# GraphicWeaver: Benchmarking Agentic Planning for Graphic Design Generation

**Dayeon Ki** ☘* **Tianyi Zhou** ☘ **Marine Carpuat** ☘
**Gang Wu** 🍃 **Puneet Mathur** 🍃 **Viswanathan Swaminathan** 🍃

University of Maryland ☘ Adobe Research 🍃
dayeonki@umd.edu

## Abstract

Vision-language model (VLM)-powered agents are increasingly enabling new forms of automation across various human tasks. While prior work has primarily focused on well-defined problems with explicit goals, the capabilities of agents in creative graphic design, where goals are inherently open-ended and subjective, remain largely underexplored. To bridge this gap, we introduce GraphicWeaver, a planning benchmark for graphic design comprising 1,079 diverse user queries and associated images spanning four design categories. Comprehensive experiments with six models reveal that current VLM-based agents struggle to handle such complex planning tasks, which require taking into account both explicit design constraints specified in queries and implicit commonsense design principles. We attribute these failures to challenges in (1) retrieving appropriate parameters for tool usage, (2) understanding spatial relationships across design components, and (3) coordinating dependencies across agents. We envision GraphicWeaver as a challenging yet valuable testbed for advancing VLM agent planning in creative design contexts.[1]

## 1 Introduction

Recent advancements in Vision-Language Models (VLMs) have expanded their potential as general-purpose agents capable of automating a wide range of human tasks. Prior work has evaluated VLM agents in diverse domains, including web navigation and interaction (Zheng et al., 2024; He et al., 2024a; Tian et al., 2025), travel planning (Xie et al., 2024; Jandial et al., 2025), item detection (Kelly et al., 2024), embodied scenarios (Zheng et al., 2023), and online shopping (Koh et al., 2024).

On the other hand, research on the planning capabilities of VLM agents for *creative* design tasks remains limited, primarily due to underspecified open-ended goals from users (Guo et al., 2024; Ge et al., 2025). They require delicate planning that translates a high-level user request into a structured plan composed of executable sub-tasks that collectively produce the final design. This is inherently complex, posing multiple challenges: **(1)** A complex design often requires collaboration involving multiple agents; **(2)** Design planning is usually *long-horizon*, involving a sequence of decisions for expert selection, tool calls, and tool uses, with an expansive tool space to explore (Xie et al., 2024); **(3)** A design plan must accommodate both explicit constraints from user queries (e.g., "the title text color must be white") and implicit constraints inferred through commonsense reasoning (e.g., "the background should contrast with the color of text elements") since user queries are often incomplete with unspecified details (Qian et al., 2024b); **(4)** Assessing design outcomes is inherently subjective, as the notion of what constitutes a *better* design vary across individuals. These challenges raise a key question: To what extent can VLM agents generate cohesive plans for creative design tasks when provided only with open-ended user queries?

We focus on graphic design, a task that remains challenging even for humans as it demands specialized knowledge of professional design tools, often requiring substantial time and effort to learn (Bedford et al., 2006). In this work, we introduce **GraphicWeaver**, a planning benchmark comprising 1,079 realistic user queries paired with associated images. The dataset spans four representative design categories: book cover, business card, postcard, and poster, chosen to capture a broad range of design goals varying in layout composition, textual arrangement, and overall visual organization (§3).

We comprehensively evaluate six VLM-based agents, ranging from smaller open-weight models to larger closed-source ones (§4). As shown in Figure 1, each agent is assessed on its ability to reason

*Work done during internship at Adobe Research.
[1] https://github.com/dayeonki/graphicweaver

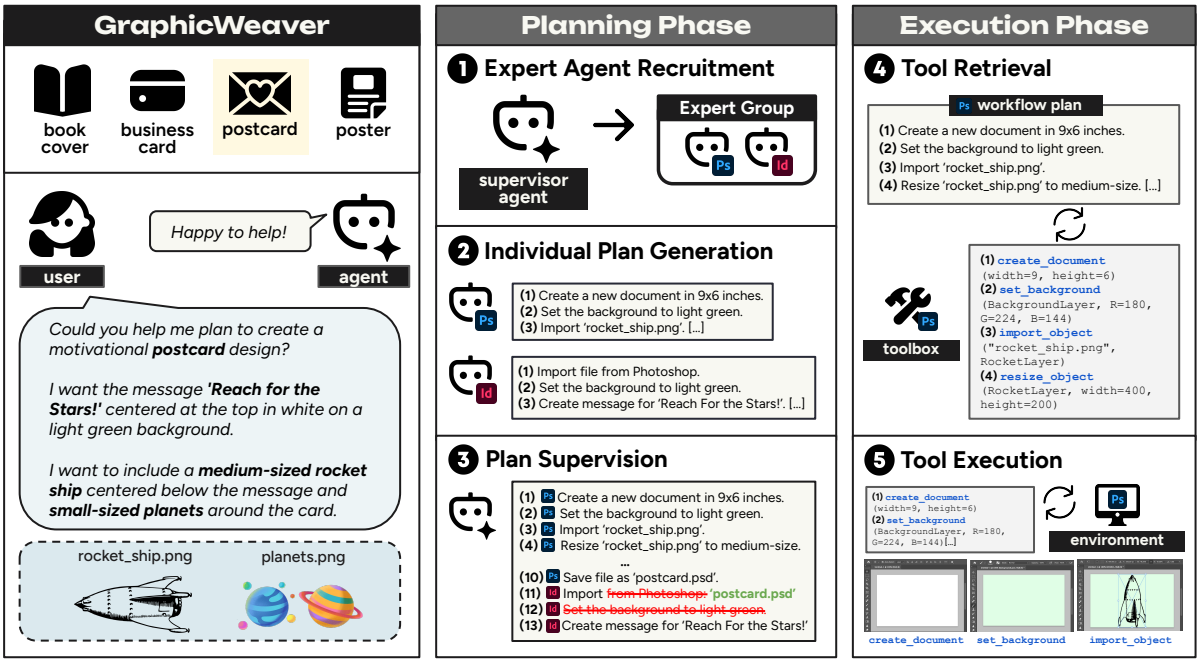


**Figure 1: Overview of GraphicWeaver.** Given a user query and associated image(s), vision-language agents collaboratively *plan* and *execute* the requested task. They generate a plan that satisfies the user's requirements expressed in the query while adhering to implicit commonsense design principles. Based on the finalized plan, each agent retrieves the appropriate tools and executes them within its respective environment.

about task requirements and devise actionable plans (Planning; §5.2) and to retrieve and execute appropriate tools to achieve the desired goal (Execution; §5.3). Our key findings are summarized as follows:

- All evaluated VLM-based agents struggle with complex planning tasks in GraphicWeaver, particularly in integrating both explicit user-specified requirements and implicit commonsense design principles into the generated plans.
- Execution success rates are generally low, with GPT-4.1 (the highest performing model) successfully executing only 62.5% of its generated plans. The resulting design outcomes receive low scores for both alignment with user queries and images, as well as for overall creativity.
- Further analyses reveal three recurring failure modes: (1) retrieval of invalid parameters for tool usage, (2) difficulty in reasoning about spatial relationships of design components, and (3) inadequate coordination across agents.

## 2 Related Work

**VLM-Based Agents.** Leveraging the strengths of Large Vision-Language Models (LVLMs), VLM-based agents have shown strong performance in automating human tasks through tool use (Schick et al., 2023; Qin et al., 2023) and advanced reasoning strategies (Yao et al., 2022; Shinn et al., 2023). Further inspired by human collaboration and its role in improving work efficiency (O'Reilly et al., 1997; Woolley et al., 2015), recent research has proposed multi-agent frameworks in which multiple agents coordinate to solve a shared task (Ding et al., 2023; Shen et al., 2023; Dong et al., 2024; Chen et al., 2024). In particular, studies suggest that assigning specialized roles to agents improves their effectiveness on complex problems (Li et al., 2023; Talebirad and Nadiri, 2023; Du et al., 2024; Hong et al., 2024; Qian et al., 2024a). Similarly, we evaluate VLM agents in a collaborative setting, but in the context of an underexplored problem in this space: graphic design generation.

**Graphic Design Generation.** Graphic design is a form of visual art that combines multimodal elements (e.g., images, texts, and vector symbols) to create aesthetic compositions that effectively communicate user's intent (Cheng et al., 2024). Most prior work has examined specific design *subtasks*—including layout generation (Li et al., 2019; Gupta et al., 2021; Jiang et al., 2023), typography generation (Zhao et al., 2018; Jiang et al., 2019), attribute recognition (Lin et al., 2024), and colorization (Yuan et al., 2021; Qiu et al., 2023)—as well as *single-shot* generation (Hsu et al., 2023; Seol et al., 2024; Yang et al., 2024b). In contrast,

| Category | Example User Query | Example Image(s) | # Train/Test |
|---|---|---|---|
| Book Cover | We need to create a cooking book cover design titled 'Quick & Easy Weekday Meals' with the author's name 'Sarah James' at the top center, above the title, and a subtle green background. The title should be in white, top center, below the author's name, and feature a delicious pasta plate centered in the middle, below the title text. Include the tagline 'Delicious Recipes for Busy Lives' in white, below the title. | | 5/260 |
| Business Card | Create a business card design in teal background for a software company named 'OceanSoft'. Please include the company name in huge white font at the top center. I want to include the tagline 'Sailing to Success' in medium white font placed below the company name and a large wave icon at the bottom center. | | 5/203 |
| Postcard | Could you draft a plan to create a wedding announcement postcard design with the message 'Save the Date' centered at the top in white on a navy blue background? I want to include medium-sized golden wedding rings centered below the message and small elegant florals in the corners. | | 5/260 |
| Poster | I need to create a poster design for a music festival named 'Rhythm Beats' on a green background, featuring a large illustration of a colorful guitar in the center, a catchy title 'Rhythm Beats' in huge white font at the top center, and a tagline 'Feel the Music in Your Soul' in large white font at the bottom center. | | 5/336 |

**Table 1: Actual examples of user queries and images in GraphicWeaver.** Each user query is paired with one or more relevant images that serve as inputs to the design planning process. Train split examples are used as in-context examples.

GraphicWeaver evaluates VLM agents on their ability to plan end-to-end designs through multi-step generation across multiple agents and web-based environments, more closely mirroring real-world graphic design practices (Inoue et al., 2024).

## 3 GraphicWeaver

### 3.1 Overview

We introduce GraphicWeaver, a benchmark for evaluating VLM agents on complex planning and tool use across multiple web-based design environments. To reflect real-world design planning, GraphicWeaver incorporates diverse constraints, including those explicitly specified in the user queries (Figure 1) and those arising from commonsense design principles, such as choosing background colors that contrast with text elements. The benchmark comprises 1,079 query-image pairs across four graphic design categories: book cover, business card, postcard, and poster. It is split into training and test sets: the training set contains 5 queries per category with human-annotated reference plans (20 in total), and the test set contains the remaining 1,059 queries, with detailed examples and statistics reported in Table 1.

### 3.2 Benchmark Construction Pipeline

This section outlines the construction pipeline of GraphicWeaver, which consists of the following steps: environment setting, diverse user query design, image pairing, and human quality checking. All prompts are outlined in Appendix D.1.

