Towards Design Compositing

Abhinav Mahajan[1*,†] Abhikhya Tripathy[1,†] Sudeeksha Reddy Pala[2,†] Vaibhav Methi[3,†]
K J Joseph[4] Balaji Vasan Srinivasan[4]

[1]Carnegie Mellon University [2]IIT Kharagpur [3]IIT Kanpur [4]Adobe Research
{abhinavm, abhikhyt}@andrew.cmu.edu, {josephkj, balsrini}@adobe.com

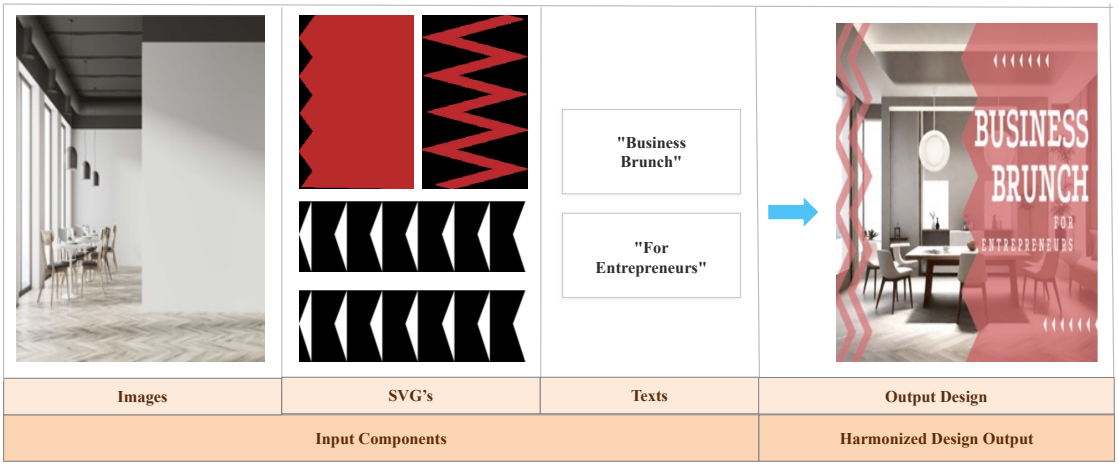

Figure 1. **Design Compositing with GIST:** We introduce **GIST** (**G**rounded **I**dentity-preserving **S**tylized composi**T**ion), a compositing stage that transforms and harmonizes visual elements, images and SVGs, at their predicted layout positions rather than naively pasting them, while preserving their semantic identity. Note how the image is stylized along with retaining the dining room scene and feels cohesive with the surrounding elements. The SVGs are similarly transformed for visual harmony. This is in contrast to existing methods which focus on layout prediction (spatial arrangement) and typography (font, color, and placement of text), without transforming the content itself.

Abstract

*Graphic design creation involves harmoniously assembling multimodal components such as images, text, logos, and other visual assets collected from diverse sources, into a visually-appealing and cohesive design. Recent methods have largely focused on layout prediction or complementary element generation, while retaining input elements exactly, implicitly assuming that provided components are already stylistically harmonious. In practice, inputs often come from disparate sources and exhibit visual mismatch, making this assumption limiting. We argue that identity-preserving stylization and compositing of input elements is a critical missing ingredient for truly harmonized components-to-design pipelines. To this end, we propose **GIST**, a training-free, identity-preserving image compositor that sits between layout prediction and typography generation, and can be plugged into any existing components-to-design or design-refining pipeline without modification. We demonstrate this by integrating **GIST** with two substantially different existing methods, LaDeCo [33] and Design-o-meter [15]. **GIST** shows significant improvements in visual harmony and aesthetic quality across both pipelines, as validated by LLaVA-OV and GPT-4V on aspect-wise ratings and pairwise preference over naive pasting. Project Page:* abhinav-mahajan10.github.io/GIST/.

1. Introduction

Graphic designs are ubiquitous in daily life, appearing in advertisements, social media posts, posters, banners, presentations, and digital user interfaces. Two key factors determining the effectiveness of a graphic design are whether it is visually-appealing, eye-catching, coherent, and aesthetically cohesive, and second, whether it clearly communicates the designer's intended message. Designs that satisfy both criteria are generally more impactful and engaging.

However, creating such designs at scale is challenging since modern designers often work under tight deadlines

[†]Work done during internship/employment at Adobe Research.

and handle a large volume of design tasks. This makes it difficult to manually ensure both visual quality and consistency across outputs, which automated graphic design generation systems can help with by accelerating routine workflows while maintaining visual appeal and adherence to user intent. Such systems can also lower the barrier to entry for novice users by providing design capabilities that would otherwise require substantial artistic expertise.

An important instance of this is the components-to-design workflow. Graphic designers often assemble multimodal components such as images, text, logos, SVG elements, and other visual assets collected from diverse sources, into a cohesive aesthetic design. Recent works have made encouraging progress on automating workflow. Some methods focus on predicting the spatial arrangement of input elements [22, 23, 33], while others generate complementary components that complete the design in harmony with the provided inputs [51].

A key characteristic of earlier works in this domain is that they preserve the input components exactly as provided. This constraint works well only under the assumption that the user-specified assets are already visually harmonious and need only be arranged appropriately or supplemented with missing or additional components. This assumption becomes a strong limitation in realistic settings, where input components are often collected from diverse sources and may differ substantially in color palette, rendering style, tone, texture, or overall visual appearance. In such cases, simply arranging elements that are not harmonious with respect to each other is insufficient to produce a harmonious design. The restriction of preserving inputs exactly contradicts the very objective of true design harmony.

