# DESIGNX: BRIDGING CODING AND GENERATION FOR RECOMMENDATION-BASED VISUAL DESIGN

**Anonymous authors**
Paper under double-blind review




Figure 1: **Gallery of DesignX.** Designs generated by our recommendation-based visual design agent can be decomposed into raster layers, editable text, and vector graphics, supporting over a hundred layers. Examples in red boxes are decomposed into editable elements, where text and vector shapes can be directly modified in PowerPoint, including properties such as text attributes and color fills.

## ABSTRACT

Despite recent advances in AI visual generation, professional design still faces two challenges: enabling non-experts to create professional-quality designs and converting raster images into unified, editable representations of raster images, vector graphics, and text. To address these challenges, we present **DesignX**, a recommendation-based visual design agent that enables both professional design creation and raster image decomposition. For the visual design agent, we construct a large-scale template library covering diverse visual design categories and develop three specialized skills: a cataloging skill for organizing and continuously expanding the template library, a coarse-to-fine retrieval skill for efficiently identifying relevant templates, and a generation skill that adapts templates according to user requirements. Beyond raster-only generation, we introduce a raster image decomposition framework consisting of a vision-language model (VLM), an RGBA-based diffusion transformer (DiT), and a text-attribute predictor. Specifically, we design a structured decomposition protocol that enables the VLM to parse visual design into six editable elements with positions and descriptions while directly generating code for simple vector primitives. The RGBA-based DiT extracts individual layers from VLM descriptions and completes occluded regions,



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

while the text-attribute predictor jointly predicts typography attributes through a multi-head Transformer architecture. Furthermore, we introduce synthetic collage data augmentation to improve model robustness. Finally, we establish a comprehensive visual design benchmark spanning diverse categories to facilitate future research. Extensive experiments demonstrate that DesignX significantly improves visual design quality, and our decomposition framework surpasses existing SOTA commercial solutions. We have released the code and template library.

## 1 INTRODUCTION

Visual design is becoming an increasingly important application of generative AI (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022; Esser et al., 2024), spanning diverse scenarios such as posters, presentation slides, infographics, academic figures, social media graphics, and user interfaces, among others. Currently, visual design is primarily driven by two paradigms: coding-based approaches, exemplified by coding agents such as Claude Design, and generation-based approaches, represented by visual generation models such as GPT-Image-2. With the rapid advancement of large language models (LLMs) and visual generation models, both paradigms have achieved remarkable progress and gained widespread recognition and adoption in practical design applications.

Coding agents can create and edit presentation slides or HTML-based interfaces from instructions, while recent visual generation models have made remarkable progress in instruction following, visual composition, and text rendering, often producing more visually compelling designs. This suggests a promising future workflow: users first create high-quality designs with generative models, then convert the resulting raster images into editable, code-based representations for existing design platforms such as PowerPoint, where they can be further refined manually or with coding agents.

Visual design is highly subjective, yet non-expert users often have limited exposure to diverse design examples and may struggle to articulate their preferred visual style. As a result, they typically provide only functional requirements and leave the entire design process to generative models, which often leads to generic visual outputs. If a visual design agent can understand the user's intent and recommend several stylistically diverse, high-quality design references, it can help alleviate this problem. By previewing these professional templates, users can more easily discover their preferences and adapt a selected design, without having to search for suitable references themselves. Directly searching the Internet for templates, however, is inefficient and unreliable: low-quality examples greatly outnumber high-quality ones, many design platforms restrict automated access, and current models still have limited ability to reliably assess visual quality. Consequently, web-based search can consume substantial time and tokens while still producing unsatisfactory results.

To this end, we build and maintain a high-quality template library covering diverse design categories and application scenarios, where each template can be viewed as a reusable visual design skill. Given the scale of the library, it is impractical to load all skill summaries into the model context for automatic selection, as in Codex. We therefore develop three complementary skills on top of the library (Fig. 2): **(1) Cataloging**, which assigns multi-dimensional metadata to each template for efficient retrieval by both users and the visual design agent; **(2) Retrieval**, which understands the user's requirements and employs a coarse-to-fine search strategy over the design library to narrow the search space and identify a small set of relevant candidates for user selection; and **(3) Generation**, which combines the selected template with the user's requirements to produce a satisfactory design.

However, our visual design agent currently produces only raster images, limiting its use in professional workflows that often combine raster and vector elements, such as PowerPoint. Editable visual designs can also work with coding agents to automate design modification and enable applications like web design. Existing studies (Yin et al., 2026; Liu et al., 2026) mainly focus on decomposing a raster image into multiple raster layers, while other essential aspects of editability, such as reconstructing editable text and simple vector graphics, remain largely unexplored.