**Environment Setting.** In GraphicWeaver, we construct a static, closed sandbox environment for evaluation, ensuring that all VLM agents operate under the same fixed set of resources and eliminate any variability for fair comparison. It also avoids the overhead of building and maintaining custom environments, which is particularly challenging in graphic design domains. We consider three complementary design environments: Adobe Photoshop, Illustrator, and InDesign, each offering distinct capabilities. For each environment, we build a database with 46 tools in which every tool is linked to an executable JavaScript code corresponding to a single mouse or keyboard operation (e.g., creating a new document) (He et al., 2024b) and parameterized only by predefined input fields. As summarized in Table 2, we group tools into four categories: basic operations, drawing, text-related, and object manipulation functions.[2] For additional details on the toolset and environment configuration, refer to Appendix B.1.

**User Query Construction.** To construct diverse queries for GraphicWeaver, we identify key design components and sub-components for each design category, as summarized in Table 3. We begin by collecting 20 screenshots of design projects (5 per category) shared by practitioners on the Behance platform,[3] ensuring that the benchmark reflects authentic, real-world design needs. Three graduate students familiar with Adobe software then collaboratively write realistic user queries and corresponding execution plans, and carry out these plans within our sandbox environment to produce final designs that closely resemble the reference screen-

[2] We derive the tools from Adobe's official tutorials, reflecting commonly used operations among design practitioners.

[3] https://www.behance.net/

| Category | Tool | Input Parameters | Description | Env. |
|---|---|---|---|---|
| **Basic** | `SetBackgroundColor` | `red, green, blue` | Set the background color to desired RGB color. | Ps Ai Id |
| | `SaveDocument` | `fileName, format` | Save the current document into desired format. | Ps Ai Id |
| **Drawing** | `OpacityDrawing` | `layerName, opacity` | Adjust opacity of a drawing. | Ai |
| | `ResizeDrawing` | `layerName, width, height` | Resize a drawing to desired width and height. | Ai |
| **Text** | `ApplyFont` | `layerName, fontName` | Apply font to text. | Ps Ai Id |
| | `RotateText` | `layerName, angle` | Rotate text to desired angle. | Ps Ai Id |
| **Object** | `ImportObject` | `fileName, layerName` | Import an image or object from file path. | Ps Ai Id |
| | `PhotoFilter` | `layerName, filterType, density` | Apply a photo filter to an object with desired density. | Ps |

**Table 2: Subset of available tools in GraphicWeaver.** Each tool is parameterized by predefined input fields for execution. **Env.**: Environment which supports the execution of a specific tool. Full list of available tools is in Appendix Table 6.

| Category | Design Components | Required? |
|---|---|---|
| **Book Cover** | Background color | ✓ |
| | Title (content, size, color, position) | ✓ |
| | Author Name (content, size, color, position) | ✓ |
| | Subtitle (content, size, color, position) | ✗ |
| | Tagline (content, size, color, position) | ✗ |
| | Image (size, position, image URL, caption) | ✓ |
| **Business Card** | Background color | ✓ |
| | Brand Name (content, size, color, position) | ✓ |
| | Tagline (content, size, color, position) | ✗ |
| | Contact Details (content, size, color, position) | ✗ |
| | Image (size, position, image URL, caption) | ✓ |
| **Postcard** | Background color | ✓ |
| | Message (content, size, color, position) | ✓ |
| | Image (size, position, image URL, caption) | ✓ |
| **Poster** | Background color | ✓ |
| | Title (content, size, color, position) | ✓ |
| | Tagline (content, size, color, position) | ✗ |
| | Image (size, position, image URL, caption) | ✓ |

**Table 3: Key design components and sub-components for each design category.** Sub-components are listed in parentheses. **Required?:** Whether the component is required in the user query during query construction process.

shots (see Appendix B.2 for more details).

For each design category, the identified design components are used as placeholders to construct query skeletons, which serve as prompt templates (Qian et al., 2024b; Xie et al., 2024; Yoran et al., 2024). We then prompt GPT-4 (Achiam et al., 2023) to randomly fill these placeholders and manually introduce additional variation in query openers (e.g., "Please help me create a design [...]", "Could you provide me a design [...]") to better reflect the range of natural phrasing in real user queries, as illustrated in Table 1.

Directly using model-generated queries often results in many with highly similar design concepts (i.e., multiple postcards themed around "Birthday"). To diversify this, we remove near-duplicates by (1) discarding queries with overlapping bi-grams in any design components, and (2) filtering out semantically similar pairs with SentenceBERT similarity scores above 0.8 (Reimers and Gurevych, 2019).

**Image Pairing.** Each validated user query includes a brief description of the image(s) required for the design (see Figure 1). To ground these descriptions in concrete visual assets, we construct an image retrieval pool by collecting vector illustrations from OpenCLIPArt[4] and Public Domain Vectors.[5] In total, we gather 274K caption-image URL pairs as our retrieval pool. For each image, we then retrieve the top-3 candidates whose captions have the highest SentenceBERT similarity scores with the query's image description.[6]

**Quality Control.** We first conduct an automatic evaluation to assess the quality of the user queries and the top-3 retrieved images. For each query, we prompt GPT-o1[7] to: (Q1) identify the key design components and rate how well each contributes to the coherence of the final design on a five-point Likert scale (1:Not at all, 5:Completely), and (Q2) rank the three retrieved image candidates from 1 (best fit) to 3 (least fit) based on their relevance.

To validate these automatic annotations, we conduct a human verification study on a stratified random sample of 200 user queries (50 per design category) using the same criteria. We observe substantial agreement between GPT-o1 and human judgments (Cohen's $\kappa$=0.586 for Q1 and Kendall's $\tau$=0.671 for Q2). We discard queries in which any design component receives a rating of 1 or 2 and retain only the image ranked as the best fit. Further details are provided in Appendix B.3.

### 3.3 Evaluation

We evaluate both the plans and the execution outcomes offered by agents along multiple dimensions.

[4] https://openclipart.org/

[5] https://publicdomainvectors.org/

[6] All images will be released under the Creative Commonsense Zero (CC0) license. The average text/image counts per query are: book covers (3.05/1.00), business cards (2.15/1.33), postcards (1.03/1.28), and posters (1.99/1.04).

[7] https://openai.com/o1/

Detailed prompts are provided in Appendix D.2.

**(1) Planning Evaluation**

- **Delivery Rate:** This assesses whether agents can successfully deliver a final plan within a limited number of steps, determined by difficulty: Easy (1 expert, 10 steps), Medium (2 experts, 20 steps), Hard (3 experts, 30 steps).[8] Plans that exceed the limit are counted as failures (Xie et al., 2024).
- **Design Pass Rate:** This measures if the plan correctly reflects both *explicit* user-specified constraints and *implicit* commonsense principles. We prompt GPT-5[9] to score color, text, and imagery alignment on a five-point Likert scale.
- **Step Efficiency:** This metric represents the proportion of unique (non-duplicate) to total steps.
- **Expert Use Efficiency:** This metric captures how effectively a plan minimizes switching between expert agents. For a plan $p$ with $N$ steps and $E$ unique experts:

$$\textbf{Eff.}(p) = \frac{E-1}{\max\left(1, \sum_{i=1}^{N} \mathbb{1}(\text{expert}_i \neq \text{expert}_{i-1})\right)} \quad (1)$$

**(2) Execution Evaluation**

- **Execution Success Rate:** This measures the proportion of plans executable without errors.
- **Fidelity:** This metric captures whether the required user images appear in the final outcome, measured via template matching (`opencv`).
- **Content Similarity:** Semantic alignment between the user query and the final outcome, measured using CLIPScore (Hessel et al., 2021).
- **VQA Pass Rate:** This metric measures how well the final design outcome reflects the components specified in the user query, using Visual Question-Answering (VQA) (Agrawal et al., 2016). For each query, we use GPT-4 to generate component-based questions,[10] and then answer them with LLAVA-1.5 13B (Liu et al., 2024). Pass rate is the average Yes/No accuracy (Zhao et al., 2024).
- **Creativity:** Following Torrance (1966); Runco and Jaeger (2012), we assess Originality (uniqueness) and Elaboration (extent to which the design expands on the user query by adding meaningful details) on a five-point Likert scale using GPT-5.

[8] Step limits are based on human-annotated plans.

[9] https://openai.com/gpt-5/

[10] On average, 9.07, 10.0, 7.89, 8.70 questions are generated per user query for book covers, business cards, postcards, and posters, respectively. Examples are in Appendix B.4.

## 4 Experiment Setup

**Models.** We focus on VLMs with input context lengths of at least 32K tokens due to the extensive information required for planning. We evaluate four open-weight models of varying sizes and families: QWEN-2.5-VL 7B and 32B (Bai et al., 2025), GEMMA-3 12B and 27B (Team et al., 2025), and two closed-source models: GPT-O4-MINI[11] and GPT-4.1.[12] We use temperature of 0.0.[13]

**Planning Strategies.** To study the impact of agentic planning, we compare two strategies: direct and agentic. In the direct mode, a single VLM agent generates the entire design plan without invoking the multi-step planning process illustrated in Figure 1. In the agentic mode, we adopt a hierarchical framework where a supervisor VLM agent $a_s$ coordinates a group of expert agents $a_i \in \mathcal{A}$ for planning (Fourney et al., 2024; Zhang et al., 2025).

Specifically, for each user query, the supervisor agent $a_s$ assembles an expert group $\mathcal{A}$ based on predefined role descriptions and assigns a high-level goal to each expert agent $a_i \in \mathcal{A}$. We instantiate three design experts, each aligned with one of our three design environments (§3.2) and prompted with distinct expertise and responsibilites:

- **Photo Editor:** An agent with an expertise in Adobe Photoshop, responsible for image editing, color correction, and applying filters.
- **Vector Graphic Editor:** An agent with an expertise in Adobe Illustrator, focused on creating and editing vector illustrations.
- **Layout Designer:** An agent with an expertise in Adobe InDesign, responsible for customizing layout templates, exporting files, and integrating text with visual elements.

Each expert VLM agent $a_i$ then proposes its own plan $p_i$ conditioned on its assigned goal. To emulate human problem-solving process (Zhu et al., 2023), each $a_i$ is instructed to plan with a sequence of actionable steps (Yang et al., 2024a; Wu et al., 2024b; Zheng et al., 2025), which further facilitates accurate tool retrieval (Huang et al., 2024). The supervisor then overlooks these individual plans in terms of the overall goal and merges them into a single cohesive plan $p_s$, which is then executed sequentially, yielding the final design outcome.