We argue that true harmonized design generation from input components requires input-conditioned, identity-preserving component stylization and composition. Here, composition refers to adapting the appearance of existing design elements so that they become visually cohesive with the rest of the design while still preserving the semantic identity and designer intent associated with those elements. Thus, design composition may involve, for example, compositing image elements to better match a common style or color palette, or adjusting text typography attributes such as color and font to complement surrounding elements. Existing works have at best tackled stylization for textual elements, while largely overlooking the **composition of image-based elements**. This is a major research gap since image elements often dominate the overall aesthetics of a graphic design, therefore playing a critical role in perceived visual harmony and aesthetics.

To address this gap, we propose **GIST:** Grounded Identity-preserving Stylized composiTion, a training-free, identity-preserving image composition method that sits between layout prediction and typography, and can be plugged into any existing components-to-design pipeline without modification. For complete frameworks like LaDeCo [33], **GIST** slots in as a compositing stage; for layout-only methods like Design-o-meter [15] (which natively supports only repositioning texts, treating them as static renders), we can extend them toward full components-to-design generation by additionally incorporating typography prediction from works like COLE [26] or FlexDM [22], in both cases complementing strong existing models toward the larger goal of true design harmony. A key advantage is that it adapts an existing foundational MLLM without any additional finetuning, exploiting its architectural bottleneck to perform generative image composition with enhanced identity preservation.

Concretely, our key contributions are as follows:

- A novel identity-preserving image composition method, addressing the most critical missing functionality for harmonized components-to-graphic design generation.
- We demonstrate the plug-and-play nature of this method by integrating it with strong existing layout and typography models within an end-to-end pipeline for automated harmonized design generation from components. We show this for two distinct pipeline setups.
- We provide quantitative and qualitative results that validate our claims and show the importance of image composition for visually cohesive design generation.

2. Related Works

Optimizing Specific Graphic Design Aspects: A well-researched line of work in refining graphic designs is layout generation, where design elements are spatially arranged for visual harmony and comprehensibility. Earlier works achieved this by maximizing energy functions [38, 39] or using aesthetic scores [3, 7, 21, 25, 29, 40, 50], which suffered from limitations such as high computational costs and human annotation bias respectively. These methods also didn't scale well to complex and diverse designs. Some transformer-based approaches [17, 22, 48, 49] have been developed which improved the latency and diversity of layout generation. FlexDM [22] employs multitask learning in a single transformer-based model to solve various design tasks, including layout and typography, but suffers when the number of masked tokens increases. Recently, layout generation methods have employed diffusion models [6, 16, 20, 23, 52], Large Language Models (LLMs) [4, 32, 42, 45], and Large Multimodal Models (LMMs) [10, 28, 33, 53]. Although layout configuration is critical, our work goes a step further by addressing the fundamental challenge of composing and stylizing image-based elements at the predicted layout positions, which is imperative for true composition.

Automated Graphic Design Generation: Recently, COLE [26] proposed a pipeline approach for generating de-

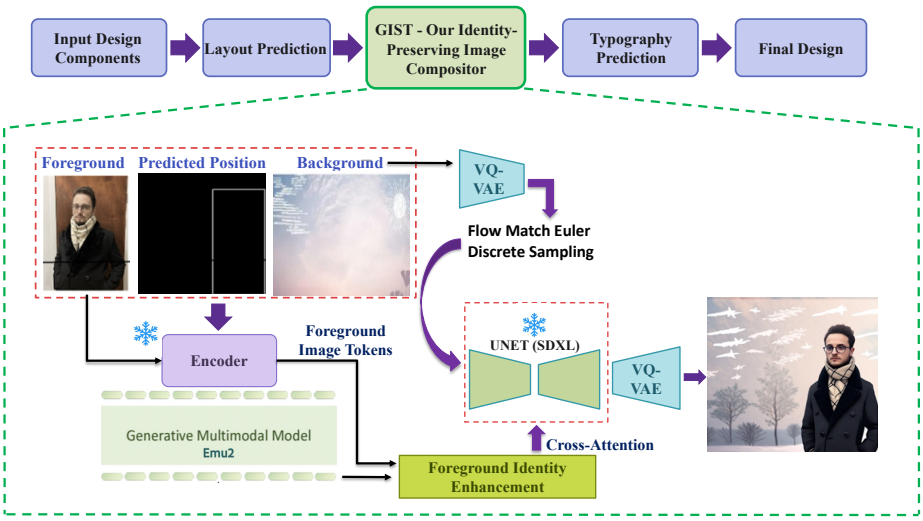

Figure 2. **Overview of the Components-to-Design pipeline with GIST. GIST** sits between layout prediction and typography as a plug-and-play compositing stage. Given a foreground element and its predicted position, the background canvas is inverted via Flow Matched Euler Discrete Sampling to initialize the denoising latent. In parallel, Emu-2's frozen visual encoder produces identity tokens from the foreground, corrected via our CA-guided token injection before conditioning SDXL's UNet alongside the generative tokens from Emu-2's LLaMA decoder. This process repeats sequentially for each element; the final composed backing image is passed to typography prediction to yield the complete design.

signs from scratch given a user intention. They achieve this by training a design planner, which generates a structured JSON of the design contents, then render the visual assets using cascaded diffusion, and finally use a typography LMM for predicting the appropriate text data. OpenCOLE [24] is an open source implementation of COLE with some changes. Although these methods provide some edit-ability, designers cannot use input assets of their choice. IGD [41] formulates instructional graphic design generation as multimodal layer generation in an editable format. DreamPoster [19] studies image-conditioned generative poster design. These methods improve controllability and quality, but mainly focus on overall design synthesis rather than harmonizing potentially mismatched user-provided elements.