To this end, in DesignX, we further study a framework for converting raster images into editable visual designs composed of raster layers, editable text, and simple vector graphics. Our framework integrates a VLM, an RGBA-based DiT, and a text-attribute predictor. Specifically, **(1)** we design a structured decomposition protocol that enables the VLM to parse a visual design into six editable element categories and predict their spatial locations, layer order, and content. For simple vector elements, such as lines and geometric shapes commonly used in PowerPoint, the VLM directly

generates executable code for reconstruction. **(2)** To improve typography recovery, we introduce a Transformer-based multi-head text-attribute predictor that jointly models correlated font attribute, such as font family, font size, and other typographic properties. We find that it predicts text attributes more accurately and efficiently than directly using a VLM. **(3)** For the RGBA-based DiT, we train a description-guided extraction mechanism that recovers target layers from local image crops, completes occluded regions, and removes irrelevant objects from the results. **(4)** For training, we develop a hybrid training scheme that combines synthetic collage data with real-world design data, improving robustness and generalization across diverse visual designs. Additionally, existing studies have largely overlooked the joint reconstruction of raster elements, editable text, and vector graphics. We therefore introduce a benchmark spanning multiple design categories and combining both generated and real-world data. We further develop a comprehensive evaluation protocol and package it as an evaluation skill to assess different aspects of this task. Our contributions are:

- We build and maintain a large-scale, high-quality template library covering diverse design categories and scenarios. Building upon this library, we introduce a visual design agent with automatic template recommendation, enabling users to create high-quality designs.
- We propose a structured decomposition protocol for jointly recovering raster images, editable text, and simple vector graphics. Based on this protocol, we develop a unified decomposition framework consisting of a VLM, an RGBA-based DiT, and a text-attribute predictor. We further introduce synthetic collage data to improve model robustness.
- We introduce the DesignX benchmark spanning diverse visual design categories, together with a comprehensive evaluation protocol for the underexplored task of jointly decomposing visual designs into raster images, editable text, and simple vector graphics.
- Our DesignX bridges AI-based image generation and professional editable design, enabling an end-to-end workflow from design creation to editable asset decomposition while outperforming leading proprietary commercial systems.

## 2 Related Work

**Agentic Visual Design.** Early visual design agents, such as LLMGA (Xia et al., 2024), explored how agents could improve general visual design and user interaction. After that, PPTAgent (Zheng et al., 2025) extracts page functions and content patterns from reference presentations to generate new slides, while AutoFigure (Zhu et al., 2026b) and PaperBanana (Zhu et al., 2026a) generate scientific illustrations through planning, rendering, and iterative refinement. Recent works (He et al., 2026; Feng et al., 2026) further improve generation through multimodal retrieval, optimized search strategies, and memory-augmented reasoning. With the rapid progress of foundation models and harness engineering, agent systems are evolving into general-purpose platforms such as Codex and Claude Code, where new capabilities can be incorporated as modular skills. Visual design agents should follow this trend. Besides, existing approaches (Zhang et al., 2026) often retrieve design knowledge or references from the open web, where noisy information and poor template quality make it difficult to reliably recommend high-quality designs. To address this issue, our DesignX builds a large-scale, high-quality visual design template library spanning diverse categories and scenarios, providing a curated knowledge base for template retrieval and recommendation. We further introduce three specialized skills—cataloging, retrieval, and generation—to continuously expand the library, efficiently identify templates that match user requirements, and generate professional designs guided by selected templates. Our recommendation-driven visual agent substantially lowers the barrier to professional visual design.

**Raster Image Decomposition.** Early AI-based graphic design research (Jia et al., 2023) mainly focused on generating multi-layer compositions from scratch. For example, LayerDiffuse (Zhang & Agrawala, 2024) jointly synthesizes foreground and background through shared attention, and ART (Pu et al., 2025) jointly predicts composite images and transparent foreground layers conditioned on global text and regional layouts. However, these methods cannot decompose raster images into editable structures. Existing approaches (Suzuki et al., 2025; Liu et al., 2026; Chen et al., 2025a) generally follow either iterative top-layer extraction with background inpainting, or joint prediction of multiple RGBA raster image layers within a diffusion model (Yin et al., 2026; Tang et al., 2026; Lungu-Stan et al., 2026; Rowles et al., 2026). In addition, recent works (Lin et al., 2026; Yun et al., 2026) employ agent-based pipelines to improve decomposition quality, while others (Tudosiu et al.,