[11] https://openai/com/o3-and-o4

[12] https://openai.com/index/gpt-4-1/

[13] HuggingFace model names are in Appendix A.

| Model | Planning | | | | Execution | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Delivery Rate (%) | Design Pass Rate (%) | Step Eff. | Expert Use Eff. | Success Rate (%) | Fidelity | Content Similarity | VQA Pass Rate (%) | Creative (O) | Creative (E) |
| ***Direct mode*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 27.1 | 22.5 | 45.5 | 0.98 | 10.3 | 0.01 | 5.70 | 16.0 | 1.05 | 0.64 |
| **QWEN-2.5-VL 32B** | 57.3 | 19.7 | 56.0 | **1.00** | 17.5 | 0.04 | 6.62 | 19.0 | 1.18 | 1.07 |
| **GEMMA-3 12B** | 37.7 | 19.0 | 63.8 | 0.99 | 14.9 | 0.03 | 6.93 | 13.6 | 1.18 | 0.76 |
| **GEMMA-3 27B** | **67.8** | 24.7 | 70.9 | **1.00** | 22.3 | 0.09 | 9.30 | 17.7 | 1.09 | 1.10 |
| **GPT-O4-MINI** | 46.2 | 23.3 | 79.5 | **1.00** | 28.0 | 0.11 | 8.70 | 21.5 | 1.14 | 1.13 |
| **GPT-4.1** | 30.3 | **30.1** | **82.5** | **1.00** | **29.5** | **0.13** | **13.1** | **23.3** | **1.39** | **1.36** |
| ***Agentic mode*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 15.2 | 51.8 | 92.0 | **1.00** | 39.4 | 0.15 | 22.5 | 37.6 | 1.77 | 1.59 |
| **QWEN-2.5-VL 32B** | 39.6 | 49.7 | 92.4 | **1.00** | 61.3 | 0.20 | 22.3 | 35.5 | 1.89 | 1.65 |
| **GEMMA-3 12B** | 27.3 | 54.5 | 96.5 | **1.00** | 58.1 | 0.17 | 21.0 | 36.2 | 2.01 | 1.68 |
| **GEMMA-3 27B** | **51.4** | 54.4 | 93.1 | **1.00** | 55.4 | 0.20 | 26.7 | 44.7 | 1.98 | 2.04 |
| **GPT-O4-MINI** | 20.8 | 51.1 | 97.1 | **1.00** | 56.5 | 0.20 | 28.3 | **44.9** | 1.78 | 1.99 |
| **GPT-4.1** | 13.9 | **56.8** | **94.7** | **1.00** | **62.5** | **0.21** | **29.3** | 44.7 | **2.02** | **2.06** |

**Table 4: Planning and execution results of different VLMs and planning strategies on GraphicWeaver.** For each planning mode, column-wise best scores are **bolded** and second-best scores are underlined. All metrics are interpreted as higher values for better performance (↑). Scores are aggregated over the four design categories; per category results are provided in Appendix C.1.

# 5 Results

We discuss the performance of various VLM agents across planning strategies (§5.1) on GraphicWeaver in terms of planning (§5.2) and execution (§5.3).

## 5.1 Direct vs. Agentic Mode

As shown in Table 4, all VLM agents perform worse in the direct mode than in the agentic mode on nearly all metrics, with the exception of delivery rate. We attribute the higher delivery rate in direct mode to the fact that directly generated plans are typically shorter, and thus more easily satisfy the maximum number of steps constraint. Expert use efficiency also remains close to perfect in the direct mode, since most models tend to persist with a single expert agent throughout the entire plan rather than switching between experts.

Other planning and execution metrics are substantially lower in the direct mode, with the largest gap reaching over 30%. For instance, design pass rate drops from 53.1% (agentic) to 23.2% (direct) on average, step efficiency from 94.3% to 66.3%, and execution success rate from 55.5% to 20.4%. Together, these results highlight the importance of an agentic framework for complex design planning, suggesting that decomposing the task and coordinating specialized experts is more effective than asking VLMs to plan in one-shot.

## 5.2 Planning

In the agentic mode, all evaluated VLM agents still struggle to deliver plans within the step limit on GraphicWeaver. GEMMA-3 27B achieves the highest rate at only 51.4%, with all other models remaining below 40%. Design pass rates are also modest, indicating difficulty in jointly satisfying explicit user-specified requirements and implicit commonsense design constraints: GPT-4.1 achieves the highest design pass rate at 56.8%, while QWEN-2.5-VL 32B scores the lowest (49.7%). In contrast, step efficiency and expert use efficiency remain relatively high, suggesting that the generated plans contain only a few redundant steps and avoid unnecessary switching between expert agents.

Across the four design categories, poster plans achieve the highest design pass rate (54.8%), whereas business card plans exhibit the lowest delivery and design pass rates. This is likely due to the higher number of text elements and, for some queries, the added complexity of planning designs for both the front and back of the card. Detailed per-category results are reported in Appendix C.1.

## 5.3 Execution

Execution metrics are also low across all evaluated VLM agents. The proportion of plans that fully execute ranges from 39.4% for QWEN-2.5-VL 32B to at most 62.5% for GPT-4.1. Even among successfully executed plans, the resulting design outcomes

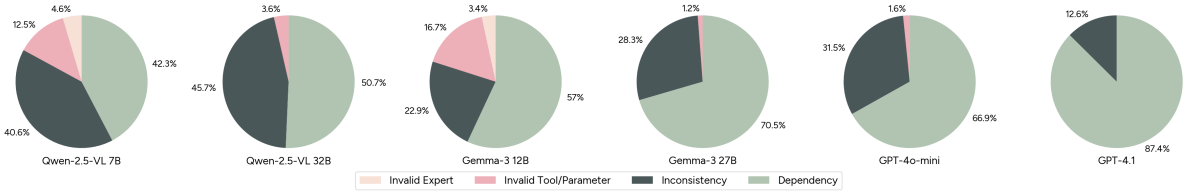


**Figure 2: Error distribution for each VLM.** For all tested models, majority of the errors stem from inconsistent planning across design components (Inconsistency) or failing to resolve dependencies across steps in design plans (Dependency).

receive low scores on measures of alignment with both user queries and images. For example, fidelity to the user images remains around 0.10-0.20 across models, with the best score of 0.21 achieved by GPT-4.1. Semantic similarity between user queries and final designs is likewise limited, ranging from 21.0% for GEMMA-3 12B to a maximum of 29.3% for GPT-4.1. None of the VLMs achieve a VQA pass rate above random chance (50% for Yes/No accuracy), with the highest score being 44.9% for GPT-O4-MINI. Additionally, the overall creativity of the generated design outcomes, as judged by GPT-5, remains low on both originality (O) and elaboration (E), with all models scoring roughly between 1.5 and 2.0 on these scales.

Taken together, these results show that GraphicWeaver presents a substantial challenge for current VLM agents: even the state-of-the-art VLMs, when equipped with agentic planning strategy, still fall short of planning and successful execution in complex graphic design tasks.

## 6 Analysis

We present further analysis on the agentic mode.

### 6.1 Expert & Tool-Use Analysis

We visualize the distribution and flow of expert agents in the generated plans in Figure 3. On average, 2.11 expert agents are recruited per user query. Across models, the Photo Editor and Layout Designer are most frequently paired, reflecting their complementary roles, whereas the Vector Graphic Editor is used less often since most queries already specify images to incorporate into the planning. The Layout Designer is responsible for the largest share of steps (15.2 on average), followed by the Vector Graphic Editor (10.4) and the Photo Editor (8.24). In terms of agent ordering, models exhibit clear preferences: the most common transition is Photo Editor → Layout Designer (43.3% of transitions), followed by Layout Designer → Photo Editor (21.5%) and plans that use only the Layout Designer accounting for 17.8% of cases.

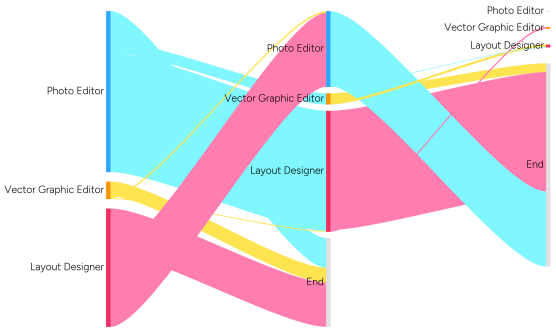


**Figure 3: Aggregated expert agent usage ordering.** Most common order is Photo Editor → Layout Designer.

As further detailed in Appendix C.2, expert agents consistently rely on only a small subset of their available tools: the Photo Editor primarily performs object manipulation, while Layout Designer focuses on text operations. The most frequent tool usage sequences closely resemble workflows in human-annotated references (e.g., document creation → set background color → import and manipulate images → text), indicating that previously observed human-like reasoning patterns (Wei et al., 2023) possibly extend to graphic design planning.

### 6.2 Error Analysis

We categorize step-level errors in the generated plans into four types, as illustrated in Figure 2: **(1) Invalid Expert**, where a non-existent agent is assigned; **(2) Invalid Tool/Parameter**, where tools outside the defined toolset or non-defined parameters are selected; **(3) Inconsistency**, where a step's assumptions conflict with design components, and **(4) Dependency**, where an operation references an object that is unavailable or uninitialized. We summarize our main findings as follows:

1. **Dependency errors dominate**, accounting for 62.5% of all errors. These include both *local* (within-agent) and *global* (across-agent) dependencies. All models particularly struggle with

global dependencies, often failing to correctly reference objects instantiated by other agents. We also observe that the proportion of dependency errors among all errors tends to increase for larger, closed-source models.

2. **Inconsistency errors are also common**, accounting for 30.3% of errors across all tested models. These typically occur when a plan contains mutually incompatible assumptions about the target design—for example, fist specifying that an element should be centered, then later introducing steps that place other elements in the same position. This is consistent with previously reported spatial reasoning limitations of VLMs (Yamada et al., 2024; Wu et al., 2024a).
3. **Invalid tool or parameter errors** are particularly common in smaller, open-weight models (the QWEN-2.5-VL and GEMMA-3 families). These errors often stem from hallucinated tools (e.g., `CreateTextAndResize`) or incorrect tool choices (e.g., mapping "*Move the title to the center.*" to `RepositionObject` instead of `RepositionText`), even when the full toolset is provided, underscoring the need for more reliable tool retrieval mechanisms.

Overall, VLM agents primarily struggle with global dependency handling and spatial reasoning, highlighting the need for more robust reasoning over design components and improved dependency resolution approaches.

## 7 Conclusion

We introduce GraphicWeaver, a planning benchmark grounded in real-world graphic design needs, to assess the complex design planing and tool-use capabilities of current vision-language agents. Our comprehensive evaluation of six VLMs show that GraphicWeaver remains highly challenging even for state-of-the-art models with an agentic planning strategy: agents struggle both to reason over user queries and to devise actionable plans, as well as to retrieve and execute appropriate tools to achieve the target design outcomes. Error analysis further reveals systematic weaknesses in tool selection, understanding spatial relationships across design components, and, most critically, recognizing and handling global dependencies across agents.

We envision GraphicWeaver as a stepping stone toward building more capable graphic design assistants, grounded in realistic tasks and constraints. We hope our work spurs future research on advancing planning in more open-ended creative tasks.

