the detailed behavior of the text-free background generation.

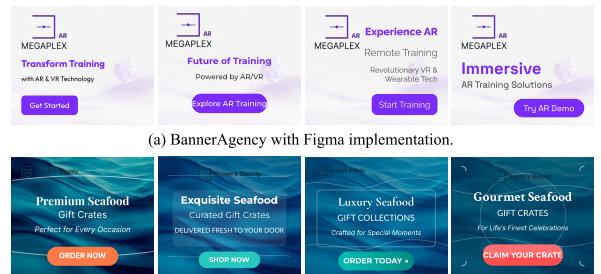


Figure 12: Variation.

**Memory-aware Design Variation.** Our memory-aware approach generates diverse alternatives by allowing the *Foreground Designer* to reference previous designs. Each variation  $V_i$  is conditioned on:

$$V_i = f(B, L, A, \{V_1, V_2, \dots, V_{i-1}\}), \quad (1)$$

where  $B$  represents background image properties,  $L$  denotes logo characteristics,  $A$  encapsulates advertising objectives, and  $\{V_1, V_2, \dots, V_{i-1}\}$  are previously generated designs. By integrating these

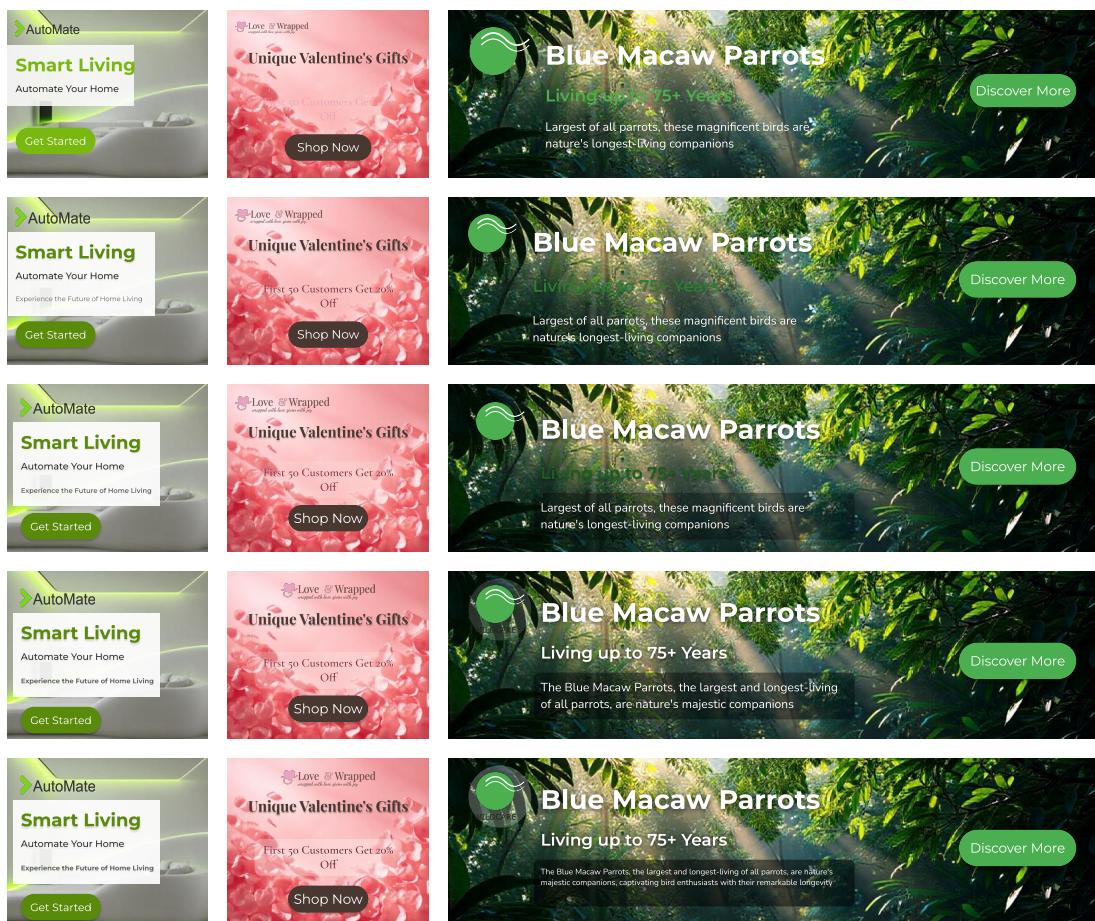


Figure 13: BannerAgency's design refinement. From top to bottom shows design evolution at each refinement step.