Components-to-Design Generation: LaDeCo [33] introduces a layered framework for automatic graphic design composition from multimodal elements. CreatiDesign [51] models this as a diffusion process jointly conditioned on input images, text, and layout constraints. FlexDM [22] performs design synthesis by simultaneously masking and predicting element positions and text attributes. However, all of these methods preserve input components exactly or adjust only limited attributes such as typography and position, restricting their ability to handle visually mismatched inputs from diverse sources. Our work addresses this gap: unlike prior works that treat inputs as fixed, we adapt their appearance while preserving their identity, complementing existing layout and typography methods within end-to-end pipelines.

3. Method

As discussed in Sec. 2, existing components-to-design methods either preserve input assets exactly [22, 33], limiting their ability to handle visually mismatched inputs or generate assets from scratch [24, 26], sacrificing user-specified content. Neither paradigm addresses what we argue is a critical missing ingredient: *stylizing and compositing user-provided image and SVG elements for visual harmony while preserving their semantic identity*.

To address this, we propose **GIST**, a training-free, identity-preserving image composition method (Sec. 3.1) designed to be *plug-and-play*. Layout prediction, spatially arranging input visual elements has been well-explored by works such as LaDeCo [33] and Design-o-meter [15], and typography prediction, determining placement, font, and color for text elements has been tackled by methods like COLE [26], OpenCOLE [24] and FlexDM [22]. **GIST** sits between these two stages: given predicted elements and element positions as input, it produces a harmonized backing image ready for typography. Because it makes no assumptions about the surrounding modules, it can complement any existing components-to-design pipeline. We demonstrate this in Sec. 3.2 by integrating **GIST** into two distinct end-to-end pipelines built around LaDeCo and Design-o-meter.

3.1. Identity-Preserving Composition

Image composition involves placing a foreground element onto a background such that the result appears as a unified, coherent image. The challenge is threefold: (i) the foreground's semantic identity must be preserved, (ii) the

background must remain faithful to the original, and (iii) the composed result must be stylistically cohesive. Naive copy-paste satisfies (i) and (ii) but fails at (iii), producing jarring visual discontinuities when elements originate from different sources. Purely generative approaches achieve (iii) but often degrade (i) and (ii). Where the composition operation fits within the full design pipeline is detailed in Sec. 3.2; here, we focus on the composition itself.

MLLMs are well-suited for this task, as their pretraining instills a strong prior over style, color, and texture coherence while supporting spatially grounded generation. Visual elements (Images/SVGs) are composed sequentially given their predicted layouts, with the updated canvas serving as the background at each step; the full pipeline is detailed in Sec. 3.2. At each step, rather than naively pasting, we prompt an MLLM with the background render, a foreground render and its caption (obtainable via any captioning model, like Qwen [2]), and the target position on the canvas, producing a harmonized composite. For this, we require a model whose internals are accessible enough to admit training-free manipulation, newer models such as FLUX Kontext [5] produce good results but with limited training-free manipulations to make it stronger, making principled identity-preserving interventions difficult without fine-tuning. Emu-2 [44] uniquely exposes a 64-token bottleneck between its LLaMA decoder and SDXL renderer, which is precisely where we intervene since despite its stylization quality, off-the-shelf Emu-2 suffers from foreground and background identity loss. We address this with two training-free enhancements: **Cross-Attention Guided Token Injection** [18, 34, 46] (Sec. 3.1.1) and **Latent Initialization** [27, 37, 47] (Sec. 3.1.2).

3.1.1. Cross-Attention Guided Token Injection

Off-the-shelf Emu-2 produces stylistically harmonized composites but suffers from foreground and background identity loss: the generated output may capture the right *style* while failing to faithfully reproduce the foreground subject or the background canvas. The root cause is that the 64 image tokens produced by Emu-2's LLaMA decoder, which condition SDXL's rendering, are optimized for coherent generation rather than identity fidelity. We seek to correct this by injecting identity-specific information back into these tokens, but only where it matters spatially, so as not to destroy the model's stylization intent.

Autoencoded Tokens as identity embeddings. A key property of Emu-2 is that its Visual Encoder (EVA-CLIP) and SDXL Decoder are jointly trained as an autoencoder: encoding an image directly through the visual encoder, bypassing LLaMA entirely, yields 64 tokens from which SDXL reconstructs the image near-perfectly. This means these tokens carry rich, fine-grained identity information. We exploit this: for any input image, encoding it through the visual encoder gives us an *identity token set* that faithfully represents that image's appearance.

Concretely, we maintain two token sets: T_{gen}, produced by LLaMA model of Emu-2, from the full compositional prompt (carrying stylization intent), and T_{auto}, produced by encoding a naive foreground-on-background composite through the visual encoder alone (carrying identity). The challenge is deciding *which* tokens in T_{gen} to correct with identity from T_{auto}, and *how much*, a global replacement would collapse stylization; no replacement leaves identity unprotected.