## 8 Limitation

Our study comes with certain limitations:

- GraphicWeaver assumes a scenario in which user queries explicitly specify the text and image elements, as well as the precise attributes such as color and text position. However, in realistic settings, users may not always specify or even know exactly what images to include in a design, or they may express their requests at a very high-level (Ge et al., 2025). Future works can explore scenarios where user input is limited, requiring models to seek clarification through interactions with users (Qian et al., 2024b; Li et al., 2024).
- The number of tools available in GraphicWeaver is currently limited to a fixed set of 46, as each corresponding JavaScript code was manually written by the authors. This set is not exhaustive of all possible tools within the three design environments. Future works can investigate automated methods for dynamically generating and retrieving tools (Yuan et al., 2024) or integrate a retrieval-augmented generation module (Lewis et al., 2021) into tool retrieval pipeline to enable agents make more informed decisions. Additionally, GraphicWeaver is currently limited to four design categories (book cover, business card, postcard, poster) and only considers three design expert agents, which does not cover the full range of design variants in graphic design tasks.
- Our experiments primarily focus on evaluating the performance of VLMs' capabilities in design planning. Supporting Large Language Models (LLMs) would require a different setup, including prompting with image captions instead of raw images, which is available in GraphicWeaver. Moreover, our agents operate in a stateless planning environment, relying only on the user query and the provided toolset. This allows us to isolate and evaluate the VLMs' planning capabilities, independent of environmental feedback. We view extending our work to support LLMs and perceptual tools as a valuable future direction.
- The scope of our experiments is constrained by computational and API budgets: we focus on open-weight models that can be run locally and closed-source models within our cost limits. Consequently, our findings may not fully generalize to other model families or larger-scale models.

## Acknowledgments

We would like to thank our collaborators at Adobe Research for their valuable feedback, including Alexa Siu, Tong Sun, Saayan Mitra, and Stefano Petrangeli. Dayeon is especially grateful for the Adobe Research intern cohort for making the internship experience memorable, including Nishant Balepur, Dang Nguyen, Vishakh Padmakumar, Paiheng Xu, Hyunji Lee, and Yoonjoo Lee.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. Vqa: Visual question answering. *Preprint*, arXiv:1505.00468.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Tim Bedford, John Quigley, and Lesley Walls. 2006. Expert elicitation for reliable system design. *Statistical Science*, 21(4).

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. 2024. Autoagents: a framework for automatic agent generation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

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

Shiying Ding, Xinyi Chen, Yan Fang, Wenrui Liu, Yiwu Qiu, and Chunlei Chai. 2023. Designgpt: Multi-agent collaboration in design. *Preprint*, arXiv:2311.11591.

Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, and Yi Yang. 2024. Villageragent: A graph-based multi-agent framework for coordinating complex task dependencies in minecraft. *Preprint*, arXiv:2406.05720.

Zhuoyun Du, Chen Qian, Wei Liu, Zihao Xie, Yifei Wang, Yufan Dang, Weize Chen, and Cheng Yang. 2024. Multi-agent software development through cross-team collaboration. *Preprint*, arXiv:2406.08979.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *Preprint*, arXiv:2411.04468.

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

Yuxuan Guo, Shaohui Peng, Jiaming Guo, Di Huang, Xishan Zhang, Rui Zhang, Yifan Hao, Ling Li, Zikang Tian, Mingju Gao, Yutai Li, Yiming Gan, Shuai Liang, Zihao Zhang, Zidong Du, Qi Guo, Xing Hu, and Yunji Chen. 2024. Luban: Building open-ended creative agents via autonomous embodied verification. *Preprint*, arXiv:2405.15414.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. 2021. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024a. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024b. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. *Preprint*, arXiv:2308.00352.

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

Tenghao Huang, Dongwon Jung, Vaibhav Kumar, Mohammad Kachuee, Xiang Li, Puyang Xu, and Muhao Chen. 2024. Planning and editing what you retrieve for enhanced tool learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 975–988, Mexico City, Mexico. Association for Computational Linguistics.

Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. 2024. Opencole: Towards reproducible automatic graphic design generation. *Preprint*, arXiv:2406.08232.

Surgan Jandial, Yinong Oliver Wang, Andrea Bajcsy, and Fernando De la Torre. 2025. On the fine-grained planning abilities of VLM web agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25347–25380, Suzhou, China. Association for Computational Linguistics.

Shuhui Jiang, Zhaowen Wang, Aaron Hertzmann, Hailin Jin, and Yun Fu. 2019. Visual font pairing. *IEEE Transactions on Multimedia*, 22(8):2086–2097.

Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. 2023. Layoutformer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18403–18412.

Chris Kelly, Luhui Hu, Bang Yang, Yu Tian, Deshun Yang, Cindy Yang, Zaoshan Huang, Zihao Li, Jiayin Hu, and Yuexian Zou. 2024. Visiongpt: Vision-language understanding agent using generalized multimodal framework. *Preprint*, arXiv:2403.09027.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: communicative agents for "mind" exploration of large language model society. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. 2019. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*.

Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. 2024. Mediq: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Charles O'Reilly, Katherine Phillips, and Sigal Barsade. 1997. Group demography and innovation: Does diversity help? *Research on managing groups and teams*, 1.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024a. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024b. Tell me more! towards implicit user intention understanding of language model driven agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1113, Bangkok, Thailand. Association for Computational Linguistics.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Qianru Qiu, Xueting Wang, and Mayu Otani. 2023. Multimodal color recommendation in vector graphic documents. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4003–4011.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mark A Runco and Garrett J Jaeger. 2012. The standard definition of creativity. *Creativity research journal*, 24(1):92–96.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Jaejung Seol, Seojun Kim, and Jaejun Yoo. 2024. Posterllama: Bridging design ability of langauge model to contents-aware layout generation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXII*, page 451–468, Berlin, Heidelberg. Springer-Verlag.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Preprint*, arXiv:2303.17580.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *Preprint*, arXiv:2306.03314.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Shulin Tian, Ziniu Zhang, Liangyu Chen, and Ziwei Liu. 2025. MMInA: Benchmarking multihop multimodal Internet agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13682–13697, Vienna, Austria. Association for Computational Linguistics.

E Paul Torrance. 1966. Torrance tests of creative thinking. *Educational and psychological measurement*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Anita Williams Woolley, Christopher F. Chabris, Alex

Pentland, Nada Hashmi, and Thomas W. Malone. 2015. Collective intelligence and group performance. *Current Directions in Psychological Science*, 24(6):420–424.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024a. Mind's eye of LLMs: Visualization-of-thought elicits spatial reasoning in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. 2024b. Os-copilot: Towards generalist computer agents with self-improvement. *Preprint*, arXiv:2402.07456.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *Forty-first International Conference on Machine Learning*.

Yutaro Yamada, Yihan Bao, Andrew Kyle Lampinen, Jungo Kasai, and Ilker Yildirim. 2024. Evaluating spatial understanding of large language models. *Transactions on Machine Learning Research*.

Ruihan Yang, Jiangjie Chen, Yikai Zhang, Siyu Yuan, Aili Chen, Kyle Richardson, Yanghua Xiao, and Deqing Yang. 2024a. Selfgoal: Your language agents already know how to achieve high-level goals. *Preprint*, arXiv:2406.04784.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. 2024. Assistantbench: Can web agents solve realistic and time-consuming tasks? *Preprint*, arXiv:2407.15711.

Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R. Fung, Hao Peng, and Heng Ji. 2024. Craft: Customizing llms by creating and retrieving from specialized toolsets. *Preprint*, arXiv:2309.17428.

Lin-Ping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, and Huamin Qu. 2021. Infocolorizer: Interactive recommendation of color palettes for infographics. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4252–4266.

Cong Zhang, Xin Deik Goh, Dexun Li, Hao Zhang, and Yong Liu. 2025. Planning with multi-constraints via collaborative language agents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10054–10082, Abu Dhabi, UAE. Association for Computational Linguistics.

Hengyuan Zhao, Pan Zhou, Difei Gao, Zechen Bai, and Mike Zheng Shou. 2024. LOVA3: Learning to visual question answering, asking and assessment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Nanxuan Zhao, Ying Cao, and Rynson WH Lau. 2018. Modeling fonts in context: Font prediction on web designs. In *Computer Graphics Forum*, volume 37, pages 385–395. Wiley Online Library.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v(ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Sipeng Zheng, Jiazheng Liu, Yicheng Feng, and Zongqing Lu. 2023. Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. *arXiv preprint arXiv:2310.13255*.

Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Zilong Zheng, and Yitao Liang. 2025. Mcu: An evaluation framework for open-ended game agents. *Preprint*, arXiv:2310.08367.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *Preprint*, arXiv:2305.17144.

# A HuggingFace Models

HuggingFace model names for open-weight models are listed in Table 5.

| Model | HuggingFace Name |
|---|---|
| QWEN-2.5-VL 7B | `Qwen/Qwen2.5-VL-7B-Instruct` |
| QWEN-2.5-VL 32B | `Qwen/Qwen2.5-VL-32B-Instruct` |
| GEMMA-3 12B | `google/gemma-3-12b-it` |
| GEMMA-3 27B | `google/gemma-3-27b-it` |

**Table 5: HuggingFace model names for the tested open-weight models.**

# B GraphicWeaver Construction Pipeline

## B.1 Environment Setting

The complete list of 46 available tools in GraphicWeaver is provided in Table 6. All tools are derived based on Adobe's official tutorial videos, which provides a diverse range of commonly used operations by graphic design practitioners. In Figure 4, we further show an example of one executable JavaScript code for `AdjustBC` tool for the Photo Editor agent.

## B.2 Human Annotation

We present examples of human-annotated user queries for each design category in Table 7. Annotators are instructed to use the same design environment(s) as those employed in the original reference designs.

## B.3 Quality Control

We detail the human annotation process as part of constructing GraphicWeaver. We built our custom annotation interface as illustrated in Figure 5. We invited 8 students to participate and provide a compensation of 10 USD gift card each. Before the survey, we show examples of both successful and failed cases to provide some context of annotation standards to annotators.

As part of the pre-survey, annotators were asked two questions on a five-point Likert scale: **(1) Design tool usage:** How often do you use design tools in daily work and life? (1:Never, 5:Always) and **(2) Adobe software usage:** How familiar are you in using Adobe software (e.g., Photoshop, Illustrator)? (1:Not familiar at all, 5: Extremely familiar). Of the 8 annotators, for design tool usage, 3 responded "Never" (Never in the past month), 4 "Rarely" (Fewer than once a week), and 1 "Sometimes" (two or three times a week). For Adobe software usage, 3 were "Not familiar at all" (have never used it before), 2 "Slightly familiar" (have some basic knowledge but have rarely used it), and 3 "Moderately familiar" (can perform simple tasks but may need guidance for more complex features).

## B.4 VQA Examples

We present several examples of generated questions for measuring VQA pass rate as part of the execution evaluation are detailed in Table 8.

# C Detailed Results

## C.1 Planning & Execution Evaluation

We provide the full numerical results by VLM and design category for all planning and execution metrics in Table 9 (direct) and Table 10 (agentic).