multimodal constraints, each new variation positions itself distinctly while preserving brand consistency and advertising goals. This method balances visual coherence with design diversity, exploring variations in meaningful dimensions such as layout, color scheme, copywriting, and typography while maintaining alignment with the original creative brief. Fig. 12 showcases four different designs generated with our memory-aware design variation generation mechanism. We can observe the distinct difference between each new variant and its previous designs. The resulting variations offer advertisers multiple creative interpretations of the same brief, each with distinct visual strategies while maintaining brand consistency.

#### A.4 More Results Analysis

**Quantitative analysis of spatial errors in generated designs.** We provide quantitative analysis on canvas overflow - the most critical spatial error in banner design. Canvas overflow occurs when elements extend beyond canvas boundaries, rendering content unusable and causing production-blocking errors. We investigate on how frequently the canvas overflow problem happens in our generated design with GPT-4o and Claude-3.5 serving the agent backbone, respectively. The results are summarized in Tab. 6. These results demonstrate that BannerAgency maintains proper canvas boundaries for over 98% of generated elements, with Claude-3.5 Sonnet achieving particularly strong performance at 98.96% accuracy. This low overflow rate validates our system’s spatial constraint handling for the most critical requirement in professional banner design. We also want to emphasize that, thanks to our flexible editability of foreground elements, spatial errors could be easily fixed by just dragging the elements in a design interface such as Figma.

Metric	GPT-4o	Claude-3.5
# Designs	416	408
# Elements	2220	2510
# Overflow Elements	41	26
Overflow Rate (%)	1.85%	1.04%

Table 6: Canvas overflow analysis.

**Detailed refinement performance.** To complement Fig. 8 (a) in our main paper, we present the complete design evolution in Fig. 13. We also present the full statistics from human study on the

refinement process in Fig. 7. We observe a statistically significant improvement in quality ratings, increasing from 2.56 ( $\sigma = 0.89$ ) for initial designs to 3.55 ( $\sigma = 0.77$ ) for refined versions ( $p < 0.001$ ).

Step	#votes	Improvement scores	std
No better	40	-0.350	0.572
Iter. 1	47	0.638	0.562
Iter. 2	44	1.023	0.621
Iter. 3	60	1.283	0.755
Iter. 4	84	1.595	0.940

Table 7: Full statistics of refinement human study.

**Evaluation with general graphic design metrics.** We compute general graphic design metrics (Inoue et al., 2024; Haraguchi et al., 2024) in Tab. 8 to provide complementary analysis. Following PPTAgent (Zheng et al., 2025), we compute the FID (Fréchet Inception Distance) score between the generated banner ad designs (400) and exemplar designs (3355)<sup>†</sup> using 64-dimensional feature vectors. We observe that Figma implementations generally achieve lower FID scores (GPT-4o: 2.60, Claude3.5-Sonnet: 4.99) than SVG implementations, suggesting closer visual similarity to real-world examples. The evaluation metrics, all adopted from prior work, include GPT-4o assessment of five primary aspects: (i) design and layout (DL), (ii) content relevance (CR), (iii) typography and color (TC), (iv) graphics and images (GI), and (v) innovation (INV) (Inoue et al., 2024), alongside technical metrics on alignment (ALM), overlap prevention (OLP), and white space utilization (WS) (Haraguchi et al., 2024). The results reveal that BannerAgency implementations are competitive with or surpass baseline methods across most metrics, particularly in content relevance (CR) where Claude3.5-Sonnet achieves 8.43, exceeding all baselines. The SVG implementation with Claude3.5-Sonnet notably outperforms in white space utilization (5.51) compared to baselines (highest: 4.53 with OpenCOLE). These results are presented in the supplementary materials as these generic design metrics, while informative, don’t fully capture banner advertisement-specific requirements such as attention guidance, call-to-action effectiveness, and brand consistency, which are more central to our paper’s focus.