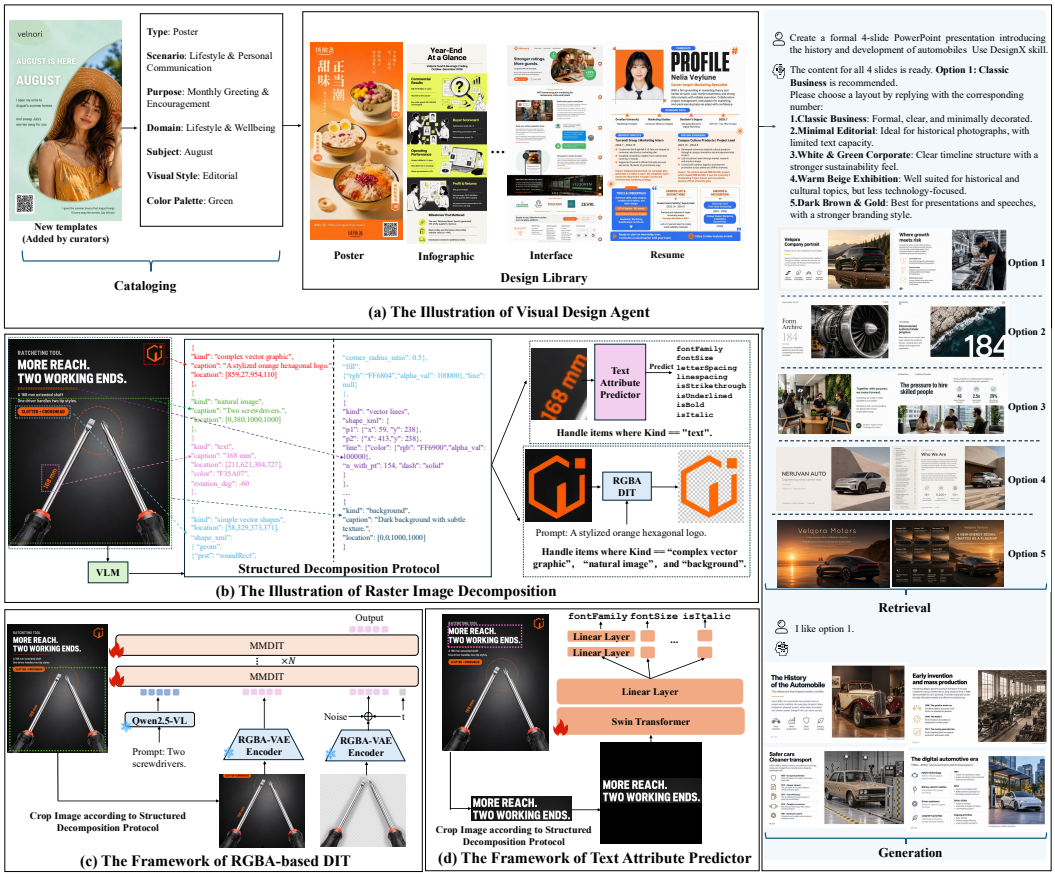



Figure 2: Overview of DesignX. **(a)** Our recommendation-based visual design agent is built on a large-scale design library covering diverse categories and scenarios. Three skills support its use: *Cataloging* assigns semantic tags, *Retrieval* performs coarse-to-fine template search, and *Generation* combines the selected template with user instructions to produce the final design. **(b)** Our decomposition framework parses a design into six element categories. Simple vector shapes and lines are reconstructed from VLM-generated code, while complex vector graphics, natural images, and backgrounds are extracted by an RGBA-based DiT guided by VLM-generated descriptions. Text is reconstructed using a text-attribute predictor. **(c)** The RGBA-based DiT operates on cropped target regions to reduce computation. **(d)** The text-attribute predictor uses a shared backbone with multiple heads to jointly predict correlated typography attributes.

2024; Chen et al., 2025b) enhance decomposition performance by constructing high-quality synthetic training data. However, prior work primarily focuses on decomposing a raster image into multiple raster layers, overlooking editable text and vector graphics, which are equally important in practical design workflows. To address this limitation, we propose a framework that decomposes raster designs into raster elements, vector graphics, and editable text.

# 3 Methodology

## 3.1 Recommendation-Based Visual Design Agent

Visual design is highly subjective, yet non-expert users often lack exposure to professional design patterns and must rely heavily on generative models, leading to limited user involvement and generic results. Alternatively, the agent may search the open web for relevant design templates. However, online resources are highly noisy, and the quality of available templates varies substantially. Moreover, many websites impose access restrictions, while current agents still have limited ability to reliably assess template quality and relevance. Thus, web-based retrieval often requires substantial token consumption and long search latency, yet still struggles to identify high-quality templates

that closely match user requirements. Therefore, it is essential to build and maintain a high-quality, easily searchable template library covering diverse design categories and scenarios. As shown in Fig. 2 (a), we construct such a library by carefully screening our internal dataset, covering presentations, e-commerce posters, product detail pages, infographics, and many other design types.