## C.2 Tool-Use Analysis

We present the top-3 most common tool-use sequences for each VLM in Table 11. We find that many of the sequences closely follow human-annotated workflows, typically starting with document creation (`CreateDocument` or `CreateDocumentCustom`), setting the background color (`SetBackgroundColor`), importing images as objects (`ImportObject`), manipulating the imported object (such as using `ResizeObject`, `RepositionObject`, etc.), saving the document (`SaveDocument`), and manipulating text elements (`CreateText`, `ApplyFont`, `ColorText`, etc.).

| Category | Tool | Input Parameters | Description | Env. |
|---|---|---|---|---|
| **Basic** | CreateDocument | docType | Create new document with pre-defined dimensions. | Ps Ai Id |
| | CreateDocumentCustom | width, height | Create new document with desired width and height values. | Ps Ai Id |
| | SetBackgroundColor | red, green, blue | Set the background color to desired RGB color. | Ps Ai Id |
| | SaveDocument | fileName, format | Save the current document into desired format. | Ps Ai Id |
| **Drawing** | DrawCircle | layerName, radius, red, green, blue | Draw a circle of desired radius and RGB color. | Ai |
| | DrawEllipse | layerName, majorRadius, minorRadius, red, green, blue | Draw an ellipse of desired radius and RGB color. | Ai |
| | DrawLine | layerName, startX, startY, endX, endY, strokeWidth, red, green, blue | Draw a line of desired length, stroke, and RGB color. | Ai |
| | DrawPolygon | layerName, sides, radius, red, green, blue | Draw a polygon of desired number of sides, radius, and RGB color. | Ai |
| | DrawRectangle | layerName, width, height, red, green, blue | Draw a rectangle of desired size and RGB color. | Ai |
| | DrawStar | layerName, numPoints, radius, red, green, blue | Draw a star of desired number of points, radius, and RGB color. | Ai |
| | DrawTriangle | layerName, base, height, red, green, blue | Draw a triangle of desired size and RGB color. | Ai |
| | OpacityDrawing | layerName, opacity | Adjust opacity of a drawing. | Ai |
| | RemoveDrawing | layerName | Remove a drawing. | Ai |
| | RepositionDrawing | layerName, posX, posY | Reposition a drawing to desired x and y-axis position. | Ai |
| | ResizeDrawing | layerName, width, height | Resize a drawing to desired width and height. | Ai |
| | RotateDrawing | layerName, angle | Rotate a drawing to desired angle. | Ai |
| | StorkeDrawing | layerName, strokeWidth, red, green, blue | Adjust stroke of a drawing with desired width and RGB color. | Ai |
| **Text** | AlignText | layerName, alignment | Align text to desired alignment (left, center, right). | Ps Ai Id |
| | ApplyFont | layerName, fontName | Apply font to text. | Ps Ai Id |
| | ArrangeText | layerName, arrangement | Arrange text to desired arrangement (front, frontward, back, backward). | Ps Ai Id |
| | ColorText | layerName, red, green, blue | Color text to desired RGB color. | Ps Ai Id |
| | CreateText | layerName, textString | Create a new text (default to Arial font). | Ps Ai Id |
| | OpacityText | layerName, opacity | Adjust opacity of text. | Ps Ai |
| | RemoveText | layerName | Remove text. | Ps Ai Id |
| | RepositionText | layerName, posX, posY | Reposition text to desired x and y-axis position. | Ps Ai Id |
| | ResizeText | layerName, fontSize | Resize text to desired font size. | Ps Ai Id |
| | RotateText | layerName, angle | Rotate text to desired angle. | Ps Ai Id |
| | StrokeText | layerName, strokeWidth, red, green, blue | Adjust stroke of text with desired width and RGB color. | Ai Id |
| **Object** | ImportObject | fileName, layerName | Import an image or object from file path. | Ps Ai Id |
| | OpacityObject | fileName, opacity | Adjust opacity of an object. | Ps Ai |
| | RemoveObject | fileName | Remove an object. | Ps Ai Id |
| | RepositionObject | fileName, posX, posY | Reposition an object to desired x and y-axis position. | Ps Ai Id |
| | ResizeObject | fileName, width, height | Resize an object to desired width and height. | Ps Ai Id |
| | RotateObject | fileName, angle | Rotate an object to desired angle. | Ps Ai Id |
| | GenerateQRObject | layerName, linkURL | Generate a QR code with desired URL embedded. | Id |
| | AdjustBC | layerName, brightness, contrast | Adjust brightness and contrast level of an object. | Ps |
| | AdjustBW | layerName | Change an object to black & white. | Ps |
| | AdjustHSL | layerName | Adjust hue, saturation, and lightness level of an object. | Ps |
| | BlurObject | layerName, blurAmount | Blur an object to desired amount. | Ps |
| | PhotoFilter | layerName, filterType, density | Apply a photo filter to an object with desired density. | Ps |
| | GlassFilter | layerName, distortion, smoothness, scaling | Apply a glass filter to an object with the specified parameters. | Ps |
| | GlowFilter | layerName, graininess, glowAmount, clearAmount | Apply a glow filter to an object with the specified parameters. | Ps |
| | OceanRippleFilter | layerName, rippleSize, rippleMagnitude | Apply an ocean ripple filter to an object with the specified parameters. | Ps |
| | StainedGlassFilter | layerName, cellSize, borderThickness, lightIntensity | Apply a stained glass filter to an object with the specified parameters. | Ps |
| | PatchWorkFilter | layerName, squareSize, relief | Apply a patchwork filter to an object with the specified parameters. | Ps |
| | WatercolorFilter | layerName, brushDetail, shadowIntensity, texture | Apply a watercolor filter to an object with the specified parameters. | Ps |

**Table 6: Complete list of available tools in GraphicWeaver.** Each tool requires specific parameters for execution. **Experts:** The expert agent(s) which supports the execution of a specific tool. For numerical parameters, we provide reference ranges (*e.g.,* `angle` as [0, 360], `brightness` as [-150, +150]). For parameters in filter-related functions, we provide a short description.

JavaScript for `AdjustBC` action (Photo Editor agent)

```
function promptForLayerName() {
    var layerName = arguments[0];

    if (layerName == null || layerName == "") {
        throw new Error("Layer with the name '" + layerName + "' does not exist.");
    }
    return layerName;
}

function promptForAdjustmentValues() {
    var brightness = parseInt(arguments[1], 10);
    var contrast = parseInt(arguments[2], 10);

    if (isNaN(brightness) || isNaN(contrast)) {
        throw new Error("Invalid input provided. Please run the script again and provide valid numbers.");
    }
    return { brightness: brightness, contrast: contrast };
}

function layerExists(layerName) {
    var ref = new ActionReference();
    ref.putName(charIDToTypeID("Lyr "), layerName);
    try {
        var desc = executeActionGet(ref);
        return true;
    } catch (e) {
        return false;
    }
}

function selectLayerByName(layerName) {
    var idselect = charIDToTypeID("slct");
    var desc = new ActionDescriptor();
    var idnull = charIDToTypeID("null");
    var ref = new ActionReference();
    var idLyr = charIDToTypeID("Lyr ");
    ref.putName(idLyr, layerName);
    desc.putReference(idnull, ref);
    var idMkVs = charIDToTypeID("MkVs");
    desc.putBoolean(idMkVs, false);
    executeAction(idselect, desc, DialogModes.NO);
}

function applyBrightnessContrastAdjustment(brightness, contrast) {
    var idBrtC = charIDToTypeID("BrgC");
    var desc = new ActionDescriptor();
    desc.putUnitDouble(charIDToTypeID("Brgh"), charIDToTypeID("#Prc"), brightness);
    desc.putUnitDouble(charIDToTypeID("Cntr"), charIDToTypeID("#Prc"), contrast);
    executeAction(idBrtC, desc, DialogModes.NO);
}

function adjustBrightnessContrast() {
    if (!layerExists(layerName)) {
        throw new Error("Layer with the name '" + layerName + "' does not exist.");
    }

    var layerName = promptForLayerName();
    if (layerName == null) {
        throw new Error("Layer name does not exist.");
    }

    var adjustments = promptForAdjustmentValues();
    if (adjustments == null) {
        throw new Error("Parameter values are not provided.");
    }

    selectLayerByName(layerName);
    applyBrightnessContrastAdjustment(adjustments.brightness, adjustments.contrast);
}

adjustBrightnessContrast();
```

**Figure 4: JavaScript code snippet for `AdjustBC` for the Photo Editor agent.** Each tool corresponds to an executable function in the design environment that takes predefined parameter values as input.

| Category | Example User Query | Reference | Design | Env. |
|---|---|---|---|---|
| Book Cover | Create a book cover design for a romance novel titled 'Love \n Story' featuring a silhouette illustration of a couple in a romantic pose against a pink moonlit background. The title should be at the top center, the author's name 'A Novel By \n Olivia Wilson' below the title, and the tagline 'Best Selling Book of the Year' above the title, all in white. | | | Ps Id |
| Business Card | Create a one-sided business card design with a light yellow background for the bookstore'CACTUS'. Replace the 'T' in 'CACTUS' with a cactus-shaped illustration in green font, centered and add a tagline 'Livros Novos e Usados' in green font below the bookstore name. | | | Id |
| Postcard | Create a postcard design with the message 'Think Happy!' in a red, curly font on a floral background featuring a mix of warm-toned roses. Place a semi-transparent white box behind the message. | | | Ai Id |
| Poster | Create a poster design with a light yellow background, featuring a large jellyfish illustration centered within a black rectangular box. Add a bold, black title 'JELLYFISH' at the top and place a brief informative sentence about jellyfish in white font at the bottom left corner. | | | Ai Id |

**Table 7: Examples of human-annotated user queries, reference designs, and resulting design outcomes for each design category. Env.:** Design environment(s) used for planning and execution.

| Category | Example User Query | Example Questions |
|---|---|---|
| Book Cover | Please create a self-help book cover design titled 'Achieve Your Dreams' with the author's name 'Nathan White', featuring a person climbing a mountain centered in the lower half against a sunrise background. The title should be at the top center in white, the author's name below the title in white, and the tagline 'Climb Higher, Dream Bigger.' below the author's name in white. | ["Is there a text 'Achieve Your Dreams'?", "Is there a text 'Nathan White'?", "Is there a person climbing a mountain?", "Is the background a sunrise?", "Is the text 'Achieve Your Dreams' at the top center in white?", "Is the text 'Nathan White' below the title in white?", "Is the text 'Climb Higher, Dream Bigger.' below the author's name in white?", "Is the person climbing a mountain centered in the lower half?"] |
| Business Card | I need to create a business card design for 'Sparkle Jewelry' with a royal blue background. Please include the company name in large gold font centered, and a medium-sized diamond icon placed above the company name. The contact details of 'Phone: +1 987 654 3210 \n Email: info@sparklejewelry.com \n Address: 12 Gem St, Los Angeles, CA, USA' should be in small gold font, placed bottom right. | ["Is the background of the business card royal blue?", "Is the company name 'Sparkle Jewelry' in large gold font?", "Is the company name centered?", "Is there a medium-sized diamond icon?", "Is the diamond icon placed above the company name?", "Are the contact details in small gold font?", "Are the contact details placed at the bottom right?", "Is the phone number '+1 987 654 3210' included in the contact details?", "Is the email 'info@sparklejewelry.com' included in the contact details?", "Is the address '12 Gem St, Los Angeles, CA, USA' included in the contact details?"] |
| Postcard | Please create a motivational postcard design with the message 'Stay Positive, Work Hard' at the top in red on a yellow background featuring a large lion roaring at the bottom. | ["Is there a text 'Stay Positive, Work Hard'?", "Is the text 'Stay Positive, Work Hard' at the top?", "Is the text 'Stay Positive, Work Hard' in red?", "Is the background yellow?", "Is there an illustration of a lion?", "Is the lion roaring?", "Is the lion illustration large?", "Is the lion illustration at the bottom?"] |
| Poster | Could you help create a promotional poster design for a jazz festival on a deep blue background, featuring a large image of a saxophonist playing in the center, a huge bold title 'JAZZFEST' in gold at the top center, and event details 'Jazz Festival \| May 5-7, 2023 \| Central Park, New York' in medium golden text at the bottom right? | ["Is there a deep blue background?", "Is there a large image of a saxophonist playing in the center?", "Is there a huge bold title 'JAZZFEST'?", "Is the title 'JAZZFEST' in gold?", "Is the title 'JAZZFEST' at the top center?", "Is there event details 'Jazz Festival \| May 5-7, 2023 \| Central Park, New York'?", "Is the event details in medium golden text?", "Is the event details at the bottom right?"] |

**Table 8: Examples of the generated questions using GPT-4 for each design category.** We use the questions to compute the VQA pass rate as part of execution evaluation.