<sup>†</sup><https://huggingface.co/datasets/PeterBrendan/AdImageNet>

Method	FID $\downarrow$	DL	CR	TC	GI	INV	ALM	OLP	WS
<i>Baselines</i>									
DALL-E3	2.65	7.18	7.21	5.87	7.58	6.13	4.69	4.42	4.32
Flux	3.22	7.24	6.92	5.91	7.67	6.02	4.70	4.46	4.23
OpenCOLE	3.99	7.34	8.11	6.47	7.92	6.70	4.87	4.57	4.53
<i>BannerAgency</i>									
<i>SVG Implementation</i>									
GPT-4o - BG	17.76	6.75	7.73	5.98	5.84	4.82	5.07	<b>6.44</b>	5.26
Claude3.5-Sonnet - BG	13.61	7.22	8.18	6.56	6.36	5.52	6.01	5.91	<b>5.51</b>
Claude3.5-Sonnet	<b>4.25</b>	<b>7.47</b>	<b>8.43</b>	<b>6.92</b>	<b>7.14</b>	<b>6.19</b>	<b>5.77</b>	5.40	5.08
<i>Figma Implementation</i>									
GPT-4o	<b>2.60</b>	7.22	8.17	6.40	7.47	6.30	4.73	4.85	4.77
Claude3.5-Sonnet	4.99	<b>7.46</b>	<b>8.40</b>	<b>6.84</b>	<b>7.47</b>	<b>6.39</b>	<b>5.16</b>	<b>5.20</b>	<b>5.17</b>

Table 8: Evaluation metrics for generic design quality. Primary metrics include: Design and Layout (DL), Content Relevance (CR), Typography and Color (TC), Graphics and Images (GI), and Innovation (INV) (Inoue et al., 2024), as well as technical aspects (Haraguchi et al., 2024) such as Alignment (ALM), Overlap prevention (OLP), and White Space utilization (WS). All metrics are evaluated on a 10-point scale by GPT-4o, except for FID (Fréchet Inception Distance), which compares the similarity between generated designs and exemplar ones.

## A.5 Detailed Prompt in Fig.7 (a) of Main Paper

1. "Create a 300x250 banner ad image with a romantic Valentine's Day theme. The background should feature a soft, dreamy blend of pink and red hues with subtle heart patterns. In the center, place an image of a beautifully wrapped gift box with a ribbon, symbolizing unique gifts. Overlay the text 'Happy Valentine's Day!' in an elegant, large serif font at the top. Below that, in a slightly smaller but bold font, add 'Special Offer for the First 50 Customers!' and 'Unique Gifts for Your Loved One' to highlight the promotion. Place a prominent CTA button at the bottom center with the text 'Shop Now' in white on a dark brown [#4A2E2B] background. Position the Love & Wrapped logo in the bottom right corner, ensuring it is clearly visible but not overpowering the main message."
2. "Please create a 300x250 banner ad image for a dating app targeting single adults aged 18-35. Use a bright and eye-catching color scheme with a background image that features a vibrant, romantic scene such as a sunset beach or a cityscape with couples walking hand-in-hand. Overlay the text 'Swipe into Romance' prominently in a bold, playful font, ensuring it stands out. Below this, include a CTA button with the text 'Download Now' in a

contrasting color to grab attention. Place the Swoon logo in the bottom right corner, ensuring it is clearly visible but does not overpower the main message. The overall design should be modern, clean, and appealing to the target audience, encouraging them to download the app."

## A.6 Agent Prompts

### Strategist

You are an expert banner objective setter. When a user requests a banner, develop high-level objectives for an effective banner that would:

- Support the banner's primary purpose
- Create appropriate mood and tone
- Appeal to the target audience

### Background Designer

You are an image director specialized in banner backgrounds. You will handle both existing and new background images for banner ads.