With the rapid advancement of foundation models and the growing adoption of harness engineering, platform-based agents such as Codex have become increasingly prevalent, with new capabilities commonly organized as modular skills. Visual design agents should follow the same paradigm. In our system, each template in the large-scale design library can be regarded as a design skill. However, conventional skill loading in systems typically relies on providing the model with brief descriptions of all available skills in advance, which does not scale to a massive design skill library. To address this issue, we introduce three higher-level skills: cataloging, retrieval, and generation.

Specifically, our cataloging skill is designed to support efficient coarse retrieval from a large template library while enabling the library to evolve as new templates are added. As shown in Fig. 2 (a), it systematically indexes each template along seven ordered facets: *Type*, *Scenario*, *Purpose*, *Domain*, *Subject*, *Visual Style*, and *Color Palette*. These facets respectively describe the visual format, application context, communication objective, broad domain, specific topic, visual style, and dominant colors. For example, a smartphone advertisement may be cataloged as Poster, E-commerce & Retail Marketing, Product Promotion & Sales, Consumer Electronics, Smartphone, Modern, and Blue, respectively. This structured organization allows both users and models to quickly narrow down relevant templates through keyword-based retrieval and ensures that newly added templates are consistently placed within the library for future search. For the retrieval skill, we adopt a coarse-to-fine retrieval strategy. Based on the user's requirements and template tags, the system first narrows down the candidate set, then carefully inspects a small number of promising templates and selects the five most relevant ones. These templates are presented with concise recommendation rationales, substantially reducing the effort required for users to browse and select templates. For the generation skill, given the selected template, the model analyzes its design style together with the user's requirements and adapts the template accordingly to produce a professional, customized design. In addition, we develop an interactive interface that allows users to directly search the template library based on their requirements, complementing the automated retrieval process.

### 3.2 Raster Image Decomposition

Although DesignX can generate high-quality visual designs, its outputs remain raster images and are therefore difficult to edit. In contrast, professional design workflows, such as those in PowerPoint, typically represent designs as editable combinations of raster images, vector graphics, and text. To bridge this gap, DesignX further supports raster image decomposition. Specifically, as shown in Fig. 2 (b), we introduce a structured decomposition protocol that guides a VLM to parse each design into seven element types: simple vector shapes, vector lines, natural images, complex vector graphics, editable text, and the background. Simple vector shapes, such as polygons and circles in PowerPoint, are usually defined by a small number of geometric parameters and can therefore be directly reconstructed by the VLM through code, preserving editable properties such as fill color, stroke width, and line style in PowerPoint. Vector lines, such as arrows and connectors, are similarly represented with explicit geometric structures and can also be recovered in an editable form in PowerPoint. Natural images refer to photographic or painted content. Complex vector graphics denote intricate digital illustrations that are difficult to faithfully reconstruct through simple code, including highly stylized lettering. Editable text elements are limited to content that can be reliably rendered with standard fonts, while the background represents the underlying design canvas. For each element, the VLM predicts its type and bounding box, with an additional rotation angle for text. It then generates type-specific intermediate representations: OCR text for textual elements, executable code for simple vector shapes and lines, and textual descriptions for natural images, complex vector graphics, and backgrounds, which are subsequently used for reconstruction.

After the VLM parses the design, we use an RGBA-based DiT (illustrated in Fig. 2 (c)) to handle the complex vector graphics, natural image, and background. Specifically, we crop each target region according to its predicted position and then condition the RGBA-based DiT on the corresponding caption to extract the desired layer content. Compared with methods such as Qwen-Image-Layer, which predict each layer from the full image, our region-based strategy avoids generating large

amounts of irrelevant transparent areas. By restricting generation to localized regions, it reduces unnecessary computation and enables more efficient layer reconstruction.

For text elements, we collect $6,279$ font families. Given this large label space, VLMs struggle to accurately identify fine-grained font categories. We therefore introduce a multi-head text attribute predictor (Fig. 2 (d)). Since font family is strongly correlated with other typographic attributes, such as font size, letter spacing, and boldness, we jointly predict these attributes within a shared representation rather than modeling them independently. This multi-task formulation allows different attribute heads to exploit their underlying correlations, leading to more coherent and accurate predictions of font family, size, and style. Notably, absolute font size, line spacing, and letter spacing are scale-dependent and cannot be directly predicted from cropped text regions. We therefore normalize these attributes with respect to the corresponding crop size. Given a text crop with height $H_c$ and width $W_c$, an absolute typography attribute $a$ is normalized as

$$\hat{a} = \frac{a}{\sqrt{H_c W_c}}, \tag{1}$$

where $H_c$ and $W_c$ denote the height and width of the original text crop, $a$ denotes the absolute attribute value, and $\hat{a}$ is its normalized representation. Each crop is isotropically resized such that its longer side is 768 pixels and then padded to $768 \times 768$ for network prediction. Since the normalization is invariant to isotropic resizing, the original absolute attribute can be recovered as