**User Query:**

We need to create a business card design for a travel agency named 'Adventure Awaits' with a sky blue background. Please include the agency name in huge white font at the top center. I want to include the tagline 'Your Journey Begins Here' in medium white font below the agency name, and a mountain icon of medium size centered below the tagline. Also, add the contact details 'Phone: +1 800-555-7890 Email: info@adventureawaits.com Website: www.adventureawaits.com' in small white font at the bottom left.

**Design Choices:**

- **Background Color:** sky blue
- **Text:**
  - **Agency_name**
    - Content: "Adventure Awaits"
    - Size: large
    - Color: white
    - Position: top center
  - **Tagline**
    - Content: "Your Journey Begins Here"
    - Size: small
    - Color: white
    - Position: below the agency name
  - **Contact_details**
    - Content: "Phone: +1 800-555-7890 Email: info@adventureawaits.com Website: www.adventureawaits.com"
    - Size: small
    - Color: white
    - Position: bottom left
- **Images:**
  - **Mountain icon**
    - Size: medium
    - Position: center below the tagline

---

**[Question 1]**

Is each design choice **aligned** with the user query?

*(Here, **"aligned"** means that each design element fits the user's specifications in the query and contributes to the overall coherence of the final design.)*

| | **Not aligned at all** (key elements are missing) | **Slightly aligned** (some important elements misplaced/incorrectly implemented) | **Moderately aligned** (capture general intent but need adjustment to fully meet user's specifications) | **Aligned well** (only need minor adjustments) | **Completely aligned** (perfectly match user query) |
|---|---|---|---|---|---|
| Background color | ○ | ○ | ○ | ○ | ○ |
| Agency_name | ○ | ○ | ○ | ○ | ○ |
| Tagline | ○ | ○ | ○ | ○ | ○ |
| Contact_details | ○ | ○ | ○ | ○ | ○ |

**[Question 2]**

If you answered "Not aligned at all" or "Slightly aligned" to the previous question, please explain your reasoning in 1-2 sentences. You may also suggest changes to improve the alignment of the design choices with the user query.

Enter here

**[Question 3]**

Below are the three image candidates for **"mountain icon"**. Please rank them based on how well they fit the user query. Assign a rank from 1 (best fit) to 3 (least fit).

**Image 1**

Rank: Select rank

**Image 2**

Rank: Select rank

**Image 3**

Rank: Select rank

**Figure 5: Screenshot of human annotation interface.** For each user query, annotators are asked to (Q1) evaluate how well each design component aligns with the query on a five-point Likert scale and (Q2) rank the three images from 1 to 3 based on their relevance to the query. Additionally, they have the option to provide free-form feedback.

| Model | Planning | | | | Execution | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Delivery Rate (%) | Design Pass Rate (%) | Step Eff. | Expert Use Eff. | Success Rate (%) | Fidelity | Content Simi-larity | VQA Pass Rate (%) | Creative (O) | Creative (E) |
| ***Direct mode (aggregated)*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 27.1 | 22.5 | 45.5 | 0.98 | 10.3 | 0.01 | 5.70 | 16.0 | 1.05 | 0.64 |
| **QWEN-2.5-VL 32B** | 57.2 | 19.7 | 55.9 | **1.00** | 17.5 | 0.04 | 6.62 | 19.0 | 1.18 | 1.07 |
| **GEMMA-3 12B** | 37.7 | 19.0 | 63.8 | 0.99 | 14.9 | 0.03 | 6.93 | 13.6 | 1.18 | 0.76 |
| **GEMMA-3 27B** | **67.8** | 24.8 | 70.9 | **1.00** | 22.3 | 0.09 | 9.36 | 17.7 | 1.09 | 1.09 |
| **GPT-O4-MINI** | 46.2 | 23.3 | 79.5 | **1.00** | 27.9 | 0.10 | 8.72 | 21.5 | 1.14 | 1.12 |
| **GPT-4.1** | 30.3 | **30.1** | **82.5** | **1.00** | **29.5** | **0.13** | **13.1** | **23.3** | **1.39** | **1.35** |
| ***Book Cover*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 9.34 | 25.5 | 38.3 | 0.98 | 3.5 | 0.00 | 5.72 | 21.6 | 1.15 | 0.64 |
| **QWEN-2.5-VL 32B** | 52.8 | 21.1 | 65.1 | **1.00** | 16.8 | 0.00 | 5.12 | **24.8** | **1.44** | 1.02 |
| **GEMMA-3 12B** | 23.8 | 20.0 | 66.5 | **1.00** | 14.4 | 0.09 | 7.08 | 14.5 | 1.18 | 0.97 |
| **GEMMA-3 27B** | **68.9** | 25.5 | 75.7 | **1.00** | 25.9 | 0.09 | 9.50 | 19.9 | 1.06 | **1.32** |
| **GPT-O4-MINI** | 45.6 | 23.1 | **81.4** | **1.00** | 27.4 | 0.09 | 10.4 | 20.7 | 1.08 | 1.09 |
| **GPT-4.1** | 30.5 | **28.1** | 80.3 | **1.00** | **29.9** | **0.12** | **14.2** | 21.8 | 1.25 | **1.32** |
| ***Business Card*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 18.7 | **27.4** | 44.5 | 0.97 | 10.3 | 0.00 | 5.00 | 15.7 | 0.98 | 0.77 |
| **QWEN-2.5-VL 32B** | 30.6 | 16.1 | 46.3 | **1.00** | 12.5 | 0.00 | 4.67 | 16.2 | 1.02 | 0.94 |
| **GEMMA-3 12B** | 20.9 | 15.6 | 68.4 | 0.98 | 12.9 | 0.00 | 5.72 | 11.8 | 1.05 | 0.64 |
| **GEMMA-3 27B** | **45.6** | 19.3 | 65.4 | **1.00** | 22.8 | 0.05 | 7.45 | 15.6 | 1.07 | 0.97 |
| **GPT-O4-MINI** | 37.8 | 21.1 | 74.5 | **1.00** | 24.4 | 0.10 | 7.75 | 19.3 | 1.04 | 0.97 |
| **GPT-4.1** | 18.9 | 25.7 | **81.4** | **1.00** | **26.1** | **0.12** | **11.1** | **21.0** | **1.43** | **1.09** |
| ***Postcard*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 45.6 | 21.9 | 55.8 | 0.98 | 15.3 | 0.01 | 6.95 | 14.5 | 0.96 | 0.43 |
| **QWEN-2.5-VL 32B** | 66.7 | 23.4 | 55.6 | **1.00** | 18.9 | 0.08 | 8.01 | 19.4 | 1.13 | 1.23 |
| **GEMMA-3 12B** | 72.4 | 21.0 | 61.1 | **1.00** | 16.9 | 0.03 | 7.33 | 15.6 | 1.25 | 0.77 |
| **GEMMA-3 27B** | **82.1** | 27.9 | 70.9 | **1.00** | 19.2 | 0.11 | 9.58 | 17.8 | 1.09 | 0.93 |
| **GPT-O4-MINI** | 40.6 | 23.8 | 82.2 | **1.00** | **31.0** | 0.11 | 8.51 | 23.4 | 1.18 | 1.12 |
| **GPT-4.1** | 20.7 | **34.3** | **83.5** | **1.00** | 30.9 | **0.14** | **13.8** | **24.6** | **1.57** | **1.47** |
| ***Poster*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 34.9 | 15.2 | 43.2 | **1.00** | 12.0 | 0.02 | 5.12 | 12.2 | 1.12 | 0.71 |
| **QWEN-2.5-VL 32B** | **78.9** | 18.3 | 56.8 | **1.00** | 21.7 | 0.09 | 8.66 | 15.6 | 1.11 | 1.08 |
| **GEMMA-3 12B** | 33.6 | 19.4 | 59.2 | **1.00** | 15.5 | 0.02 | 7.58 | 12.6 | 1.25 | 0.67 |
| **GEMMA-3 27B** | 74.5 | 26.3 | 71.4 | **1.00** | 21.3 | 0.11 | 10.9 | 17.4 | 1.15 | 1.16 |
| **GPT-O4-MINI** | 60.7 | 25.3 | 80.0 | **1.00** | 29.0 | 0.12 | 8.21 | 22.6 | 1.24 | 1.32 |
| **GPT-4.1** | 51.1 | **32.3** | **84.7** | **1.00** | **31.2** | **0.13** | **13.3** | **25.7** | **1.32** | **1.54** |

**Table 9: Planning and execution results of different VLMs on GraphicWeaver for direct mode (aggregated and per category).** For each section, column-wise best scores are **bolded** and second-best scores are underlined. All metrics are interpreted as higher values for better performance (↑).