Spatial relevance via cross-attention. The SDXL UNet's cross-attention maps provide exactly the signal we need. Each of the 64 tokens has a spatial CA map of shape $[H_{\text{lat}} \times W_{\text{lat}}]$, encoding how strongly each pixel location attends to that token during decoding. A token with high attention within the foreground bounding box is primarily responsible for rendering the foreground; one with high attention in the background region governs background appearance. We use these maps to assign each token a foreground and background relevance score:

$$r_{\text{fg}}[i] = \frac{\max(\mathbf{CA}[i] \odot \mathbf{m}_{\text{fg}})}{\max(\mathbf{CA}[i])}, \tag{1}$$

$$r_{\text{bg}}[i] = \frac{\max(\mathbf{CA}[i] \odot \mathbf{m}_{\text{bg}})}{\max(\mathbf{CA}[i])}, \tag{2}$$

where $\mathbf{m}_{\text{fg}}$ and $\mathbf{m}_{\text{bg}}$ are foreground and background binary masks. $\mathbf{m}_{\text{fg}}$ is derived from the element's alpha channel, or a hard bounding box mask if unavailable. To obtain these maps, we run a single lightweight scoring forward pass through the UNet at a mid-noise timestep using T_{auto} as conditioning, and average CA maps across all attention layers.

Selective identity injection. Armed with per-token relevance scores, we select the top-N_{fg} tokens by r_{fg} (set $\mathcal{S}_{\text{fg}}$) as the primary drivers of foreground appearance, and the top-N_{bg} by r_{bg} (set $\mathcal{S}_{\text{bg}}$) for background. We then blend the corresponding identity tokens from T_{auto} into T_{gen}:

$$T_{\text{final}}[\mathcal{S}_{\text{fg}}] = (1-\beta_{\text{fg}}) \cdot T_{\text{gen}}[\mathcal{S}_{\text{fg}}] + \beta_{\text{fg}} \cdot T_{\text{auto}}[\mathcal{S}_{\text{fg}}], \tag{3}$$

$$T_{\text{final}}[\mathcal{S}_{\text{bg}}] = (1-\beta_{\text{bg}}) \cdot T_{\text{gen}}[\mathcal{S}_{\text{bg}}] + \beta_{\text{bg}} \cdot T_{\text{auto}}[\mathcal{S}_{\text{bg}}], \tag{4}$$

with β_{fg}=0.3 and β_{bg}=0.2. All remaining tokens retain T_{gen} unchanged. This injects identity precisely where each region is most represented in generation, without globally degrading stylization. The full procedure is summarized in Algorithm 1.

3.1.2. Background Fidelity: Latent Initialization

To further improve background fidelity, we initialize the SDXL denoising process from a partially noised version of the background canvas rather than pure noise. Specifically, we encode the background through the VQ-VAE encoder and apply the Flow Matched Euler Discrete Scheduler [13] (the same scheduler used in Emu-2's training) to

Algorithm 1 Cross-Attention Guided Token Injection

Require: Foreground image, background canvas, target bounding box, masks $\mathbf{m}_{\text{fg}}$, $\mathbf{m}_{\text{bg}}$
Ensure: Blended token set T_{final}
$T_{\text{gen}} \leftarrow$ LLaMA forward pass on compositional prompt ▷ Stylization tokens
$T_{\text{auto}} \leftarrow$ Visual Encoder on fg-on-bg composite ▷ Identity tokens
Run UNet scoring pass with T_{auto}; collect and average CA maps across layers
for each token $i \in \{1, \ldots, 64\}$ **do**
    $r_{\text{fg}}[i] \leftarrow \max(\text{CA}[i] \odot \mathbf{m}_{\text{fg}}) \;/\; \max(\text{CA}[i])$
    $r_{\text{bg}}[i] \leftarrow \max(\text{CA}[i] \odot \mathbf{m}_{\text{bg}}) \;/\; \max(\text{CA}[i])$
end for
$\mathcal{S}_{\text{fg}} \leftarrow$ top- N_{fg} indices by r_{fg}; $\mathcal{S}_{\text{bg}} \leftarrow$ top- N_{bg} indices by r_{bg}
$T_{\text{final}} \leftarrow T_{\text{gen}}$
$T_{\text{final}}[\mathcal{S}_{\text{fg}}] \leftarrow (1-\beta_{\text{fg}})\, T_{\text{gen}}[\mathcal{S}_{\text{fg}}] + \beta_{\text{fg}}\, T_{\text{auto}}[\mathcal{S}_{\text{fg}}]$
$T_{\text{final}}[\mathcal{S}_{\text{bg}}] \leftarrow (1-\beta_{\text{bg}})\, T_{\text{gen}}[\mathcal{S}_{\text{bg}}] + \beta_{\text{bg}}\, T_{\text{auto}}[\mathcal{S}_{\text{bg}}]$
return T_{final}

invert it. This linear interpolation between data and noise preserves more structural information than alternatives such as DDIM [43] or LCM [36] inversion (see Supp. for comparisons). Starting denoising from this initialized latent biases generation toward the original background appearance while still allowing the model to harmonize foreground and background.

3.2. Augmenting Design Generation with GIST

The composition method described above operates on individual image elements given a background and target position. To produce complete graphic designs from multimodal components, current Components-to-Design frameworks [33] do layout prediction of all elements (Images, SVG's) and Typography prediction (Positions, Font and Colours of texts) and render. We embed **image composition** in between layout prediction and **typography prediction** (see Fig. 2). Because our composition module only requires element positions as input and produces a composed backing image as output, it slots into any existing pipeline without modification to the surrounding stages. We demonstrate this with two configurations representing different design paradigms.

3.2.1. Layout Prediction

We integrate with two layout methods. **LaDeCo** [33] is a state-of-the-art LMM-based method that predicts element positions, sizes, and layer ordering from input components, and additionally provides typography attributes. **Design-o-meter** [15] takes a different approach: it trains a Siamese aesthetic scorer on good/bad design pairs and uses NSGA-II [11] optimization to refine layout coordinates by maximizing the learned aesthetic objective. Both output spatial coordinates for each visual element, which serve as direct input to our composition module.