$$a = \hat{a}\sqrt{H_c W_c}. \tag{2}$$

For each text element, the text attribute predictor jointly estimates the font family, font size, letter spacing, line spacing, and several binary style attributes, including bold, italic, underline, and strikethrough. We formulate font-family prediction as a classification task, continuous attributes as regression tasks, and binary style attributes as binary classification tasks. The training objective is:

$$\mathcal{L}_{\text{text}} = \lambda_{\text{CE}}\mathcal{L}_{\text{CE}}^{\text{font}} + \sum_{a \in \mathcal{A}_{\text{reg}}} \mathcal{L}_{\text{L1}}^{a} + \sum_{a \in \mathcal{A}_{\text{bin}}} \mathcal{L}_{\text{BCE}}^{a}, \tag{3}$$

where

$$\mathcal{A}_{\text{reg}} = \{\text{fontSize}, \text{letterSpacing}, \text{lineSpacing}\}, \quad \mathcal{A}_{\text{bin}} = \{\text{bold}, \text{italic}, \text{underline}, \text{strikethrough}\}. \tag{4}$$

Here, $\mathcal{L}_{\text{CE}}$, $\mathcal{L}_{\text{L1}}$, and $\mathcal{L}_{\text{BCE}}$ denote the cross-entropy, $L_1$, and binary cross-entropy losses, respectively. We set $\lambda_{\text{CE}} = 0.1$, while all remaining loss weights are set to 1.

Our training data consists of 2.2M real layered designs and 2.1M synthetic collages. For synthetic data, we build an asset pool containing high-quality RGBA foregrounds, including complex vector graphics and natural images, basic PowerPoint vector shapes and lines with their parameters, and 6,279 vector font families. Backgrounds are sampled from solid-color or gradient vector graphics and natural images. We then randomly compose raster, vector, and text elements by varying their positions, scales, and element-specific attributes, while ensuring that no foreground element fully occludes an underlying element. Training data is converted into our decomposition protocol for VLM training. For the text attribute predictor and RGBA-based DiT, we further extract the corresponding elements from the decomposition data, resulting in 6M and 4M training samples, respectively. The RGBA-based DiT is subsequently distilled with DMD (Yin et al., 2024) to enable 4-step inference. We use Qwen3.6-35B-A3B (Yang et al., 2025) as the VLM. The RGBA-based DiT is initialized from Qwen-Image (Wu et al., 2025) and adopts the RGBA VAE from Qwen-Image-Layer, while the text attribute predictor is built upon the Swin Transformer (Liu et al., 2021).

Joint training on real and synthetic data yields the best performance. Real designs contain strong biases toward frequently used layouts, fonts, and graphic elements, which can lead to a substantial domain gap on AI-generated designs with more diverse distribution. Synthetic collages reduce such biases and improve robustness by increasing compositional and appearance diversity. However, synthetic data alone cannot reliably capture appropriate layer granularity, which must still be learned from real data to avoid under- or over-segmentation.

### 3.3 BENCHMARK

To evaluate the visual design agent, we collect 500 design requests covering common real-world scenarios. Since modern foundation models such as GPT-Image-2 already exhibit strong instruction-following and rendering capabilities, differences in higher-level design quality are often subjective

Table 2: Ablation study on the use of real and synthetic training data for visual decomposition.

| Method | Real Data | Synthetic Data | LPIPS↓ | Valid Image Layers↑ | Valid Text Layers↑ | Valid Vector Layers↑ | Total Valid Layers↑ |
|---|---|---|---|---|---|---|---|
| DesignX V1 | ✓ | | 0.1592 | 2,132 | 1,570 | 1,042 | 4,294 |
| DesignX V2 | | ✓ | 0.1968 | 1,750 | 2,613 | 1,733 | 6,096 |
| **DesignX V3 (Ours)** | ✓ | ✓ | **0.1400** | **2,150** | **3,713** | **2,233** | **8,096** |

Table 3: Comparison of decomposition performance and inference efficiency.

| Method | Runtime (s)↓ | LPIPS↓ | Valid Image Layers↑ | Valid Text Layers↑ | Valid Vector Layers↑ | Total Valid Layers↑ |
|---|---|---|---|---|---|---|
| Qwen-Image-Layer | 366 | 0.2237 | 1,056 | 0 | 0 | 1,056 |
| GPT6-Astra Ultra | 283 | 0.1698 | 14 | 2,338 | 2,345 | 4,697 |
| MagicLayer | – | 0.1414 | 1,884 | 1,987 | 0 | 3,871 |
| **DesignX (Ours)** | **47** | **0.1382** | **2,150** | **3,713** | **2,233** | **8,096** |

and difficult to capture with predefined metrics. We therefore adopt a human preference evaluation: given a design request and the corresponding outputs from different methods, evaluators are simply asked which result they would prefer to use in that scenario.