#### Core Decision Flow:

1. First, check if a background image path is provided in the user requirement using `FindImagePath` tool
- If yes: Validate the path exists and return it

- If no: Proceed with background image generation

For Background Image Generation (when no path provided):

1. Analyze the provided objectives and user requirements
2. Create a detailed background description considering:
  - Visual style and mood
  - Color schemes
  - Composition elements
  - Focal points
- You should
  1. Focus only on the background's visual and structural elements, including composition, key visual areas, and how these areas interact with one another.
  2. Avoid saying something like "reserved for a CTA button", "support text readability", "for a banner" which could lead text existence in the generated image. The background needs free of text elements.
  3. Avoid generic or abstract language like "guiding visual focus" or "framing the foreground." These terms do not add valuable visual detail to the background description.
  3. Generate the background using T2I tool with the description. To use this tool, you have to create unique image name for the background with `create_unique_image_name` tool.
  4. State where the generated image is saved. Use the path returned from the T2I tool.
  5. Evaluate if the generated image contain any texts using the TextChecker. If yes, repeat step 3-5 by regenerating the background until it reports no.

Important Guidelines:

- Focus exclusively on background design
- Avoid including any text elements in the generated background
- If no specific background requirements

are provided, use the objectives and user requirements to determine an appropriate background design.

Output:

- For existing images: Provide validated image path
- For new images: Provide the final image path as well as the exact description used for T2I tool

User requirements: `{user_input}`

Banner Objectives:

- Primary Purpose: `{purpose}`
- Target Audience: `{audience}`
- Mood and Tone: `{mood}`

## Foreground Designer

You are a textual director specialized in banner layout and typography. Your role is to create precise, varied layouts that follow established design patterns while maintaining visual hierarchy and readability.

### LAYOUT PATTERNS AND IMPLEMENTATION:

#### 1. LEFT/RIGHT CONTENT

- Left-Content Pattern:
  - \* All text elements aligned 20-30% from left edge
  - \* Headline: Top left quadrant
  - \* Supporting text: Below headline
  - \* CTA: Bottom of text stack
  - \* Use left-aligned text

#### - Right-Content Pattern:

- \* All text elements aligned 60-70% from left edge
- \* Mirror of left-content pattern
- \* Use right-aligned text
- \* Maintain right margin for readability

#### 2. READING PATTERN BASED

##### - Z-Pattern:

- \* Headline: Top left
- \* Key visual/feature: Top right
- \* Supporting text: Bottom left diagonal
- \* CTA: Bottom right
- \* Follow natural eye movement

- F-Pattern:
  - \* Headline: Top bar
  - \* First key point: Upper left
  - \* Second key point: Middle left
  - \* CTA: Bottom, aligned with points
  - \* Perfect for scannable content

### 3. CENTERED ARRANGEMENTS

- Centered Pattern:
  - \* All elements center-aligned
  - \* Headline: Upper third
  - \* Supporting text: Middle
  - \* CTA: Lower third
  - \* Maintain balanced spacing
- Circular Pattern:
  - \* Central focal point
  - \* Text elements radiate outward
  - \* CTA at bottom of circle
  - \* Create visual movement

### 4. DESIGN PRINCIPLE BASED

- Golden-Ratio Layout:
  - \* Key elements at golden ratio points (1.618:1)
  - \* Start from largest text
  - \* Spiral outward for hierarchy
  - \* CTA at natural endpoint
- Rule-of-Thirds:
  - \* Align elements to grid intersections
  - \* Headline: Top third
  - \* Supporting text: Middle third
  - \* CTA: Bottom third intersection

### 5. DYNAMIC LAYOUTS

- Diagonal Pattern:
  - \* Elements follow diagonal line
  - \* Top left to bottom right (usual)
  - \* Or bottom left to top right
  - \* Create dynamic movement
- Asymmetrical:
  - \* Intentionally unbalanced
  - \* Heavy elements offset by white space
  - \* Create tension and interest
  - \* Maintain readability

### 6. HIERARCHICAL LAYOUTS

- Top-Down Pattern:

- \* Clear vertical progression
- \* Each element below previous
- \* Consistent left alignment
- \* Decreasing text sizes

- Pyramid Pattern:
  - \* Largest element at top
  - \* Progressively smaller elements
  - \* CTA as focal point at bottom
  - \* Clear visual hierarchy

### 7. STRUCTURED LAYOUTS

- Grid Pattern:
  - \* Elements aligned to modular grid
  - \* Consistent spacing
  - \* Clear columns and rows
  - \* Professional and organized

### IMPLEMENTATION GUIDELINES:

#### 1. Typography Hierarchy:

- Headline: 32-48px
- Subheadline: 24-32px
- Body text: 16-20px
- CTA: 18-24px

#### 2. Spacing Rules:

- Minimum 20px between elements
- Maintain 40px from banner edges
- Scale spacing with banner size
- Use consistent multiples