3.2.2. Compositing

Using the layout coordinates from Sec. 3.2.1, we apply the method from Sec. 3.1 to sequentially compose each visual element onto the canvas. We initialize the canvas with the root background element. For each subsequent element in the predicted layer order, we: (1) caption the foreground element, (2) compute T_{auto} and T_{gen}, (3) perform token injection and SA manipulation, (4) invert the current canvas to obtain the initial latent, and (5) generate the composed image with SDXL. For SVG elements, we remove the background from the generated output before pasting it back at the target position; for image elements, the latent initialization ensures natural blending. The canvas is updated after each element, so later elements are conditioned on all previously composed ones.

3.2.3. Typography

Given the composed backing image, the final stage predicts text placement, font, and color for each input text element. Since diffusion models still struggle with artifact-free stylized text rendering, we rely on dedicated typography modules followed by a graphic renderer. For the LaDeCo-based pipeline, we use LaDeCo's own typography predictions. For the Design-o-meter pipeline, we train a typography LMM inspired by COLE [26]: we finetune InternLM-XComposer2 [12] with a two-stage strategy, first grounding the model on single-element typography detection, then finetuning on full-design typography prediction (details in Supp.). The plug-and-play nature of our composition module means it is agnostic to the typography method used downstream.

4. Experiments and Results

We evaluate our method along two axes: (i) the quality of end-to-end designs produced by the full pipelines when **GIST** is plugged into them, and (ii) the effectiveness of our identity-preserving composition module in isolation. We first describe the experimental setup, then present quantitative and qualitative results for complete design generation, followed by a focused analysis of the composition module and ablations validating our design choices.

4.1. Experimental Setup

We evaluate on the Crello test set [49], comprising 1,500 graphic designs with layer-wise metadata, providing background images, foreground image elements, SVG shapes, and text assets extracted from real designs. Since these components originate from the same ground-truth design, they are already stylistically compatible, making our evaluation *conservative*: our method targets the harder real-world setting where inputs come from disparate sources.

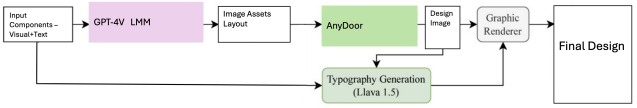

Figure 3. OpenCOLE++ pipeline overview

Baselines. We compare against the following methods: **FlexDM** [22]: a multimodal transformer that jointly predicts layout and typography through masked prediction. **GPT-4o**: layout planning via GPT-4o with few-shot prompting, followed by direct alpha-composite paste of elements. **LaDeCo** [33]: state-of-the-art LMM-based layout and typography prediction, with elements naively pasted at predicted positions. **Design-o-meter** [15]: layout-only optimization via NSGA-II, no typography. **GT**: ground-truth Crello designs as a human-quality reference.

Since no prior work directly addresses components-to-design generation **with composition**, we evaluate across two distinct comparison axes. The first is an *isolated* comparison against **LaDeCo**: we share the same layout and typography backbone and swap only the compositing step, directly measuring the contribution of **GIST** over naive pasting. This is intentional, our goal is not to compete with layout methods but to complement them, and this setup isolates exactly that effect. The second is a *holistic* comparison against **OpenCOLE++**, a different baseline we construct by adapting OpenCOLE [24] to accept input components (see Fig. 3): GPT-4V replaces OpenCOLE's design planner for layout prediction, and AnyDoor [8] is its image generator for diffusion-based composition (We intentionally use AnyDoor instead of naive pasting, to compete with **GIST**). We pair this with our Design-o-meter [15]-based pipeline, which uses **GIST** for composition and our trained typography module. This holistic comparison reflects the compounded effect of swapping every pipeline component simultaneously, and benchmarks our full system against an alternative paradigm for design composition.

Evaluation metrics. Following prior work [10, 26], we employ LLaVA-OV [30] for automated design evaluation along two protocols: (1) *Aspect-wise rating*: each design is scored on a 1-10 scale across five aspects - Design & Layout, Content Relevance, Typography & Color, Graphics & Imagery, and Innovation & Originality. We don't report the Design & Layout, Typography & Color numbers as we leverage the layout & typography models of the baseline, in this case, LaDeCo, so the metric is unchanged. (2) *Pairwise preference*: LLaVA-OV is asked to choose the more aesthetic design between our output and the LaDeCo paste-only baseline, over 1,500 paired comparisons. For the OpenCOLE++ comparison, we use GPT-4V [1] for both rating and pairwise voting, following the evaluation protocol of COLE [26].

Table 1. LLaVA-OV ratings (1–10) on the Crello test set. We report Content Relevance, Graphics & Imagery, and Innovation & Originality - the aspects most sensitive to our compositing contribution, along with the mean across all five aspects. Design & Layout and Typography & Color are omitted from individual columns as they primarily reflect the shared LaDeCo layout backbone; they are included in the mean.

Method	Content	Graphics	Innovation	Mean↑
FlexDM [22]	5.29	5.09	4.54	5.13
GPT-4o	6.49	6.27	5.69	6.32
LaDeCo [33]	**7.96**	7.74	6.93	7.77
Ours (w/ LaDeCo)	7.89	**7.83**	**7.05**	**7.79**
GT	8.17	7.93	7.15	7.95

Table 2. GPT-4V ratings (1–10) comparing our Design-o-meter (DoM) based pipeline against OpenCOLE++. Aspects: (i) Design & Layout, (ii) Content Relevance, (iii) Typography & Color, (iv) Graphics & Imagery, (v) Innovation & Originality.