Existing raster decomposition methods, such as Qwen-Image-Layer, primarily focus on separating an image into multiple RGBA layers. However, practical visual designs also contain editable text, shapes, and line elements, whose reconstruction remains largely unexplored and lacks dedicated evaluation benchmarks. To bridge this gap, we introduce the DesignX Decomposition Benchmark, consisting of 150 test samples spanning diverse design scenarios. The benchmark contains both human-designed and AI-generated visual designs.

Since layered decomposition generally admits multiple valid solutions and thus has no unique ground truth, we develop an evaluation skill based on GPT-6 Astra. The evaluation considers both the number of valid decomposed elements and their reconstruction quality across three element types: **(1) Raster layers.** A layer is considered valid if it contains semantically meaningful content rather than spurious artifacts and does not duplicate content already present in other layers. Its reconstruction quality is considered successful if no obvious artifacts or severe blurring are observed. **(2) Text layers.** A valid text layer must correspond to an actual text element and must not remain embedded in raster layers. Reconstruction quality is evaluated by comparing its font family, font size, position, letter spacing, line spacing, and other typographic attributes with the source design. **(3) Simple vector layers.** A valid vector layer must correspond to an actual simple vector element and must not be duplicated in other layers. Its reconstruction is considered successful when its appearance, color, stroke width, and size approximately match the source element. Overall, better decomposition extracts more valid editable elements while maintaining reconstruction quality.

## 4 Experiments

**Evaluation on Visual Design Agent.** Given the same instruction, we compare direct generation with GPT-Image-2 against our visual design agent, which first retrieves the most suitable template and then generates the design accordingly. Human preference results in Tab. 1 show that agent-generated designs are preferred in 82.96% of comparisons, demonstrating the effectiveness of template guidance. This result is achieved with fully automatic template selection; allowing users to choose preferred templates could improve both design quality and user experience.

Table 1: Human preference.

| Agent | Original |
|---|---|
| 82.96% | 17.04% |

**Evaluation on Raster Image Decomposition.** We compare DesignX with the open-source Qwen-Image-Layer and the commercial models MagicLayer and GPT6-Astra Ultra. Quantitative results are reported in Tab. 3, using the evaluation protocol in Sec. 3.3. Qwen-Image-Layer only decomposes raster images into raster layers, while MagicLayer additionally supports text but not simple vector graphics. GPT6-Astra Ultra, like DesignX, supports both text and vector decomposition. However, without a dedicated font library and text-attribute predictor, its font reconstruction is less accurate. It also relies on segmentation-based cutout rather than an efficient RGBA-based DiT, which often causes adjacent elements to remain attached.

As shown in Tab. 3, DesignX achieves the lowest LPIPS and the largest numbers of valid image, text, and vector layers, indicating better reconstruction consistency and more complete decomposition. It also requires substantially less inference time per image than existing open-source and commercial




Figure 3: Visual comparison of raster image decomposition. Our DesignX significantly outperforms current open-source models and commercial models (GPT-6-Astra Ultra and MagicLayer).

methods. Qualitative results in Fig. 3 further support these observations. Qwen-Image-Layer often leaves multiple elements entangled in the same layer, while MagicLayer suffers from blurry reconstruction and missed layers. GPT6-Astra Ultra provides stronger multimodal understanding, but its segmentation-based extraction and weaker font matching still limit separation quality in complex foreground-background regions.

**Collage Synthetic Data and Real Data.** As shown in Tab. 2, real data is particularly beneficial for image layer decomposition, while synthetic data contributes more substantially to text and vector reconstruction. We attribute this difference to data distribution. Although real data is generally of higher quality, it exhibits strong biases in text styles and geometric layouts, which may limit generalization to less common patterns. In contrast, synthetic data provides more balanced and diverse element combinations, improving robustness to underrepresented distributions.

**Validation on Multi-head Text Predictor.** As shown in Tab. 4, we compare our multi-head text predictor, which jointly predicts font family, font size, letter spacing, line spacing, boldness, and italics, with independent per-attribute networks. To ensure a fair comparison, we approximately

Table 4: Comparison between multi-head prediction and independent prediction of text attributes. AC denotes accuracy, while RE denotes relative error.