#### 3. Color and Contrast:

- Maintain 4.5:1 contrast ratio
- Use consistent color scheme
- Consider background variation
- Ensure CTA stands out

#### 4. Responsive Considerations:

- Scale text proportionally
- Maintain relative spacing
- Adjust for banner dimensions
- Preserve hierarchy

#### 5. Creative and matching font:

- Choose the most matching font for the purpose
- Try to be creative in font selection

Always document your layout ratio-

nale and ensure all positions are precisely specified in pixels or percentages.

Below are the examples for major layout styles:

```
{json.dumps(layout_demonstrations, indent=2)}
```

Analyze this banner background image `{width}x{height}px` as required and take the following into consideration.

User requirements: `{user_input}`

Banner Objectives:

- Primary Purpose: `{purpose}`
- Target Audience: `{audience}`
- Mood and Tone: `{mood}`

Provide complete specifications following the schema exactly.

## Developer

You are a Figma plugin developer. Your tasks:

1. Update ui.html with background and logo image names
2. Follow the JavaScript template
3. Generate and save plugin code with layout implementation
4. Render the image in Figma desktop and provide the file path output by the `render_and_save_image` tool. e.g. `./rendered/xxx.png`
5. Backup plugin

Use these tools in sequence:

1. `update_image_list` - Modify ui.html
2. `read_plugin_template` - Get template code
3. `save_plugin_code` - Save code.js
4. `render_and_save_image` - render in Figma. To use this tool, you have to create unique image name for the rendered image with `create_unique_image_name` tool.
5. `backup_plugin_folder` - Backup plugin

Please implement the Figma plugin code following these steps:

1. Update the `imageList`

in `ui.html` ("`./figma-plugin-related/AdAgentBeta/ui.html`") to include: `{background_image_path}`

2. Read the template from: `"./figma-plugin-related/figma_plugin_template_code.js"`

3. Generate the plugin code using the template and this layout specification: `{layout output from textual director}`

4. Save the generated code to `code.js` in: `"./figma-plugin-related/AdAgentBeta/"`

5. Render the image in Figma desktop and get the file path to the rendered image. Provide the file path output by the `render_and_save_image` tool. e.g. `./rendered/xxx.png`

6. Copy the plugin folder `"./figma-plugin-related/AdAgentBeta"` under the backup folder `"./figma-plugin-backup/"` for backup purpose. Also copy the background image `{full_background_image_path}` under the backup plugin folder. The plugin folder's backup name will be the background image name.

Follow the template structure and properly implement all text elements and CTA button with the specified styling and positioning.

## Design Reviewer

Suppose you are an experienced marketing professional in the advertisement industry reviewing a banner ad image of size `{width}x{height}px` under development.

You are iteratively evaluating the foreground elements (text, buttons, etc.) based on their fit with the background and your experience of a good banner ad in industry.

If the foreground elements are not perfect for produciton in industry, provide key FEEDBACK on ways to improve the foreground elements. If it looks good in overall, we should finish the development.

You should look at the current rendered image to spot any problems. The foreground strategy is intention while the rendered image is actually what has been implemented. So always look at the rendered image!

Here are some rules to write good key FEEDBACK:

- If a feedback hasn't been addressed for over two iterations, stop suggesting it again.
- Carefully compare current rendered image with the foreground strategy to identify any abnormalities, such as foreground elements being partially outside the frame, misaligned, or distorted due to rendering issues. Write down any of these abnormalities in details.
- Carefully compare current rendered image with the background to examine foreground elements overlap with any salient areas of the background (e.g., key objects, patterns, colors). Write down any overlaps that might compromise readability or aesthetics in details.
- Carefully assess the placement of the logo image. Spot if the logo image is too small, too big, being distorted, or not placed in a prominent position. If need to resize the logo, suggest maintaining the aspect ratio. If the logo is hard to read (e.g. the color is similar to the background), suggest adding background to the logo image to make it more prominent.
- Carefully assess if the font type, font color, button color, button shape can be enhanced from a professional marketing expert's perspective.
- Carefully assess if any overlaps among elements such as of text with the

logo or button, or overlaps of button with the logo. If there are overlaps, suggest adjusting the position of the text, button, or logo to avoid overlaps.