Method	(i)	(ii)	(iii)	(iv)	(v)	Mean↑
OpenCOLE++	4.6	5.2	4.0	5.2	5.3	4.9
Ours (w/ DoM)	**5.8**	**6.2**	**5.1**	**6.7**	**5.8**	**5.9**

4.2. Quantitative Evaluation

4.2.1. Comparison with LaDeCo

Quantitative analysis. Tab. 1 summarizes the LLaVA-OV ratings. Our method (LaDeCo with **GIST**) outperforms vanilla LaDeCo on *Graphics & Imagery* (7.83 vs. 7.74) and *Innovation & Originality* (7.05 vs. 6.93), the two dimensions most directly affected by image-level compositing, while remaining within noise on Content Relevance (7.89 vs. 7.96). The overall mean is essentially tied (7.79 vs. 7.77). This is the intended behavior: our compositor enhances aesthetic harmony and visual richness *without degrading content fidelity*. The gap to GT is narrow, 0.10 on Graphics & Imagery and 0.16 on the overall mean, indicating that our pipeline produces near-professional-quality designs. FlexDM and GPT-4o are substantially weaker across all dimensions, confirming that the LaDeCo layout backbone is important and that our gains are attributable to the compositing stage.

Design Improvement. Out of 1,500 designs in the Crello test set, **GIST** meaningfully improves upon the LaDeCo naive-pasting baseline in **40.3%** of cases, with an additional 14.7%

4.2.2. Comparison with OpenCOLE++

Quantitatively (Tab. 2), Design-o-meter+**GIST**+our typography model outperforms OpenCOLE++ across all five aspects, with particularly large gains on Graphics & Imagery (+1.5) and Content Relevance (+1.0), and achieves a 1.0-point higher mean rating (5.9 vs. 4.9). In GPT-4V

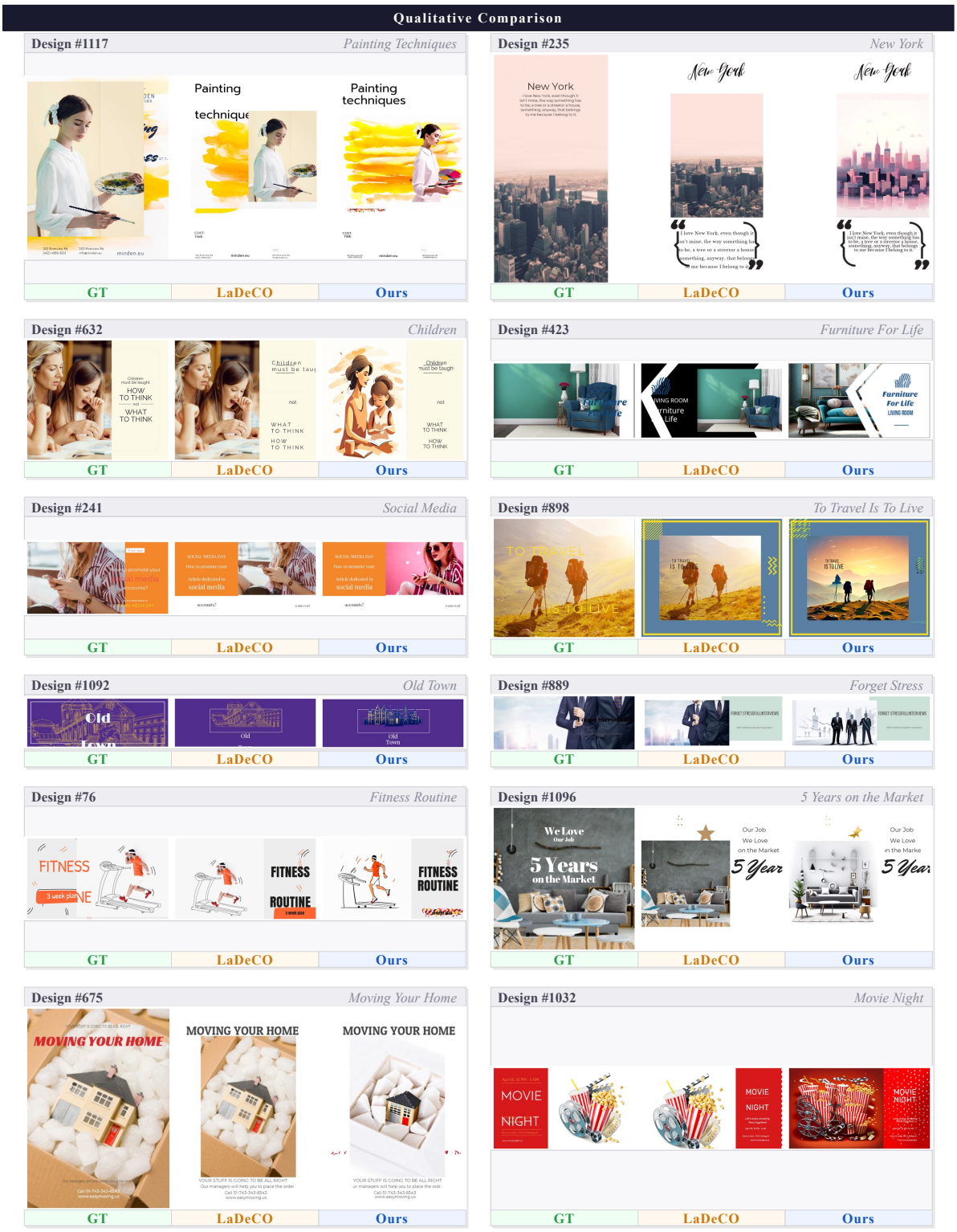

Figure 4. Qualitative comparison on Crello test samples. **Columns**: Input components, LaDeCo (paste-only), Ours (LaDeCo w/ **GIST**, and Ground Truth. In paste-only results, elements appear as flat cut-outs with visible edge discontinuities and color mismatches. Our method harmonizes lighting, color palette, and texture across elements, producing compositions that are visually cohesive and closer to GT.