| Method | Param | Font Family (AC%)↑ | Font Size (RE%)↓ | Letter Spacing (RE%)↓ | Line Spacing (RE%)↓ | Is Strikethrough (AC%)↑ | Is Underlined (AC%)↑ | Is Bold (AC%)↑ | Is Italic (AC%)↑ |
|---|---|---|---|---|---|---|---|---|---|
| Text Predictor (V1) | 25.0M | 93.25 | - | - | - | - | - | - | - |
| Text Predictor (V2) | 15.0M | - | 5.74 | - | - | - | - | - | - |
| Text Predictor (V3) | 15.0M | - | - | 6.68 | - | - | - | - | - |
| Text Predictor (V4) | 15.0M | - | - | - | 2.19 | - | - | - | - |
| Text Predictor (V5) | 10M | - | - | - | - | 100.0 | - | - | - |
| Text Predictor (V6) | 10M | - | - | - | - | - | 100.0 | - | - |
| Text Predictor (V7) | 10M | - | - | - | - | - | - | 94.35 | - |
| Text Predictor (V8) | 10M | - | - | - | - | - | - | - | 97.71 |
| Multi-Head Text Predictor (Ours) | 87.9M | **93.73** | **1.28** | **2.97** | **0.76** | **99.60** | **99.85** | **98.40** | **99.16** |

Table 5: Comparison between the VLM and our multi-head text predictor for text attribute prediction. AC denotes accuracy, while RE denotes relative error.

| Method | Param | Font Family (AC%)↑ | Font Size (RE%)↓ | Letter Spacing (RE%)↓ | Line Spacing (RE%)↓ | Is Strikethrough (AC%)↑ | Is Underlined (AC%)↑ | Is Bold (AC%)↑ | Is Italic (AC%)↑ |
|---|---|---|---|---|---|---|---|---|---|
| VLM | 35B | 57.90 | 10.59 | 14.30 | 8.82 | 94.67 | 92.18 | 90.05 | 90.12 |
| Multi-Head Text Predictor (Ours) | 87.9M | **93.73** | **1.28** | **2.97** | **0.76** | **99.60** | **99.85** | **98.40** | **99.16** |

match the overall parameter budgets, while allocating relatively larger models to the independent predictors for the more challenging tasks of font family and font size estimation. Despite using a unified backbone, the multi-head formulation consistently improves prediction accuracy across all attributes. We attribute these gains to the strong correlations among typographic properties. For example, font family and size are closely related to character appearance and spatial layout, while letter spacing and line spacing are jointly reflected in the overall text arrangement. Jointly modeling these attributes allows their predictions to share complementary visual cues and mutually constrain one another, leading to more accurate and coherent typography reconstruction.

**Comparsion on Text predictor and VLM.** As shown in Tab. 5, we compare two schemes for text attribute prediction: directly predicting typographic attributes (e.g., font family and size) with a VLM, and using a dedicated Text Attribute Predictor. On our benchmark, the specialized text attribute predictor consistently outperforms the VLM in both large-scale font classification and fine-grained attribute regression, such as font size prediction. This is because VLMs are better suited for tasks requiring high-level understanding and flexible output, such as layer decomposition planning and code generation. In contrast, text attribute prediction is essentially a visual classification and regression problem, where a dedicated predictor achieves higher accuracy with greater efficiency.

## 5 Conclusion

To lower the barrier to professional visual design and enable users to easily edit AI-generated results with existing design tools, we propose DesignX. We first construct and maintain a high-quality visual design template library covering diverse categories and scenarios. To efficiently utilize this large-scale library, we introduce three design skills: a cataloging skill that organizes templates with structured attributes; a retrieval skill that performs coarse-to-fine search to identify templates aligned with user requirements; and a generation skill that adapts retrieved templates to user intent. However, the visual agent still produces raster outputs, limiting wider usage. To this end, we further study raster image decomposition. Unlike previous methods that mainly decompose images into raster layers, DesignX introduces a unified representation for raster elements, vector graphics, and text. Our framework consists of three key components: a VLM, an RGBA-based DiT, and a text-attribute predictor. The VLM parses raster designs according to our structured protocol and directly generates code for simple vector shapes. The RGBA-based DiT performs description-guided extraction of complex visual assets while completing occluded regions. The text-attribute predictor jointly predicts typography attributes through a multi-head transformer architecture. We further introduce synthetic collage data to improve model robustness and generalization. Finally, we present DesignX benchmark for comprehensive evaluation of visual design agents and raster image decomposition. Extensive experiments demonstrate the effectiveness of DesignX.

## AI USE STATEMENT

AI-assisted tools were used solely for language editing and manuscript-level quality control. This included improving clarity and readability, correcting grammatical and typographical errors, and identifying potential factual inaccuracies, ambiguous statements, and inconsistencies across sections. All suggestions were reviewed and verified by the authors before incorporation. AI tools were not used to generate experimental results. The authors take full responsibility for this paper.