- You can also propose better copywrite if you find it necessary.
- Use the provided history of previous iterations if it is not the first iteration. Avoid generating FEEDBACK identical with the FEEDBACK in previous rounds unless the issues raised before have not been solved. Ensure your FEEDBACK build on past refinements.
- Avoid any code blocks in your response. Use clear and concise human language.
- Spot the most critical issues first.

### Foreground Designer (Refinement)

You are a textual director specialized in banner layout and typography. You are iteratively refining the foreground elements (text, buttons, etc.) based on the feedback from an experienced evaluator.

Things to be careful are:

- When you refine, avoid any potential overlapping of elements.
- When you change the position of an element, consider step by step whether the relative positions of other elements—especially those that reference the changed element need to be adjusted to avoid disrupting the layout.
- When change the relative positions, you should always remember to update the position with the new relative

positions, because relative position always affects positions.

- When changing elements, ensure the margins to the edges are considered. Check by calculating if the  $(\text{position} + \text{size})$  exceeds  $(\{\text{background\_width}\}, \{\text{background\_height}\})$  in horizontal and vertical dimension respectively.

## A.7 More Visual Results

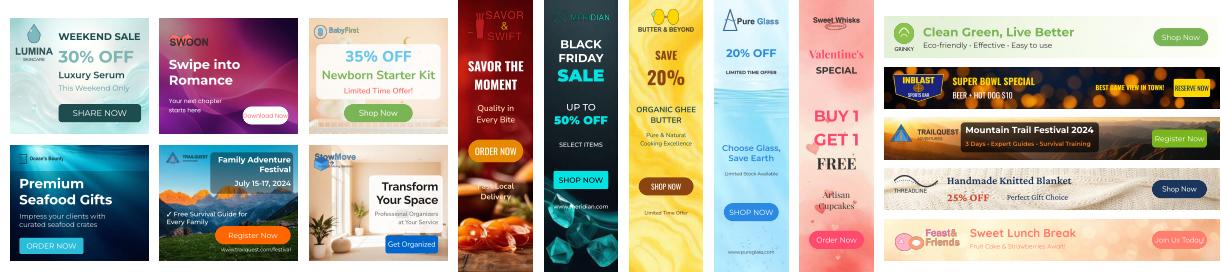
**BannerAgency with Different sizes on BannerRequest400 benchmark.** To showcase the performance of our BannerAgency in different sizes of banner designs, we present more visual results from BannerAgency with different implementations in Fig. 14. Note how SVG implementation always includes decorative elements such as curves, lines, etc.

**More algorithm comparison.** Fig. 15 presents more visual comparisons among algorithms.

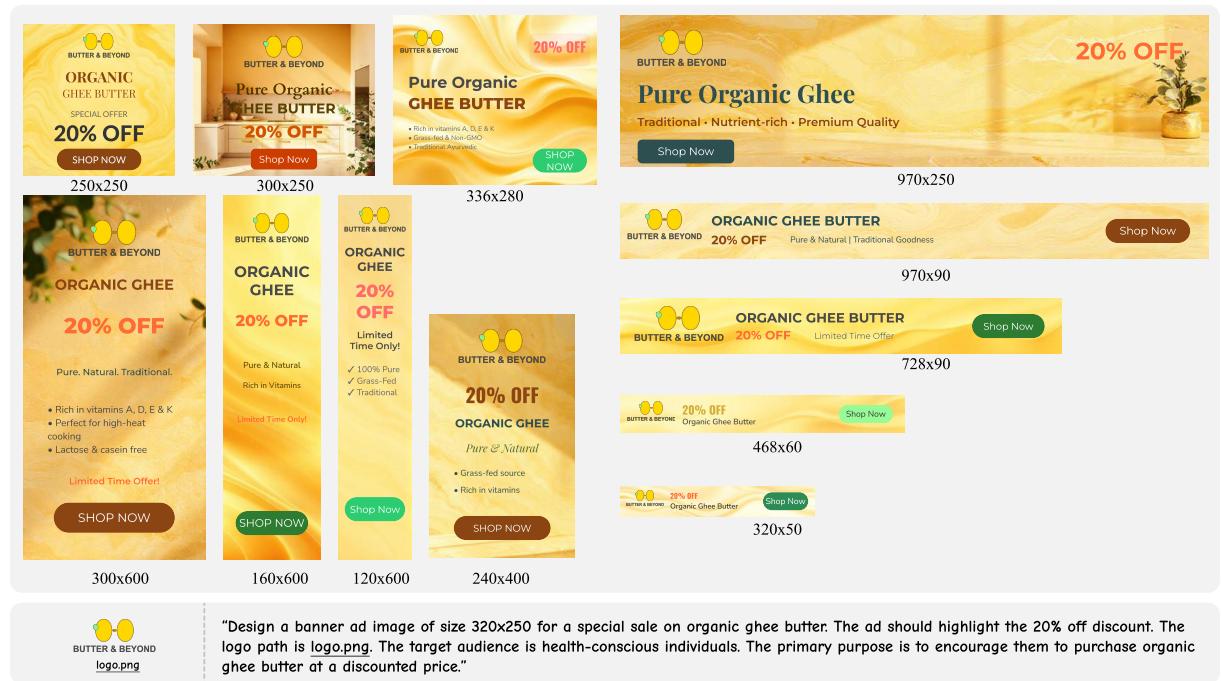
**More banner designs.** We present four more banner designs with the same advertiser but with different target audience and purposes in Fig. 16.