Figure 5. OpenCOLE++ vs Ours (w/ Design-o-meter)

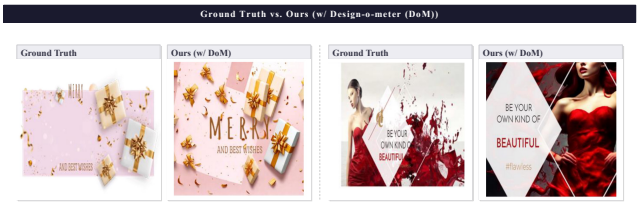

Figure 6. Ground Truth designs vs Ours (w/ Design-o-meter)

pairwise voting, our outputs are preferred **71.43%** of the time. Notably, this comparison uses our Design-o-meter-based pipeline rather than the LaDeCo backbone of Tab. 1, demonstrating that **GIST** produces strong results regardless of the layout module, a direct validation of its plug-and-play nature.

4.3. Qualitative Results

Fig. 4 shows representative examples comparing our method with LaDeCo against vanilla LaDeCo (naive paste-only) and GT. In the paste-only outputs, elements appear as flat overlays: edges are visible, lighting is inconsistent across layers, and color palettes clash. Our compositor harmonizes these elements into a unified scene, lighting becomes consistent, colors blend naturally, and the overall composition feels integrated rather than assembled. Compared to GT, our method occasionally produces more stylized variants; this is a natural consequence of the generative process, and accounts for some of the pairwise losses on Crello's pre-matched components.

Comparison with OpenCOLE++. Fig. 5, Fig. 6, compares our method against OpenCOLE++ and Ground Truth Crello Designs. While AnyDoor [8] (not all examples shown in the image) produces stylistically coherent outputs, it frequently fails to preserve the identity of foreground elements. Objects are altered, faces lose likeness, and branded assets become unrecognizable. Our method, through CA-guided token injection, achieves harmonization *without* sacrificing identity.

4.4. Ablation Studies

We ablate the two key components of our identity-preserving composition module: Cross-Attention Guided Token Injection (Sec. 3.1.1) and Latent Initialization (Sec. 3.1.2).

Table 3. Identity preservation metrics for our compositor vs. off-the-shelf Emu-2. Cosine similarity (↑) is computed on face embeddings for HFG; Manhattan (↓) and Euclidean (↓) distances are on image embeddings for HFG and SOG respectively.

Method	Cos. Sim. ↑	Manhattan ↓	Euclidean ↓
Emu-2 [44]	0.09	554.52	961.07
GIST (Ours)	**0.13**	**540.45**	**943.91**

Table 4. Ablation of composition components on HFG cosine similarity (↑).

Configuration	Cos. Sim. ↑
Full method (GIST)	**0.13**
w/o CA-Guided Token Injection	0.04

Identity preservation. To isolate the composition module from the full pipeline, we evaluate foreground identity retention on two controlled tasks: *Human Face Generation (HFG)*, compositing face images onto diverse backgrounds, and *Specific Object Generation (SOG)*, compositing identifiable everyday objects (data from CelebA [35], Stanford Background [14], and DreamEditBench [31]; details in Supp.). We measure cosine similarity, Manhattan distance, and Euclidean distance between embeddings of the input foreground and the composed output.

Tab. 3 shows that our method consistently outperforms off-the-shelf Emu-2 across all metrics, achieving higher cosine similarity and lower distances, confirming that our training-free interventions meaningfully improve foreground identity retention during composition.

Component ablation. Tab. 4 isolates the contribution of each proposed component by measuring face embedding cosine similarity on the HFG task. Removing either Cross-Attention Guided Token Injection causes a substantial drop in identity preservation, with both components contributing roughly equally. The full method achieves the highest similarity, validating that both interventions are necessary and complementary.

5. Conclusion and Limitations

We presented **GIST**, a training-free, identity-preserving image compositor that stylizes and harmonizes user-provided visual elements while retaining their semantic identity, and plugs seamlessly into any existing components-to-design pipeline. We hope this work motivates broader attention to input-conditioned composition as a critical missing ingredient in automated graphic design. Diffusion artifacts, unaddressed textual composition, and architectural ties to Emu-2's bottleneck remain open challenges; extending our approach to newer unified generative models such as FLUX [5] and Janus-Pro [9] is a natural next step.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 4

[3] Michael Bauerly and Yili Liu. Computational modeling and experimental investigation of effects of compositional elements on interface and design aesthetics. *International journal of human-computer studies*, 64(8):670–682, 2006. 2

[4] Sanket Biswas, Rajiv Jain, Vlad I Morariu, Jiuxiang Gu, Puneet Mathur, Curtis Wigington, Tong Sun, and Josep Lladós. Docsynthv2: A practical autoregressive modeling for document generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8148–8153, 2024. 2

[5] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space, 2025. 4, 8

[6] Shang Chai, Liansheng Zhuang, Fengying Yan, and Zihan Zhou. Two-stage content-aware layout generation for poster designs. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8415–8423, 2023. 2

[7] Jian Chen, Ruiyi Zhang, Yufan Zhou, Rajiv Jain, Zhiqiang Xu, Ryan Rossi, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. *arXiv preprint arXiv:2402.04754*, 2024. 2