| Model | Planning | | | | Execution | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Delivery Rate (%) | Design Pass Rate (%) | Step Eff. | Expert Use Eff. | Success Rate (%) | Fidelity | Content Simi-larity | VQA Pass Rate (%) | Creative (O) | Creative (E) |
| ***Agentic mode (aggregated)*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 15.2 | 51.8 | 92.0 | **1.00** | 39.4 | 0.14 | 22.5 | 37.6 | 1.77 | 1.59 |
| **QWEN-2.5-VL 32B** | 39.6 | 49.7 | 92.4 | **1.00** | 61.3 | 0.20 | 22.3 | 35.4 | 1.89 | 1.65 |
| **GEMMA-3 12B** | 27.3 | 54.5 | 96.5 | **1.00** | 58.1 | 0.17 | 21.0 | 36.2 | 2.01 | 1.68 |
| **GEMMA-3 27B** | **51.4** | 54.4 | 93.1 | **1.00** | 55.4 | 0.20 | 26.7 | 44.7 | 1.98 | 2.04 |
| **GPT-O4-MINI** | 20.8 | 51.1 | **97.1** | **1.00** | 56.5 | 0.20 | 28.3 | **44.8** | 1.78 | 1.99 |
| **GPT-4.1** | 13.9 | **56.8** | 94.7 | **1.00** | **62.5** | **0.21** | **29.3** | 44.7 | **2.02** | **2.06** |
| ***Book Cover*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 6.40 | 52.4 | 92.6 | **1.00** | 10.6 | 0.10 | 17.2 | 37.9 | 1.75 | 1.57 |
| **QWEN-2.5-VL 32B** | 40.0 | 56.1 | 95.4 | **1.00** | 66.0 | 0.18 | 17.4 | 34.1 | 1.99 | 1.73 |
| **GEMMA-3 12B** | 17.5 | 55.9 | **97.2** | **1.00** | 60.8 | 0.17 | 18.3 | 33.0 | **2.22** | 1.99 |
| **GEMMA-3 27B** | **57.4** | 53.6 | 90.7 | **1.00** | 53.5 | 0.19 | 23.6 | **46.4** | 2.07 | **2.14** |
| **GPT-O4-MINI** | 23.9 | 48.9 | 97.0 | **1.00** | 55.0 | 0.18 | **27.3** | 41.0 | 1.91 | 1.96 |
| **GPT-4.1** | 4.90 | **56.9** | 95.4 | **1.00** | **66.4** | **0.22** | 24.0 | 38.3 | 2.00 | 2.07 |
| ***Business Card*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 12.3 | 48.8 | 94.3 | **1.00** | 56.0 | 0.12 | 19.4 | 30.7 | 1.61 | 1.53 |
| **QWEN-2.5-VL 32B** | **46.8** | 44.2 | 90.5 | **1.00** | **62.1** | **0.20** | 18.5 | 29.8 | 1.73 | 1.64 |
| **GEMMA-3 12B** | 1.70 | 53.3 | 96.5 | **1.00** | 58.1 | 0.15 | 19.2 | 34.5 | 1.81 | 1.64 |
| **GEMMA-3 27B** | 9.60 | 53.4 | 94.4 | **1.00** | 54.6 | 0.16 | 24.8 | 39.2 | 1.91 | 1.98 |
| **GPT-O4-MINI** | 2.60 | 51.2 | **97.0** | **1.00** | 56.1 | 0.19 | **32.1** | **47.7** | 1.85 | 1.79 |
| **GPT-4.1** | 0.00 | **55.8** | 93.8 | **1.00** | 60.7 | **0.20** | 27.8 | 46.9 | **2.12** | **2.03** |
| ***Postcard*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 21.2 | 53.6 | 91.0 | **1.00** | 48.9 | 0.18 | 30.9 | 48.7 | 1.98 | 1.91 |
| **QWEN-2.5-VL 32B** | 61.1 | 43.1 | 90.9 | **1.00** | **62.8** | 0.21 | 31.0 | 46.9 | 2.06 | 1.75 |
| **GEMMA-3 12B** | 68.0 | 55.2 | 94.4 | **1.00** | 56.4 | 0.18 | 23.3 | 40.1 | 2.13 | 1.73 |
| **GEMMA-3 27B** | **82.3** | 54.8 | 95.4 | **1.00** | 59.5 | **0.22** | 31.2 | **50.2** | 2.14 | 2.20 |
| **GPT-O4-MINI** | 15.4 | 50.3 | **96.8** | **1.00** | 53.6 | 0.20 | 28.3 | 45.0 | 1.70 | 2.18 |
| **GPT-4.1** | 8.00 | **56.5** | 94.2 | **1.00** | 57.5 | 0.21 | **33.6** | 48.4 | **2.20** | **2.22** |
| ***Poster*** | | | | | | | | | | |
| **QWEN-2.5-VL 7B** | 20.9 | 52.4 | 90.0 | **1.00** | 42.2 | 0.18 | 22.5 | 33.3 | 1.75 | 1.37 |
| **QWEN-2.5-VL 32B** | 10.6 | 55.4 | 92.9 | **1.00** | 54.3 | 0.19 | 22.1 | 30.9 | 1.77 | 1.47 |
| **GEMMA-3 12B** | 21.9 | 53.6 | **97.9** | **1.00** | 57.2 | 0.18 | 23.2 | 37.3 | **1.88** | 1.37 |
| **GEMMA-3 27B** | **56.1** | 55.6 | 91.7 | **1.00** | 53.9 | 0.21 | 27.0 | 43.1 | 1.79 | 1.85 |
| **GPT-O4-MINI** | 41.4 | 53.9 | 97.5 | **1.00** | 61.3 | 0.21 | 25.5 | **45.6** | 1.67 | **2.04** |
| **GPT-4.1** | 42.7 | **57.9** | 95.4 | **1.00** | **65.3** | **0.22** | **31.8** | 45.3 | 1.75 | 1.91 |

**Table 10: Planning and execution results of different VLMs on GraphicWeaver for agentic mode (aggregated and per category).** For each section, column-wise best scores are **bolded** and second-best scores are underlined. All metrics are interpreted as higher values for better performance (↑).

| Model | Frequency | Tool-Use Sequence |
|---|---|---|
| **QWEN-2.5-VL 7B** | 34 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ApplyFont → ColorText → AlignText → ResizeText → RepositionText → ExportDocument → AdjustHSL → SaveDocument |
| | 22 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → ApplyFont → ColorText → AlignText → ResizeText → ExportDocument |
| | 17 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → ResizeText → ColorText → AlignText → RepositionText → ExportDocument |
| **QWEN-2.5-VL 32B** | 45 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ApplyFont → ColorText → AlignText → ExportDocument |
| | 22 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ApplyFont → ColorText → AlignText → RepositionText → ExportDocument |
| | 12 | CreateDocumentCustom → SetBackgroundColor → CreateText → ColorText → AlignText → ImportObject → ResizeObject → RepositionObject → ExportDocument |
| **GEMMA-3 12B** | 28 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ResizeText → ColorText → AlignText → RepositionText → ExportDocument |
| | 26 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → SaveDocument → CreateText → ResizeText → ColorText → AlignText → ExportDocument |
| | 14 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ApplyFont → ColorText → AlignText → ResizeText → ExportDocument → AdjustHSL → SaveDocument |
| **GEMMA-3 27B** | 59 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ColorText → AlignText → ExportDocument |
| | 48 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ResizeText → ColorText → AlignText → ExportDocument |
| | 44 | CreateDocumentCustom → ImportObject → ResizeObject → RepositionObject → SaveDocument → SetBackgroundColor → CreateText → ApplyFont → ColorText → AlignText → ExportDocument |
| **GPT-O4-MINI** | 78 | CreateDocument → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → ColorText → AlignText → ExportDocument |
| | 53 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateDocument → CreateText → ColorText → AlignText → ExportDocument |
| | 42 | CreateDocumentCustom → SetBackgroundColor → CreateText → ColorText → AlignText → ImportObject → ResizeObject → RepositionObject → ExportDocument |
| **GPT-4.1** | 37 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ColorText → AlignText → ExportDocument |
| | 33 | CreateDocumentCustom → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → CreateText → ResizeText → ColorText → AlignText → ExportDocument |
| | 21 | CreateDocument → SetBackgroundColor → ImportObject → ResizeObject → RepositionObject → AdjustHSL → SaveDocument → CreateText → AlignText → ExportDocument |

**Table 11: Top-3 most common tool-use sequences per model, ordered by frequency.**

# D Prompts

## D.1 GraphicWeaver Prompts

We show prompt templates used for constructing user queries in GraphicWeaver and prompting VLMs for in the agentic mode below.

Prompt D.1.1: Query Construction (Book Cover)

**Task:** You are a design expert. Given the template for generating a book cover, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a book cover design with a [background color] background, featuring [images]. The title [title] should be placed at [position] in [color] and the author name [author name] at [position] in [color].
You may also include an optional [subtitle] or [tagline] if needed.

**** Example Starts ***
Create a book cover design for a romance novel titled 'Love \n Story' featuring a silhouette illustration of a couple in a romantic pose against a pink moonlit background. The title should be at the top center, the author's name 'A Novel By \n Olivia Wilson' below the title, and the tagline 'Best Selling Book of the Year' above the title, all in white.
**** Example Ends ***

**Examples:**

Prompt D.1.2: Query Construction (Business Card)

**Task:** You are a design expert. Given the template for generating a business card, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a [side]-sided business card design with a [background color] background, featuring [images]. The name [brand name] should be placed at [position] in [color].
You may also include an optional [contact details] or [tagline] if needed.

**** Example Starts ***
Create a one-sided business card design with a light yellow background for the bookstore'CACTUS'. Replace the 'T' in 'CACTUS' with a cactus-shaped illustration in green font, centered and add a tagline 'Livros Novos e Usados' in green font below the bookstore name.
**** Example Ends ***

**Examples:**

Prompt D.1.3: Query Construction (Postcard)

**Task:** You are a design expert. Given the template for generating a postcard, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a postcard design with a [background color] background, featuring [images]. The message [message] should be placed at [position] in [color].

**** Example Starts ***
Create a postcard design with the message 'Think Happy!' in a red, curly font on a floral background featuring a mix of warm-toned roses. Place a semi-transparent white box behind the message.
**** Example Ends ***

**Examples:**

Prompt D.1.4: Query Construction (Poster)

**Task:** You are a design expert. Given the template for generating a poster, fill in the placeholders with appropriate design components. Generate 5 diverse examples in a Python list of strings. Be as creative as possible.

**Template:** Create a poster design with a [background color] background, featuring [images]. The title [title] should be placed at [position] in [color].
You may also include an optional [tagline] if needed.

**** Example Starts ***
Create a poster design with a light yellow background, featuring a large jellyfish illustration centered within a black rectangular box. Add a bold, black title 'JELLYFISH' at the top and place a brief informative sentence about jellyfish in white font at the bottom left corner.
**** Example Ends ***

**Examples:**

Prompt D.1.5: Expert Agent Recruitment

**Task:** Your task is to recruit the necessary experts to complete a design outlined in the user query. Create a recruitment status in JSON list format.

**User query:** `{user query}`
**User image file(s):** `{user image files}`

You can recruit from the three experts with the following profiles:

- **Photo Editor**
  - Job Responsibilities:
    - Image editing: Cropping, adjusting composition, correcting lighting, and retouching images or illustrations.
    - Color correction: Adjusting brightness and contrast or adjusting hue and saturation.
    - Apply filters: Apply different filters (e.g., photo, glass, ocean ripple, watercolor) to images.
- **Vector Graphic Editor**
  - Job Responsibilities:
    - Draw shapes: Drawing simple shapes (circle, polygon, square, star) on canvas.
- **Layout Designer**
  - Job Responsibilities:
    - Customize layout templates: Create grid systems for books, brochures, cards, and magazines to organize the layout.
    - Export files: Export documents to any format, in print or digital.
    - Combine text and visual elements: Combine visual elements from other apps with text into a completed design.