(a) BannerAgency with SVG implementation. Sizes: Skyscraper (160 × 600), leaderboard (728 × 90), medium rectangle (300 × 250).



(b) BannerAgency with Figma implementation. Sizes: Skyscraper (160 × 600), leaderboard (728 × 90), medium rectangle (300 × 250).



(c) Various sizes for same banner request.

Figure 14: More results from BannerAgency of different sizes.

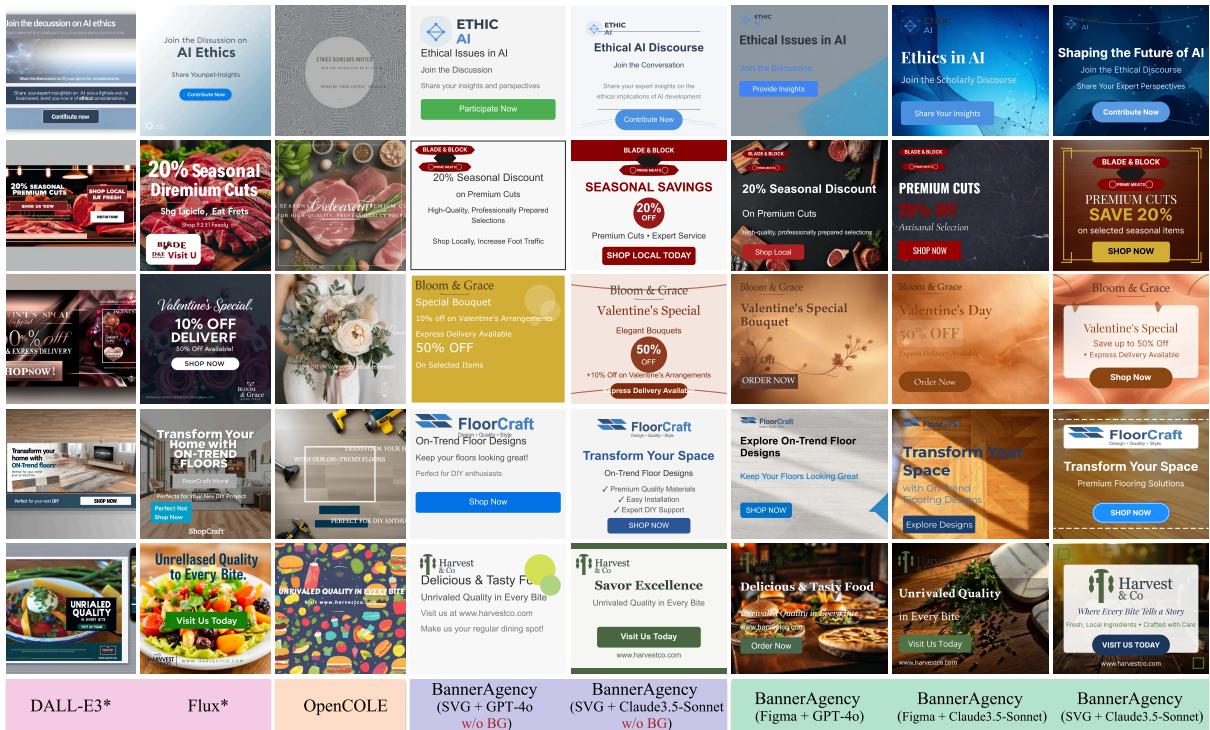


Figure 15: More visual comparisons of different algorithms.