## REFERENCES

Jingye Chen, Zhaowen Wang, Nanxuan Zhao, Li Zhang, Difan Liu, Jimei Yang, and Qifeng Chen. Rethinking layered graphic design generation with a top-down approach. In *ICCV*, 2025a. 3

Junwen Chen, Heyang Jiang, Keming Wu, Ji Li, Chao Zhang, Keiji Yanai, Dong Chen, and Yuhui Yuan. Prismlayers: Open data for high-quality multi-layer transparent image generative models. 2025b. 4

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 2

Kaituo Feng, Manyuan Zhang, Shuang Chen, Yunlong Lin, Kaixuan Fan, Yilei Jiang, Hongyu Li, Dian Zheng, Chenyang Wang, and Xiangyu Yue. Gen-searcher: Reinforcing agentic search for image generation. *arXiv preprint arXiv:2603.28767*, 2026. 3

Jun He, Junyan Ye, Zilong Huang, Dongzhi Jiang, Chenjue Zhang, Leqi Zhu, Renrui Zhang, Xiang Zhang, and Weijia Li. Mind-brush: Integrating agentic cognitive search and reasoning into image generation. *arXiv preprint arXiv:2602.01756*, 2026. 3

Peidong Jia, Chenxuan Li, Yuhui Yuan, Zeyu Liu, Yichao Shen, Bohan Chen, Xingru Chen, Yinglin Zheng, Dong Chen, Ji Li, et al. Cole: A hierarchical generation framework for multi-layered and editable graphic design. *arXiv preprint arXiv:2311.16974*, 2023. 3

Zhen Lin, Qiujie Xie, Minjun Zhu, Shichen Li, Qiyao Sun, Enhao Gu, Yiran Ding, Ke Sun, Fang Guo, Panzhong Lu, et al. Autofigure-edit: Generating editable scientific illustrations via reference-guided styling. In *ACL*, 2026. 3

Cheng Liu, Yiren Song, Haofan Wang, and Mike Zheng Shou. Ominpsd: Layered psd generation with diffusion transformer. In *CVPR*, 2026. 2, 3

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021. 6

Vlad-Constantin Lungu-Stan, Ionuț Mironică, and Mariana-Iuliana Georgescu. Lade: Unified multi-layered graphic media generation and decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6031–6040, 2026. 3

Yifan Pu, Yiming Zhao, Zhicong Tang, Ruihong Yin, Haoxing Ye, Yuhui Yuan, Dong Chen, Jianmin Bao, Sirui Zhang, Yanbin Wang, et al. Art: Anonymous region transformer for variable multi-layer transparent image generation. In *CVPR*, 2025. 3

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

Ciara Rowles, Reshinth Adithyan, Nikhil Pinnaparaju, Vikram Voleti, and Mark Boss. Stable-layers: Fine-tuning image layer decomposition models with vlm-scored reinforcement learning. *arXiv preprint arXiv:2605.30257*, 2026. 3

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 2

Tomoyuki Suzuki, Kang-Jun Liu, Naoto Inoue, and Kota Yamaguchi. Layerd: Decomposing raster graphic designs into layers. In *ICCV*, 2025. 3

Zhicong Tang, Zhao Zhang, Jingye Chen, Mohan Zhou, Yifan Pu, Yuchi Liu, Yalong Bai, Ethan Smith, and Yuhui Yuan. Mrt: Masked region transformer for layered image generation and editing at scale. *arXiv preprint arXiv:2605.27235*, 2026. 3

Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. Mulan: A multi layer annotated dataset for controllable text-to-image generation. In *CVPR*, 2024. 3

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 6

Bin Xia, Shiyin Wang, Yingfan Tao, Yitong Wang, and Jiaya Jia. Llmga: Multimodal large language model based generation assistant. In *ECCV*, 2024. 3

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 6

Shengming Yin, Zekai Zhang, Zecheng Tang, Kaiyuan Gao, Xiao Xu, Kun Yan, Jiahao Li, Yilei Chen, Yuxiang Chen, Heung-Yeung Shum, et al. Qwen-image-layered: Towards inherent editability via layer decomposition. In *CVPR*, 2026. 2, 3

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 6

Jooyeol Yun, Jintae Park, Hyesu Lim, Junha Hyung, Hyungjin Chung, and Jaegul Choo. Redesign: Recovering editable design structures from images via agentic decomposition. In *ECCV*, 2026. 3

Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024. 3

Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, et al. Qwen-image-agent: Bridging the context gap in real-world image generation. *arXiv preprint arXiv:2606.26907*, 2026. 3

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. Pptagent: Generating and evaluating presentations beyond text-to-slides. In *EMNLP*, 2025. 3

Dawei Zhu, Rui Meng, Yale Song, Xiyu Wei, Sujian Li, Tomas Pfister, and Jinsung Yoon. Paperbanana: Automating academic illustration for ai scientists. In *ICML*, 2026a. 3

Minjun Zhu, Zhen Lin, Yixuan Weng, Panzhong Lu, Qiujie Xie, Yifan Wei, Sifan Liu, Qiyao Sun, and Yue Zhang. Autofigure: Generating and refining publication-ready scientific illustrations. *arXiv preprint arXiv:2602.03828*, 2026b. 3