*** Output Format ***
Each object in the JSON list should follow:

```
{
    "expert": "Name of the expert (Photo Editor, Vector Graphic Editor, Layout
        Desinger).",
    "task": "High-level task that can be performed by the expert."
}
```

*** Example Starts ***

```
[
    {"expert": "Photo Editor", "task": "Add the provided images to create a deep
         purple night sky background with a large dreamy moon centered,
        surrounded by small twinkling stars spread across the top half of the
        cover."},
    {"expert": "Layout Designer", "task": "Combine the edited image with the
        title 'Moonlit Fantasies', the author name 'J.K. Stellar', and the
        tagline 'A Journey Through the Night Sky.' to create the book cover
        design."}
]
```

*** Example Ends ***

**** Key Requirements ***
- Only recruit each expert one.
- The name of the expert must match those in the expert profiles.
- For task description, explain how the expert can contribute towards the final product. Summarize in one sentence.
- In order to achieve the task in the design outline, experts should work together and their task will be dependent to each other. Arrange in the order of which expert should finish first.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Recruitment status:**

Prompt D.1.6: Individual Plan Generation

**Task:** You are a proficient `{expert}`. You are recruited to collaborate on a design project with other experts.

You are assigned to complete the following task: `{task}`. Please plan a sequence of detailed, low-level subtasks required to accomplish this task and output them as a JSON list.

**** Output Format ***
Each object in the JSON list should follow:

```
{
    "id": "ID of the subtask, starting from 1.",
    "expert": "Name of the expert.",
    "description": "Description of the subtask in one sentence."
}
```

*** Example Starts ***

```
[
    {"id": 1, "expert": "Photo Editor", "description": "Create a new document
        with book cover dimensions."},
    {"id": 2, "expert": "Photo Editor", "description": "Set the background color
         to light pink."},
    {"id": 3, "expert": "Photo Editor", "description": "Import the pink moonlit
        image from 'static/pink_moonlit.png'."},
    {"id": 4, "expert": "Photo Editor", "description": "Resize the pink moonlit
        image to medium size, covering the bottom part of the document."},
    {"id": 5, "expert": "Photo Editor", "description": "Reposition the pink
        moonlit image to the bottom-center of the document."},
    {"id": 6, "expert": "Photo Editor", "description": "Import the couple
        silhouette illustration from 'static/couple_silhouette.png'."},
    {"id": 7, "expert": "Photo Editor", "description": "Resize the couple
        silhouette illustration to span across the lower half of the cover."},
    {"id": 8, "expert": "Photo Editor", "description": "Reposition the couple
        silhouette illustration to be centered in the bottom-middle part."},
    {"id": 9, "expert": "Photo Editor", "description": "Adjust the background
        colors to match the light pink moonlit theme."},
    {"id": 10, "expert": "Photo Editor", "description": "Save the document in a
        psd format suitable for further editing by the Layout Designer."},
]
```

*** Example Ends ***

**** Key Requirements ***
- First step should always be creating a new document and the last step should always be saving the document in appropriate file format.
- Use the exact image URLs the user provided when importing images.
- Do not include very basic operations such as opening the software or closing the software.
- Do not include new expert in the plan.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Sequence of subtasks:**

Prompt D.1.7: Plan Supervision

**Task:** You are the supervisor of a design project that requires collaboration among various design experts.

The following experts have been recruited for the project. Use as reference:
`{recruitment status}`

Each expert has submitted their proposed workflow plans:
`{workflow plans}`

Your task is to combine these proposed workflow plans into a cohesive sequence of tasks in a JSON list format.

**** Output Format ***
Each object in the JSON list should follow:

```
{
    "id": "ID of the subtask, starting from 1.",
    "expert": "Name of the expert.",
    "description": "Description of the subtask in one sentence."
}
```

*** Example Starts ***

```
[
    {"id": 1, "expert": "Photo Editor", "description": "Create a new document
        with book cover dimensions."},
    {"id": 2, "expert": "Photo Editor", "description": "Set the background color
         to light pink."},
    ...
    {"id": 11, "expert": "Layout Designer", "description": "Create a new
        document with book cover dimensions."},
    {"id": 12, "expert": "Layout Designer", "description": "Import the edited
        image from the Photo Editor: 'moonlit_illustration_edited.psd'."},
    {"id": 13, "expert": "Layout Designer", "description": "Resize the edited
        image to cover the entire document."},
    {"id": 14, "expert": "Layout Designer", "description": "Create text for the
        title 'LOVE\nSTORY'."},
    {"id": 15, "expert": "Layout Designer", "description": "Apply the Andale
        Mono font to the title text."},
    ...
    {"id": 31, "expert": "Layout Designer", "description": "Reposition the
        tagline text above the title."},
    {"id": 32, "expert": "Layout Designer", "description": "Export the final
        book cover design as a PDF file."}
]
```

*** Example Ends ***

**** Key Requirements ***
- Do NOT repeat any steps that are already completed in previous step.
- For each expert, first step should always be creating a new document and the last step should always be saving the document in appropriate file format.
- When switching experts, use the output from the previous expert as input for the next.
- Once an expert is used and switched to another expert, it should not be used again.
- You should output only one list of workflow plan.
- Start the id from 1 to the number of steps in the workflow.
- Arrange each subtask in a chronological order.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Supervised sequence of subtasks:**

Prompt D.1.8: Tool Retrieval

**Task:** You are a proficient `{expert}`. You are recruited to collaborate on a design project with other experts. Use your available list of tools to map each step in the sequence of subtasks to a tool.

**Sequence of subtasks:** `{workflow plan}`

Your available tools are as below:
`{list of tools}`

**** Output Format ***
Each object in the JSON list should follow:
```
{
    "id": "ID of the subtask, starting from 1.",
    "expert": "Name of the expert.",
    "description": "Description of the subtask in one sentence.",
    "tool": "Name of the mapped tool.",
    "parameters": "Dictionary of parameter keys and corresponding values."
}
```
*** Example Starts ***
```
[
    {"id": 1, "expert": "Photo Editor", "description": "Create a new document
        with book cover dimensions.", "skill": "CreateDocument", "parameters":
        {"docType": "book cover"}},
    {"id": 2, "expert": "Photo Editor", "description": "Set the background color
         to light pink.", "skill": "SetBackgroundColor", "parameters": {"red":
        255, "green": 179, "blue": 238}},
    ...
    {"id": 8, "expert": "Photo Editor", "description": "Reposition the couple
        silhouette illustration to be centered in the bottom-middle part.", "
        parameters": {"layerName": "SilhouetteLayer", "posX": 267, "posY":
        1052}},
    {"id": 9, "expert": "Photo Editor", "description": "Adjust the background
        colors to match the light pink moonlit theme.", "skill": "AdjustHSL", "
        parameters": {"layerName": "MoonlitLayer", "hue": 18, "saturation": -18,
         "light": 0}},
    {"id": 10, "expert": "Photo Editor", "description": "Save the document in a
        format suitable for further editing by the Layout Designer.", "skill": "
        SaveDocument", "parameters": {"fileName": "moonlit_illustration_edited",
         "format": "psd"}},
]
```
*** Example Ends ***

**** Key Requirements ***
- For any file name that appears in the design outline, use exact file names in your sequence of subtasks.
- Each step should only be mapped to one tool. If a step of the workflow is not able to be mapped to one tool, it means the step can be decomposed further into multiple steps. You can reformat, reorder, add, edit steps of the workflow if needed to be directly mapped to tools.
- Each step should have a tool and a dictionary of parameter values.
- For layerName, try to name it as to end as Layer (e.g., BackgroundLayer, TitleLayer).
- For detailed numeric values (e.g., height, width, x-axis position, y-axis position), consider the document's dimensions, imagine, and propose a likely value.
- Arrange each subtask in a chronological order.
- Output should be in a list of JSON objects format.
- Do NOT include further explanation other than in the JSON list.
- Be as concise and brief as possible.

**Sequence of subtasks:**

## D.2 Evaluation Prompts

We present the prompts used for evaluating both planning and execution, including design pass rate (color, text, image), VQA pass rate, and creativity (originality and elaboration).

Prompt D.2.1: Design Pass Rate Evaluation (Color)

**Task:** Evaluate if the workflow plan (1) correctly applies the background color and (2) the background and the text color are contrasting. Return a score between 1 to 5 according to the scoring rubric.

**Background color:** `{background color}`
**Text elements:** `{text}`
**Workflow plan:** `{workflow plan}`

**** Scoring Rubric ***
- 1: Workflow plan fails to reflect all of the color constraints specified.
- 3: Workflow plan reflects approximately half of the color constraints specified.
- 5: Workflow plan reflects all of the color constraints specified.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.
**Score:**

Prompt D.2.2: Design Pass Rate Evaluation (Text)

**Task:** Evaluate if the workflow plan adequately applies the text elements (e.g., title, tagline) specified. Return a score between 1 to 5 according to the scoring rubric.

**Text elements:** `{text}`
**Workflow plan:** `{workflow plan}`

**** Scoring Rubric ***
- 1: Workflow plan fails to reflect all of the text elements specified.
- 3: Workflow plan reflects approximately half of the text elements specified.
- 5: Workflow plan reflects all of the text elements specified.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.
**Score:**

Prompt D.2.3: Design Pass Rate Evaluation (Image)

**Task:** Evaluate if the workflow plan adequately applies the image elements (e.g., size, position) specified. Return a score between 1 to 5 according to the scoring rubric.

**Image elements:** `{image}`
**Workflow plan:** `{workflow plan}`

**** Scoring Rubric ***
- 1: Workflow plan fails to reflect all of the image elements specified.
- 3: Workflow plan reflects approximately half of the image elements specified.
- 5: Workflow plan reflects all of the image elements specified.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.
**Score:**

Prompt D.2.4: VQA Evaluation

**Instruction:** Look at the image and answer the question with 'Yes' or 'No'.

**Question:** `{quetion}`
**Answer:**

Prompt D.2.5: Creativity Evaluation (Originality)

**Instruction:** Evaluate the originality of the image generated based on the user query. Originality measures the uniqueness of the ideas generated. Original ideas are those that are rare or unconventional, differing from the norm. Return a score between 1 to 5 according to the scoring rubric.

**User query:** `{user query}`

**** Scoring Rubric ***
- 1: Image is highly conventional and predictable. No significant signs of creative thinking is shown.
- 2: Image shows minimal originality and mostly align with typical or common responses. Few novel elements are present.
- 3: Image is somewhat original, with a mix of conventional and unique elements.
- 4: Image is noticeable original and uncommon. It shows creative thinking and depart meaningfully from conventional norms.
- 5: Image is highly unique, rare, and stand out as unconventional. They demonstrate a strong departure from typical or expected approaches.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.
**Score:**

Prompt D.2.6: Creativity Evaluation (Elaboration)

**Instruction:** Evaluate the elaboration of the image generated based on the user query. Elaboration refers to the ability to expand upon, refine, and embellish an idea. It involves adding details, developing nuances, and building upon a basic concept to make it more intricate or complex. Return a score between 1 to 5 according to the scoring rubric.

**User query:** `{user query}`

**** Scoring Rubric ***
- 1: Image is presented in a simpler or vague manner with no meaningful development or supporting detail.
- 2: Image is minimally expanded, with few details or refinements added.
- 3: Image includes expansion of some details, but elaboration is somewhat surface-level.
- 4: Image well-expands the user query with several added details and refinements.
- 5: Image thoroughly expands the user query with rich, specific details or refinements added beyond the core concept.

Score should strictly be a number between 1 to 5. Do not include any further explanation other than the score.
**Score:**