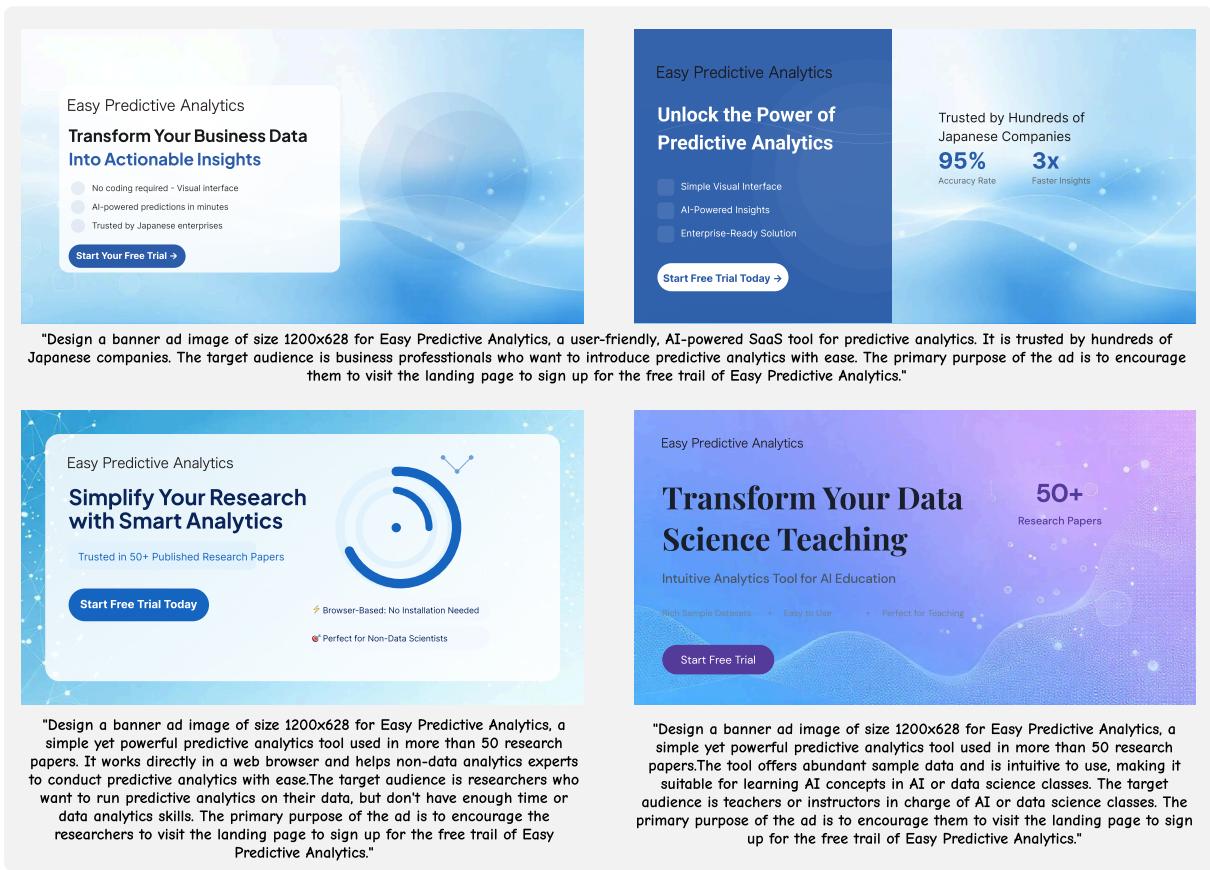


Figure 16: More example banner designs from BannerAgency with the same advertiser but different target audiences and purposes.