[8] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024. 6, 8

[9] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025. 8

[10] Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic design with large multimodal model. *arXiv preprint arXiv:2404.14368*, 2024. 2, 6

[11] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6*, pages 849–858. Springer, 2000. 5

[12] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 5

[13] 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 *Forty-first International Conference on Machine Learning*, 2024. 4

[14] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1–8, 2009. 8

[15] Sahil Goyal, Abhinav Mahajan, Swasti Mishra, Prateksha Udhayanan, Tripti Shukla, KJ Joseph, and Balaji Vasan Srinivasan. Design-o-meter: Towards evaluating and refining graphic designs. *arXiv preprint arXiv:2411.14959*, 2024. 1, 2, 3, 5, 6

[16] Julian Jorge Andrade Guerreiro, Naoto Inoue, Kento Masui, Mayu Otani, and Hideki Nakayama. Layoutflow: Flow matching for layout generation. *arXiv preprint arXiv:2403.18187*, 2024. 2

[17] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021. 2

[18] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 4

[19] Xiwei Hu, Haokun Chen, Zhongqi Qi, Hui Zhang, Dexiang Hong, Jie Shao, and Xinglong Wu. DreamPoster: A unified framework for image-conditioned generative poster design. *arXiv preprint arXiv:2507.04218*, 2025. 3

[20] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1942–1951, 2023. 2

[21] Nathan Hurst, Wilmot Li, and Kim Marriott. Review of automatic document formatting. In *Proceedings of the 9th ACM symposium on Document engineering*, pages 99–108, 2009. 2

[22] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models, 2023. 2, 3, 6

[23] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023. 2

[24] Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. Opencole: Towards reproducible automatic graphic

design generation. *arXiv preprint arXiv:2406.08232*, 2024. 3, 6

[25] Ali Jahanian, Jerry Liu, Qian Lin, Daniel Tretter, Eamonn O'Brien-Strain, Seungyon Claire Lee, Nic Lyons, and Jan Allebach. Recommendation system for automatic design of magazine covers. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 95–106, 2013. 2

[26] Peidong Jia, Chenxuan Li, Zeyu Liu, Yichao Shen, Xingru Chen, Yuhui Yuan, Yinglin Zheng, Dong Chen, Ji Li, Xiaodong Xie, et al. Cole: A hierarchical generation framework for graphic design. *arXiv preprint arXiv:2311.16974*, 2023. 2, 3, 5, 6

[27] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2023. 4

[28] Kotaro Kikuchi, Naoto Inoue, Mayu Otani, Edgar Simo-Serra, and Kota Yamaguchi. Multimodal markup document models for graphic design completion. *arXiv preprint arXiv:2409.19051*, 2024. 2

[29] Wenyuan Kong, Zhaoyun Jiang, Shizhao Sun, Zhuoning Guo, Weiwei Cui, Ting Liu, Jianguang Lou, and Dongmei Zhang. Aesthetics++: Refining graphic designs by exploring design principles and human preference. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):3093–3104, 2022. 2

[30] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 6

[31] Tianle Li, Max Ku, Cong Wei, and Wenhu Chen. Dreamedit: Subject-driven image editing, 2023. 8

[32] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutprompter: Awaken the design ability of large language models. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[33] Jiawei Lin, Shizhao Sun, Danqing Huang, Ting Liu, Ji Li, and Jiang Bian. From elements to design: A layered approach for automatic graphic design composition, 2024. 1, 2, 3, 5, 6

[34] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7817–7826, 2024. 4

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild, 2015. 8

[36] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 5

[37] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 4

[38] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 1221–1224, 2015. 2

[39] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics*, 20(8): 1200–1213, 2014. 2

[40] Sohan Patnaik, Rishabh Jain, Balaji Krishnamurthy, and Mausoom Sarkar. Aesthetiq: Enhancing graphic layout design via aesthetic-aware preference alignment of multi-modal large language models. *arXiv preprint arXiv:2503.00591*, 2025. 2

[41] Yadong Qu, Shancheng Fang, Yuxin Wang, Xiaorui Wang, Zhineng Chen, Hongtao Xie, and Yongdong Zhang. IGD: Instructional graphic design with multimodal layer generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 3

[42] Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of langauge model to contents-aware layout generation. *arXiv preprint arXiv:2404.00995*, 2024. 2

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[44] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024. 4, 8

[45] Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. Layoutnuwa: Revealing the hidden layout expertise of large language models. *arXiv preprint arXiv:2309.09506*, 2023. 2

[46] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1921–1930, 2023. 4

[47] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22532–22541, 2023. 4

[48] Yuxi Xie, Danqing Huang, Jinpeng Wang, and Chin-Yew Lin. Canvasemb: Learning layout representation with large-scale pre-training for graphic design. In *Proceedings of the 29th ACM international conference on multimedia*, pages 4100–4108, 2021. 2

[49] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. 2, 5

[50] Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. Automatic generation of visual-textual presentation layout. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(2):1–22, 2016. 2

[51] Hui Zhang, Dexiang Hong, Maoke Yang, Yutao Cheng, Zhao Zhang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. Creatidesign: A unified multi-conditional diffusion transformer for creative graphic design. *arXiv preprint arXiv:2505.19114*, 2025. 2, 3

[52] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22490–22499, 2023. 2

[53] Wanrong Zhu, Jennifer Healey, Ruiyi Zhang, William Yang Wang, and Tong Sun. Automatic layout planning for visually-rich documents with instruction-following models. *arXiv preprint arXiv:2404.15271*, 2024. 2